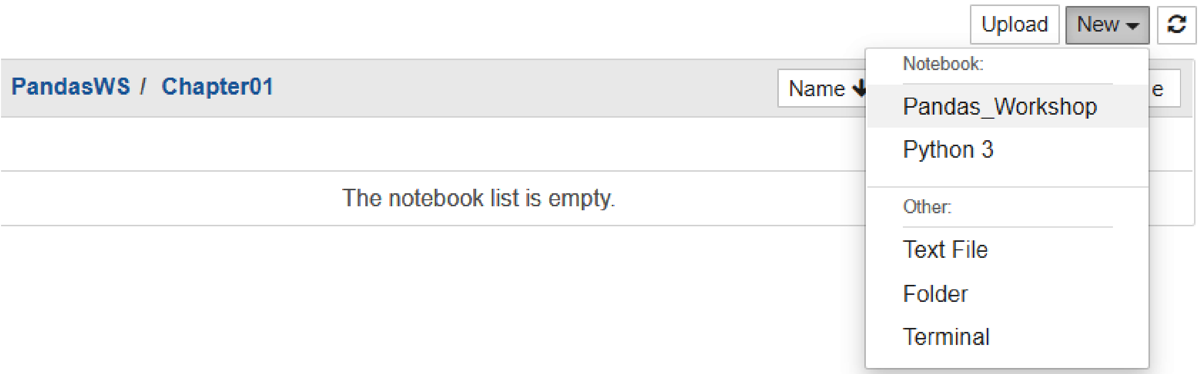
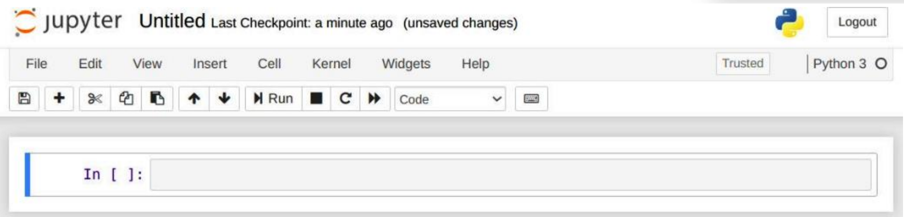
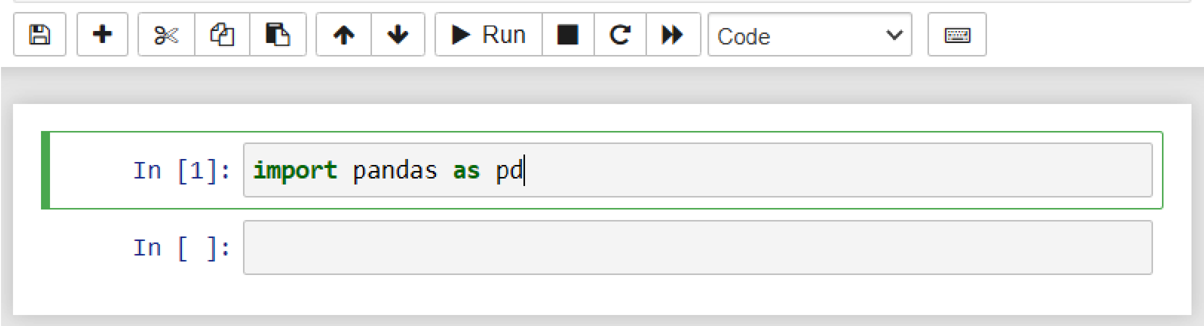
Chapter 1: Introduction to Pandas








```
0     10
1     20
2     30
3     40
dtype: int64
```

```
0                                [10, 20]
1                     [30, 40.5, series]
2                                [50, 55]
3    {'Name': 'Tess', 'Org': 'Packt'}
dtype: object
```

	0
0	30
1	50
2	20

```
Shape of new data frame (2, 3)
```

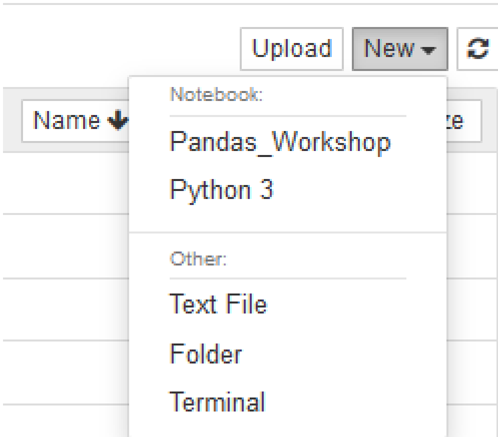
	V1
0	30
1	50
2	20

	V1
R1	30
R2	50
R3	20

	0	1	2
0	10	15	20
1	100	200	300

	V1	V2	V3
R1	10	15	20
R2	100	200	300

```
school;sex;age;address;famsize;Pstatus;Medu;Fedu;Mjob;Fjob;reason;guardian;traveltime;studyti
GP;"F";18;"U";"GT3";"A";4;4;"at_home";"teacher";"course";"mother";2;2;0;"yes";"no";"no";"no";"
GP;"F";17;"U";"GT3";"T";1;1;"at_home";"other";"course";"father";1;2;0;"no";"yes";"no";"no";"no";
GP;"F";15;"U";"LE3";"T";1;1;"at_home";"other";"other";"mother";1;2;0;"yes";"no";"no";"no";"yes";
```



	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	4	0	11	11
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	2	9	11	11
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	6	12	13	12
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	0	14	14	14
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	0	11	13	13

5 rows × 33 columns

A	B	C	D	E	F	G	H	I	J	K
	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob
0	GP	F	18	U	GT3	A	4	4	at_home	teacher
1	GP	F	17	U	GT3	T	1	1	at_home	other
2	GP	F	15	U	LE3	T	1	1	at_home	other
3	GP	F	15	U	GT3	T	4	2	health	services
4	GP	F	16	U	GT3	T	3	3	other	other

```
Out[7]: school        object
        sex           object
        age            int64
        address       object
        famsize       object
        Pstatus       object
        Medu           int64
        Fedu           int64
        Mjob          object
        Fjob          object
        reason        object
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 649 entries, 0 to 648
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   school      649 non-null    object
 1   sex         649 non-null    object
 2   age         649 non-null    int64
 3   address     649 non-null    object
 4   famsize     649 non-null    object
 5   Pstatus     649 non-null    object
 6   Medu        649 non-null    int64
 7   Fedu        649 non-null    int64
 8   Mjob        649 non-null    object
```

```
Out[13]: school        object
         sex           object
         age            int64
         address       object
         famsize       object
         Pstatus       object
         Medu         float64
         Fedu           int64
         Mjob          object
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4.0	4	at_home	teacher	...	4	3	4	1	1	3	4	0	11	11
1	GP	F	17	U	GT3	T	1.0	1	at_home	other	...	5	3	3	1	1	3	2	9	11	11
2	GP	F	15	U	LE3	T	1.0	1	at_home	other	...	4	3	2	2	3	3	6	12	13	12
3	GP	F	15	U	GT3	T	4.0	2	health	services	...	3	2	2	1	1	5	0	14	14	14
4	GP	F	16	U	GT3	T	3.0	3	other	other	...	4	3	2	1	2	5	0	11	13	13

5 rows × 33 columns

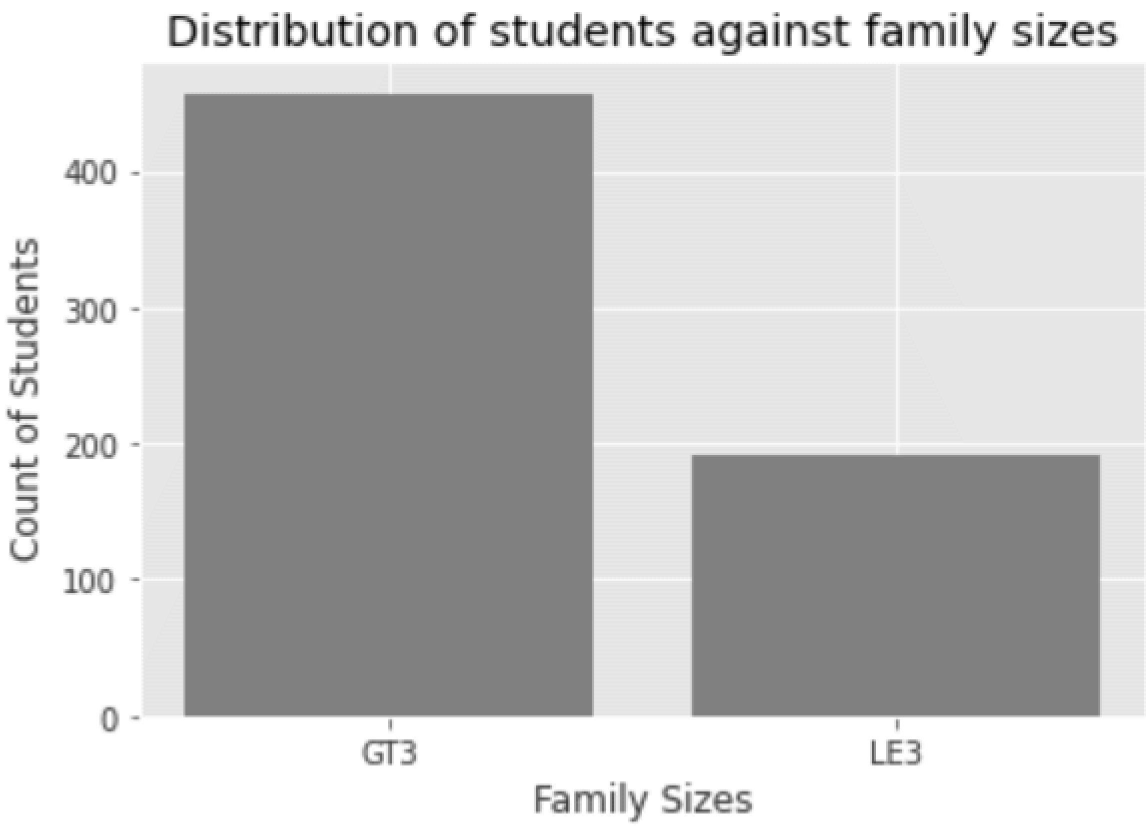
```
0      18
1      17
2      15
3      15
4      16
       ..
644    19
645    18
646    18
647    17
648    18
Name: age, Length: 649, dtype: int64
```

	age	address	famsize
0	18	U	GT3
1	17	U	GT3
2	15	U	LE3
3	15	U	GT3
4	16	U	GT3
...	...	...	...
644	19	R	GT3
645	18	U	LE3
646	18	U	GT3
647	17	U	LE3
648	18	R	LE3

649 rows × 3 columns

```
famsize
GT3    457
LE3    192
Name: famsize, dtype: int64
```

```
famsize
GT3    457
LE3    192
Name: famsize, dtype: int64
```




```
DatetimeIndex(['2021-02-09', '2021-02-10', '2021-02-11', '2021-02-12',
               '2021-02-13', '2021-02-14', '2021-02-15'],
              dtype='datetime64[ns]', freq='D')
```

	G1	G2	G3
0	5	16	16
1	14	16	16
2	17	18	17
3	19	19	19
4	16	18	18

	G1	G2	G3
0	5	16	16
1	14	16	16
2	17	18	17
3	19	19	19
4	16	18	18

```
array([3.38456318, 1.76608323, 2.14843901, 2.95586157, 2.4149523 ,
       2.3740889 , 1.50526992, 1.91944094, 0.03591338, 3.45030228])
```

axis = 0

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

axis = 1

```
sepal_length                                  24.3
sepal_width                                   16.4
petal_length                                     7
petal_width                                      1
```

```
0    10.2
1     9.5
2     9.4
3     9.4
4    10.2
```

	Product	Sales
0	C	45
1	A	60
2	A	26
3	C	57
4	C	81

```
0     45
1     60
2     26
Name: Sales, dtype: int64
```

```
Product      C
Sales       57
Name: 3, dtype: object
```

	Product	Sales
0	C	45
1	A	60
2	A	26
3	C	57
4	C	81

```
0      True
1      False
2      False
3      False
4      False
```

```
ProductSales
0      C      45
```

	V1	V2	V3
0	1.914369	6.926164	1.351655
1	3.997345	-0.933664	0.700947
2	3.282978	7.136794	2.100351
3	1.493705	12.794912	1.344148
4	2.421400	4.926122	0.996846

```
V1      302.710907
V2      494.139331
V3       95.243434
dtype:  float64
```

```
V1      3.027109
V2      4.941393
V3      0.952434
dtype:  float64
```

```
0      3.397396
1      1.254876
2      4.173374
3      5.210922
4      2.781456
         ...
95     0.228614
96     2.681817
97     2.503941
98     2.811963
99     2.792288
Length: 100, dtype: float64
```

	V1	V2	V3
0	0 0.0 1 1.0 2 1.0 3 0.0 4 ...	0 2.0 1 -1.0 2 2.0 3 4.0 4 ...	0 0.0 1 0.0 2 0.0 3 0.0 4 ...
1	0 1.914369 1 0.997345 2 0.282978 3...	0 0.926164 1 2.066336 2 1.136794 3...	0 1.351655 1 0.700947 2 2.100351 3...

	V1	V2	V3
0	(0.0, 1.9143693966994388)	(2.0, 0.9261640678154937)	(0.0, 1.3516550589033651)
1	(1.0, 0.9973454465835858)	(-1.0, 2.0663362054386534)	(0.0, 0.7009473343295873)
2	(1.0, 0.28297849805199204)	(2.0, 1.1367939064115546)	(0.0, 2.1003510496085642)
3	(0.0, 1.493705286081908)	(4.0, 0.7949117818079436)	(0.0, 1.3441484651110442)
4	(0.0, 2.4213997480314635)	(1.0, 1.9261220557055578)	(0.0, 0.9968463745430639)
...	...	...	...
95	(1.0, 1.031114458921742)	(-2.0, 1.3068349762420635)	(0.0, 1.347893659569804)
96	(0.0, 1.9154320879942335)	(1.0, 1.1921195307477372)	(0.0, 1.9379002733568176)
97	(0.0, 1.6365284553814157)	(1.0, 1.6674478368409789)	(0.0, 1.207847269946217)
98	(1.0, 0.37940061207813613)	(1.0, 0.9762148513802424)	(0.0, 1.080272210739859)
99	(0.0, 2.6208235654274477)	(1.0, 1.3461612136911745)	(0.0, 1.4098803048050945)

100 rows × 3 columns

```
0      15
1      40
2      15
3      10
4      15
       ..
95     15
96     10
97     35
98     40
99     20
Length: 100, dtype: object
```

```
0       15
1       40
2       15
3       10
4       15
        ..
95      15
96      10
97      35
98      40
99      20
Length: 100, dtype: int64
```

	V1	V2	V3	V4
0	1.914369	6.926164	1.351655	15
1	3.997345	-0.933664	0.700947	40
2	3.282978	7.136794	2.100351	15
3	1.493705	12.794912	1.344148	10
4	2.421400	4.926122	0.996846	15
...	...	...	...	...
95	4.031114	-4.693165	1.347894	15
96	1.915432	4.192120	1.937900	10
97	1.636528	4.667448	1.207847	35
98	3.379401	3.976215	1.080272	40
99	2.620824	4.346161	1.409880	20

100 rows × 4 columns

```
array([[ 1.91436940e+00,  6.92616407e+00,  1.35165506e+00,
         1.50000000e+01],
       [ 3.99734545e+00, -9.33663795e-01,  7.00947334e-01,
         4.00000000e+01],
       [ 3.28297850e+00,  7.13679391e+00,  2.10035105e+00,
         1.50000000e+01],
       [ 1.49370529e+00,  1.27949118e+01,  1.34414847e+00,
         1.00000000e+01],
       [ 2.42139975e+00,  4.92612206e+00,  9.96846375e-01,
         1.50000000e+01],
       [ 4.65143654e+00,  5.10242639e+00,  8.96668848e-01,
         1.50000000e+01],
       [ 5.73320757e-01,  5.53864845e+00,  9.56738857e-01,
         2.00000000e+01],
       [ 2.57108737e+00, -5.85927132e-01,  5.42346465e-01,
         1.50000000e+01],
       [ 4.26593626e+00,  6.27843992e+00,  9.52398730e-01,
         1.50000000e+01],
       [ 2.13325960e+00,  1.83770768e-01,  1.13934176e+00,
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
376	GP	F	18	U	GT3	T	1.0	1	other	other	...	4	5	5	1	2	2	0	14	14	14
142	GP	M	18	U	LE3	T	3.0	1	services	services	...	3	3	4	4	5	4	2	11	11	12
43	GP	M	15	U	GT3	T	2.0	2	services	services	...	5	4	1	1	1	1	0	9	10	10
162	GP	M	15	U	LE3	A	2.0	1	services	other	...	4	5	5	2	5	5	0	12	11	11
351	GP	M	20	U	GT3	A	3.0	2	services	other	...	5	5	3	1	1	5	0	14	15	15

5 rows × 33 columns

```
0     16
1     44
2     21
3     62
4     23
5     14
6     58
7     78
8     81
9     52
10    53
11    30
12    27
13    99
14    41
dtype: int64
```

```
0     C
1     B
2     C
3     B
4     A
5     B
6     C
7     C
8     C
9     C
10    C
11    B
12    A
13    B
14    A
dtype: object
```

```
0     16
1     44
2     21
3     62
4     23
5     14
6     58
7     78
8     81
9     52
10    53
11    30
12    27
13    99
14    41
dtype: int64
```

	Product	Sales
0	C	16
1	B	44
2	C	21
3	B	62
4	A	23
5	B	14
6	C	58
7	C	78
8	C	81
9	C	52
10	C	53
11	B	30
12	A	27
13	B	99
14	A	41

```
0     C
1     A
2     A
3     C
4     C
5     B
6     C
7     B
8     C
9     B
10    B
11    B
12    B
13    B
14    B
dtype: object
```

```
0     45
1     60
2     26
3     57
4     81
5     66
6     53
7     41
8     87
9     68
10    64
11    95
12    38
13    11
14    75
dtype: int64
```

	Product	Sales
0	C	45
1	A	60
2	A	26
3	C	57
4	C	81
5	B	66
6	C	53
7	B	41
8	C	87
9	B	68
10	B	64
11	B	95
12	B	38
13	B	11
14	B	75

```
0      True
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12    False
13    False
14    False
Name: Sales, dtype: bool
```

	Product	Sales
0	C	45

```
0        True
1        True
2        True
3       False
4        True
5        True
6       False
7       False
8        True
9        True
10       True
11       True
12       True
13      False
14       True
Name: Sales, dtype: bool
```

	Product	Sales
0	C	45
1	A	60
2	A	26
4	C	81
5	B	66
8	C	87
9	B	68
10	B	64
11	B	95
12	B	38
14	B	75

```
   ProductSales
3      C      57
6      C      53
7      B      41
13     B      11
```

```
   ProductSales
2      A      26
5      B      66
6      C      53
8      C      87
```

```
         Sales
Product
A           91
B          249
C          359
         Sales
Product
A           86
B          458
C          323
```

Product	Sales1	Sales2
A	91	86
B	249	458
C	359	323

	Months	Grocery_sales	Stationary_sales
0	Jan	16	57
1	Jan	44	139
2	Jan	15	85
3	Jan	59	8
4	Jan	36	106

Chapter 2: Working with Data Structures

	A	B
1	date	GDP
2	2017-03-31	19190.4
3	2017-06-30	19356.6
4	2017-09-30	19611.7
5	2017-12-31	19918.9
6	2018-03-31	20163.2
7	2018-06-30	20510.2
8	2018-09-30	20749.8
9	2018-12-31	20897.8
10	2019-03-31	21098.8
11	2019-06-30	21340.3
12	2019-09-30	21542.5
13	2019-12-31	21729.1

	date	GDP
0	2017-03-31	19190.4
1	2017-06-30	19356.6
2	2017-09-30	19611.7
3	2017-12-31	19918.9
4	2018-03-31	20163.2
5	2018-06-30	20510.2
6	2018-09-30	20749.8
7	2018-12-31	20897.8
8	2019-03-31	21098.8
9	2019-06-30	21340.3
10	2019-09-30	21542.5
11	2019-12-31	21729.1

```
Out[4]:
```

	col1	col2
0	0	1
1	1	3
2	2	5
3	3	7
4	4	9

Out[5]:

	col1	col2
95	95	191
96	96	193
97	97	195
98	98	197
99	99	199

In [6]:
```
?pd.DataFrame
```

```
Init signature:
pd.DataFrame(
    data=None,
    index: Union[Collection, NoneType] = None,
    columns: Union[Collection, NoneType] = None,
    dtype: Union[str, numpy.dtype, ForwardRef('ExtensionDtype'), NoneType] = None,
    copy: bool = False,
)
Docstring:
Two-dimensional, size-mutable, potentially heterogeneous tabular data.

Data structure also contains labeled axes (rows and columns).
Arithmetic operations align on both row and column labels. Can be
thought of as a dict-like container for Series objects. The primary
pandas data structure.

Parameters
----------
data : ndarray (structured or homogeneous), Iterable, dict, or DataFrame
    Dict can contain Series, arrays, constants, or list-like objects.
```

```
   col1  col2
0     0     1
1     1     3
2     2     5
3     3     7
4     4     9
    col1  col2
95    95   191
96    96   193
97    97   195
98    98   197
99    99   199
```

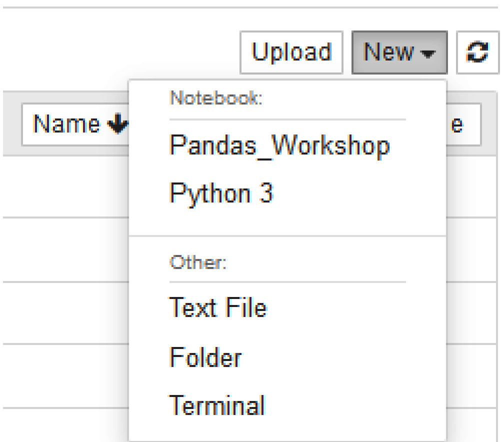
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28,
29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 5
5, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99] [1, 3, 5, 7, 9, 11, 13, 15, 17, 1
9, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71,
73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99, 101, 103, 105, 107, 109, 111, 113, 115, 117, 119,
121, 123, 125, 127, 129, 131, 133, 135, 137, 139, 141, 143, 145, 147, 149, 151, 153, 155, 157, 159, 161, 1
63, 165, 167, 169, 171, 173, 175, 177, 179, 181, 183, 185, 187, 189, 191, 193, 195, 197, 199]
```

```
   col1  col2
0     0     1
1     1     3
2     2     5
3     3     7
4     4     9
    col1  col2
95    95   191
96    96   193
97    97   195
98    98   197
99    99   199
```

```
   col1  col2
0     0     1
1     1     3
2     2     5
3     3     7
4     4     9
    col1  col2
95    95   191
96    96   193
97    97   195
98    98   197
99    99   199
```

Out[35]:

	0	1	2	3	4
0	0	1	2	3	4
1	2	3	4	5	6



Out[19]:

	time	measurement
0	0.0	0.000000e+00
1	0.1	5.877853e-01
2	0.2	9.510565e-01
3	0.3	9.510565e-01
4	0.4	5.877853e-01
5	0.5	1.224647e-16
6	0.6	-5.877853e-01
7	0.7	-9.510565e-01
8	0.8	-9.510565e-01
9	0.9	-5.877853e-01
10	1.0	-2.449294e-16

Out[33]:

	letter
21	v
22	w
23	x
24	y
25	aa
26	bb
27	cc
28	dd

```
RangeIndex(start=0, stop=100, step=1)
Index(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n',
       'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'aa', 'bb', 'cc',
       'dd', 'ee', 'ff', 'gg', 'hh', 'ii', 'jj', 'kk', 'll', 'mm', 'nn', 'oo',
       'pp', 'qq', 'rr', 'ss', 'tt', 'uu', 'vv', 'ww', 'xx', 'yy', 'aaa',
       'bbb', 'ccc', 'ddd', 'eee', 'fff', 'ggg', 'hhh', 'iii', 'jjj', 'kkk',
       'lll', 'mmm', 'nnn', 'ooo', 'ppp', 'qqq', 'rrr', 'sss', 'ttt', 'uuu',
       'vvv', 'www', 'xxx', 'yyy', 'aaaa', 'bbbb', 'cccc', 'dddd', 'eeee',
       'ffff', 'gggg', 'hhhh', 'iiii', 'jjjj', 'kkkk', 'llll', 'mmmm', 'nnnn',
       'oooo', 'pppp', 'qqqq', 'rrrr', 'ssss', 'tttt', 'uuuu', 'vvvv', 'wwww',
       'xxxx', 'yyyy'],
      dtype='object', name='letter')
```

Out[48]:

	col1	col2
letter		
uuuu	95	191
vvvv	96	193
wwww	97	195
xxxx	98	197
yyyy	99	199

Out[129]:

	col1	col2	animal_type
letter			
a	0	1	cat
b	1	3	cat
c	2	5	cat
d	3	7	cat
e	4	9	cat
...	...	...	...
uuuu	95	191	dog
vvvv	96	193	dog
wwww	97	195	dog
xxxx	98	197	dog
yyyy	99	199	dog

100 rows × 3 columns

Out[133]:

	col1	col2
animal_type		
cat	0	1
cat	1	3
cat	2	5
cat	3	7
cat	4	9
...	...	...
dog	95	191
dog	96	193
dog	97	195
dog	98	197
dog	99	199

100 rows × 2 columns

Out[34]:

	good	bad
animal_type		
cat	0	1
cat	1	3
cat	2	5
cat	3	7
cat	4	9

Out[35]:

animal_type	good	bad
cat	0	1
cat	1	3
cat	2	5
cat	3	7
cat	4	9
cat	5	11
cat	6	13
cat	7	15
cat	8	17
cat	9	19

```
            product  wholesale_price   msrp  qty_ordered  qty_shipped
0      skippys_dream             8.99  18.38          100          100
1      just_the_beef             4.99  10.43          200          195
2  potatos_and_lamb              5.19  11.43           50           50
           product  wholesale_price   msrp  qty_ordered  qty_shipped
0      cat_delight             4.95   9.98           50            0
1    tuna_surprise             7.17  15.27          100          100
2  hint_of_catnip              3.99   8.23           25           25
            product  wholesale_price   msrp  qty_ordered  qty_shipped animal
0      skippys_dream             8.99  18.38          100          100    dog
1      just_the_beef             4.99  10.43          200          195    dog
2  potatos_and_lamb              5.19  11.43           50           50    dog
           product  wholesale_price   msrp  qty_ordered  qty_shipped animal
0      cat_delight             4.95   9.98           50            0    cat
1    tuna_surprise             7.17  15.27          100          100    cat
2  hint_of_catnip              3.99   8.23           25           25    cat
```

Out[8]:

	product	wholesale_price	msrp	qty_ordered	qty_shipped	animal
0	skippys_dream	8.99	18.38	100	100	dog
1	just_the_beef	4.99	10.43	200	195	dog
2	potatos_and_lamb	5.19	11.43	50	50	dog
3	turkey_and_cranberries	5.98	12.00	50	50	dog
4	roasted_duck	9.59	17.48	15	15	dog
0	cat_delight	4.95	9.98	50	0	cat
1	tuna_surprise	7.17	15.27	100	100	cat
2	hint_of_catnip	3.99	8.23	25	25	cat
3	roast_chicken	5.57	12.08	30	30	cat
4	lamb_w_rice	5.83	11.68	30	30	cat

Out[9]:

	product	wholesale_price	msrp	qty_ordered	qty_shipped
animal					
dog	skippys_dream	8.99	18.38	100	100
dog	just_the_beef	4.99	10.43	200	195
dog	potatos_and_lamb	5.19	11.43	50	50
dog	turkey_and_cranberries	5.98	12.00	50	50
dog	roasted_duck	9.59	17.48	15	15
cat	cat_delight	4.95	9.98	50	0
cat	tuna_surprise	7.17	15.27	100	100
cat	hint_of_catnip	3.99	8.23	25	25
cat	roast_chicken	5.57	12.08	30	30
cat	lamb_w_rice	5.83	11.68	30	30

Out[44]:

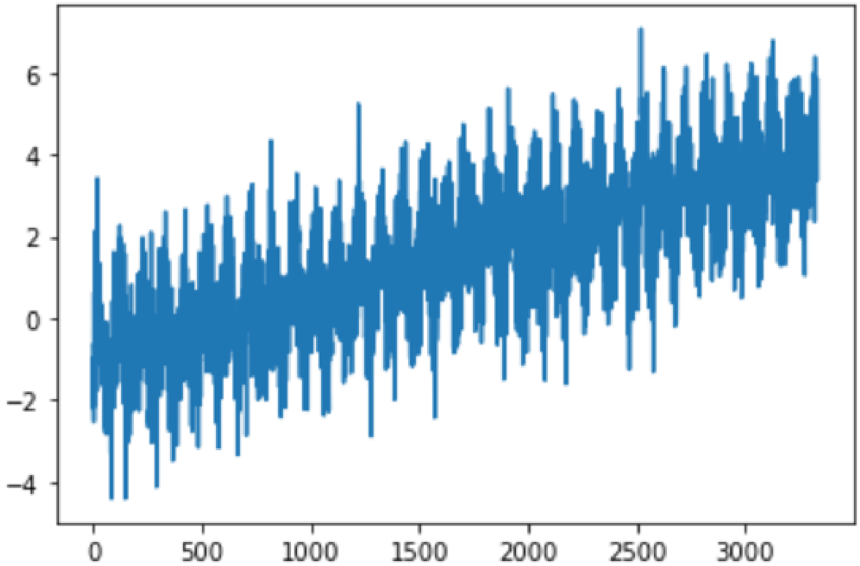
	food_consumption	taste_index
0	60.0	3.5
1	40.0	8.0

Out[168]:

	food_cons	taste
0	60.0	3.5
1	40.0	8.0

```
Init signature:
pd.Series(
    data=None,
    index=None,
    dtype=None,
    name=None,
    copy=False,
    fastpath=False,
)
Docstring:
One-dimensional ndarray with axis labels (including time series).
```

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x2c0ab0d5248>
```




```
Out[26]: 0    0
         1    1
         2    2
         3    3
         4    4
         dtype: int64
```

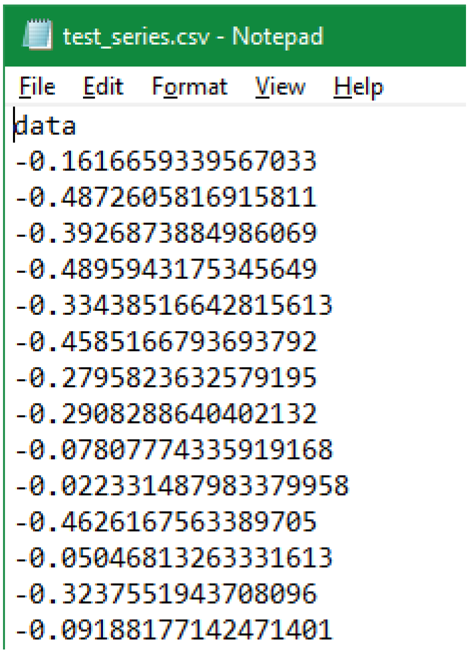
```
0    a
1    b
2    c
3    d
4    e
dtype: object
 21    v
22    w
23    x
24    y
25    z
dtype: object
```

```
Out[33]: n    13
         o    14
         p    15
         q    16
         r    17
         s    18
         t    19
         u    20
         v    21
         w    22
         x    23
         y    24
         z    25
         dtype: int64
```

```
Out[35]: 0                  1
         1                cat
         2          yesterday
         3          [1, 2, 3]
         4    (0, 1, 2, 3, 4)
         dtype: object
```

```
1  is type  <class 'int'>
cat  is type  <class 'str'>
yesterday  is type  <class 'str'>
[1, 2, 3]  is type  <class 'list'>
range(0, 5)  is type  <class 'range'>
```

```
Out[2]: 0      -0.161666
        1      -0.487261
        2      -0.392687
        3      -0.489594
        4      -0.334385
                  ...
        360    -0.386292
        361    -0.311027
        362    -0.437957
        363    -0.569164
        364    -0.727658
        Name: data, Length: 365, dtype: float64
```


test_series.csv - Notepad

File Edit Format View Help

```
data
-0.1616659339567033
-0.4872605816915811
-0.3926873884986069
-0.4895943175345649
-0.33438516642815613
-0.4585166793693792
-0.2795823632579195
-0.2908288640402132
-0.07807774335919168
-0.022331487983379958
-0.4626167563389705
-0.05046813263331613
-0.3237551943708096
-0.09188177142471401
```

```
Out[3]: 0       0.000000
        1       0.068242
        2       0.136167
        3       0.203456
        4       0.269797
                  ...
        360    -0.519584
        361    -0.460065
        362    -0.398401
        363    -0.334880
        364    -0.269797
        Length: 365, dtype: float64
```

Out[4]:

	col1	col2
0	-0.161666	0.000000
1	-0.487261	0.068242
2	-0.392687	0.136167
3	-0.489594	0.203456
4	-0.334385	0.269797

Out[19]:

	date	data
0	2017-01-01	0
1	2017-01-02	1
2	2017-01-03	2
3	2017-01-04	3
4	2017-01-05	4
5	2017-01-06	5

Out[31]:

	data
date	
2017-01-01	0
2017-01-02	1
2017-01-03	2
2017-01-04	3
2017-01-05	4
2017-01-06	5

Out[25]:

	data
date	
2017-01-01 00:00:00	0.0
2017-01-01 12:00:00	0.5
2017-01-02 00:00:00	1.0
2017-01-02 12:00:00	1.5
2017-01-03 00:00:00	2.0
2017-01-03 12:00:00	2.5
2017-01-04 00:00:00	3.0
2017-01-04 12:00:00	3.5
2017-01-05 00:00:00	4.0
2017-01-05 12:00:00	4.5
2017-01-06 00:00:00	5.0

Out[28]:

	time	1/31/2019	2/28/2019	3/31/2019	4/30/2019	5/31/2019	6/30/2019	7/31/2019	8/31/2019	9/30/2019	10/31/2019	11/30/2019	12/31/2019
0	12:00:00 AM	2312.22	2403.93	2285.59	1841.71	1144.73	579.97	184.34	217.88	609.83	1098.53	1832.15	2409.02
1	12:15:00 AM	2357.01	2503.56	2319.69	1863.97	1183.33	511.77	225.56	158.63	531.24	1132.16	1797.98	2354.98
2	12:30:00 AM	2298.20	2475.26	2386.27	1875.62	1259.22	555.14	167.05	199.51	536.58	1126.26	1725.46	2336.46
3	12:45:00 AM	2359.41	2615.92	2368.70	1825.99	1139.68	525.37	117.55	149.68	482.08	1087.29	1816.17	2374.96
4	1:00:00 AM	2328.82	2565.09	2298.29	1802.28	1178.65	586.78	212.88	129.09	551.16	1145.26	1802.78	2318.55

Out[29]:

```
time           object
1/31/2019     float64
2/28/2019     float64
3/31/2019     float64
4/30/2019     float64
5/31/2019     float64
6/30/2019     float64
7/31/2019     float64
8/31/2019     float64
9/30/2019     float64
10/31/2019    float64
11/30/2019    float64
12/31/2019    float64
dtype: object
```

Out[22]:

	time	1/31/2019	2/28/2019	3/31/2019	4/30/2019	5/31/2019	6/30/2019	7/31/2019	8/31/2019	9/30/2019	10/31/2019	11/30/2019	12/31/2019
0	0 days 00:00:00	2312.22	2403.93	2285.59	1841.71	1144.73	579.97	184.34	217.88	609.83	1098.53	1832.15	2409.02
1	0 days 00:15:00	2357.01	2503.56	2319.69	1863.97	1183.33	511.77	225.56	158.63	531.24	1132.16	1797.98	2354.98
2	0 days 00:30:00	2298.20	2475.26	2386.27	1875.62	1259.22	555.14	167.05	199.51	536.58	1126.26	1725.46	2336.46
3	0 days 00:45:00	2359.41	2615.92	2368.70	1825.99	1139.68	525.37	117.55	149.68	482.08	1087.29	1816.17	2374.96
4	0 days 01:00:00	2328.82	2565.09	2298.29	1802.28	1178.65	586.78	212.88	129.09	551.16	1145.26	1802.78	2318.55
...	...	...	...	...	...	...	...	...	...	...	...	...	...
91	0 days 22:45:00	2347.70	2549.58	2351.71	1850.14	1064.03	534.39	208.35	176.65	580.45	1100.67	1821.87	2263.76
92	0 days 23:00:00	2234.47	2570.41	2296.87	1778.81	1180.03	584.29	108.65	243.64	477.31	1214.94	1816.56	2231.82
93	0 days 23:15:00	2302.04	2469.22	2273.06	1865.61	1146.12	535.60	112.78	137.46	554.88	1131.31	1894.77	2360.27
94	0 days 23:30:00	2276.66	2401.19	2326.91	1801.10	1125.49	535.32	156.38	242.52	585.82	1121.86	1786.20	2293.01
95	0 days 23:45:00	2338.81	2521.09	2317.67	1825.98	1164.15	561.30	177.96	106.36	574.28	1116.48	1834.12	2305.17

Out[23]:

	1/31/2019	2/28/2019	3/31/2019	4/30/2019	5/31/2019	6/30/2019	7/31/2019	8/31/2019	9/30/2019	10/31/2019	11/30/2019	12/31/2019
time												
0 days 00:00:00	2312.22	2403.93	2285.59	1841.71	1144.73	579.97	184.34	217.88	609.83	1098.53	1832.15	2409.02
0 days 00:15:00	2357.01	2503.56	2319.69	1863.97	1183.33	511.77	225.56	158.63	531.24	1132.16	1797.98	2354.98
0 days 00:30:00	2298.20	2475.26	2386.27	1875.62	1259.22	555.14	167.05	199.51	536.58	1126.26	1725.46	2336.46
0 days 00:45:00	2359.41	2615.92	2368.70	1825.99	1139.68	525.37	117.55	149.68	482.08	1087.29	1816.17	2374.96
0 days 01:00:00	2328.82	2565.09	2298.29	1802.28	1178.65	586.78	212.88	129.09	551.16	1145.26	1802.78	2318.55

Out[6]:

	1/31/2019	2/28/2019	3/31/2019	4/30/2019	5/31/2019	6/30/2019	7/31/2019	8/31/2019	9/30/2019	10/31/2019	11/30/2019	12/31
time												
00:00:00	2312.220000	2403.930000	2285.590000	1841.710000	1144.730000	579.970000	184.340000	217.880000	609.830000	1098.530000	1832.150000	2409.0
00:05:00	2327.150000	2437.140000	2296.956667	1849.130000	1157.596667	557.236667	198.080000	198.130000	583.633333	1109.740000	1820.760000	2391.0
00:10:00	2342.080000	2470.350000	2308.323333	1856.550000	1170.463333	534.503333	211.820000	178.380000	557.436667	1120.950000	1809.370000	2372.9
00:15:00	2357.010000	2503.560000	2319.690000	1863.970000	1183.330000	511.770000	225.560000	158.630000	531.240000	1132.160000	1797.980000	2354.9
00:20:00	2337.406667	2494.126667	2341.883333	1867.853333	1208.626667	526.226667	206.056667	172.256667	533.020000	1130.193333	1773.806667	2348.8

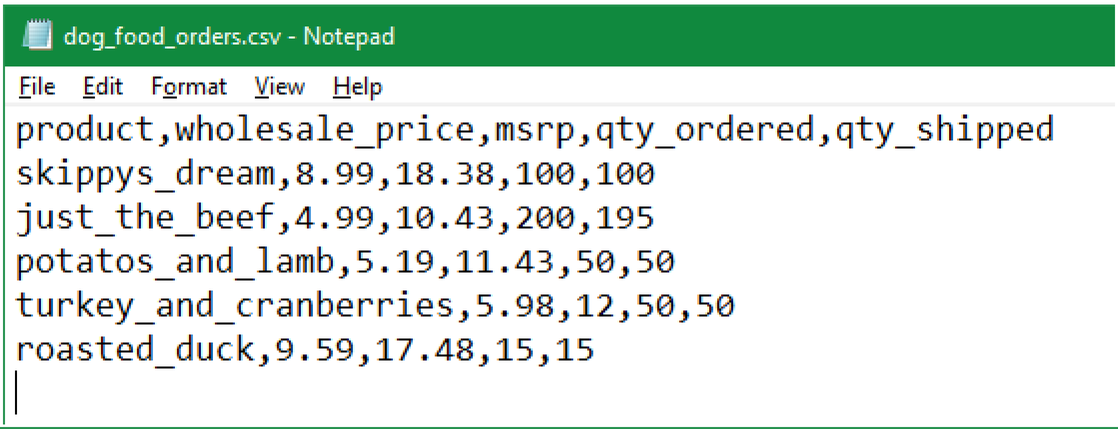
Out[4]:

	GDP
date	
2017-03-31	19190.4
2017-06-30	19356.6
2017-09-30	19611.7
2017-12-31	19918.9
2018-03-31	20163.2

Chapter 3: Data I/O

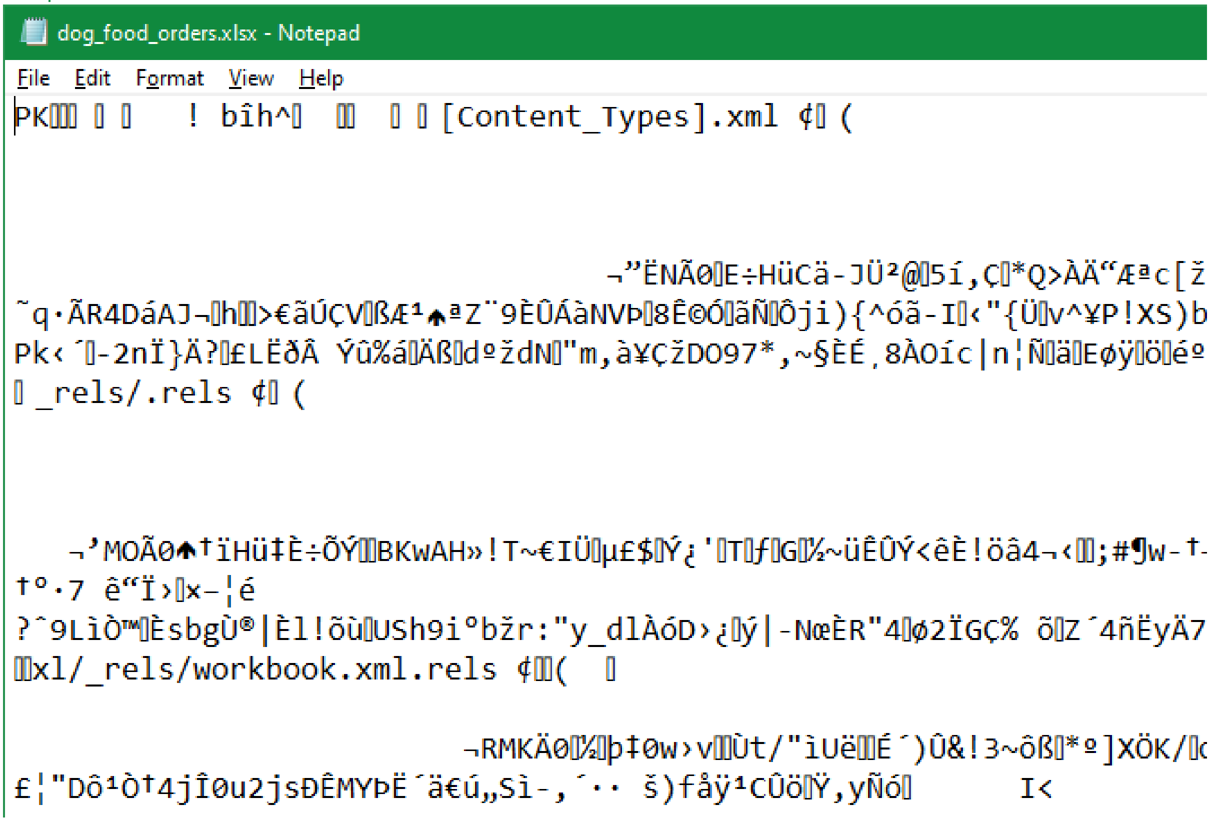
Type of Data	File / system	Input	Output	dependencies
text	**CSV**	**read_csv**	**to_csv**	
text	**JSON**	**read_json**	**to_json**	
text	**HTML**	**read_html**	**to_html**	**lxml or bs4/html5lib**
text	**XML**			**pandas-read-xml**
text	Local clipboard	read_clipboard	to_clipboard	
text	Fixed-Width Text File	read_fwf		
binary	**Matlab / Octave**			**scipy.io**
binary	**Excel**	**read_excel**	**to_excel**	**xlrd or openpyxl**
binary	**HDF5**	**read_hdf**	**to_hdf**	**zlib, lzo, etc.**
binary	**Stata**	**read_stata**	**to_stata**	**pyreadstat**
binary	**SAS**	**read_sas**		
binary	OpenDocument	read_excel		
binary	Feather	read_feather	to_feather	
binary	Parquet	read_parquet	to_parquet	
binary	ORC	read_orc		
binary	Msgpack	read_msgpack	to_msgpack	
binary	**SPSS**	**read_spss**		
binary	Pickle	read_pickle	to_pickle	
SQL	**SQL**	**read_sql**	**to_sql**	**sqlite3**
SQL	**BigQuery (Google)**	read_gbq	to_gbq	**pandas-gbq** **google-cloud-bigquery**

```
Out[18]: {'__header__': b'MATLAB 5.0 MAT-file Platform: nt, Created on: Tue Feb  2 14:21:02 2021',
          '__version__': '1.0',
          '__globals__': [],
          'storage': array([[0.00000000e+00],
                 [3.60020368e-04],
                 [7.26299303e-04],
                 ...,
                 [1.36616373e-05],
                 [1.35810556e-05],
                 [1.36134929e-05]]),
          'T1': array([[475.5],
                 [475.5],
                 [475.4],
                 ...,
                 [476.8],
                 [476.8],
                 [476.8]]),
          'time': array([[10256548.8],
                 [10256549. ],
                 [10256549.2],
                 ...,
                 [10273672.4],
                 [10273672.6],
                 [10273672.8]]),
          'value': array([[10256548.8       ],
                 [10256550.09106825],
                 [10256550.31226313],
                 ...,
                 [10273670.63541315],
                 [10273672.1572869 ],
                 [10273672.87393071]])}
```


dog_food_orders.csv - Notepad

File Edit Format View Help

```
product,wholesale_price,msrp,qty_ordered,qty_shipped
skippys_dream,8.99,18.38,100,100
just_the_beef,4.99,10.43,200,195
potatos_and_lamb,5.19,11.43,50,50
turkey_and_cranberries,5.98,12,50,50
roasted_duck,9.59,17.48,15,15
```


dog_food_orders.xlsx - Notepad

File Edit Format View Help

PK ! bîh^ [Content_Types].xml ¢ (
¬”ËNÃ0E÷HüCä-JÜ²@5í,Ç*Q>ÀÄ“Æªc[ž
˜q·ÃR4DáAJ¬h>€ãÚÇVßÆ¹ªZ¨9ÈÛÁàNVÞ8Ê©ÓãÑÔji){^óã-I‹"{Üvˆ¥P!XS)b
Pk‹´-2nÏ}Ä?£LËðÂ Ýû%áÄßdºždN"m,à¥ÇžDO97*,~§ÈÉ¸8ÀOíc|n¦ÑäEøÿöéº
_rels/.rels ¢ (
¬’MOÃ0†ïHü‡È÷ÕÝBKwAH»!T~€IÜµ£$Ý¿'TƒG½~üÊÛÝ<êÈ!öâ4¬‹;#¶w-†-
†º·7 ê“Ï›x–¦é
?ˆ9LìÒ™ÈsbgÙ®|Èl!õùUSh9iºbžr:"y_dlÀóD›¿ý|-NœÈR"4ø2ÏGÇ% õZ´4ñËyÄ7
xl/_rels/workbook.xml.rels ¢(
¬RMKÄ0½þ‡0w›vÙt/"ìUëÉ´)Û&!3~ôß*º]XÖK/
£¦"Dô¹Ò†4jÎ0u2jsÐÊMYÞË´ä€ú„Sì-,´·· š)fåÿ¹CÛöŸ,yÑó I<

Out[19]:

	product	wholesale_price	msrp	qty_ordered	qty_shipped
0	skippys_dream	8.99	18.38	100	100
1	just_the_beef	4.99	10.43	200	195
2	potatos_and_lamb	5.19	11.43	50	50
3	turkey_and_cranberries	5.98	12.00	50	50
4	roasted_duck	9.59	17.48	15	15

Wind power

From Wikipedia, the free encyclopedia

"wind energy" redirects here. For the academic journal, see Wind Energy (journal).

For other types of wind turbines used for direct mechanical power, see windmill and windpump.

Wind power or **wind energy** is the use of wind to provide mechanical power through wind turbines to turn electric generators for electrical power. Wind power is a popular sustainable, renewable source of power that has a much smaller impact on the environment compared to burning fossil fuels.

Wind farms consist of many individual wind turbines, which are connected to the electric power transmission network. Onshore wind is an inexpensive source of electric power, competitive with or in many places cheaper than coal or gas plants. Onshore wind farms have a greater visual impact on the landscape than other power stations, as they need to be spread over more land and need to be built away from dense population. Offshore wind is steadier and stronger than on land and offshore farms have less visual impact, but construction and maintenance costs are significantly higher. Small onshore wind farms can feed some energy into the grid or provide power to isolated off-grid locations.

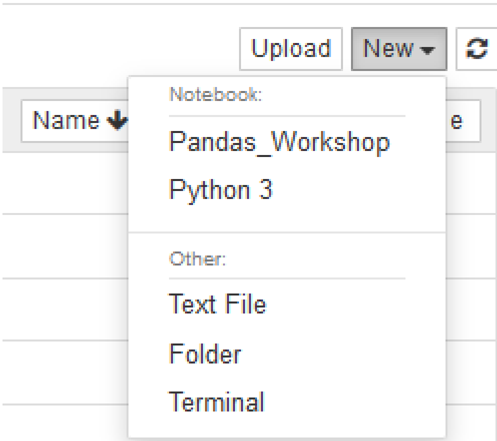
Large onshore wind farms

Wind farm	Capacity (MW)	Country	Refs
Gansu Wind Farm	7,965	China	[18][19]
Muppandal wind farm	1,500	India	[20]
Alta (Oak Creek-Mojave)	1,320	United States	[21]
Jaisalmer Wind Park	1,064	India	[22]
Shepherds Flat Wind Farm	845	United States	[23]
Roscoe Wind Farm	782	United States	
Horse Hollow Wind Energy Center	736	United States	[24][25]
Capricorn Ridge Wind Farm	662	United States	[24][25]
Fântânele-Cogealac Wind Farm	600	Romania	[26]
Fowler Ridge Wind Farm	600	United States	[27]
Whitelee Wind Farm	539	United Kingdom	[28]

```
Out[6]: [                                                0
0                            Part of a series about
1                                Sustainable energy
2                                          Overview
3             Carbon-neutral fuel Fossil fuel phase-out
4                               Energy conservation
5   Cogeneration Efficient energy use Energy stora...
6                                  Renewable energy
7   Hydroelectricity Solar Wind Bioenergy Geotherm...
8                             Sustainable transport
9         Electric vehicle Green vehicle Plug-in hybrid
10          Renewable energy portal  Environment portal
11  .mw-parser-output .navbar{display:inline;font-...,
                          Wind farm  Capacity(MW)         Country        Refs
0                   Gansu Wind Farm          7965           China    [18][19]
1                Muppandal wind farm          1500           India        [20]
2            Alta (Oak Creek-Mojave)          1320   United States        [21]
3                Jaisalmer Wind Park          1064           India        [22]
4           Shepherds Flat Wind Farm           845   United States        [23]
5                  Roscoe Wind Farm           782   United States         NaN
6   Horse Hollow Wind Energy Center           736   United States    [24][25]
7          Capricorn Ridge Wind Farm           662   United States    [24][25]
8       Fântânele-Cogealac Wind Farm           600         Romania        [26]
9            Fowler Ridge Wind Farm           600   United States        [27]
10               Whitelee Wind Farm           539  United Kingdom       [28],
```

Out[9]:

	Wind farm	Capacity(MW)	Country	Refs
0	Gansu Wind Farm	7965	China	[18][19]
1	Muppandal wind farm	1500	India	[20]
2	Alta (Oak Creek-Mojave)	1320	United States	[21]
3	Jaisalmer Wind Park	1064	India	[22]
4	Shepherds Flat Wind Farm	845	United States	[23]
5	Roscoe Wind Farm	782	United States	NaN
6	Horse Hollow Wind Energy Center	736	United States	[24][25]
7	Capricorn Ridge Wind Farm	662	United States	[24][25]
8	Fântânele-Cogealac Wind Farm	600	Romania	[26]
9	Fowler Ridge Wind Farm	600	United States	[27]
10	Whitelee Wind Farm	539	United Kingdom	[28]

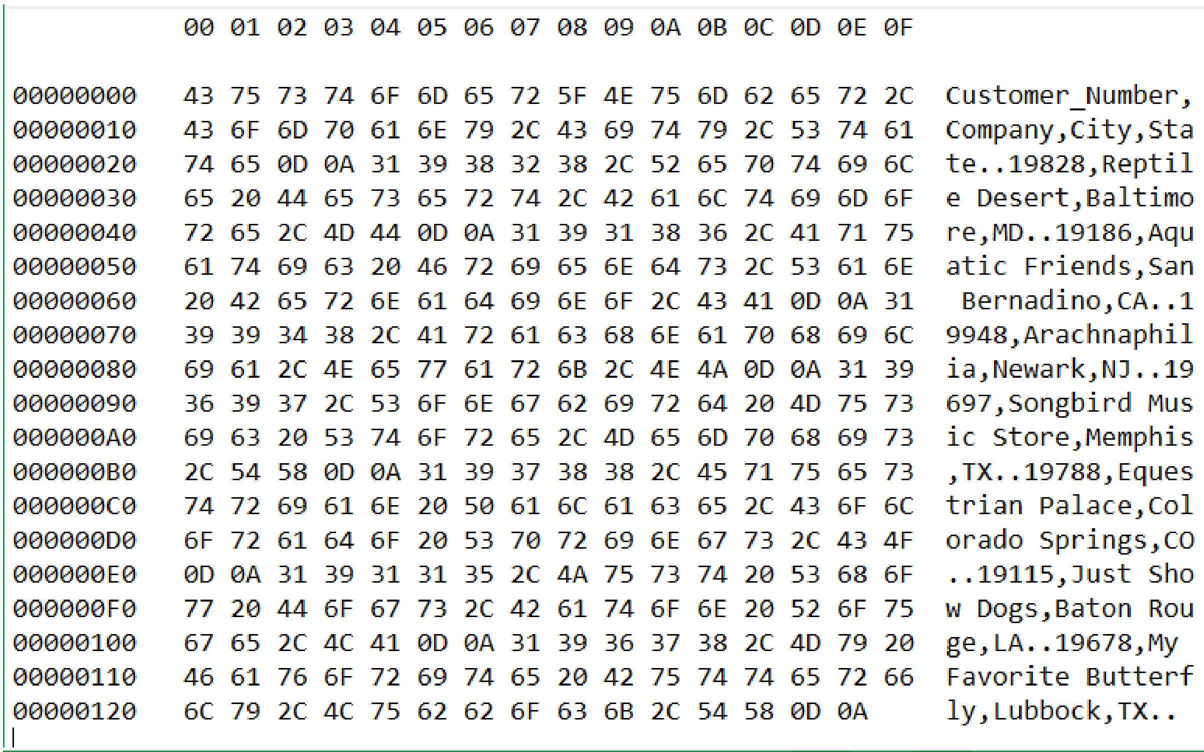



```
Out[3]: [                                                  0
         0                           Part of a series about
         1                               Sustainable energy
         ...
         10       Renewable energy portal  Environment portal
         11  .mw-parser-output .navbar{display:inline;font-...,
             Solar Electricity Generation                         \
                                     Year                 Energy (TWh)
         0                           2004                          2.6
         1                           2005                          3.7
         2                           2006                          5.0
         ...

         12                         1.31%
         13                         1.73%
         14                         2.68%
         15  Sources:[32][33][34][35][36]  ,
                                     Name        Country  CapacityMWp  \
         0            Pavagada Solar Park          India         2050
         1      Tengger Desert Solar Park          China         1547
         2              Bhadla Solar Park          India         1515
         3     Kurnool Ultra Mega Solar Park          India         1000
         4  Datong Solar Power Top Runner Base          China         1000
         5       Longyangxia Dam Solar Park          China          850
         6            Rewa Ultra Mega Solar          India          750
         7        Kamuthi Solar Power Project          India          648
         8            Solar Star (I and II)  United States          579
         9                Topaz Solar Farm  United States          550

           GenerationGWh p.a.   Sizekm2  Year                 Ref
         0                NaN        53  2017        [2][52][53]
         1                NaN        43  2016           [54][55]
         2                NaN        40  2017       [56][57][58]
         3                NaN        24  2017               [59]
         4                NaN       NaN  2016       [60][61][62]
         5                NaN        23  2015  [63][64][65][66][67]
         6                NaN       NaN  2018               [68]
         7                NaN      10.1  2016           [69][70]
         8             1664.0        13  2015           [71][72]
         9             1301.0  24.6[73]  2014       [74][75][76]  ,
```

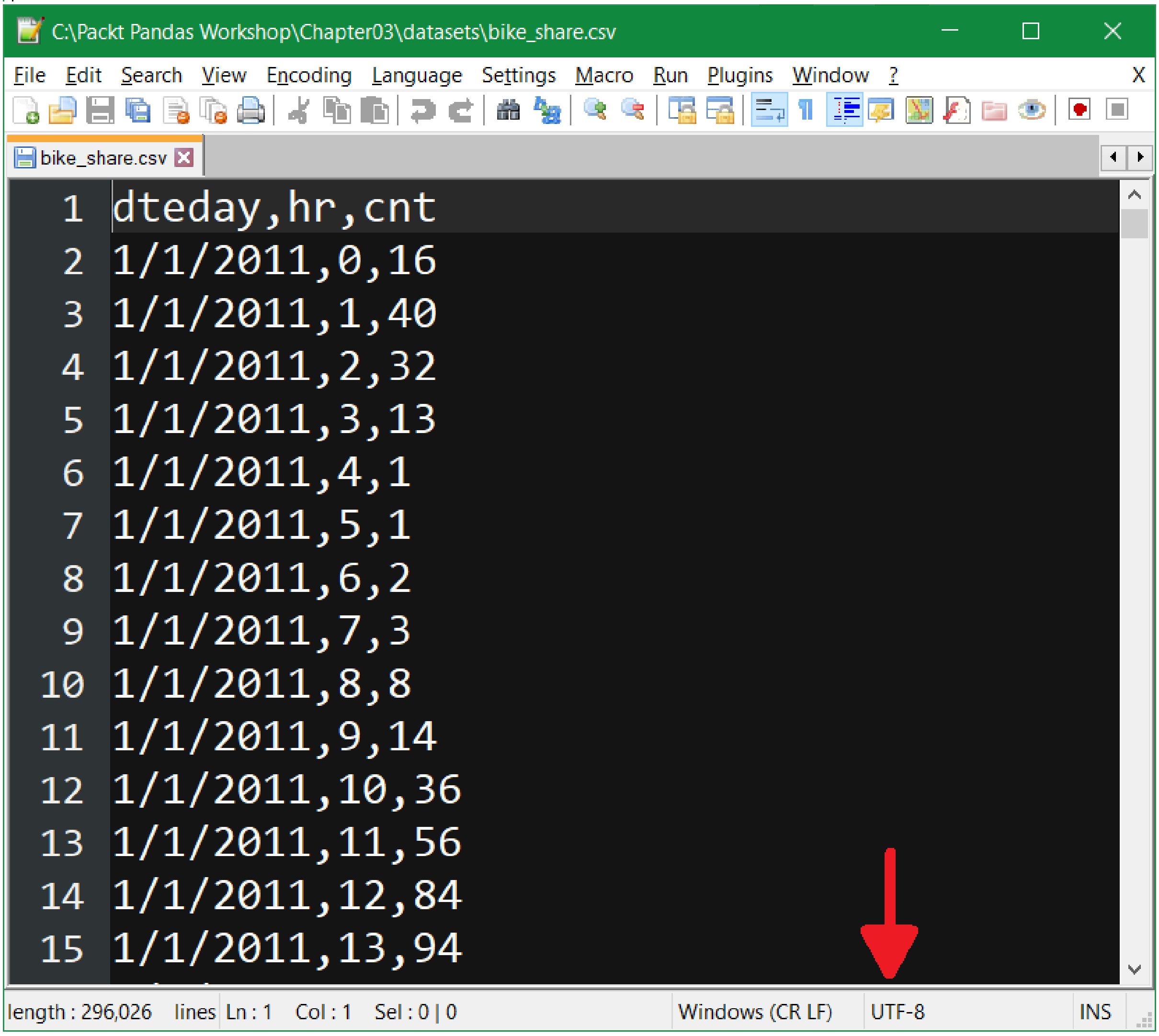
Out[7]:

	Name	Country	CapacityMWp	GenerationGWh p.a.	Sizekm2	Year	Ref
0	Pavagada Solar Park	India	2050	NaN	53	2017	[2][52][53]
1	Tengger Desert Solar Park	China	1547	NaN	43	2016	[54][55]
2	Bhadla Solar Park	India	1515	NaN	40	2017	[56][57][58]
3	Kurnool Ultra Mega Solar Park	India	1000	NaN	24	2017	[59]
4	Datong Solar Power Top Runner Base	China	1000	NaN	NaN	2016	[60][61][62]
5	Longyangxia Dam Solar Park	China	850	NaN	23	2015	[63][64][65][66][67]
6	Rewa Ultra Mega Solar	India	750	NaN	NaN	2018	[68]
7	Kamuthi Solar Power Project	India	648	NaN	10.1	2016	[69][70]
8	Solar Star (I and II)	United States	579	1664.0	13	2015	[71][72]
9	Topaz Solar Farm	United States	550	1301.0	24.6[73]	2014	[74][75][76]

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]



00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 43 75 73 74 6F 6D 65 72 5F 4E 75 6D 62 65 72 2C Customer_Number,
00000010 43 6F 6D 70 61 6E 79 2C 43 69 74 79 2C 53 74 61 Company,City,Sta
00000020 74 65 0D 0A 31 39 38 32 38 2C 52 65 70 74 69 6C te..19828,Reptil
00000030 65 20 44 65 73 65 72 74 2C 42 61 6C 74 69 6D 6F e Desert,Baltimo
00000040 72 65 2C 4D 44 0D 0A 31 39 31 38 36 2C 41 71 75 re,MD..19186,Aqu
00000050 61 74 69 63 20 46 72 69 65 6E 64 73 2C 53 61 6E atic Friends,San
00000060 20 42 65 72 6E 61 64 69 6E 6F 2C 43 41 0D 0A 31 Bernadino,CA..1
00000070 39 39 34 38 2C 41 72 61 63 68 6E 61 70 68 69 6C 9948,Arachnaphil
00000080 69 61 2C 4E 65 77 61 72 6B 2C 4E 4A 0D 0A 31 39 ia,Newark,NJ..19
00000090 36 39 37 2C 53 6F 6E 67 62 69 72 64 20 4D 75 73 697,Songbird Mus
000000A0 69 63 20 53 74 6F 72 65 2C 4D 65 6D 70 68 69 73 ic Store,Memphis
000000B0 2C 54 58 0D 0A 31 39 37 38 38 2C 45 71 75 65 73 ,TX..19788,Eques
000000C0 74 72 69 61 6E 20 50 61 6C 61 63 65 2C 43 6F 6C trian Palace,Col
000000D0 6F 72 61 64 6F 20 53 70 72 69 6E 67 73 2C 43 4F orado Springs,CO
000000E0 0D 0A 31 39 31 31 35 2C 4A 75 73 74 20 53 68 6F ..19115,Just Sho
000000F0 77 20 44 6F 67 73 2C 42 61 74 6F 6E 20 52 6F 75 w Dogs,Baton Rou
00000100 67 65 2C 4C 41 0D 0A 31 39 36 37 38 2C 4D 79 20 ge,LA..19678,My
00000110 46 61 76 6F 72 69 74 65 20 42 75 74 74 65 72 66 Favorite Butterf
00000120 6C 79 2C 4C 75 62 62 6F 63 6B 2C 54 58 0D 0A ly,Lubbock,TX..



C:\Packt Pandas Workshop\Chapter03\datasets\bike_share.csv
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
bike_share.csv
1 dteday,hr,cnt
2 1/1/2011,0,16
3 1/1/2011,1,40
4 1/1/2011,2,32
5 1/1/2011,3,13
6 1/1/2011,4,1
7 1/1/2011,5,1
8 1/1/2011,6,2
9 1/1/2011,7,3
10 1/1/2011,8,8
11 1/1/2011,9,14
12 1/1/2011,10,36
13 1/1/2011,11,56
14 1/1/2011,12,84
15 1/1/2011,13,94
length : 296,026 lines Ln : 1 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS

Out[2]:

	dteday	hr	cnt
0	1/1/2011	0	16
1	1/1/2011	1	40
2	1/1/2011	2	32
3	1/1/2011	3	13
4	1/1/2011	4	1
...	...	...	...
17374	12/31/2012	19	119
17375	12/31/2012	20	89
17376	12/31/2012	21	90
17377	12/31/2012	22	61
17378	12/31/2012	23	49

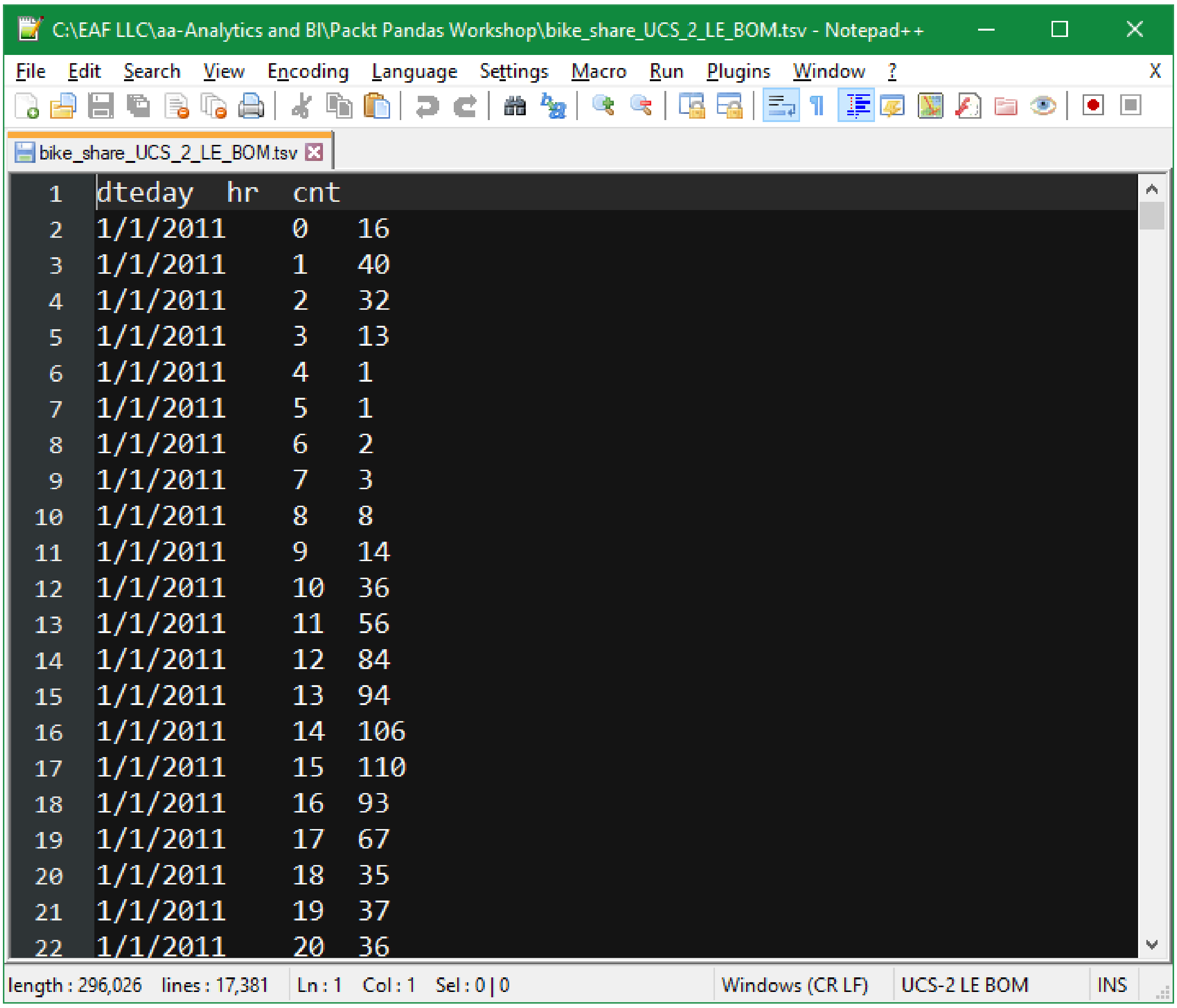
17379 rows × 3 columns

```
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xff in position 0: invalid start byte
```

Out[8]:

	dteday	hr	cnt
0	1/1/2011	0	16
1	1/1/2011	1	40
2	1/1/2011	2	32
3	1/1/2011	3	13
4	1/1/2011	4	1
...	...	...	...
17374	12/31/2012	19	119
17375	12/31/2012	20	89
17376	12/31/2012	21	90
17377	12/31/2012	22	61
17378	12/31/2012	23	49

17379 rows × 3 columns


C:\EAF LLC\aa-Analytics and BI\Packt Pandas Workshop\bike_share_UCS_2_LE_BOM.tsv - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
bike_share_UCS_2_LE_BOM.tsv
dteday hr cnt
1/1/2011 0 16
1/1/2011 1 40
1/1/2011 2 32
1/1/2011 3 13
1/1/2011 4 1
1/1/2011 5 1
1/1/2011 6 2
1/1/2011 7 3
1/1/2011 8 8
1/1/2011 9 14
1/1/2011 10 36
1/1/2011 11 56
1/1/2011 12 84
1/1/2011 13 94
1/1/2011 14 106
1/1/2011 15 110
1/1/2011 16 93
1/1/2011 17 67
1/1/2011 18 35
1/1/2011 19 37
1/1/2011 20 36
length : 296,026 lines : 17,381 Ln : 1 Col : 1 Sel : 0 | 0
Windows (CR LF)
UCS-2 LE BOM
INS

Out[10]:

	dteday	hr	cnt
0	1/1/2011	0	16
1	1/1/2011	1	40
2	1/1/2011	2	32
3	1/1/2011	3	13
4	1/1/2011	4	1
...	...	...	...
17374	12/31/2012	19	119
17375	12/31/2012	20	89
17376	12/31/2012	21	90
17377	12/31/2012	22	61
17378	12/31/2012	23	49

17379 rows × 3 columns

```
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xff in position 0: invalid start byte
```

Out[3]:

	age\tsex\ton_thyroxine\tquery_on_thyroxine\ton_antithyroid_medicatio ... 4U\tFTI_measured\tFTI\tTBG_measured\tTBG\treferral_source\tresult\tvalue
0	41\tF\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\t...
1	23\tF\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\t...
2	46\tM\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\t...
3	70\tF\tt\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\t...
4	70\tF\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\t...
...	...
2795	70\tM\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\t...
2796	73\tM\tf\tt\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\t...
2797	75\tM\tf\tf\tf\tf\tf\tf\tf\tt\tf\tf\tf\tf\t...
2798	60\tF\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\t...
2799	81\tF\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\tf\t...

2800 rows × 1 columns

Out[4]:

	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_medication	sick	pregnant	thy
0	41	F	f	f	f	f	f	
1	23	F	f	f	f	f	f	
2	46	M	f	f	f	f	f	
3	70	F	t	f	f	f	f	
4	70	F	f	f	f	f	f	
...	...	...	...	...	...	...	...	
2795	70	M	f	f	f	f	f	
2796	73	M	f	t	f	f	f	
2797	75	M	f	f	f	f	f	
2798	60	F	f	f	f	f	f	
2799	81	F	f	f	f	f	f	

2800 rows × 31 columns

Customers		
Customer_ID	Address	Credit_Limit
02349	1324 S. My Way	10000
13795	2987 West St.	13000
93298	3756 East Ave.	9500
39873	12 North Gary Ln.	13500
...		

Orders			
Order_ID	Customer_ID	Item	Qty
347991	02349	23-0495	1000
347991	02349	17-0311	200
269981	13795	99-0000	1
459812	93298	45-2391	237
...			

select * from RENTALS | Enter a SQL expression to filter results (use Ctrl+Space)

	index	dteday	hr	cnt
1	0	1/1/2011	0	16
2	1	1/1/2011	1	40
3	2	1/1/2011	2	32
4	3	1/1/2011	3	13
5	4	1/1/2011	4	1
6	5	1/1/2011	5	1
7	6	1/1/2011	6	2
8	7	1/1/2011	7	3
9	8	1/1/2011	8	8
10	9	1/1/2011	9	14
11	10	1/1/2011	10	36
12	11	1/1/2011	11	56
13	12	1/1/2011	12	84
14	13	1/1/2011	13	94
15	14	1/1/2011	14	106
16	15	1/1/2011	15	110
17	16	1/1/2011	16	93

Save | Cancel | Script

200 | 200+ | Rows: 1

Out[5]:

'[["NAME","state","county"],\n["Sebastian County, Arkansas","05","131"],\n["Sevier County, Arkansas","05","133"],\n["Sharp County, Arkansas","05","135"],\n["Stone County, Arkansas","05","137"],\n["Union County, Arkansas","05","139"],\n["Van Buren County, Arkansas","05","141"],\n["Washington County, Arkansas","05","143"],\n["White County, Arkansas","05","145"],\n["Yell County, Arkansas","05","149"],\n["Colusa County, California","06","011"],\n["Butte County, California","06","007"],\n["Alameda County, California","06","001"],\n["Alpine County, California","06","003"],\n["Amador County, California","06","005"],\n["Calaveras County, California","06","009"],\n["Contra Costa County, California","06","013"],\n["Del Norte County, California","06","015"],\n["Kings County, California","06","031"],\n["Glenn County, California","06","021"],\n["Humboldt County, California","06","023"],\n["Imperial County, California","06","025"],\n["El Dorado County, California","06","017"],\n["Fresno County, California","06","019"],\n["Inyo County, California","06","027"],\n["Kern County, California","06","029"],\n["Mariposa County, California","06","043"],\n["Lake County, California","06","033"],\n["Lassen County, California","06","035"],\n["Los Angeles County, California","06","037"],\n["Madera County, California","06","039"],\n["Marin County,

```
Out[10]: [['NAME', 'state', 'county'],
 ['Sebastian County, Arkansas', '05', '131'],
 ['Sevier County, Arkansas', '05', '133'],
 ['Sharp County, Arkansas', '05', '135'],
 ['Stone County, Arkansas', '05', '137'],
 ['Union County, Arkansas', '05', '139'],
 ['Van Buren County, Arkansas', '05', '141'],
 ['Washington County, Arkansas', '05', '143'],
 ['White County, Arkansas', '05', '145'],
 ['Yell County, Arkansas', '05', '149'],
 ['Colusa County, California', '06', '011'],
 ['Butte County, California', '06', '007'],
 ['Alameda County, California', '06', '001'],
 ['Alpine County, California', '06', '003'],
 ['Amador County, California', '06', '005'],
 ['Calaveras County, California', '06', '009'],
 ['Contra Costa County, California', '06', '013'],
 ['Del Norte County, California', '06', '015'],
```

Out[11]:

	0	1	2
0	NAME	state	county
1	Sebastian County, Arkansas	05	131
2	Sevier County, Arkansas	05	133
3	Sharp County, Arkansas	05	135
4	Stone County, Arkansas	05	137
...	...	...	...
3217	Eau Claire County, Wisconsin	55	035
3218	Florence County, Wisconsin	55	037
3219	Fond du Lac County, Wisconsin	55	039
3220	Forest County, Wisconsin	55	041
3221	Jefferson County, Wisconsin	55	055

3222 rows × 3 columns

Out[6]:

	County	state_code	county_code
1	Sebastian County, Arkansas	05	131
2	Sevier County, Arkansas	05	133
3	Sharp County, Arkansas	05	135
4	Stone County, Arkansas	05	137
5	Union County, Arkansas	05	139
...	...	...	...
3217	Eau Claire County, Wisconsin	55	035
3218	Florence County, Wisconsin	55	037
3219	Fond du Lac County, Wisconsin	55	039
3220	Forest County, Wisconsin	55	041
3221	Jefferson County, Wisconsin	55	055

3221 rows × 3 columns

Large onshore wind farms

Wind farm	Capacity (MW)	Country	Refs
Gansu Wind Farm	7,965	China	[23][24]
Muppandal wind farm	1,500	India	[25]
Alta (Oak Creek-Mojave)	1,320	United States	[26]
Jaisalmer Wind Park	1,064	India	[27]
Shepherds Flat Wind Farm	845	United States	[28]
Roscoe Wind Farm	782	United States	
Horse Hollow Wind Energy Center	736	United States	[29][30]
Capricorn Ridge Wind Farm	662	United States	[29][30]
Fântânele-Cogealac Wind Farm	600	Romania	[31]
Fowler Ridge Wind Farm	600	United States	[32]
Whitelee Wind Farm	539	United Kingdom	[33]

```html
<!DOCTYPE html>
<html class="client-nojs" lang="en" dir="ltr">
<head>
<meta charset="UTF-8"/>
<title>Wind power - Wikipedia</title>
<script>document.documentElement.className="client-js";RLCONF={"wgBreakFrames":!1,"wgSeparatorTransformTable":["",""],"wgDigitTransformTable":["",
"Articles with permanently dead external links","CS1 maint: location","Articles with Spanish-language sources (es)","Articles with short descripti
"wgIsProbablyEditable":!1,"wgRelevantPageIsProbablyEditable":!1,"wgRestrictionEdit":["autoconfirmed"],"wgRestrictionMove":["sysop"],"wgFlaggedRevs
"#87ceeb","#a4a1a2"]}],"version":2,"marks":[{"type":"line","properties":{"hover":{"stroke":{"value":"red"}},"update":{"stroke":{"scale":"color","f
"format":{"parse":{"y":"number","x":"date"},"type":"json"},"name":"chart","values":[{"y":6.1,"series":"","x":"1996"},{"y":7.6,"series":"","x":"199
"wgULSPosition":"interlanguage","wgGENewcomerTasksGuidanceEnabled":!0,"wgGEAskQuestionEnabled":!1,"wgGELinkRecommendationsFrontendEnabled":!1,"wgW
"jquery.makeCollapsible","mediawiki.toc","skins.vector.legacy.js","ext.gadget.ReferenceTooltips","ext.gadget.charinsert","ext.gadget.extra-toolbar
<script>(RLQ=window.RLQ||[]).push(function(){mw.loader.implement("user.options@1hzgi",function($,jQuery,require,module){/*@nomin*/mw.user.tokens.s
});});</script>
<link rel="stylesheet" href="/w/load.php?lang=en&amp;modules=ext.cite.styles%7Cext.graph.styles%7Cext.math.styles%7Cext.timeline.styles%7Cext.tmh.
<script async="" src="/w/load.php?lang=en&amp;modules=startup&amp;only=scripts&amp;raw=1&amp;skin=vector"></script>
<meta name="ResourceLoaderDynamicStyles" content=""/>
<link rel="stylesheet" href="/w/load.php?lang=en&amp;modules=site.styles&amp;only=styles&amp;skin=vector"/>
<meta name="generator" content="MediaWiki 1.37.0-wmf.4"/>
<meta name="referrer" content="origin"/>
<meta name="referrer" content="origin-when-crossorigin"/>
<meta name="referrer" content="origin-when-cross-origin"/>
<meta property="og:image" content="https://upload.wikimedia.org/wikipedia/commons/thumb/e/e0/Wind_power_plants_in_Xinjiang%2C_China.jpg/1200px-Win
<meta property="og:title" content="Wind power - Wikipedia"/>
<meta property="og:type" content="website"/>
<link rel="preconnect" href="//upload.wikimedia.org"/>
<link rel="alternate" media="only screen and (max-width: 720px)" href="//en.m.wikipedia.org/wiki/Wind_power"/>
<link rel="apple-touch-icon" href="/static/apple-touch/wikipedia.png"/>
<link rel="shortcut icon" href="/static/favicon/wikipedia.ico"/>
<link rel="search" type="application/opensearchdescription+xml" href="/w/opensearch_desc.php" title="Wikipedia (en)"/>
<link rel="EditURI" type="application/rsd+xml" href="//en.wikipedia.org/w/api.php?action=rsd"/>
<link rel="license" href="//creativecommons.org/licenses/by-sa/3.0/"/>
<link rel="canonical" href="https://en.wikipedia.org/wiki/Wind_power"/>
<link rel="dns-prefetch" href="//login.wikimedia.org"/>
<link rel="dns-prefetch" href="//meta.wikimedia.org" />
</head>
<body class="mediawiki ltr sitedir-ltr mw-hide-empty-elt ns-0 ns-subject page-Wind_power rootpage-Wind_power skin-vector action-view skin-vector-l
<div id="mw-head-base" class="noprint"></div>
```

World's largest offshore wind farms

Wind farm	Capacity (MW)	Country	Turbines and model	Commissioned	Refs
Walney Extension	659	United Kingdom	47 x Vestas 8MW 40 x Siemens Gamesa 7MW	2018	[48]
London Array	630	United Kingdom	175 × Siemens SWT-3.6	2012	[49][50][51]
Gemini Wind Farm	600	The Netherlands	150 × Siemens SWT-4.0	2017	[52]
Gwynt y Môr	576	United Kingdom	160 × Siemens SWT-3.6 107	2015	[53]
Greater Gabbard	504	United Kingdom	140 × Siemens SWT-3.6	2012	[54]
Anholt	400	Denmark	111 × Siemens SWT-3.6–120	2013	[55]
BARD Offshore 1	400	Germany	80 BARD 5.0 turbines	2013	[56]

Out[11]:

	Wind farm	Capacity (MW)	Country	Turbines and model	Commissioned	Refs
0	Walney Extension	659	United Kingdom	47 x Vestas 8MW 40 x Siemens Gamesa 7MW	2018	[48]
1	London Array	630	United Kingdom	175 × Siemens SWT-3.6	2012	[49][50][51]
2	Gemini Wind Farm	600	The Netherlands	150 × Siemens SWT-4.0	2017	[52]
3	Gwynt y Môr	576	United Kingdom	160 × Siemens SWT-3.6 107	2015	[53]
4	Greater Gabbard	504	United Kingdom	140 × Siemens SWT-3.6	2012	[54]
5	Anholt	400	Denmark	111 × Siemens SWT-3.6–120	2013	[55]
6	BARD Offshore 1	400	Germany	80 BARD 5.0 turbines	2013	[56]

data.cityofnewyork.us/api/views/825b-niea/rows.xml

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<response>
 ▼<row>
  ▼<row _id="row-yvru.xsvq_qzbq" _uuid="00000000-0000-0000-1B32-87B29F69422E" _position="0" _address="https://data.cityofnewyork.us/resource/_825b-niea/row-yvru.xsvq_qzbq">
     <grade>3</grade>
     <year>2006</year>
     <category>Asian</category>
     <number_tested>9768</number_tested>
     <mean_scale_score>700</mean_scale_score>
     <level_1_1>243</level_1_1>
     <level_1_2>2.5</level_1_2>
     <level_2_1>543</level_2_1>
     <level_2_2>5.6</level_2_2>
     <level_3_1>4128</level_3_1>
     <level_3_2>42.3</level_3_2>
     <level_4_1>4854</level_4_1>
     <level_4_2>49.7</level_4_2>
     <level_3_4_1>8982</level_3_4_1>
     <level_3_4_2>92.0</level_3_4_2>
   </row>
  ▼<row _id="row-q8z8.q7b3.3ppa" _uuid="00000000-0000-0000-D9CE-B1F89A0D1307" _position="0" _address="https://data.cityofnewyork.us/resource/_825b-niea/row-q8z8.q7b3.3ppa">
     <grade>4</grade>
     <year>2006</year>
     <category>Asian</category>
     <number_tested>9973</number_tested>
     <mean_scale_score>699</mean_scale_score>
     <level_1_1>294</level_1_1>
     <level_1_2>2.9</level_1_2>
     <level_2_1>600</level_2_1>
```


Out[8]:

	@_id	@_uuid	@_position	@_address	grade	year	category	number_tested	mean_scale_score	le
0	row-yvru.xsvq_qzbq	00000000-0000-0000-1B32-87B29F69422E	0	https://data.cityofnewyork.us/resource/_825b-n...	3	2006	Asian	9768	700	
1	row-q8z8.q7b3.3ppa	00000000-0000-0000-D9CE-B1F89A0D1307	0	https://data.cityofnewyork.us/resource/_825b-n...	4	2006	Asian	9973	699	
2	row-i23x-4prc-46fj	00000000-0000-0000-C9EE-2418870B5F93	0	https://data.cityofnewyork.us/resource/_825b-n...	5	2006	Asian	9852	691	
3	row-7u9v-dwwy.fhw3	00000000-0000-0000-17FD-7D50A499A0E1	0	https://data.cityofnewyork.us/resource/_825b-n...	6	2006	Asian	9606	682	
4	row-64kf_k4ma_4zgq	00000000-0000-0000-6A3C-917EFD40527E	0	https://data.cityofnewyork.us/resource/_825b-n...	7	2006	Asian	9433	671	
...	...	...	...	...	...	...	...	...	...	
163	row-i6yz_wbge_khnu	00000000-0000-0000-11E2-D5CA802D0782	0	https://data.cityofnewyork.us/resource/_825b-n...	5	2011	White	10808	699	


```
▼<response>
 ▼<row>
  ▼<row _id="row-yvru.xsvq_qzbq" _uuid="00000000-0000-0000-1B32-87B29F69422E" _position="0" _address="https://data.cit
     <grade>3</grade>
     <year>2006</year>
     <category>Asian</category>
     <number_tested>9768</number_tested>
     <mean_scale_score>700</mean_scale_score>
     <level_1_1>243</level_1_1>
     <level_1_2>2.5</level_1_2>
     <level_2_1>543</level_2_1>
     <level_2_2>5.6</level_2_2>
     <level_3_1>4128</level_3_1>
     <level_3_2>42.3</level_3_2>
     <level_4_1>4854</level_4_1>
     <level_4_2>49.7</level_4_2>
     <level_3_4_1>8982</level_3_4_1>
     <level_3_4_2>92.0</level_3_4_2>
   </row>
  ▼<row _id="row-q8z8.q7b3.3ppa" _uuid="00000000-0000-0000-D9CE-B1F89A0D1307" _position="0" _address="https://data.cit
     <grade>4</grade>
     <year>2006</year>
     <category>Asian</category>
     <number_tested>9973</number_tested>
     <mean_scale_score>699</mean_scale_score>
     <level_1_1>294</level_1_1>
```

parameter (= default value)	meaning
io	the object containing the Excel data--can be a path etc.
sheet_name = 0	defaults to the first sheet
header = 0	what row, if any, contains the column names?
usecols = None	what columns to read, if not all; can be a list of letters or integers etc.

Out[20]:

	time	s1	s2	s3
0	0.95924	0.234046	3.514755	0.447823
1	0.96424	0.171669	4.837437	0.495071
2	0.96924	0.271542	4.673110	0.383604
3	0.97424	0.057020	3.048180	0.193946
4	0.97924	0.062937	5.631988	0.338150
...	...	...	...	...
10669	54.30424	15.066911	7.506722	29.028388
10670	54.30924	17.264761	10.195260	24.272862
10671	54.31424	9.744161	7.956116	10.244286
10672	54.31924	1.722525	10.254374	2.513277
10673	54.32424	10.190016	11.267764	0.942601

10674 rows × 4 columns

	A	B	C	D
1	time	s1	s2	s3
2	0.95924	0.23405	3.51476	0.44782
3	0.96424	0.17167	4.83744	0.49507
4	0.96924	0.27154	4.67311	0.3836
5	0.97424	0.05702	3.04818	0.19395
6	0.97924	0.06294	5.63199	0.33815
7	0.98424	0.19886	5.75142	0.37132
8	0.98924	0.11517	5.97284	0.27648
9	0.99424	0.08889	2.25377	0.18153
10	0.99924	0.00892	3.61314	0.3012
11	1.00424	0.22749	3.60927	0.4957
12	1.00924	0.18625	2.68281	0.3985
13	1.01424	0.14093	4.63483	0.38326
14	1.01924	0.14895	5.01276	0.25547

sensor_data

Out[10]:

	YEAR	Y	W	R	L	K
0	1948.0	1.214	0.243	0.1454	1.415	0.612
1	1949.0	1.354	0.260	0.2181	1.384	0.559
2	1950.0	1.569	0.278	0.3157	1.388	0.573
3	1951.0	1.948	0.297	0.3940	1.550	0.564
4	1952.0	2.265	0.310	0.3559	1.802	0.574

Out[11]:

	y	x1	x2	x3
0	19.5	43.1	29.1	11.9
1	24.7	49.8	28.2	22.8
2	30.7	51.9	37.0	18.7
3	29.8	54.3	31.1	20.1
4	19.1	42.2	30.9	12.9

```
data:
   yy1  y1          wgt  hhsex  age  agecl  educ  edcl  married  kids  ...  \
0    1  11  6119.779308      2   75      6    12     4        2     0  ...
1    1  12  4712.374912      2   75      6    12     4        2     0  ...

   nwcat  inccat  assetcat  ninccat  ninc2cat  nwpctlecat  incpctlecat  \
0      5       3         6        3         2          10            6
1      5       3         6        3         1          10            5

   nincpctlecat  incqrtcat  nincqrtcat
0             6          3           3
1             5          2           2

[2 rows x 351 columns]
data2:
   yy1  y1          wgt  hhsex  age  agecl  educ  edcl  married  kids  ...  \
0    1  11  6119.779308      2   75      6    12     4        2     0  ...
1    1  12  4712.374912      2   75      6    12     4        2     0  ...

   nwcat  inccat  assetcat  ninccat  ninc2cat  nwpctlecat  incpctlecat  \
0      5       3         6        3         2          10            6
1      5       3         6        3         1          10            5

   nincpctlecat  incqrtcat  nincqrtcat
0             6          3           3
1             5          2           2

[2 rows x 351 columns]
differences between rscfp2019 and rscfp2019_write:
 Empty DataFrame
Columns: []
Index: []
```

Out[21]:

	time	data
10	0.10	0.036951
11	0.11	0.040645
12	0.12	0.044337
13	0.13	0.048029
14	0.14	0.051721

Out[27]:

	name
0	Customers
1	Invoices

Out[10]:

	Customer_Number	Company	City	State
0	15846	Pet Radio	Minneapolis	MN
1	13197	Just Pets	Columbus	OH
2	11154	Love Strays	Pittsburgh	PA
3	15540	WebPet	Mesa	AZ
4	18397	Pet-ng-Zoo	San Antonio	TX
5	17293	Pet Fud	St. Paul	MN
6	19977	Canine Cravings	Henderson	NV
7	15238	Stock Ur Pet	Stockton	CA
8	15217	Kittle Lullaby	New Orleans	LA
9	17114	Big Dogs Only	Anchorage	AK
10	18448	K9s4Ever	Dallas	TX
11	13388	Bird Sanctuary	Newark	NJ
12	11485	GrrrtoPurr	Plano	TX

Out[13]:

	Customer_Number	Company	City	State
0	18397	Pet-ng-Zoo	San Antonio	TX
1	18448	K9s4Ever	Dallas	TX
2	11485	GrrrtoPurr	Plano	TX

Out[14]:

	Customer_Number	Company	City	State
4	18397	Pet-ng-Zoo	San Antonio	TX
10	18448	K9s4Ever	Dallas	TX
12	11485	GrrrtoPurr	Plano	TX

```
   index       Date  Customer_Number        Invoice   Amount
0      0  2/20/2020            18397  2020022018397  1038.95
1      1  2/25/2020            17114  2020022517114  1523.97
2      2  2/25/2020            15846  2020022515846  1535.56
    index       Date  Customer_Number        Invoice   Amount
35     35  3/19/2020            17114  2020031917114  1041.22
36     36  3/19/2020            13388  2020031913388  1043.63
37     37  3/24/2020            15217  2020032415217  1542.85
```

Out[5]:

	Date	Customer_Number	Invoice	Amount
0	3/24/2020	15846	2020032415846	1355.73
1	3/24/2020	17293	2020032417293	1375.67
2	3/24/2020	18448	2020032418448	1415.38
3	3/24/2020	11485	2020032411485	1025.46
4	3/25/2020	11154	2020032511154	1245.01
5	3/25/2020	13388	2020032513388	1055.32
6	3/25/2020	13197	2020032513197	1105.15
7	3/25/2020	15217	2020032515217	1495.33
8	3/26/2020	17114	2020032617114	1185.30
9	3/26/2020	13197	2020032613197	1290.44
10	3/26/2020	15238	2020032615238	1170.75
11	3/26/2020	18397	2020032618397	1330.36

Out[6]:

	Date	Customer_Number	Invoice	Amount
38	3/24/2020	15846	2020032415846	1355.73
39	3/24/2020	17293	2020032417293	1375.67
40	3/24/2020	18448	2020032418448	1415.38
41	3/24/2020	11485	2020032411485	1025.46
42	3/25/2020	11154	2020032511154	1245.01
43	3/25/2020	13388	2020032513388	1055.32
44	3/25/2020	13197	2020032513197	1105.15
45	3/25/2020	15217	2020032515217	1495.33
46	3/26/2020	17114	2020032617114	1185.30
47	3/26/2020	13197	2020032613197	1290.44
48	3/26/2020	15238	2020032615238	1170.75
49	3/26/2020	18397	2020032618397	1330.36

```
   index       Date  Customer_Number        Invoice   Amount
0      0  2/20/2020            18397  2020022018397  1038.95
1      1  2/25/2020            17114  2020022517114  1523.97
2      2  2/25/2020            15846  2020022515846  1535.56
3      3  2/25/2020            15540  2020022515540  1568.95
4      4  2/26/2020            18448  2020022618448  1509.51
    index       Date  Customer_Number        Invoice   Amount
45     45  3/25/2020            15217  2020032515217  1495.33
46     46  3/26/2020            17114  2020032617114  1185.30
47     47  3/26/2020            13197  2020032613197  1290.44
48     48  3/26/2020            15238  2020032615238  1170.75
49     49  3/26/2020            18397  2020032618397  1330.36
```

Out[2]:

	index	Customer_Number	Company	City	State
0	None	15846	Pet Radio	Minneapolis	MN
1	None	13197	Just Pets	Columbus	OH
2	None	11154	Love Strays	Pittsburgh	PA
3	None	15540	WebPet	Mesa	AZ
4	None	18397	Pet-ng-Zoo	San Antonio	TX
5	None	17293	Pet Fud	St. Paul	MN
6	None	19977	Canine Cravings	Henderson	NV
7	None	15238	Stock Ur Pet	Stockton	CA
8	None	15217	Kittle Lullaby	New Orleans	LA
9	None	17114	Big Dogs Only	Anchorage	AK
10	None	18448	K9s4Ever	Dallas	TX
11	None	13388	Bird Sanctuary	Newark	NJ
12	None	11485	GrrrtoPurr	Plano	TX

Out[2]:

	Customer_Number	Company	City	State
0	19828	Reptile Desert	Baltimore	MD
1	19186	Aquatic Friends	San Bernadino	CA
2	19948	Arachnaphilia	Newark	NJ
3	19697	Songbird Music Store	Memphis	TX
4	19788	Equestrian Palace	Colorado Springs	CO
5	19115	Just Show Dogs	Baton Rouge	LA
6	19678	My Favorite Butterfly	Lubbock	TX

Out[6]:

	index	Customer_Number	Company	City	State
0	None	15846	Pet Radio	Minneapolis	MN
1	None	13197	Just Pets	Columbus	OH
2	None	11154	Love Strays	Pittsburgh	PA
3	None	15540	WebPet	Mesa	AZ
4	None	18397	Pet-ng-Zoo	San Antonio	TX
5	None	17293	Pet Fud	St. Paul	MN
6	None	19977	Canine Cravings	Henderson	NV
7	None	15238	Stock Ur Pet	Stockton	CA
8	None	15217	Kittle Lullaby	New Orleans	LA
9	None	17114	Big Dogs Only	Anchorage	AK
10	None	18448	K9s4Ever	Dallas	TX
11	None	13388	Bird Sanctuary	Newark	NJ
12	None	11485	GrrrtoPurr	Plano	TX
13	None	19828	Reptile Desert	Baltimore	MD
14	None	19186	Aquatic Friends	San Bernadino	CA
15	None	19948	Arachnaphilia	Newark	NJ
16	None	19697	Songbird Music Store	Memphis	TX
17	None	19788	Equestrian Palace	Colorado Springs	CO
18	None	19115	Just Show Dogs	Baton Rouge	LA
19	None	19678	My Favorite Butterfly	Lubbock	TX

Chapter 4: Pandas Data Types

	Customer ID	Customer Name	2018 Revenue	2019 Revenue	Growth	Start Year	Start Month	Start Day	New Customer
0	1001.0	Pandas Banking	€235000	€248000	5.5%	2013	3	10	0
1	1002.0	Pandas Grocery	€196000	€205000	4.5%	2016	4	30	0
2	1003.0	Pandas Telecom	€167000	€193000	15.5%	2010	11	24	0
3	1004.0	Pandas Transport	€79000	€90000	13.9%	2018	1	15	1
4	1005.0	Pandas Insurance	€241000	€264000	9.5%	2009	6	1	0

```
0    €235000€248000
1    €196000€205000
2    €167000€193000
3      €79000€90000
4    €241000€264000
dtype: object
```

```
Customer ID      float64
Customer Name     object
2018 Revenue      object
2019 Revenue      object
Growth            object
Start Year         int64
Start Month        int64
Start Day          int64
New Customer       int64
dtype: object
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Customer ID    5 non-null      float64
 1   Customer Name  5 non-null      object
 2   2018 Revenue   5 non-null      object
 3   2019 Revenue   5 non-null      object
 4   Growth         5 non-null      object
 5   Start Year     5 non-null      int64
 6   Start Month    5 non-null      int64
 7   Start Day      5 non-null      int64
 8   New Customer   5 non-null      int64
dtypes: float64(1), int64(4), object(4)
memory usage: 488.0+ bytes
```

```
0    1001
1    1002
2    1003
3    1004
4    1005
Name: Customer ID, dtype: int32
```

```
0    235000
1    196000
2    167000
3     79000
4    241000
Name: 2018 Revenue, dtype: int64
```

```
0    248000
1    205000
2    193000
3     90000
4    264000
Name: 2019 Revenue, dtype: int64
```

```
0     5.5
1     4.5
2    15.5
3    13.9
4     9.5
Name: Growth, dtype: float64
```

```
0   2013-03-10
1   2016-04-30
2   2010-11-24
3   2018-01-15
4   2009-06-01
Name: Starting Date, dtype: datetime64[ns]
```

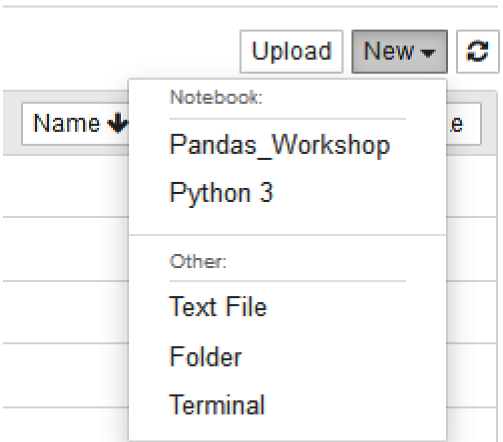
```
0    False
1    False
2    False
3     True
4    False
Name: New Customer, dtype: bool
```

```
0      Pandas Banking
1      Pandas Grocery
2      Pandas Telecom
3    Pandas Transport
4    Pandas Insurance
Name: Customer Name, dtype: category
Categories (5, object): [Pandas Banking, Pandas Grocery, Pandas Insurance, Pandas Telecom, Pandas Transport]
```

```
Customer ID                int32
Customer Name           category
2018 Revenue               int64
2019 Revenue               int64
Growth                    object
year                       int64
month                      int64
day                        int64
New Customer                bool
Starting Date     datetime64[ns]
dtype: object
```

```
0    483000
1    401000
2    360000
3    169000
4    505000
dtype: int64
```

```
0   -2732 days
1   -1585 days
2   -3569 days
3    -960 days
4   -4110 days
Name: Starting Date, dtype: timedelta64[ns]
```



	Receipt Id	Date of Purchase	Product Name	Product Weight	Total Price	Retail shop name
0	10001	24/05/20	Wheat	4.8lb	€17	Fline Store
1	10002	05/05/20	Fruit Juice	3.1lb	€19	Dello Superstore
2	10003	27/04/20	Vegetables	1.2lb	€15	Javies Retail
3	10004	05/05/20	Oil	3.1lb	€17	Javies Retail
4	10005	27/04/20	Wheat	4.8lb	€13	Javies Retail

	Receipt Id	Date of Purchase	Product Name	Product Weight	Total Price	Retail shop name
99995	109996	24/05/20	Oil	4.8lb	€25	Visco Retail
99996	109997	20/04/20	Rice	3.1lb	€12	Kelly Superstore
99997	109998	08/01/20	Fruit Juice	2.7lb	€24	Dello Superstore
99998	109999	05/05/20	Butter	3.1lb	€22	Dello Superstore
99999	110000	17/04/20	Bread	4.4lb	€27	Visco Retail

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 6 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Receipt Id        100000 non-null  int64
 1   Date of Purchase  100000 non-null  object
 2   Product Name      100000 non-null  object
 3   Product Weight    100000 non-null  object
 4   Total Price       100000 non-null  object
 5   Retail shop name  100000 non-null  object
dtypes: int64(1), object(5)
memory usage: 4.6+ MB
```

```
0        2020-05-24
1        2020-05-05
2        2020-04-27
3        2020-05-05
4        2020-04-27
            ...
99995    2020-05-24
99996    2020-04-20
99997    2020-01-08
99998    2020-05-05
99999    2020-04-17
Name: Date of Purchase, Length: 100000, dtype: datetime64[ns]
```

```
0        17
1        19
2        15
3        17
4        13
         ..
99995    25
99996    12
99997    24
99998    22
99999    27
Name: Total Price, Length: 100000, dtype: object
```

```
0        17.0
1        19.0
2        15.0
3        17.0
4        13.0
         ...
99995    25.0
99996    12.0
99997    24.0
99998    22.0
99999    27.0
Name: Total Price, Length: 100000, dtype: float64
```

```
0        4.8
1        3.1
2        1.2
3        3.1
4        4.8
        ...
99995    4.8
99996    3.1
99997    2.7
99998    3.1
99999    4.4
Name: Product Weight, Length: 100000, dtype: object
```

```
0        4.8
1        3.1
2        1.2
3        3.1
4        4.8
        ...
99995    4.8
99996    3.1
99997    2.7
99998    3.1
99999    4.4
Name: Product Weight, Length: 100000, dtype: float64
```

```
0              Wheat
1        Fruit Juice
2         Vegetables
3                Oil
4              Wheat
            ...
99995            Oil
99996           Rice
99997    Fruit Juice
99998         Butter
99999          Bread
Name: Product Name, Length: 100000, dtype: category
Categories (9, object): [Bread, Butter, Cheese, Fruit Juice, ..., Oil, Rice, Vegetables, Wheat]
```

```
0              Fline Store
1        Dello Superstore
2            Javies Retail
3            Javies Retail
4            Javies Retail
               ...
99995        Visco Retail
99996    Kelly Superstore
99997    Dello Superstore
99998    Dello Superstore
99999        Visco Retail
Name: Retail shop name, Length: 100000, dtype: category
Categories (8, object): [Dello Superstore, Fline Store, Javies Retail, Kanes Store, Kelly Superstore, Oldi Superstor
e, Rotero Retail, Visco Retail]
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 6 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Receipt Id        100000 non-null  int64
 1   Date of Purchase  100000 non-null  datetime64[ns]
 2   Product Name      100000 non-null  category
 3   Product Weight    100000 non-null  float64
 4   Total Price       100000 non-null  float64
 5   Retail shop name  100000 non-null  category
dtypes: category(2), datetime64[ns](1), float64(2), int64(1)
memory usage: 3.2 MB
```

	Customer ID	Customer Name	2018 Revenue	2019 Revenue	Growth	New Customer	Starting Date
0	NaN	Pandas Banking	235000.0	248000.0	5.5	0.0	2013-03-10
1	1002.0	Pandas Grocery	196000.0	205000.0	4.5	NaN	2016-04-30
2	1003.0	Pandas Telecom	NaN	193000.0	15.5	0.0	NaT
3	1004.0	Pandas Transport	79000.0	NaN	NaN	1.0	2018-01-15
4	1005.0	NaN	241000.0	264000.0	9.5	0.0	2009-06-01

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Customer ID    4 non-null      float64
 1   Customer Name  4 non-null      category
 2   2018 Revenue   4 non-null      float64
 3   2019 Revenue   4 non-null      float64
 4   Growth         4 non-null      float64
 5   New Customer   4 non-null      float64
 6   Starting Date  4 non-null      datetime64[ns]
dtypes: category(1), datetime64[ns](1), float64(5)
memory usage: 573.0 bytes
```

```
0       Pandas Banking
1       Pandas Grocery
2       Pandas Telecom
3     Pandas Transport
4                  NaN
Name: Customer Name, dtype: object
```

```
0    False
1     True
2    False
3     True
4    False
Name: New Customer, dtype: bool
```

```
0   -2732 days
1   -1585 days
2          NaT
3    -960 days
4   -4110 days
Name: Starting Date, dtype: timedelta64[ns]
```

	Receipt Id	Date of Purchase	Product Name	Product Weight	Total Price	Retail shop name
0	10001.0	24/05/20	Wheat	87.0	99.0	NaN
1	NaN	05/05/20	NaN	NaN	25.0	Dello Superstore
2	10003.0	27/04/20	Vegetables	19.0	37.0	Javies Retail
3	10004.0	05/05/20	Oil	99.0	44.0	Javies Retail
4	10005.0	NaN	Wheat	30.0	NaN	Javies Retail

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58 entries, 0 to 57
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Receipt Id         44 non-null     float64
 1   Date of Purchase   46 non-null     object
 2   Product Name       46 non-null     object
 3   Product Weight     51 non-null     float64
 4   Total Price        51 non-null     float64
 5   Retail shop name   45 non-null     object
dtypes: float64(3), object(3)
memory usage: 2.8+ KB
```

	Receipt Id	Date of Purchase	Product Name	Product Weight	Total Price	Retail shop name
0	10001.0	24/05/20	Wheat	87.0	99.0	NaN
1	0.0	05/05/20	NaN	0.0	25.0	Dello Superstore
2	10003.0	27/04/20	Vegetables	19.0	37.0	Javies Retail
3	10004.0	05/05/20	Oil	99.0	44.0	Javies Retail
4	10005.0	NaN	Wheat	30.0	0.0	Javies Retail

	Receipt Id	Date of Purchase	Product Name	Product Weight	Total Price	Retail shop name
0	10001.0	24/05/20	Wheat	87.0	99.0	NaN
1	0.0	05/05/20	NaN	0.0	25.0	Dello Superstore
2	10003.0	27/04/20	Vegetables	19.0	37.0	Javies Retail
3	10004.0	05/05/20	Oil	99.0	44.0	Javies Retail
4	10005.0	01/01/99	Wheat	30.0	0.0	Javies Retail

	Receipt Id	Date of Purchase	Product Name	Product Weight	Total Price	Retail shop name
0	10001.0	24/05/20	Wheat	87.0	99.0	Missing Name
1	0.0	05/05/20	Missing Name	0.0	25.0	Dello Superstore
2	10003.0	27/04/20	Vegetables	19.0	37.0	Javies Retail
3	10004.0	05/05/20	Oil	99.0	44.0	Javies Retail
4	10005.0	01/01/99	Wheat	30.0	0.0	Javies Retail

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58 entries, 0 to 57
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Receipt Id        58 non-null     int32
 1   Date of Purchase  58 non-null     datetime64[ns]
 2   Product Name      58 non-null     category
 3   Product Weight    58 non-null     int32
 4   Total Price       58 non-null     int32
 5   Retail shop name  58 non-null     category
dtypes: category(2), datetime64[ns](1), int32(3)
memory usage: 2.1 KB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1728 entries, 0 to 1727
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   buying    1728 non-null   category
 1   maint     1728 non-null   category
 2   doors     1728 non-null   int64
 3   persons   1728 non-null   int32
 4   lug_boot  1728 non-null   category
 5   safety    1728 non-null   category
 6   class     1728 non-null   category
dtypes: category(5), int32(1), int64(1)
memory usage: 29.8 KB
```

```
0   1990-05-31 10:00:00
1   1995-06-05 15:00:00
2   2020-09-09 12:00:00
dtype: datetime64[ns]
```

```
0    1990-05-31    0    10:00:00
1    1995-06-05    1    15:00:00
2    2020-09-09    2    12:00:00
dtype: object       dtype: object
```

```
0    1990        0     Thursday
1    1995        1       Monday
2    2020        2    Wednesday
dtype: int64    dtype: object
```

```
0   -10807 days
1    -8976 days
2      252 days
dtype: timedelta64[ns]
```

```
0   -933724800.0
1   -775526400.0
2     21772800.0
dtype: float64
```

	days	hours	minutes	seconds	milliseconds	microseconds	nanoseconds
0	-10807	0	0	0	0	0	0
1	-8976	0	0	0	0	0	0
2	252	0	0	0	0	0	0

```
0          Data Analysis using python with panda...
1     pandas DataFrame and pandas series are useful
2                                    PYTHON3 PANDAS
dtype: object
```

```
0    1          0    56
1    2          1    47
2    0          2    14
dtype: int64 dtype: int64
```

```
0    47          0    False
1    45          1     True
2    14          2    False
dtype: int64     dtype: bool
```

```
0    True
1    True
2    True
dtype: bool
```

	Analysis	Data	DataFrame	PANDAS	PYTHON3	and	are	great	is	pandas	python	series	useful	using	with
0	1	1	0	0	0	0	0	1	1	1	1	0	0	1	1
1	0	0	1	0	0	1	1	0	0	1	0	1	1	0	0
2	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0

	python	pandas
0	1	1
1	0	1
2	0	1

	Customer ID	Customer Name	2018 Revenue	2019 Revenue	Growth	Start Year	Start Month	Start Day	New Customer
0	1001.0	Pandas Banking	235000	248000	5.5	2013	3	10	0
1	1002.0	Pandas Grocery	196000	205000	4.5	2016	4	30	0
2	1003.0	Pandas Telecom	167000	193000	15.5	2010	11	24	0
3	1004.0	Pandas Transport	79000	90000	13.9	2018	1	15	1
4	1005.0	Pandas Insurance	241000	264000	9.5	2009	6	1	0

```
Customer ID        float64
Customer Name       object
2018 Revenue        object
2019 Revenue        object
Growth              object
Start Year           int64
Start Month          int64
Start Day            int64
New Customer         int64
dtype: object
```

	Customer Name	2018 Revenue	2019 Revenue	Growth
0	Pandas Banking	235000	248000	5.5
1	Pandas Grocery	196000	205000	4.5
2	Pandas Telecom	167000	193000	15.5
3	Pandas Transport	79000	90000	13.9
4	Pandas Insurance	241000	264000	9.5

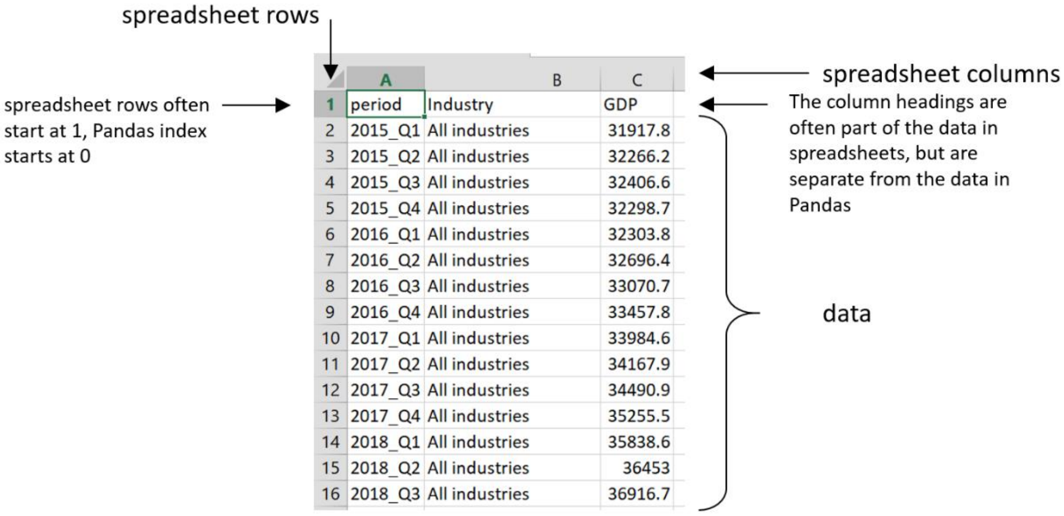
	Customer ID	Start Year	Start Month	Start Day	New Customer
0	1001.0	2013	3	10	0
1	1002.0	2016	4	30	0
2	1003.0	2010	11	24	0
3	1004.0	2018	1	15	1
4	1005.0	2009	6	1	0

	Customer ID
0	1001.0
1	1002.0
2	1003.0
3	1004.0
4	1005.0

Out[2]:

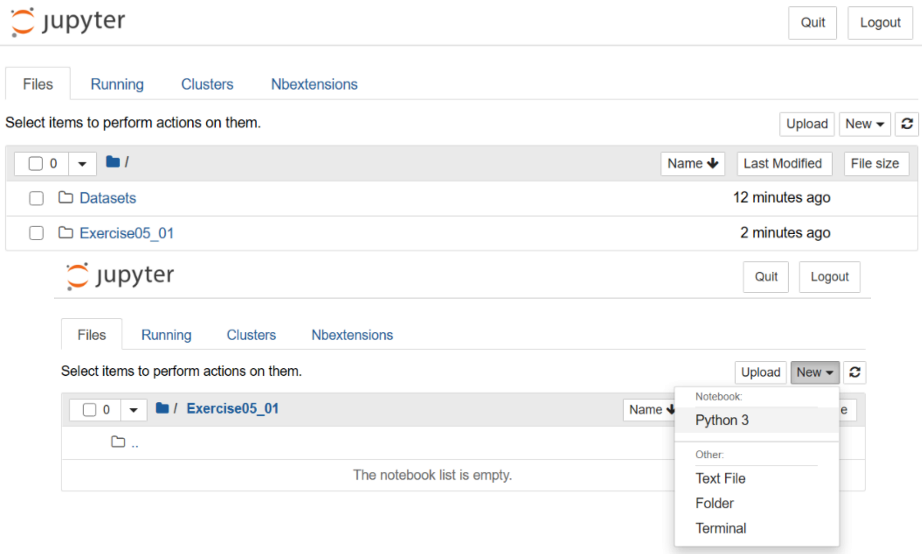
	period	Industry	GDP
0	2015_Q1	All industries	31917.8
1	2015_Q2	All industries	32266.2
2	2015_Q3	All industries	32406.6
3	2015_Q4	All industries	32298.7
4	2016_Q1	All industries	32303.8
...	...	...	...
2129	2019_Q2	Government enterprises	371.4
2130	2019_Q3	Government enterprises	373.5
2131	2019_Q4	Government enterprises	375.1
2132	2020_Q1	Government enterprises	372.8
2133	2020_Q2	Government enterprises	346.0

2134 rows × 3 columns



	A	B	C
1	period	Industry	GDP
2	(R2, C1)	(R2, C2)	(R2, C3)
3	(R3, C1)	(R3, C2)	(R3, C3)
4	(R4, C1)	(R4, C2)	(R4, C3)
5	(R5, C1)	(R5, C2)	(R5, C3)
6	(R6, C1)	(R6, C2)	(R6, C3)
7	(R7, C1)	(R7, C2)	(R7, C3)
8	(R8, C1)	(R8, C2)	(R8, C3)
9	(R9, C1)	(R9, C2)	(R9, C3)
10	(R10, C1)	(R10, C2)	(R10, C3)
11	(R11, C1)	(R11, C2)	(R11, C3)
12	(R12, C1)	(R12, C2)	(R12, C3)
13	(R13, C1)	(R13, C2)	(R13, C3)

period	Industry	GDP
(R0, C0)	(R0, C1)	(R0, C2)
(R1, C0)	(R1, C1)	(R1, C2)
(R2, C0)	(R2, C1)	(R2, C2)
(R3, C0)	(R3, C1)	(R3, C2)
(R4, C0)	(R4, C1)	(R4, C2)
(R5, C0)	(R5, C1)	(R5, C2)
(R6, C0)	(R6, C1)	(R6, C2)
(R7, C0)	(R7, C1)	(R7, C2)
(R8, C0)	(R8, C1)	(R8, C2)
(R9, C0)	(R9, C1)	(R9, C2)
(R10, C0)	(R10, C1)	(R10, C2)
(R11, C0)	(R11, C1)	(R11, C2)




```
the index is type  <class 'pandas.core.indexes.range.RangeIndex'>
while the columns are type  <class 'pandas.core.indexes.base.Index'>
the second item in the index is  1
and the second column is  Industry
```

Out[11]:

	values	names
0	6	oranges
1	1	apples
2	5	bananas
3	2	pears

Out[12]:

	values	names
citrus	6	oranges
non_citrus	1	apples
non_citrus	5	bananas
non_citrus	2	pears

```
non_citrus      apples
non_citrus     bananas
non_citrus       pears
Name: names, dtype: object

non_citrus      apples
non_citrus     bananas
non_citrus       pears
Name: names, dtype: object
```

Out[26]:

	species	name
0	feline	housecat
1	canine	wolf
2	canine	dingo
3	feline	tiger

Out[27]:

	species	name
0	feline	housecat
2	canine	dingo
3	feline	tiger

Out[28]:
```
species    canine
name        dingo
Name: 2, dtype: object
```

Out[29]:
```
species    canine
name        dingo
Name: 2, dtype: object
```

Out[32]:

	period	GDP
0	2015_Q1	31917.8
1	2015_Q2	32266.2
2	2015_Q3	32406.6
3	2015_Q4	32298.7
4	2016_Q1	32303.8

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-33-c6b03db243d3> in <module>
      3 # use the same column list of [0, 2]
      4 #
----> 5 GDP_summary = GDP_data.loc[:, [0, 2]]

~\Miniconda3\envs\keras-gpu-4\lib\site-packages\pandas\core\indexing.py in __getitem__(self, key)
   1760                 except (KeyError, IndexError, AttributeError):
   1761                     pass
-> 1762             return self._getitem_tuple(key)
   1763         else:
   1764             # we by definition only have the 0th axis

~\Miniconda3\envs\keras-gpu-4\lib\site-packages\pandas\core\indexing.py in _getitem_tuple(self, tup)
   1287                 continue
   1288 
-> 1289             retval = getattr(retval, self.name)._getitem_axis(key, axis=i)
   1290 
   1291         return retval

~\Miniconda3\envs\keras-gpu-4\lib\site-packages\pandas\core\indexing.py in _getitem_axis(self, key, axis)
   1952                     raise ValueError("Cannot index with multidimensional key")
   1953 
-> 1954                 return self._getitem_iterable(key, axis=axis)
   1955 
   1956             # nested tuple slicing

~\Miniconda3\envs\keras-gpu-4\lib\site-packages\pandas\core\indexing.py in _getitem_iterable(self, key, axi
s)
  1593         else:
  1594             # A collection of keys
-> 1595             keyarr, indexer = self._get_listlike_indexer(key, axis, raise_missing=False)
  1596             return self.obj._reindex_with_indexers(
  1597                 {axis: [keyarr, indexer]}, copy=True, allow_dups=True

~\Miniconda3\envs\keras-gpu-4\lib\site-packages\pandas\core\indexing.py in _get_listlike_indexer(self, key,
axis, raise_missing)
  1551 
  1552         self._validate_read_indexer(
-> 1553             keyarr, indexer, o._get_axis_number(axis), raise_missing=raise_missing
  1554         )
  1555         return keyarr, indexer

~\Miniconda3\envs\keras-gpu-4\lib\site-packages\pandas\core\indexing.py in _validate_read_indexer(self, key,
indexer, axis, raise_missing)
  1638             if missing == len(indexer):
  1639                 axis_name = self.obj._get_axis_name(axis)
-> 1640                 raise KeyError(f"None of [{key}] are in the [{axis_name}]")
  1641 
  1642             # We (temporarily) allow for some missing keys with .loc, except in

KeyError: "None of [Int64Index([0, 2], dtype='int64')] are in the [columns]"
```

KeyError: "None of [Int64Index([0, 2], dtype='int64')] are in the [columns]"

Out[34]:

	period	GDP
0	2015_Q1	31917.8
1	2015_Q2	32266.2
2	2015_Q3	32406.6
3	2015_Q4	32298.7
4	2016_Q1	32303.8
...	...	...
2129	2019_Q2	371.4
2130	2019_Q3	373.5
2131	2019_Q4	375.1
2132	2020_Q1	372.8
2133	2020_Q2	346.0

2134 rows × 2 columns

```
Out[37]: Index(['AT', 'AP', 'AH', 'AFDP', 'GTEP', 'TIT', 'TAT', 'TEY', 'CDP', 'CO',
               'NOX'],
              dtype='object')
```

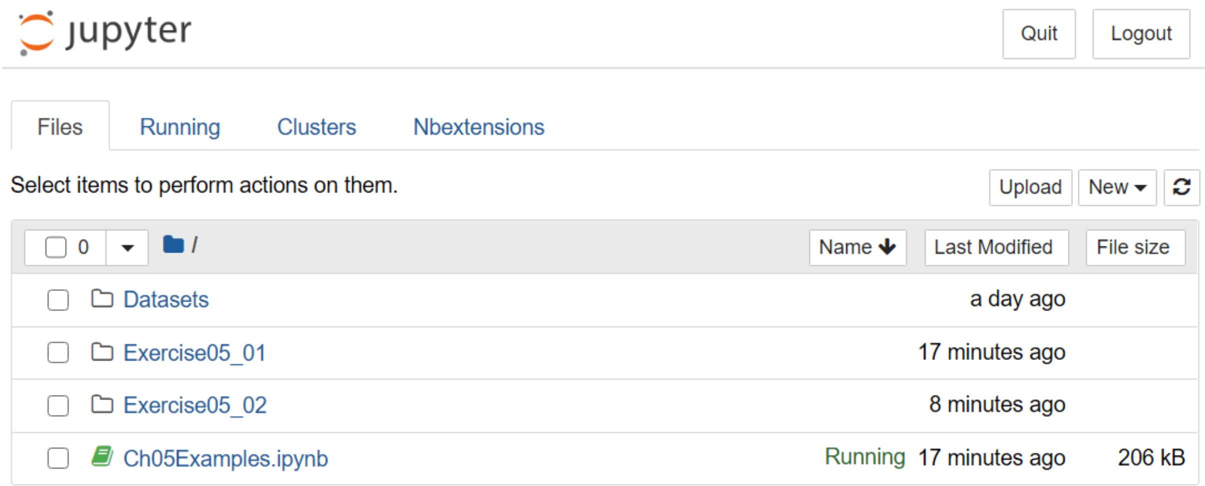
Out[38]:

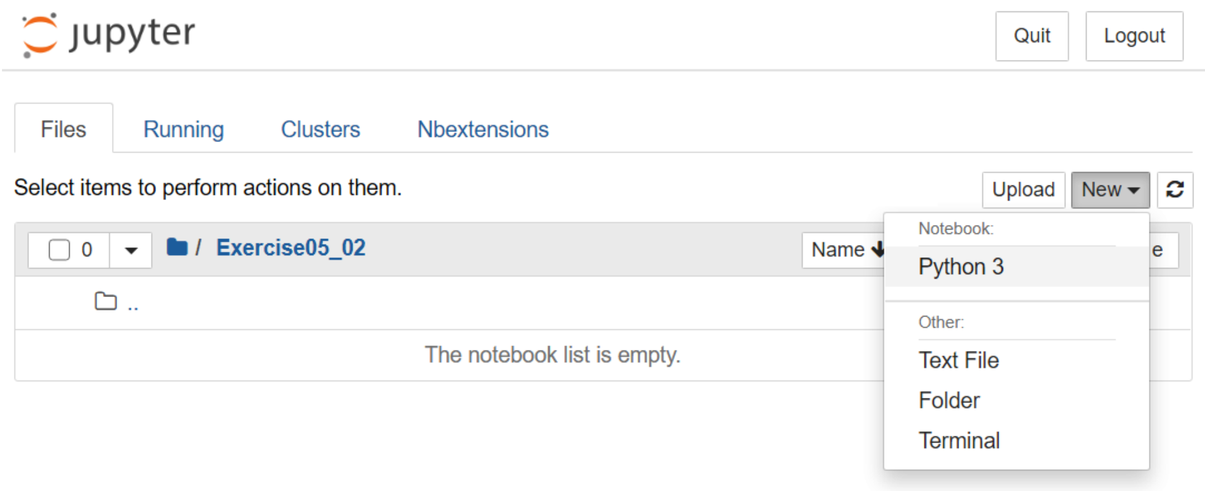
	AT	AP	AH	AFDP	GTEP	TIT	TAT	TEY	CDP
0	1.95320	1020.1	84.985	2.5304	20.116	1048.7	544.92	116.27	10.799
1	1.21910	1020.1	87.523	2.3937	18.584	1045.5	548.50	109.18	10.347
2	0.94915	1022.2	78.335	2.7789	22.264	1068.8	549.95	125.88	11.256
3	1.00750	1021.7	76.942	2.8170	23.358	1075.2	549.63	132.21	11.702
4	1.28580	1021.6	76.732	2.8377	23.483	1076.2	549.68	133.58	11.737

Out[48]:

	CO	NOX
0	7.4491	113.250
1	6.4684	112.020
2	3.6335	88.147
3	3.1972	87.078
4	2.3833	82.515
...	...	...
7379	10.9930	89.172
7380	11.1440	88.849
7381	11.4140	96.147
7382	3.3134	64.738
7383	11.9810	109.240

7384 rows × 2 columns





```
(7384, 11)
['AT', 'AP', 'AH', 'AFDP', 'GTEP', 'TIT', 'TAT', 'TEY', 'CDP', 'CO',
'NOX']
```

	CO	NOX
count	100.000000	100.000000
mean	3.774012	77.661970
std	1.774795	13.708632
min	0.475440	58.432000
25%	2.656625	64.672000
50%	3.501650	78.084000
75%	4.078250	85.121250
max	12.659000	118.270000

	CO	NOX
count	7384.000000	7384.000000
mean	3.129986	59.890509
std	2.234962	11.132464
min	0.212800	25.905000
25%	1.808175	52.399000
50%	2.533400	56.838500
75%	3.702550	65.093250
max	41.097000	119.680000

	species	location	weight	color	fur
0	dog	city	10	striped	long
1	chicken	town	11	solid	long
2	cat	city	12	striped	short
3	cat	farm	13	striped	short
4	chicken	farm	14	solid	long
...	...	...	...	...	...
95	pig	city	105	solid	short
96	chicken	city	106	solid	short
97	dog	town	107	striped	long
98	cat	town	108	solid	short
99	chicken	town	109	solid	short

100 rows × 5 columns

Out[129]:

		weight					
	color	solid		spotted		striped	
	fur	long	short	long	short	long	short
location	species						
city	cat	74.00	23.50	33.00	84.00	20.00	12.00
	chicken	44.00	91.33	63.67		103.00	87.00
	dog	15.00	44.00	64.00	75.00	39.00	51.00
	pig		86.00	40.00		39.00	
farm	cat	49.00	85.00	85.00	82.00	69.00	24.33
	chicken	59.00		81.00	43.00	16.00	75.67
	dog	102.00	99.00	49.20	58.00	44.50	
	pig	30.00	36.00	65.00	54.50	47.00	49.00
town	cat		108.00	60.50	64.75	37.00	
	chicken	52.50	81.00				44.25
	dog	61.00	71.50		51.50	79.67	55.00
	pig	101.00		95.00	60.50		

Out[131]:

period	Industry	GDP
2015_Q1	Accommodation	256.20
	Accommodation and food services	973.60
	Administrative and support services	795.70
	Administrative and waste management services	883.00
	Agriculture, forestry, fishing, and hunting	466.30
...	...	...
2020_Q2	Warehousing and storage	135.10
	Waste management and remediation services	100.60
	Water transportation	32.00
	Wholesale trade	1,810.90
	Wood products	111.90

2134 rows × 1 columns

Out[98]:

period	Industry	GDP
2017_Q2	Farms	399.7
	Federal	1124.6
	Federal Reserve banks, credit intermediation, and related activities	926.5
	Finance and insurance	2827.1

Out[121]:

	period	Industry	GDP
75	2017_Q2	Farms	399.7
1197	2017_Q2	Finance and insurance	2827.1
1219	2017_Q2	Federal Reserve banks, credit intermediation, ...	926.5
1967	2017_Q2	Federal	1124.6

Out[146]:

	period	Industry	GDP
75	2017_Q2	Farms	399.7
1197	2017_Q2	Finance and insurance	2827.1
1219	2017_Q2	Federal Reserve banks, credit intermediation, ...	926.5
1967	2017_Q2	Federal	1124.6

Out[39]:

	period	Industry	GDP
0	2015_Q1	All industries	31917.8
1	2015_Q2	All industries	32266.2
2	2015_Q3	All industries	32406.6
3	2015_Q4	All industries	32298.7
4	2016_Q1	All industries	32303.8

Out[40]:
```
MultiIndex([('2015_Q1',          'All industries'),
            ('2015_Q2',          'All industries'),
            ('2015_Q3',          'All industries'),
            ('2015_Q4',          'All industries'),
            ('2016_Q1',          'All industries'),
            ('2016_Q2',          'All industries'),
            ('2016_Q3',          'All industries'),
            ('2016_Q4',          'All industries'),
            ('2017_Q1',          'All industries'),
            ('2017_Q2',          'All industries'),
            ...
            ('2018_Q1', 'Government enterprises'),
            ('2018_Q2', 'Government enterprises'),
            ('2018_Q3', 'Government enterprises'),
            ('2018_Q4', 'Government enterprises'),
            ('2019_Q1', 'Government enterprises'),
            ('2019_Q2', 'Government enterprises'),
            ('2019_Q3', 'Government enterprises'),
            ('2019_Q4', 'Government enterprises'),
            ('2020_Q1', 'Government enterprises'),
            ('2020_Q2', 'Government enterprises')],
           names=['period', 'Industry'], length=2134)
```

Out[41]:

		GDP
period	**Industry**	
2015_Q1	**All industries**	31917.8
2015_Q2	**All industries**	32266.2
2015_Q3	**All industries**	32406.6
2015_Q4	**All industries**	32298.7
2016_Q1	**All industries**	32303.8
...	**...**	...
2019_Q2	**Government enterprises**	371.4
2019_Q3	**Government enterprises**	373.5
2019_Q4	**Government enterprises**	375.1
2020_Q1	**Government enterprises**	372.8
2020_Q2	**Government enterprises**	346.0

2134 rows × 1 columns

Out[42]:

period	Industry	GDP
2015_Q1	Accommodation	256.2
	Accommodation and food services	973.6
	Administrative and support services	795.7
	Administrative and waste management services	883.0
	Agriculture, forestry, fishing, and hunting	466.3
...	...	...
2020_Q2	Warehousing and storage	135.1
	Waste management and remediation services	100.6
	Water transportation	32.0
	Wholesale trade	1810.9
	Wood products	111.9

2134 rows × 1 columns

```
Out[12]: Index(['class', 'cap-shape', 'cap-surface', 'cap-color', 'bruises', 'odor',
       'gill-attachment', 'gill-spacing', 'gill-size', 'gill-color',
       'stalk-shape', 'stalk-root', 'stalk-surface-above-ring',
       'stalk-surface-below-ring', 'stalk-color-above-ring',
       'stalk-color-below-ring', 'veil-type', 'veil-color', 'ring-number',
       'ring-type', 'spore-print-color', 'population', 'habitat'],
      dtype='object')
```

Out[5]:

population	habitat	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	stalk-surface-above-ring	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	nu
n	g	x	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w	
	m	b	s	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	
a	g	x	s	g	f	n	f	w	b	k	t	e	s	s	w	w	p	w	
n	g	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	
	m	b	s	w	t	a	f	c	b	g	e	c	s	s	w	w	p	w	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
v	l	x	s	n	f	n	a	c	b	y	e	?	s	s	o	o	p	o	
c	l	k	s	n	f	n	a	c	b	y	e	?	s	s	o	o	p	o	
v	l	x	s	n	f	n	a	c	b	y	e	?	s	s	o	o	p	n	
c	l	f	s	n	f	n	a	c	b	n	e	?	s	s	o	o	p	o	
	l	x	s	n	f	n	a	c	b	y	e	?	s	s	o	o	p	o	

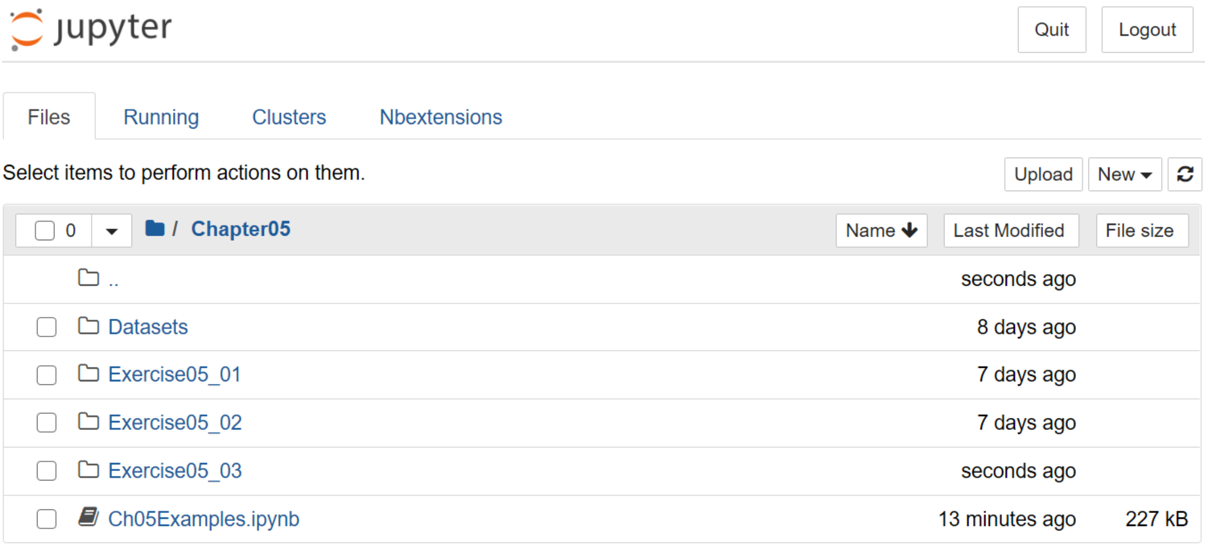
4208 rows × 20 columns

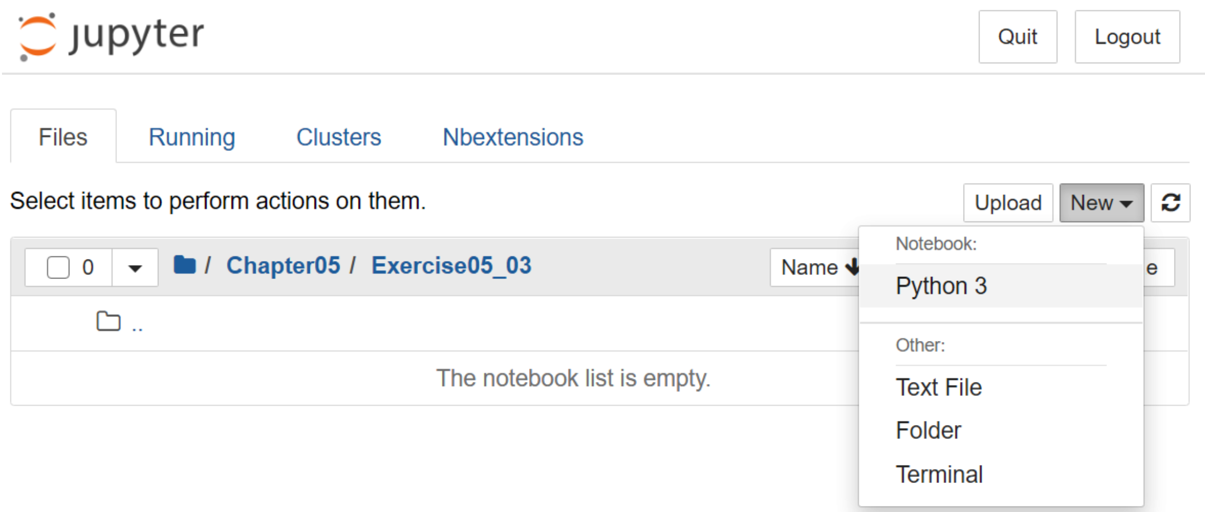
```
<class 'pandas.core.series.Series'>
 0        31917.8
1        32266.2
2        32406.6
3        32298.7
4        32303.8
          ...
2129       371.4
2130       373.5
2131       375.1
2132       372.8
2133       346.0
Name: GDP, Length: 2134, dtype: float64
<class 'pandas.core.series.Series'>
 0        31917.8
1        32266.2
2        32406.6
3        32298.7
4        32303.8
          ...
2129       371.4
2130       373.5
2131       375.1
2132       372.8
2133       346.0
Name: GDP, Length: 2134, dtype: float64
```

```
<class 'pandas.core.frame.DataFrame'>
       period      GDP
0     2015_Q1  31917.8
1     2015_Q2  32266.2
2     2015_Q3  32406.6
3     2015_Q4  32298.7
4     2016_Q1  32303.8
...       ...      ...
2129  2019_Q2    371.4
2130  2019_Q3    373.5
2131  2019_Q4    375.1
2132  2020_Q1    372.8
2133  2020_Q2    346.0

[2134 rows x 2 columns]
```

Out[231]:

	period	Industry	GDP
3	2015_Q4	All industries	32298.7
4	2016_Q1	All industries	32303.8
5	2016_Q2	All industries	32696.4
6	2016_Q3	All industries	33070.7
7	2016_Q4	All industries	33457.8
8	2017_Q1	All industries	33984.6
9	2017_Q2	All industries	34167.9


jupyter
Quit
Logout
Files
Running
Clusters
Nbextensions
Select items to perform actions on them.
Upload
New
0
/ Chapter05
Name
Last Modified
File size
..
seconds ago
Datasets
8 days ago
Exercise05_01
7 days ago
Exercise05_02
7 days ago
Exercise05_03
seconds ago
Ch05Examples.ipynb
13 minutes ago
227 kB


jupyter
Quit
Logout
Files
Running
Clusters
Nbextensions
Select items to perform actions on them.
Upload
New
0
/ Chapter05 / Exercise05_03
Name
..
The notebook list is empty.
Notebook:
Python 3
Other:
Text File
Folder
Terminal

Out[2]:

	date	sales
0	2017-03-31	199190.4
1	2017-06-30	194356.6
2	2017-09-30	191611.7
3	2017-12-31	198918.9
4	2018-03-30	200163.2
5	2018-06-30	201510.2
6	2018-09-30	209749.8
7	2019-12-31	201897.8
8	2019-03-31	200098.8
9	2019-06-30	219340.3
10	2019-09-30	211542.5
11	2019-12-31	211729.1

	date	sales
month		
03	2017-03-31	199190.4
06	2017-06-30	194356.6
09	2017-09-30	191611.7
12	2017-12-31	198918.9
03	2018-03-30	200163.2
06	2018-06-30	201510.2
09	2018-09-30	209749.8
12	2019-12-31	201897.8
03	2019-03-31	200098.8
06	2019-06-30	219340.3
09	2019-09-30	211542.5
12	2019-12-31	211729.1

```
using .iloc with index 3:
 date     2017-12-31
sales        198919
Name: 12, dtype: object

using .loc with index 03:
              date     sales
month
03      2017-03-31  199190.4
03      2018-03-30  200163.2
03      2019-03-31  200098.8
```

Out[18]:

	sales
month	
03	199817.466667
06	205069.033333
09	204301.333333
12	204181.933333

Out[234]:

	period	Industry	GDP
0	2015_Q1	All industries	31917.8
3	2015_Q4	All industries	32298.7
6	2016_Q3	All industries	33070.7
9	2017_Q2	All industries	34167.9
12	2018_Q1	All industries	35838.6
15	2018_Q4	All industries	37205.3
18	2019_Q3	All industries	37991.1
21	2020_Q2	All industries	34260.0
24	2015_Q3	Private industries	28826.0
27	2016_Q2	Private industries	29058.3
30	2017_Q1	Private industries	30263.5
33	2017_Q4	Private industries	31425.1
36	2018_Q3	Private industries	32940.9
39	2019_Q2	Private industries	33632.4
42	2020_Q1	Private industries	33685.4
45	2015_Q2	Agriculture, forestry, fishing, and hunting	457.9
48	2016_Q1	Agriculture, forestry, fishing, and hunting	443.7

Out[190]:

	period	Industry	GDP
2133	2020_Q2	Government enterprises	346.0
2132	2020_Q1	Government enterprises	372.8
2131	2019_Q4	Government enterprises	375.1
2130	2019_Q3	Government enterprises	373.5
2129	2019_Q2	Government enterprises	371.4
...	...	...	...
4	2016_Q1	All industries	32303.8
3	2015_Q4	All industries	32298.7
2	2015_Q3	All industries	32406.6
1	2015_Q2	All industries	32266.2
0	2015_Q1	All industries	31917.8

2134 rows × 3 columns

Out[194]:

	period	Industry	GDP
100	2018_Q1	Forestry, fishing, and related activities	56.0
97	2017_Q2	Forestry, fishing, and related activities	55.7
94	2016_Q3	Forestry, fishing, and related activities	51.8
91	2015_Q4	Forestry, fishing, and related activities	52.8
88	2015_Q1	Forestry, fishing, and related activities	54.6
85	2019_Q4	Farms	405.9
82	2019_Q1	Farms	392.3
79	2018_Q2	Farms	405.0
76	2017_Q3	Farms	395.8
73	2016_Q4	Farms	375.1
70	2016_Q1	Farms	390.0
67	2015_Q2	Farms	405.4
64	2020_Q1	Agriculture, forestry, fishing, and hunting	467.7
61	2019_Q2	Agriculture, forestry, fishing, and hunting	448.4
58	2018_Q3	Agriculture, forestry, fishing, and hunting	449.9
55	2017_Q4	Agriculture, forestry, fishing, and hunting	455.1
52	2017_Q1	Agriculture, forestry, fishing, and hunting	455.1

```
<class 'pandas.core.series.Series'>
 period     2015_Q1
GDP        31917.8
Name: 0, dtype: object
```


```
<class 'pandas.core.series.Series'>
 period     2015_Q1
GDP        31917.8
Name: 0, dtype: object
```

```
<class 'pandas.core.frame.DataFrame'>
     period         Industry      GDP
0  2015_Q1  All industries  31917.8
<class 'pandas.core.frame.DataFrame'>
     period      GDP
0  2015_Q1  31917.8
<class 'pandas.core.series.Series'>
 0    31917.8
Name: GDP, dtype: float64

<class 'pandas.core.series.Series'>
 0    31917.8
Name: GDP, dtype: float64
```

Out[43]:

	period	Industry	GDP
0	2015_Q1	All industries	31917.8
1	2015_Q2	All industries	32266.2
2	2015_Q3	All industries	32406.6
3	2015_Q4	All industries	32298.7
22	2015_Q1	Private industries	28392.6
...	...	...	...
2093	2015_Q4	General government	2164.7
2112	2015_Q1	Government enterprises	324.4
2113	2015_Q2	Government enterprises	326.4
2114	2015_Q3	Government enterprises	328.7
2115	2015_Q4	Government enterprises	330.2

388 rows × 3 columns

Out[44]:

	period	Industry	GDP
0	2015_Q1	All industries	36917.8
1	2015_Q2	All industries	37266.2
2	2015_Q3	All industries	37406.6
3	2015_Q4	All industries	37298.7
22	2015_Q1	Private industries	33392.6
...	...	...	...
2093	2015_Q4	General government	7164.7
2112	2015_Q1	Government enterprises	5324.4
2113	2015_Q2	Government enterprises	5326.4
2114	2015_Q3	Government enterprises	5328.7
2115	2015_Q4	Government enterprises	5330.2

388 rows × 3 columns

```
C:\Users\bbate\Miniconda3\envs\keras-gpu-2\lib\site-packages\ipykernel_
launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
```

Out[46]:

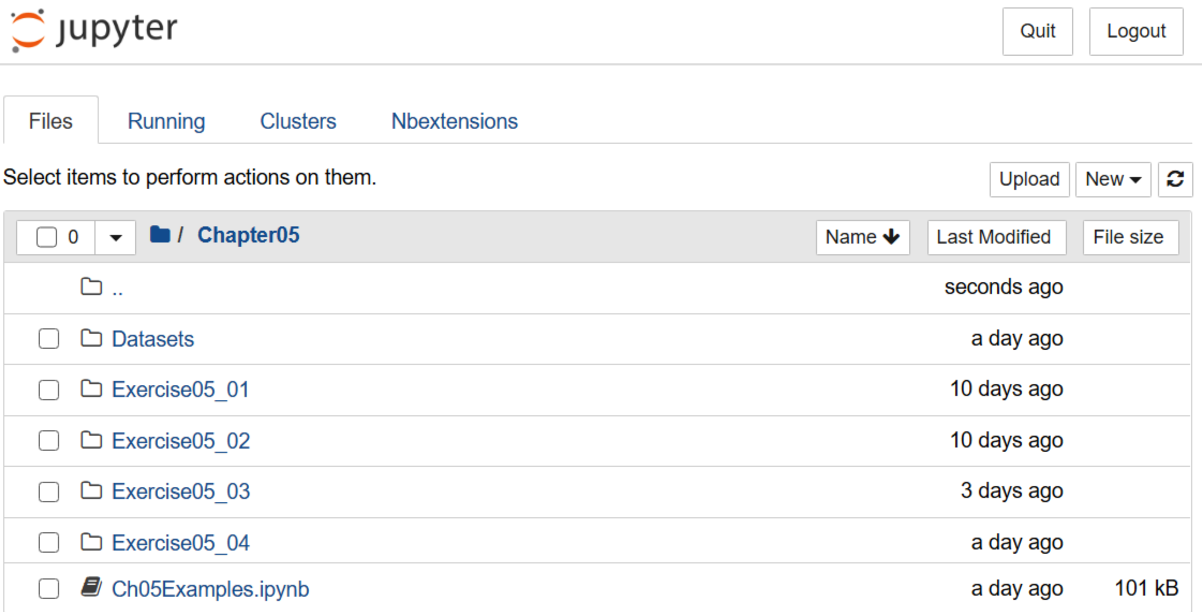
	period	Industry	GDP
0	2015_Q1	All industries	36917.8
1	2015_Q2	All industries	37266.2
2	2015_Q3	All industries	37406.6
3	2015_Q4	All industries	37298.7
22	2015_Q1	Private industries	33392.6
...	...	...	...
2093	2015_Q4	General government	7164.7
2112	2015_Q1	Government enterprises	5324.4
2113	2015_Q2	Government enterprises	5326.4
2114	2015_Q3	Government enterprises	5328.7
2115	2015_Q4	Government enterprises	5330.2

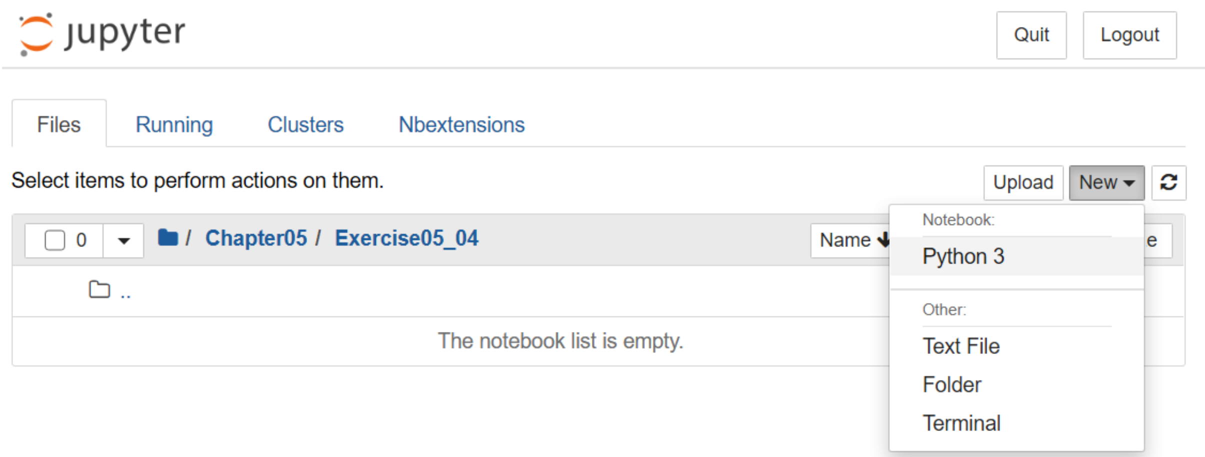
388 rows × 3 columns

Out[47]:

	period	Industry	GDP
0	2015_Q1	All industries	0.0
1	2015_Q2	All industries	0.0
2	2015_Q3	All industries	0.0
3	2015_Q4	All industries	0.0
22	2015_Q1	Private industries	0.0
...	...	...	...
2093	2015_Q4	General government	7164.7
2112	2015_Q1	Government enterprises	5324.4
2113	2015_Q2	Government enterprises	5326.4
2114	2015_Q3	Government enterprises	5328.7
2115	2015_Q4	Government enterprises	5330.2

388 rows × 3 columns





Out[3]:

	condition	date	plant-stand	precip	temp	hail	crop-hist	area-damaged	severity	seed-tmt	...	int-discolor	sclerotia	fruit-pods	fruitspots	seed	mold-growth	seed-discolor	seed-size	shr
0	diaporthe-stem-canker	6.0	0.0	2.0	1.0	0.0	1.0	1.0	1.0	0.0	...	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	
1	diaporthe-stem-canker	4.0	0.0	2.0	1.0	0.0	2.0	0.0	2.0	1.0	...	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	
2	diaporthe-stem-canker	3.0	0.0	2.0	1.0	0.0	1.0	0.0	2.0	1.0	...	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	
3	diaporthe-stem-canker	3.0	0.0	2.0	1.0	0.0	1.0	0.0	2.0	0.0	...	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	
4	diaporthe-stem-canker	6.0	0.0	2.0	1.0	0.0	2.0	0.0	1.0	0.0	...	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
302	2-4-d-injury	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
303	herbicide-injury	1.0	1.0	NaN	0.0	NaN	1.0	0.0	NaN	NaN	...	NaN	NaN	3.0	NaN	NaN	NaN	NaN	NaN	
304	herbicide-injury	0.0	1.0	NaN	0.0	NaN	0.0	3.0	NaN	NaN	...	NaN	NaN	3.0	NaN	NaN	NaN	NaN	NaN	
305	herbicide-injury	1.0	1.0	NaN	0.0	NaN	0.0	0.0	NaN	NaN	...	NaN	NaN	3.0	NaN	NaN	NaN	NaN	NaN	
306	herbicide-injury	1.0	1.0	NaN	0.0	NaN	1.0	3.0	NaN	NaN	...	NaN	NaN	3.0	NaN	NaN	NaN	NaN	NaN	

307 rows × 36 columns

```
there are 19 unique conditions ['diaporthe-stem-canker', 'charcoal-rot', 'rhizoctonia-root-rot', 'phytophthora-ro
t', 'brown-stem-rot', 'powdery-mildew', 'downy-mildew', 'brown-spot', 'bacterial-blight', 'bacterial-pustule', 'pu
rple-seed-stain', 'anthracnose', 'phyllosticta-leaf-spot', 'alternarialeaf-spot', 'frog-eye-leaf-spot', 'diaporthe
-pod-&-stem-blight', 'cyst-nematode', '2-4-d-injury', 'herbicide-injury']
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\Miniconda3\envs\keras-gpu-5\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
   2897             try:
-> 2898                 return self._engine.get_loc(casted_key)
   2899             except KeyError as err:


~\Miniconda3\envs\keras-gpu-5\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
   2898                 return self._engine.get_loc(casted_key)
   2899             except KeyError as err:
-> 2900                 raise KeyError(key) from err
   2901 
   2902         if tolerance is not None:

KeyError: 0
```

```
Out[14]: Int64Index([  7,  11,  13,  15,  16,  19,  22,  30,  43,  48,  55,  76,  77,
                     80,  90,  91,  95, 100, 101, 102, 104, 110, 114, 116, 127, 128,
                    133, 137, 142, 150, 151, 155, 157, 159, 161, 164, 166, 168, 170,
                    172, 173, 176, 177, 181, 183, 185, 190, 195, 197, 201, 203, 205,
                    206, 208, 214],
                   dtype='int64')
```

```
case  22  with condition  rhizoctonia-root-rot  is severe
case  43  with condition  phytophthora-rot  is severe
case  48  with condition  phytophthora-rot  is severe
case  55  with condition  phytophthora-rot  is severe
```

```
Out[2]: Jan    100
        Feb    125
        Mar    105
        Apr    111
        May    275
        Jun    137
        Jul     99
        Aug     10
        Sep    250
        Oct    100
        Nov    175
        Dec    200
        Name: income, dtype: int64
```

```
Out[3]: 0     100
        1     125
        2     105
        3     111
        4     275
        5     137
        6      99
        7      10
        8     250
        9     100
        10    175
        11    200
        Name: income, dtype: int64
```

```
<class 'pandas.core.series.Series'>
0    288.177459
1    316.485721
2    338.565899
3    336.866984
4    332.844765
Name: annual_cost, dtype: float64
```

```
UK_energy.loc[[2, 4, 6]]
```

```
 2    338.565899
4    332.844765
6    341.909881
Name: annual_cost, dtype: float64
```

```
UK_energy[2:7:2]

 2     338.565899
4     332.844765
6     341.909881
Name: annual_cost, dtype: float64
```

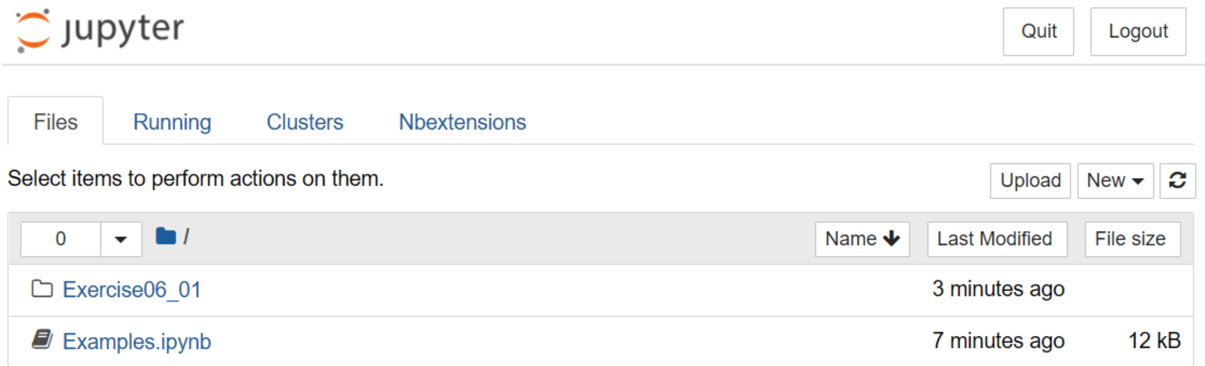
```
UK_energy[[2, 4, 6]]

 2     338.565899
4     332.844765
6     341.909881
Name: annual_cost, dtype: float64
```

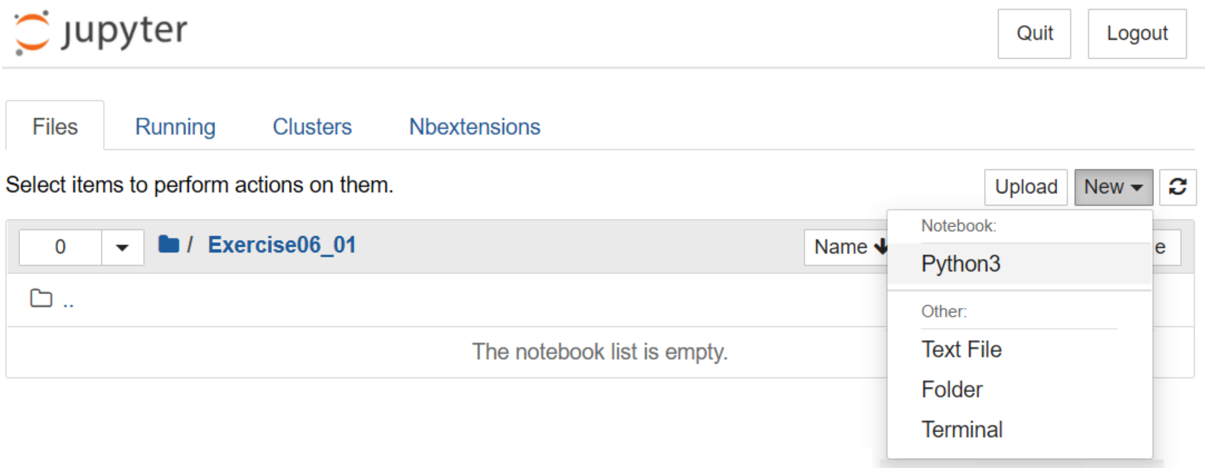
```
UK_energy.iloc[[2, 4, 6]]

 2     338.565899
4     332.844765
6     341.909881
Name: annual_cost, dtype: float64
```

```
UK_energy.iloc[2:7:2]

 2     338.565899
4     332.844765
6     341.909881
Name: annual_cost, dtype: float64
```

```
Out[27]: Index(['year_1990', 'year_1991', 'year_1992', 'year_1993', 'year_1994',
                'year_1995', 'year_1996', 'year_1997', 'year_1998', 'year_1999',
                'year_2000', 'year_2001', 'year_2002', 'year_2003', 'year_2004',
                'year_2005', 'year_2006', 'year_2007', 'year_2008', 'year_2009',
                'year_2010', 'year_2011', 'year_2012', 'year_2013', 'year_2014',
                'year_2015', 'year_2016', 'year_2017', 'year_2018', 'year_2019'],
               dtype='object')
```

```
Out[30]: year_1997    326.418454
         year_1998    306.393163
         year_1999    295.687501
         year_2000    290.333333
         year_2001    283.333333
         year_2002    281.666667
         year_2003    283.666667
         year_2004    291.666667
         year_2005    323.666667
         year_2006    382.000000
         year_2007    423.111111
         year_2008    487.333333
         year_2009    498.666667
         year_2010    484.000000
         year_2011    523.181818
         Name: annual_cost, dtype: float64
```





```
Out[5]: 0       0.783670
        1       0.293040
        2       0.111169
        3      -0.169703
        4      -0.147029
                  ...
        139     0.723983
        140     0.687518
        141     0.515671
        142     0.432008
        143     0.146747
        Name: Y, Length: 144, dtype: float64
Out[3]: 0       0.783670
        2       0.293040
        4       0.111169
        6      -0.169703
        8      -0.147029
                  ...
        278     0.723983
        280     0.687518
        282     0.515671
        284     0.432008
        286     0.146747
        Name: Y, Length: 144, dtype: float64
Out[7]: 0       0.783670
        4       0.111169
        8      -0.147029
        12     -0.032271
        16     -0.202202
                  ...
        268    -0.014538
        272     0.180167
        276     0.382172
        280     0.687518
        284     0.432008
        Name: Y, Length: 72, dtype: float64
```

Out[10]:

```
284    0.432008
280    0.687518
276    0.382172
272    0.180167
268   -0.014538
264   -0.080900
260    0.069567
256    0.153728
252    0.220703
Name: Y, dtype: float64
```

Out[6]:

	date	input_flow	input_Zinc	input_pH	input_BOD	input_COD	input_SS	input_VSS	input_SED	input_CON	...	output_COND	RD-DBO-P	RD-SS-P	RD-SED-P
0	1/1/1990	41230.0	0.35	7.6	120.0	344.0	136.0	54.4	4.5	993	...	903.0	-9999.0	62.8	93.3
1	1/2/1990	37386.0	1.40	7.9	165.0	470.0	170.0	76.5	4.0	1365	...	1481.0	-9999.0	50.0	94.4
2	1/3/1990	34535.0	1.00	7.8	232.0	518.0	220.0	65.5	5.5	1617	...	1492.0	32.6	62.4	95.0
3	1/4/1990	32527.0	3.00	7.8	187.0	460.0	180.0	67.8	5.2	1832	...	1590.0	13.2	57.6	95.5
4	1/7/1990	27760.0	1.20	7.6	199.0	466.0	186.0	74.2	4.5	1220	...	1411.0	38.2	46.6	95.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
522	10/25/1991	35400.0	0.70	7.6	156.0	364.0	194.0	63.9	5.5	1680	...	1840.0	47.3	61.3	94.0
523	10/26/1991	30964.0	3.30	7.7	220.0	540.0	184.0	62.0	3.5	1445	...	1337.0	-9999.0	38.6	93.3
524	10/27/1991	35573.0	7.30	7.6	176.0	333.0	178.0	64.0	3.5	1627	...	1799.0	-9999.0	40.4	95.0
525	10/29/1991	29801.0	1.60	7.7	172.0	400.0	136.0	70.1	1.5	1402	...	1468.0	32.4	40.4	88.0
526	10/30/1991	31524.0	1.60	7.9	-9999.0	478.0	204.0	64.7	6.0	1798	...	1568.0	-9999.0	43.9	65.3

527 rows × 39 columns

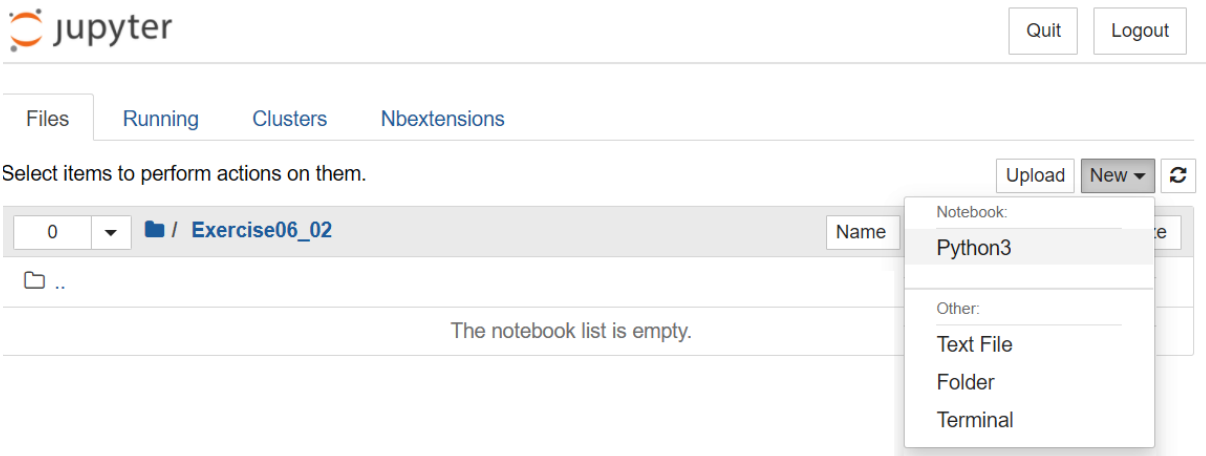
Out[41]:

	pH
date	
3/20/1990	7.4
4/13/1990	7.2
6/4/1990	7.3
6/8/1990	7.4
7/1/1990	7.3

```
   pH  flow
0  7.4   NaN
1  7.2   NaN
2  7.3   NaN
3  7.4   NaN
4  7.3   NaN
           pH     flow
8/21/1990  NaN  34352.0
8/24/1990  NaN  32802.0
8/28/1991  NaN  32922.0
8/29/1991  NaN  32190.0
8/4/1991   NaN  24978.0
```

```
pH index:  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]
acidity index:  ['3/20/1990', '4/13/1990', '6/4/1990', '6/8/1990', '7/1/1990', '7/23/1990', '7/29/1990', '8/21/1990', '8/24/199
0', '10/7/1990', '3/26/1991', '4/12/1991', '5/9/1991', '5/23/1991', '6/14/1991', '6/24/1991', '7/1/1991', '7/5/1991', '7/19/199
1', '7/21/1991', '7/30/1991', '8/1/1991', '8/4/1991', '8/18/1991', '8/28/1991', '8/29/1991', '10/5/1991']
```

```
                                         pH      flow
                           date
      pH      flow         3/20/1990    7.4   39165.0
0    7.4   39165.0         4/13/1990    7.2   34667.0
1    7.2   34667.0         6/4/1990     7.3   51520.0
2    7.3   51520.0         6/8/1990     7.4   35789.0
3    7.4   35789.0         7/1/1990     7.3   30201.0
4    7.3   30201.0                       pH      flow
       pH      flow        date
22    7.3   24978.0        8/4/1991     7.3   24978.0
23    7.3   27527.0        8/18/1991    7.3   27527.0
24    7.4   32922.0        8/28/1991    7.4   32922.0
25    7.3   32190.0        8/29/1991    7.3   32190.0
26    7.3   33695.0        10/5/1991    7.3   33695.0
```



```
pear     51
pear     92
pear     14
pear     74
pear     99
pear      2
pear     52
pear     37
pear     63
pear     59
pear     75
peach    60
peach    20
peach    82
peach    86
peach    21
peach     1
peach    87
peach    21
peach    48
dtype: int32
pear     51
pear     92
pear     14
peach    60
peach    20
peach    82
peach    86
pear     74
pear     99
pear      2
peach    21
pear     52
peach     1
peach    87
pear     37
pear     63
pear     59
pear     75
peach    21
peach    48
dtype: int32
```

```
Out[2]: fruit
orange    149
apple      98
orange     69
peach     103
peach     124
orange     81
pear      144
orange     67
peach     113
peach     127
Name: qty_ordered, dtype: int64
```

```
    top 3 changed 2.2 %
     vs. all changed 8.0 %
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-30-1b2c688411ee> in <module>
      2 # try to print using .iloc
      3 #
----> 4 print(my_list.iloc[12:23])

AttributeError: 'list' object has no attribute 'iloc'
```

Out[18]:

	period	Industry	GDP
0	2015_Q1	All industries	31917.8
22	2015_Q1	Private industries	28392.6
44	2015_Q1	Agriculture, forestry, fishing, and hunting	466.3
66	2015_Q1	Farms	411.7
88	2015_Q1	Forestry, fishing, and related activities	54.6

Out[16]:
```
{'period': {0: '2015_Q1',
  22: '2015_Q1',
  44: '2015_Q1',
  66: '2015_Q1',
  88: '2015_Q1'},
 'Industry': {0: 'All industries',
  22: 'Private industries',
  44: 'Agriculture, forestry, fishing, and hunting',
  66: 'Farms',
  88: 'Forestry, fishing, and related activities'},
 'GDP': {0: 31917.8, 22: 28392.6, 44: 466.3, 66: 411.7, 88: 54.6}}
```

```
0     2015_Q1
22    2015_Q1
44    2015_Q1
66    2015_Q1
88    2015_Q1
Name: period, dtype: object
```

Out[5]:

Sex	Rings	Length	Diameter	Height	Whole weight	.Shucked weight	Viscera weight	Shell weight
M	15	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150
	7	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070
F	9	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210
M	10	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155
I	7	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055
	8	0.425	0.300	0.095	0.3515	0.1410	0.0775	0.120
F	20	0.530	0.415	0.150	0.7775	0.2370	0.1415	0.330
	16	0.545	0.425	0.125	0.7680	0.2940	0.1495	0.260
M	9	0.475	0.370	0.125	0.5095	0.2165	0.1125	0.165
F	19	0.550	0.440	0.150	0.8945	0.3145	0.1510	0.320

```
for oysters with 16 or more rings

males weigh 0.458 vs. females weigh 0.449
males are 0.603 long  vs. females are 0.603 long
males are 0.478 in diameter  vs. females are 0.479 in diameter
males are 0.176 in height  vs. females are 0.174 in height
```

Chapter 7: Data Exploration and Data Transformation

	0	1	2	3	4	5	6	7	8
0	1001.0	Pandas Banking	235000	248000	5.5	2013	3	10	0
1	1002.0	Pandas Grocery	196000	205000	4.5	2016	4	30	0
2	1003.0	Pandas Telecom	167000	193000	15.5	2010	11	24	0
3	1004.0	Pandas Transport	79000	90000	13.9	2018	1	15	1
4	1005.0	Pandas Insurance	241000	264000	9.5	2009	6	1	0

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 9 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   0       5 non-null      float64
 1   1       5 non-null      object
 2   2       5 non-null      int64
 3   3       5 non-null      int64
 4   4       5 non-null      float64
 5   5       5 non-null      int64
 6   6       5 non-null      int64
 7   7       5 non-null      int64
 8   8       5 non-null      int64
dtypes: float64(2), int64(6), object(1)
memory usage: 488.0+ bytes
```

```
0    1001
1    1002
2    1003
3    1004
4    1005
Name: Customer ID, dtype: int64
```

```
0    483000
1    401000
2    360000
3    169000
4    505000
dtype: int64
```

	10001	24/05/20	Wheat	4.8lb	€17	Fline Store
0	10002	05/05/20	Fruit Juice	3.1lb	€19	Dello Superstore
1	10003	27/04/20	Vegetables	1.2lb	€15	Javies Retail
2	10004	05/05/20	Oil	3.1lb	€17	Javies Retail
3	10005	27/04/20	Wheat	4.8lb	€13	Javies Retail
4	10006	14/01/20	Butter	3.6lb	€27	Oldi Superstore
5	10007	20/04/20	Oil	4.8lb	€21	Dello Superstore
6	10008	05/05/20	Wheat	3.6lb	€25	Oldi Superstore
7	10009	17/04/20	Fruits	1.2lb	€24	Oldi Superstore
8	10010	15/06/20	Oil	4.4lb	€25	Kanes Store
9	10011	17/06/20	Oil	4.4lb	€16	Fline Store
10	10012	11/06/20	Cheese	2.3lb	€20	Fline Store
11	10013	19/03/20	Rice	4.4lb	€27	Kanes Store
12	10014	01/01/20	Cheese	1.2lb	€10	Fline Store
13	10015	07/07/20	Fruit Juice	3.6lb	€27	Oldi Superstore

```
['Customer ID',
 'Customer Name',
 '2018 Revenue',
 '2019 Revenue',
 'Growth',
 'Start Year',
 'Start Month',
 'Start Day',
 'New Customer']
```

	0	1	2	3	4	5
0	10001	24/05/20	Wheat	4.8lb	€17	Fline Store
1	10002	05/05/20	Fruit Juice	3.1lb	€19	Dello Superstore
2	10003	27/04/20	Vegetables	1.2lb	€15	Javies Retail
3	10004	05/05/20	Oil	3.1lb	€17	Javies Retail
4	10005	27/04/20	Wheat	4.8lb	€13	Javies Retail
5	10006	14/01/20	Butter	3.6lb	€27	Oldi Superstore
6	10007	20/04/20	Oil	4.8lb	€21	Dello Superstore
7	10008	05/05/20	Wheat	3.6lb	€25	Oldi Superstore
8	10009	17/04/20	Fruits	1.2lb	€24	Oldi Superstore
9	10010	15/06/20	Oil	4.4lb	€25	Kanes Store
10	10011	17/06/20	Oil	4.4lb	€16	Fline Store
11	10012	11/06/20	Cheese	2.3lb	€20	Fline Store
12	10013	19/03/20	Rice	4.4lb	€27	Kanes Store
13	10014	01/01/20	Cheese	1.2lb	€10	Fline Store
14	10015	07/07/20	Fruit Juice	3.6lb	€27	Oldi Superstore

	full_name	address	creation_date_time
0	Pasquale Cooper	1268 Burgoyne Promenade, San Leandro, Florida	2004-05-29 02:07:28
1	Giuseppe Wood	738 Opalo Circle, Brooklyn Center, Kansas	2008-04-24 19:42:11
2	Lindsey Garza	747 Desmond Nene, Olive Branch, Wisconsin	2013-08-23 09:41:48
3	Randy Mcpherson	171 Byron Street, Pleasanton, Vermont	2010-06-21 22:52:23
4	Cristobal Walsh	55 Crestwell Square, Oxford, Alaska	2014-12-13 09:47:34

	full_name	address	creation_date_time
0	Pasquale Cooper	1268 Burgoyne Promenade, San Leandro, Florida	2004-05-29 02:07:28
1	Giuseppe Wood	738 Opalo Circle, Brooklyn Center, Kansas	2008-04-24 19:42:11
2	Lindsey Garza	747 Desmond Nene, Olive Branch, Wisconsin	2013-08-23 09:41:48
3	Randy Mcpherson	171 Byron Street, Pleasanton, Vermont	2010-06-21 22:52:23
4	Cristobal Walsh	55 Crestwell Square, Oxford, Alaska	2014-12-13 09:47:34

	full_name	address	creation_date_time	first_name	last_name
0	Pasquale Cooper	1268 Burgoyne Promenade, San Leandro, Florida	2004-05-29 02:07:28	Pasquale	Cooper
1	Giuseppe Wood	738 Opalo Circle, Brooklyn Center, Kansas	2008-04-24 19:42:11	Giuseppe	Wood
2	Lindsey Garza	747 Desmond Nene, Olive Branch, Wisconsin	2013-08-23 09:41:48	Lindsey	Garza
3	Randy Mcpherson	171 Byron Street, Pleasanton, Vermont	2010-06-21 22:52:23	Randy	Mcpherson
4	Cristobal Walsh	55 Crestwell Square, Oxford, Alaska	2014-12-13 09:47:34	Cristobal	Walsh

	address	creation_date_time	first_name	last_name
0	1268 Burgoyne Promenade, San Leandro, Florida	2004-05-29 02:07:28	Pasquale	Cooper
1	738 Opalo Circle, Brooklyn Center, Kansas	2008-04-24 19:42:11	Giuseppe	Wood
2	747 Desmond Nene, Olive Branch, Wisconsin	2013-08-23 09:41:48	Lindsey	Garza
3	171 Byron Street, Pleasanton, Vermont	2010-06-21 22:52:23	Randy	Mcpherson
4	55 Crestwell Square, Oxford, Alaska	2014-12-13 09:47:34	Cristobal	Walsh

	creation_date_time	first_name	last_name	street	city	state
0	2004-05-29 02:07:28	Pasquale	Cooper	1268 Burgoyne Promenade	San Leandro	Florida
1	2008-04-24 19:42:11	Giuseppe	Wood	738 Opalo Circle	Brooklyn Center	Kansas
2	2013-08-23 09:41:48	Lindsey	Garza	747 Desmond Nene	Olive Branch	Wisconsin
3	2010-06-21 22:52:23	Randy	Mcpherson	171 Byron Street	Pleasanton	Vermont
4	2014-12-13 09:47:34	Cristobal	Walsh	55 Crestwell Square	Oxford	Alaska

```
0    2004-05-29 02:07:28
1    2008-04-24 19:42:11
2    2013-08-23 09:41:48
3    2010-06-21 22:52:23
4    2014-12-13 09:47:34
Name: creation_date_time, dtype: datetime64[ns]
```

	id	city	state	city	state
0	1	Hutchinson	Texas	Hutchinson	Texas
1	2	Yorkville	South Dakota	Yorkville	South Dakota
2	1	Hutchinson	Texas	Hutchinson	Texas
3	3	Round Lake	Kansas	Round Lake	Kansas
4	4	Orinda	Montana	Orinda	Montana
5	3	Round Lake	Kansas	Round Lake	Kansas

	id	city	state
0	1	Hutchinson	Texas
1	2	Yorkville	South Dakota
3	3	Round Lake	Kansas
4	4	Orinda	Montana

	id	city	state	city	state
0	1	Hutchinson	Texas	Hutchinson	Texas
1	2	Yorkville	South Dakota	Yorkville	South Dakota
2	1	Hutchinson	Texas	Hutchinson	Texas
3	3	Round Lake	Kansas	Round Lake	Kansas
4	4	Orinda	Montana	Orinda	Montana
5	3	Round Lake	Kansas	Round Lake	Kansas

```
array([False, False, False,  True,  True])
```

	id	city	state
0	1	Hutchinson	Texas
1	2	Yorkville	South Dakota
2	1	Hutchinson	Texas
3	3	Round Lake	Kansas
4	4	Orinda	Montana
5	3	Round Lake	Kansas

	id	city	state
0	1	Hutchinson	Texas
1	2	Yorkville	South Dakota
3	3	Round Lake	Kansas
4	4	Orinda	Montana

	0	1
0	Vernia Anthony	1051 Balceta Square, Reedley, Michigan
1	Daren Underwood	982 Duboce Gardens, Peachtree City, Georgia
2	Stanley Marks	541 Merrill Stravenue, Talladega, Pennsylvania
3	Shad Ruiz	1018 Whiting Line, North Platte, New Jersey
4	Danny Mooney	1301 Grand View Crescent, Oviedo, Washington

```
['full_name', 'address']
```

	full_name	address
0	Vernia Anthony	1051 Balceta Square, Reedley, Michigan
1	Daren Underwood	982 Duboce Gardens, Peachtree City, Georgia
2	Stanley Marks	541 Merrill Stravenue, Talladega, Pennsylvania
3	Shad Ruiz	1018 Whiting Line, North Platte, New Jersey
4	Danny Mooney	1301 Grand View Crescent, Oviedo, Washington

	full_name	address	first_name	last_name
0	Vernia Anthony	1051 Balceta Square, Reedley, Michigan	Vernia	Anthony
1	Daren Underwood	982 Duboce Gardens, Peachtree City, Georgia	Daren	Underwood
2	Stanley Marks	541 Merrill Stravenue, Talladega, Pennsylvania	Stanley	Marks
3	Shad Ruiz	1018 Whiting Line, North Platte, New Jersey	Shad	Ruiz
4	Danny Mooney	1301 Grand View Crescent, Oviedo, Washington	Danny	Mooney

	address	first_name	last_name
0	1051 Balceta Square, Reedley, Michigan	Vernia	Anthony
1	982 Duboce Gardens, Peachtree City, Georgia	Daren	Underwood
2	541 Merrill Stravenue, Talladega, Pennsylvania	Stanley	Marks
3	1018 Whiting Line, North Platte, New Jersey	Shad	Ruiz
4	1301 Grand View Crescent, Oviedo, Washington	Danny	Mooney

	first_name	last_name	street	city	state
0	Vernia	Anthony	1051 Balceta Square	Reedley	Michigan
1	Daren	Underwood	982 Duboce Gardens	Peachtree City	Georgia
2	Stanley	Marks	541 Merrill Stravenue	Talladega	Pennsylvania
3	Shad	Ruiz	1018 Whiting Line	North Platte	New Jersey
4	Danny	Mooney	1301 Grand View Crescent	Oviedo	Washington

	YEAR	M0-24	M25-54	M55	F0-24	F25-54	F55
0	2018	282	812	993	712	466	373
1	2019	243	196	365	340	969	659

	YEAR	M0-24	M25-54	M55	F0-24	F25-54	F55
0	2018	282	812	993	712	466	373
1	2019	243	196	365	340	969	659

	YEAR	demographic	sales
0	2018	M0-24	282
1	2019	M0-24	243
2	2018	M25-54	812
3	2019	M25-54	196
4	2018	M55	993
5	2019	M55	365
6	2018	F0-24	712
7	2019	F0-24	340
8	2018	F25-54	466
9	2019	F25-54	969
10	2018	F55	373
11	2019	F55	659

	YEAR	sales	gender	age_group
0	2018	282	M	0-24
1	2019	243	M	0-24
2	2018	812	M	25-54
3	2019	196	M	25-54
4	2018	993	M	55
5	2019	365	M	55
6	2018	712	F	0-24
7	2019	340	F	0-24
8	2018	466	F	25-54
9	2019	969	F	25-54
10	2018	373	F	55
11	2019	659	F	55

	store_id	sales
0	1	282
1	2	243
2	3	391
3	4	973

	store_id	sales
0	1	272
1	2	370
2	3	178
3	4	622

	store_id	sales	year
0	1	282	2018
1	1	272	2019
2	2	243	2018
3	2	370	2019
4	3	391	2018
5	3	178	2019
6	4	973	2018
7	4	622	2019

	store_id	sales
0	1	282
1	2	243
2	3	391
3	4	973

	store_id	sales
0	1	272
1	2	370
2	3	178
3	4	622

	store_id	sales	year
0	1	282	2018
1	2	243	2018
2	3	391	2018
3	4	973	2018

	store_id	sales	year
0	1	272	2019
1	2	370	2019
2	3	178	2019
3	4	622	2019

	store_id	sales	year
0	1	282	2018
1	2	243	2018
2	3	391	2018
3	4	973	2018
0	1	272	2019
1	2	370	2019
2	3	178	2019
3	4	622	2019

	store_id	sales	year
0	1	282	2018
0	1	272	2019
1	2	243	2018
1	2	370	2019
2	3	391	2018
2	3	178	2019
3	4	973	2018
3	4	622	2019

	store_id	sales	year
0	1	282	2018
1	1	272	2019
2	2	243	2018
3	2	370	2019
4	3	391	2018
5	3	178	2019
6	4	973	2018
7	4	622	2019

	store_id	M0-24	M25-54	M55	F0-24	F25-54	F55
0	1	34	27	60	54	17	98
1	2	54	73	89	25	12	78
2	3	86	66	68	81	32	75
3	4	19	58	55	37	70	12
4	5	91	17	46	67	19	14

	store_id	M0-24	M25-54	M55	F0-24	F25-54	F55
0	1	46	16	28	62	98	76
1	2	44	92	60	26	86	50
2	3	53	85	50	84	34	44
3	4	88	71	45	48	19	34
4	5	37	18	45	45	10	11

	store_id	M0-24	M25-54	M55	F0-24	F25-54	F55	year
0	1	34	27	60	54	17	98	2018
1	2	54	73	89	25	12	78	2018
2	3	86	66	68	81	32	75	2018
3	4	19	58	55	37	70	12	2018
4	5	91	17	46	67	19	14	2018

	store_id	M0-24	M25-54	M55	F0-24	F25-54	F55	year
0	1	46	16	28	62	98	76	2019
1	2	44	92	60	26	86	50	2019
2	3	53	85	50	84	34	44	2019
3	4	88	71	45	48	19	34	2019
4	5	37	18	45	45	10	11	2019

	store_id	M0-24	M25-54	M55	F0-24	F25-54	F55	year
0	1	34	27	60	54	17	98	2018
1	2	54	73	89	25	12	78	2018
2	3	86	66	68	81	32	75	2018
3	4	19	58	55	37	70	12	2018
4	5	91	17	46	67	19	14	2018
0	1	46	16	28	62	98	76	2019
1	2	44	92	60	26	86	50	2019
2	3	53	85	50	84	34	44	2019
3	4	88	71	45	48	19	34	2019
4	5	37	18	45	45	10	11	2019

	year	store_id	demographic	sales
0	2018	1	M0-24	34
1	2018	2	M0-24	54
2	2018	3	M0-24	86
3	2018	4	M0-24	19
4	2018	5	M0-24	91
5	2019	1	M0-24	46

	year	store_id	demographic	sales	gender	age_group
0	2018	1	M0-24	34	M	0-24
1	2018	2	M0-24	54	M	0-24
2	2018	3	M0-24	86	M	0-24
3	2018	4	M0-24	19	M	0-24
4	2018	5	M0-24	91	M	0-24
5	2019	1	M0-24	46	M	0-24

	year	store_id	sales	gender	age_group
0	2018	1	34	M	0-24
1	2018	2	54	M	0-24
2	2018	3	86	M	0-24
3	2018	4	19	M	0-24
4	2018	5	91	M	0-24

	year	store_id	sales	gender	age_group
30	2018	1	54	F	0-24
40	2018	1	17	F	25-54
50	2018	1	98	F	55
0	2018	1	34	M	0-24
10	2018	1	27	M	25-54

	store_id	age_group	gender	year	sales
30	1	0-24	F	2018	54
40	1	25-54	F	2018	17
50	1	55	F	2018	98
0	1	0-24	M	2018	34
10	1	25-54	M	2018	27

	store_id	age_group	gender	year	sales
0	1	0-24	F	2018	54
1	1	25-54	F	2018	17
2	1	55	F	2018	98
3	1	0-24	M	2018	34
4	1	25-54	M	2018	27

	id	city	state	population
0	1.0	Hutchinson	Texas	20938.0
1	NaN	Yorkville	Illinois	20119.0
2	3.0	Round Lake	Illinois	NaN
3	4.0	Orinda	California	19926.0

	city	state
0	Hutchinson	Texas
1	Yorkville	Illinois
2	Round Lake	Illinois
3	Orinda	California

	id	city	state	population
0	1.0	Hutchinson	Texas	20938.0
1	NaN	Yorkville	Illinois	20119.0
2	3.0	Round Lake	Illinois	NaN
3	4.0	Orinda	California	19926.0

	id	city	state	population
0	1.0	Hutchinson	Texas	20938.0
3	4.0	Orinda	California	19926.0

	id	city	state	population
0	1.0	Hutchinson	Texas	20938.0
1	NaN	Yorkville	Illinois	20119.0
3	4.0	Orinda	California	19926.0

	city	state
0	Hutchinson	Texas
1	Yorkville	Illinois
2	Round Lake	Illinois
3	Orinda	California

	id	city	state	population
0	1.0	Hutchinson	Texas	20938.0
1	NaN	Yorkville	Illinois	20119.0
2	3.0	Round Lake	Illinois	NaN
3	4.0	Orinda	NaN	19926.0

	id	city	state	population
0	1.0	Hutchinson	Texas	20938.0
1	-999.0	Yorkville	Illinois	20119.0
2	3.0	Round Lake	Illinois	-999.0
3	4.0	Orinda	Missing Value	19926.0

	id	city	state	population
0	1.0	Hutchinson	Texas	20938.0
1	NaN	Yorkville	Illinois	20119.0
2	3.0	Round Lake	Illinois	NaN
3	4.0	Orinda	NaN	19926.0

	id	city	state	population
1	NaN	Yorkville	Illinois	20119.0
2	3.0	Round Lake	Illinois	NaN
3	4.0	Orinda	NaN	19926.0

	id	city	state	population
0	1.0	Hutchinson	Texas	20938.0
1	NaN	Yorkville	Illinois	20119.0
2	3.0	Round Lake	Illinois	NaN
3	4.0	Orinda	Missing Value	19926.0

	id	city	state	population
0	1.0	Hutchinson	Texas	20938.0
1	-999.0	Yorkville	Illinois	20119.0
2	3.0	Round Lake	Illinois	NaN
3	4.0	Orinda	Missing Value	19926.0

	id	city	state	population
0	1.0	Hutchinson	Texas	20938.0
1	-999.0	Yorkville	Illinois	20119.0
2	3.0	Round Lake	Illinois	-999.0
3	4.0	Orinda	Missing Value	19926.0

	id	city	state	population
0	1.0	Hutchinson	Texas	20938.0
1	-999.0	Yorkville	Illinois	20119.0
2	3.0	Round Lake	Illinois	20328.0
3	4.0	Orinda	Missing Value	19926.0

	store_id	sales	year
0	1	282	2018
1	1	272	2019
2	2	243	2018
3	2	370	2019
4	3	391	2018
5	3	178	2019
6	4	973	2018
7	4	622	2019

	sales							
	count	mean	std	min	25%	50%	75%	max
store_id								
1	2.0	277.0	7.071068	272.0	274.50	277.0	279.50	282.0
2	2.0	306.5	89.802561	243.0	274.75	306.5	338.25	370.0
3	2.0	284.5	150.613744	178.0	231.25	284.5	337.75	391.0
4	2.0	797.5	248.194480	622.0	709.75	797.5	885.25	973.0

	store_id	sales	year
0	1	282	2018
1	1	272	2019
2	2	243	2018
3	2	370	2019
4	3	391	2018
5	3	178	2019
6	4	973	2018
7	4	622	2019

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7fddf8d71dd0>
```

	sales
store_id	
1	554
2	613
3	569
4	1595

	sales
store_id	
1	277.0
2	306.5
3	284.5
4	797.5

	sales
store_id	
1	272
2	243
3	178
4	622

	sales
store_id	
1	282
2	370
3	391
4	973

	sales
store_id	
1	7.071068
2	89.802561
3	150.613744
4	248.194480

	sales
store_id	
1	50.0
2	8064.5
3	22684.5
4	61600.5

	sales							
	count	mean	std	min	25%	50%	75%	max
store_id								
1	2.0	277.0	7.071068	272.0	274.50	277.0	279.50	282.0
2	2.0	306.5	89.802561	243.0	274.75	306.5	338.25	370.0
3	2.0	284.5	150.613744	178.0	231.25	284.5	337.75	391.0
4	2.0	797.5	248.194480	622.0	709.75	797.5	885.25	973.0

	sales				
	sum	mean	min	max	std
store_id					
1	554	277.0	272	282	7.071068
2	613	306.5	243	370	89.802561
3	569	284.5	178	391	150.613744
4	1595	797.5	622	973	248.194480

	brand	type	sales	units	year
0	Pandas	Product A	476	46	2010
1	Pandas	Product B	794	39	2010
2	Pandas	Product C	199	62	2010
3	Pandas	Product A	686	26	2011
4	Pandas	Product B	207	93	2011
5	Pandas	Product C	199	62	2011
6	Python	Product A	300	33	2010
7	Python	Product B	949	51	2010
8	Python	Product C	168	30	2010
9	Python	Product A	921	51	2011
10	Python	Product B	266	24	2011
11	Python	Product C	674	39	2011

		sum		min		max	
		sales	units	sales	units	sales	units
brand	type						
Pandas	Product A	1162	72	476	26	686	46
	Product B	1001	132	207	39	794	93
	Product C	398	124	199	62	199	62
Python	Product A	1221	84	300	33	921	51
	Product B	1215	75	266	24	949	51
	Product C	842	69	168	30	674	39
Total		5839	556	168	24	949	93

	brand	type	sales	units	year
0	Pandas	Product A	476	46	2010
1	Pandas	Product B	794	39	2010
2	Pandas	Product C	199	62	2010
3	Pandas	Product A	686	26	2011
4	Pandas	Product B	207	93	2011
5	Pandas	Product C	199	62	2011
6	Python	Product A	300	33	2010
7	Python	Product B	949	51	2010
8	Python	Product C	168	30	2010
9	Python	Product A	921	51	2011
10	Python	Product B	266	24	2011
11	Python	Product C	674	39	2011

	sales
brand	
Pandas	426.833333
Python	546.333333

	sales
brand	
Pandas	2561
Python	3278

	sum	min	max
	sales	sales	sales
brand			
Pandas	2561	199	794
Python	3278	168	949

	sum		min		max	
	sales	units	sales	units	sales	units
brand						
Pandas	2561	328	199	26	794	93
Python	3278	228	168	24	949	51

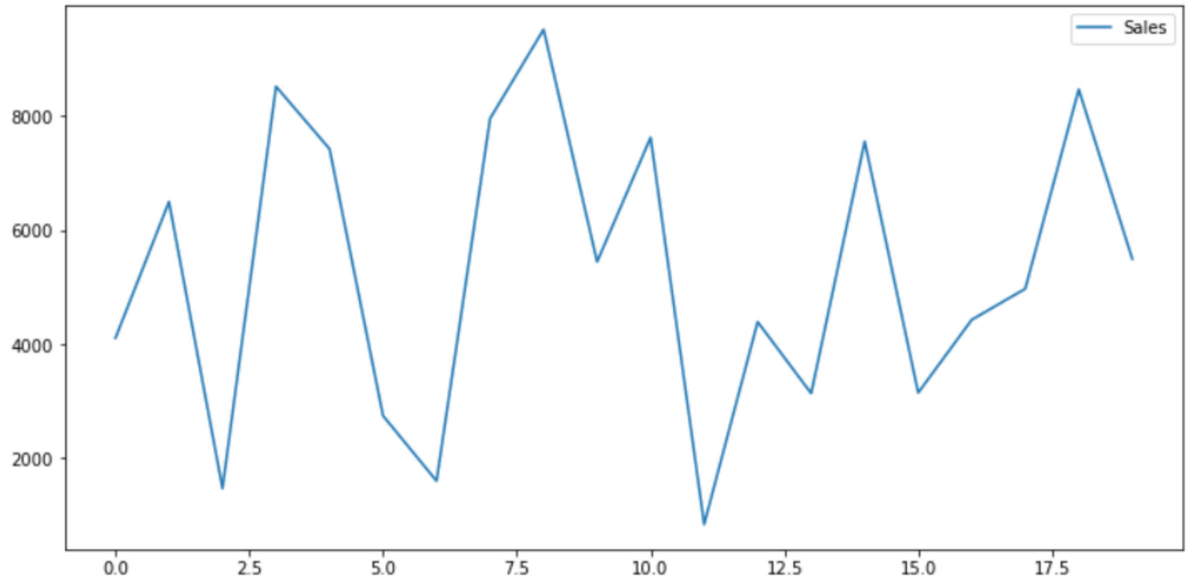
		sum		min		max	
		sales	units	sales	units	sales	units
type	brand						
Product A	Pandas	1162	72	476	26	686	46
	Python	1221	84	300	33	921	51
Product B	Pandas	1001	132	207	39	794	93
	Python	1215	75	266	24	949	51
Product C	Pandas	398	124	199	62	199	62
	Python	842	69	168	30	674	39

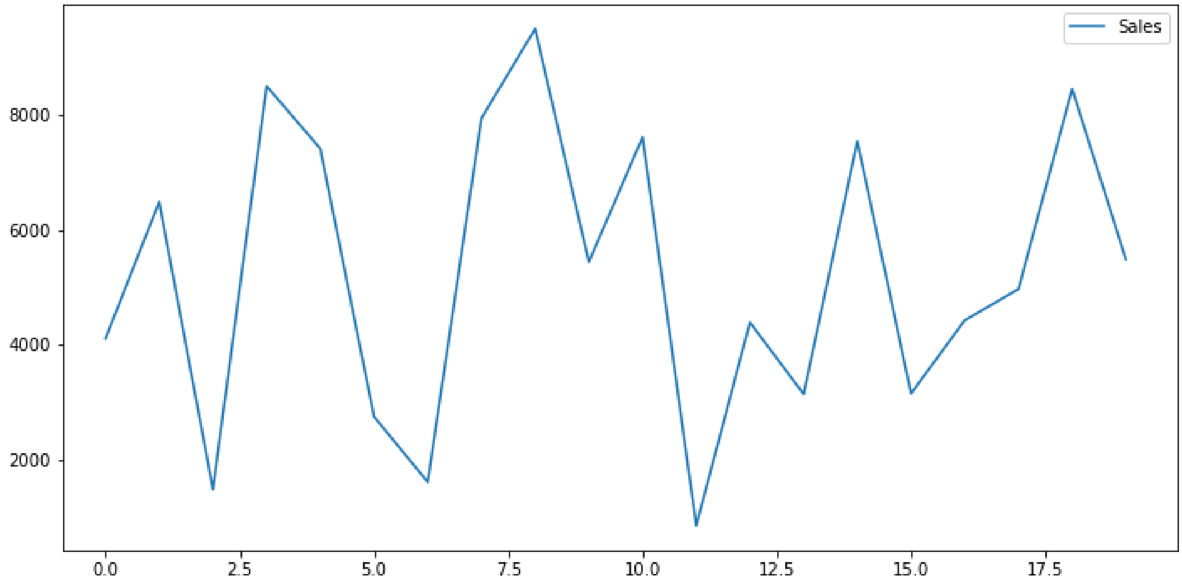
		sum		min		max	
		sales	units	sales	units	sales	units
brand	type						
Pandas	Product A	1162	72	476	26	686	46
	Product B	1001	132	207	39	794	93
	Product C	398	124	199	62	199	62
Python	Product A	1221	84	300	33	921	51
	Product B	1215	75	266	24	949	51
	Product C	842	69	168	30	674	39
Total		5839	556	168	24	949	93

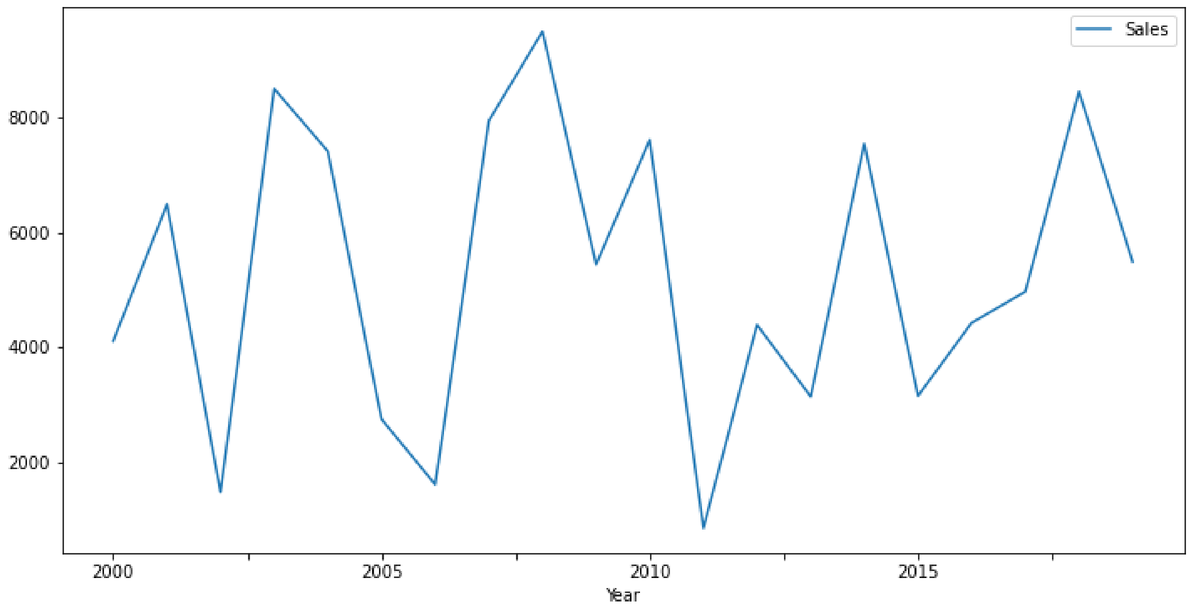
Chapter 8: Understanding Data Visualization

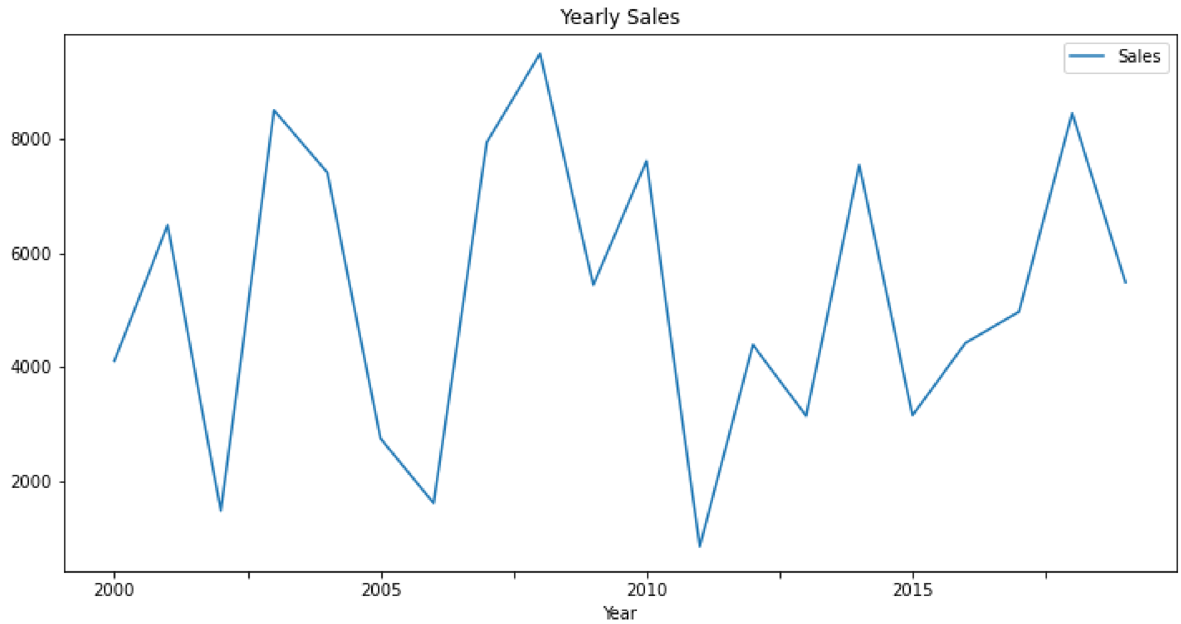
	Year	Sales
0	2000	4107
1	2001	6492
2	2002	1476
3	2003	8508
4	2004	7416
5	2005	2747
6	2006	1606
7	2007	7947
8	2008	9506
9	2009	5441
10	2010	7617
11	2011	847
12	2012	4389
13	2013	3139
14	2014	7546
15	2015	3150
16	2016	4426
17	2017	4969
18	2018	8457
19	2019	5491

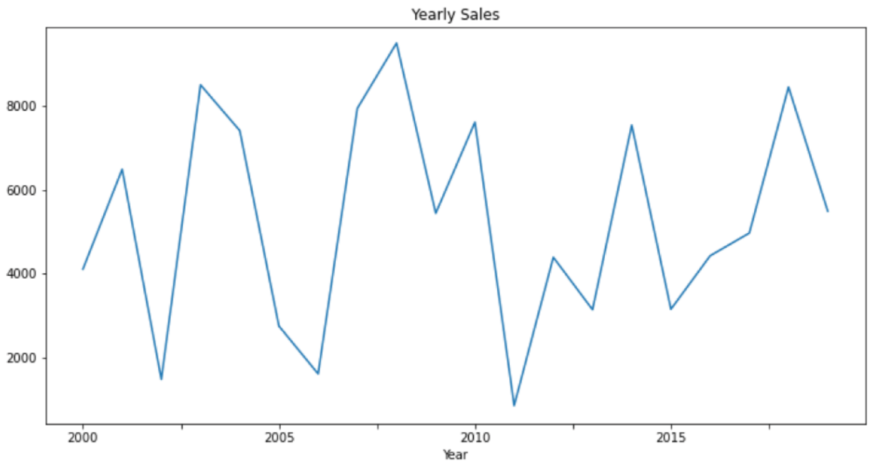
```
<AxesSubplot:>
```

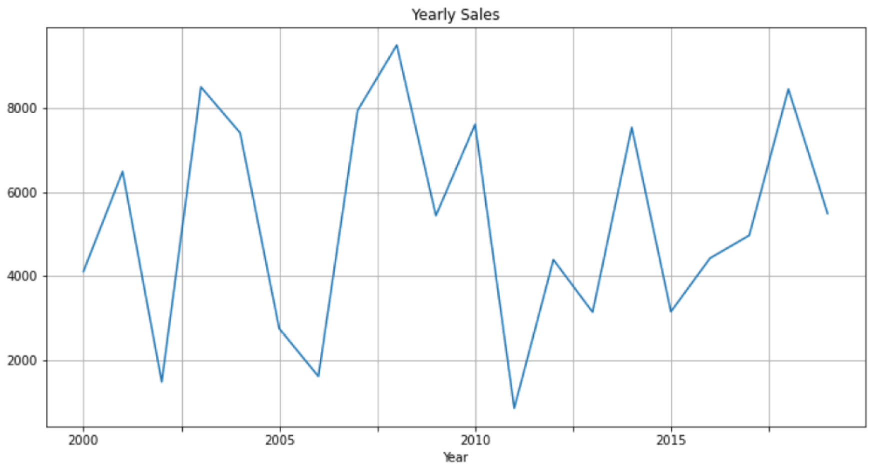


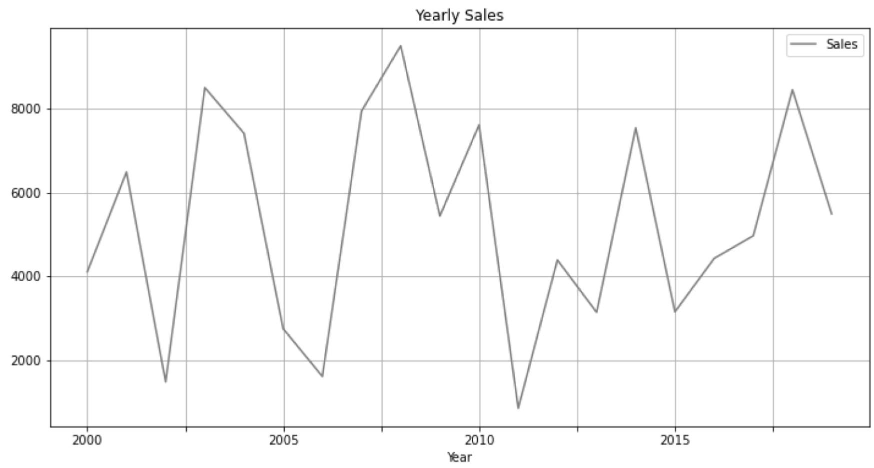

Sales
8000
6000
4000
2000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

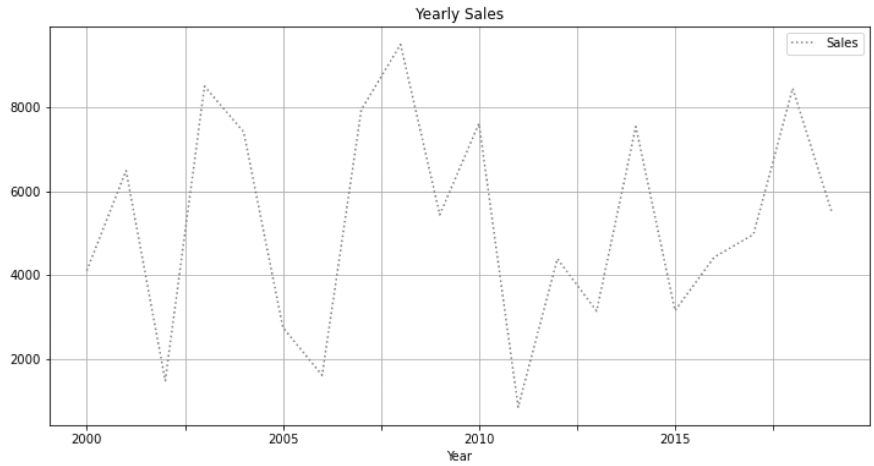

Sales
8000
6000
4000
2000
2000
2005
2010
2015
Year

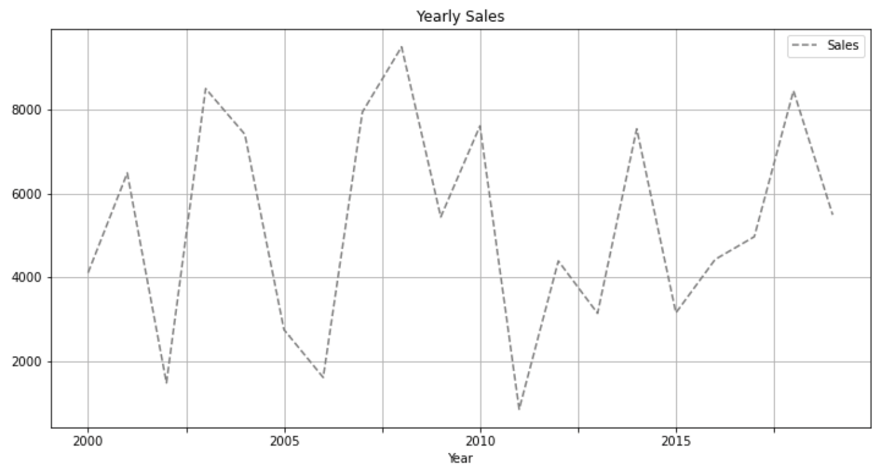

Yearly Sales
Sales
8000
6000
4000
2000
2000
2005
2010
2015
Year

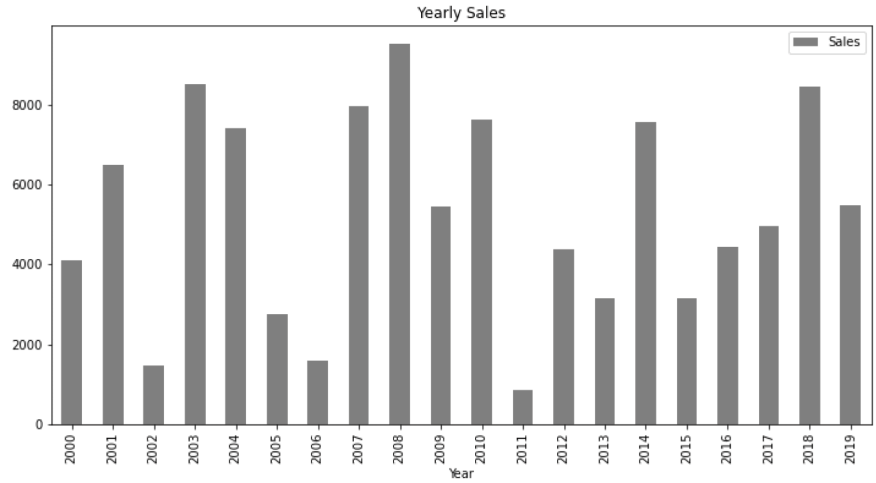

Yearly Sales
8000
6000
4000
2000
2000
2005
2010
2015
Year

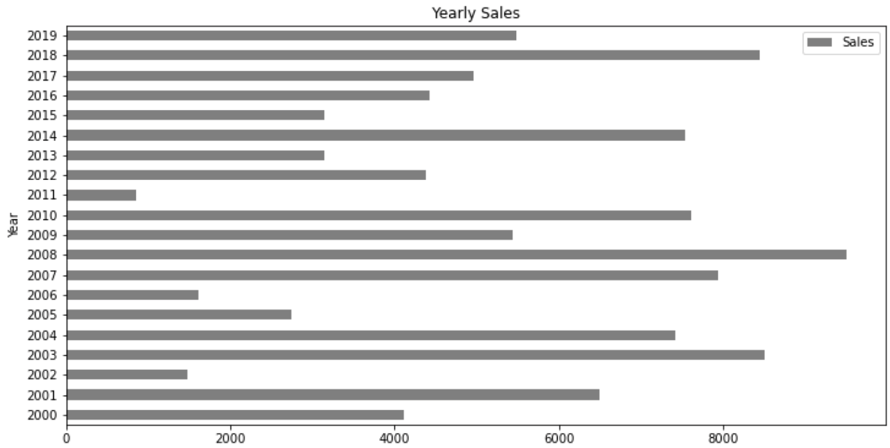

Yearly Sales
8000
6000
4000
2000
2000
2005
2010
2015
Year

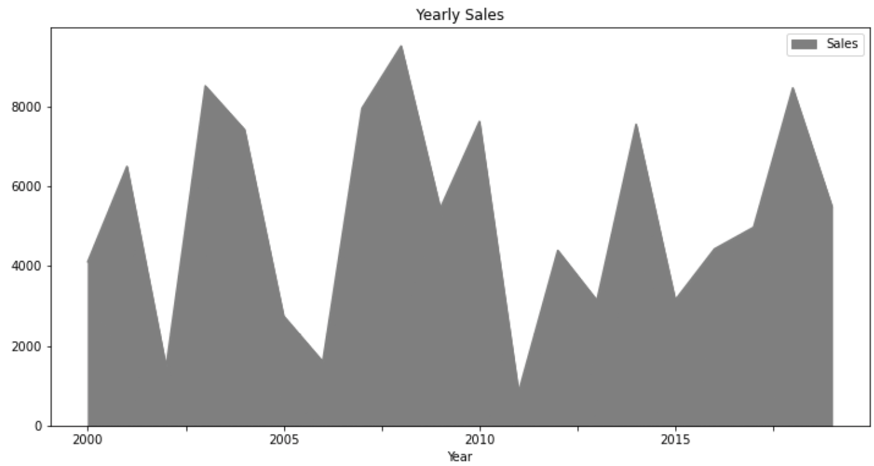

Yearly Sales
Sales
8000
6000
4000
2000
2000
2005
2010
2015
Year


Yearly Sales
Sales
8000
6000
4000
2000
2000
2005
2010
2015
Year


Yearly Sales
Sales
8000
6000
4000
2000
2000
2005
2010
2015
Year



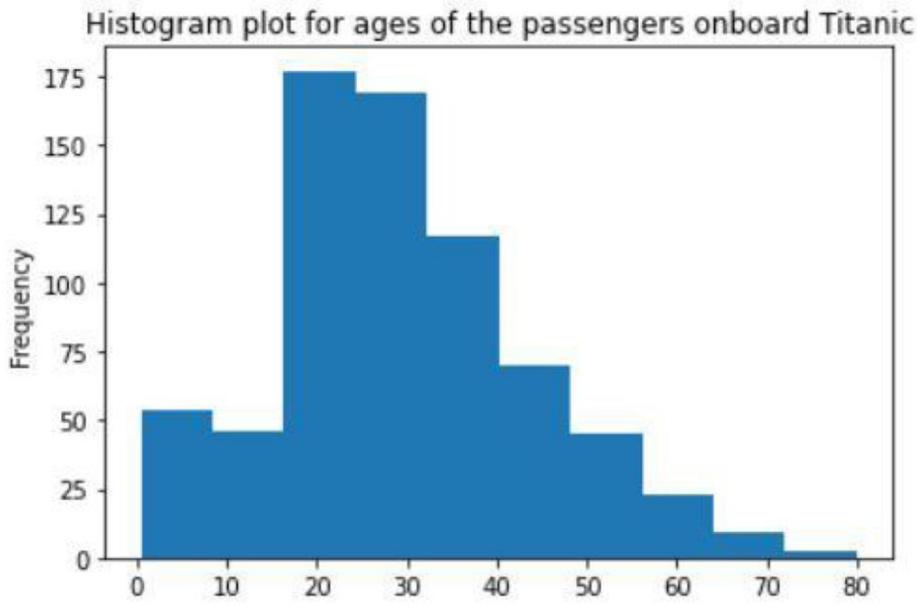


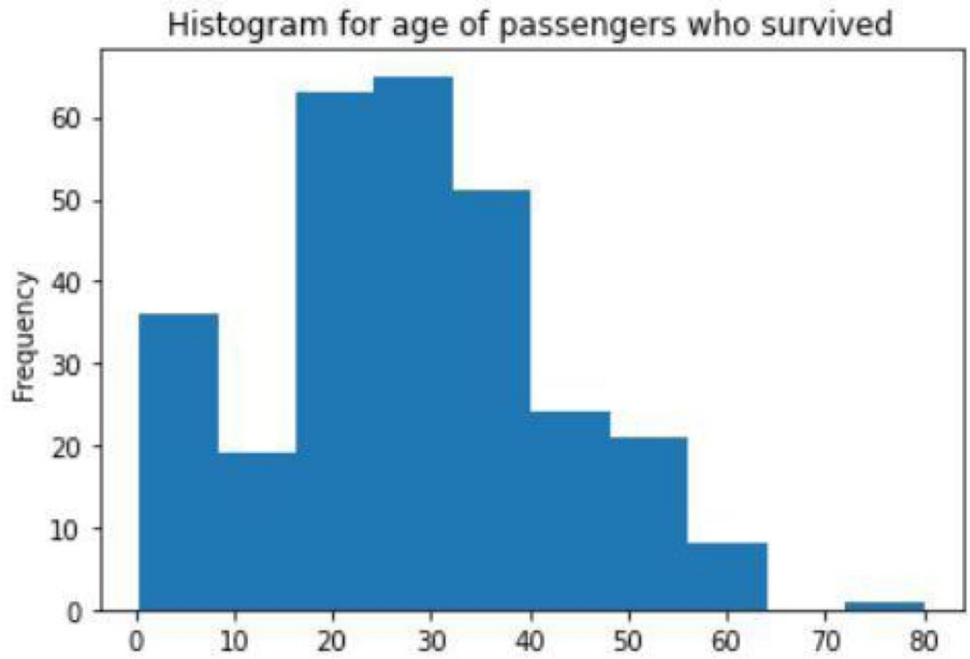


	survived	ticket_class	gender	age	number_sibling_spouse	number_parent_children	passenger_fare	port_of_embarkation	age_group
0	0	3	male	22.0	1	0	7.2500	S	18-59
1	1	1	female	38.0	1	0	71.2833	C	18-59
2	1	3	female	26.0	0	0	7.9250	S	18-59
3	1	1	female	35.0	1	0	53.1000	S	18-59
4	0	3	male	35.0	0	0	8.0500	S	18-59

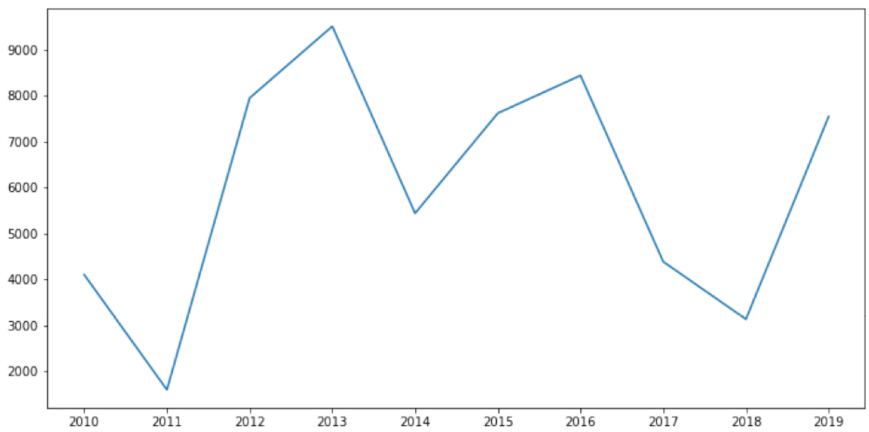
	survived	ticket_class	gender	age	number_sibling_spouse	number_parent_children	passenger_fare	port_of_embarkation	age_group
0	0	3	male	22.0	1	0	7.2500	S	18-59
1	1	1	female	38.0	1	0	71.2833	C	18-59
2	1	3	female	26.0	0	0	7.9250	S	18-59
3	1	1	female	35.0	1	0	53.1000	S	18-59
4	0	3	male	35.0	0	0	8.0500	S	18-59
...	...	...	...	...	...	...	...	...	...
885	0	3	female	39.0	0	5	29.1250	Q	18-59
886	0	2	male	27.0	0	0	13.0000	S	18-59
887	1	1	female	19.0	0	0	30.0000	S	18-59
889	1	1	male	26.0	0	0	30.0000	C	18-59
890	0	3	male	32.0	0	0	7.7500	Q	18-59

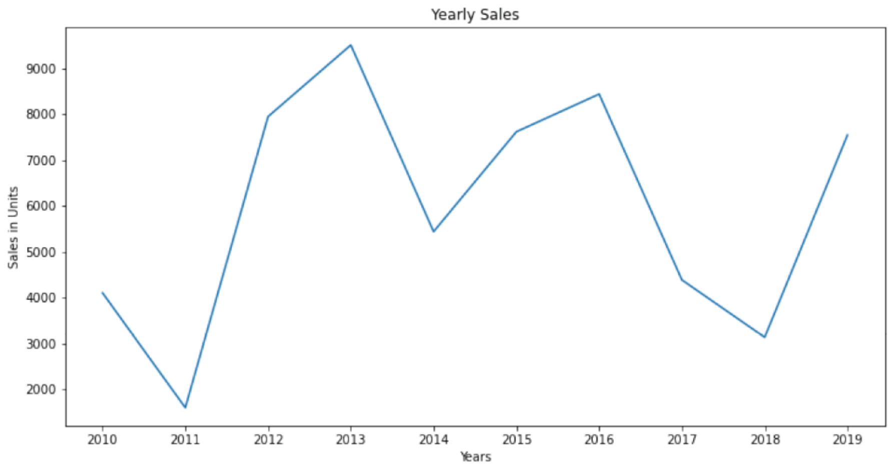
712 rows × 9 columns

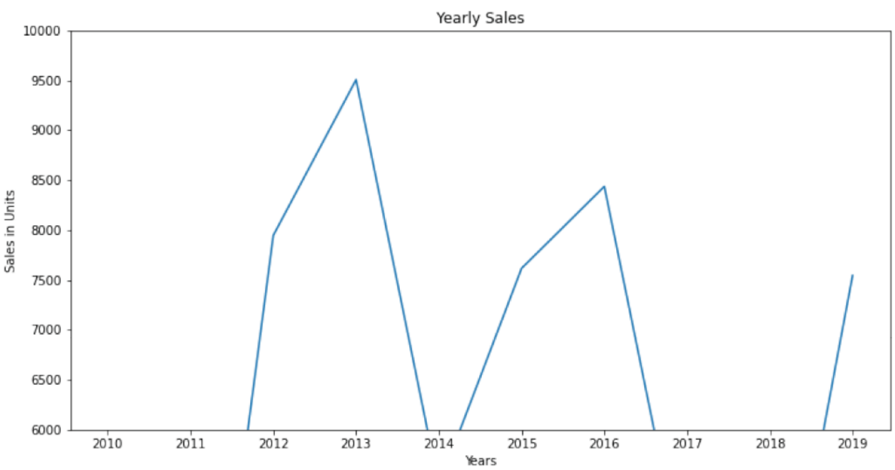


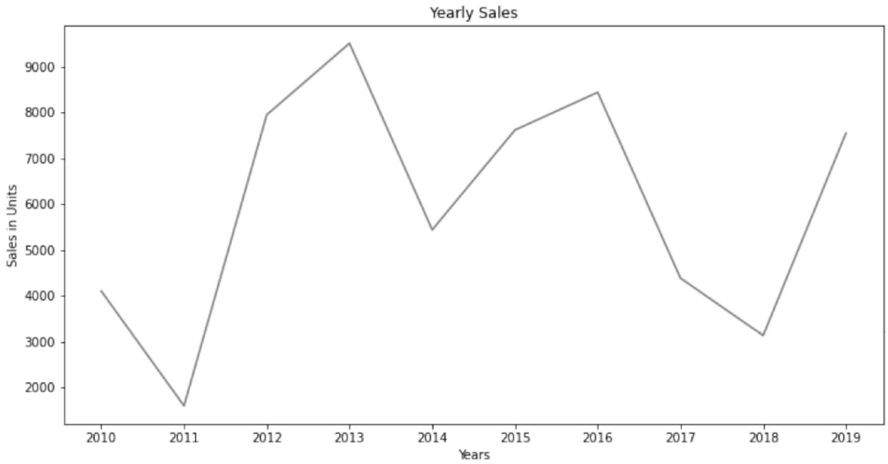


	Year	Sales
0	2010	4107
1	2011	1606
2	2012	7947
3	2013	9506
4	2014	5441
5	2015	7617
6	2016	8437
7	2017	4389
8	2018	3139
9	2019	7546

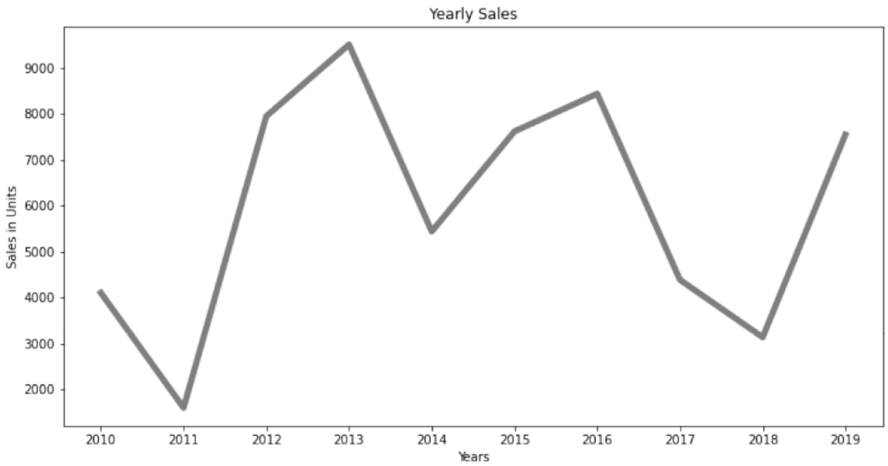




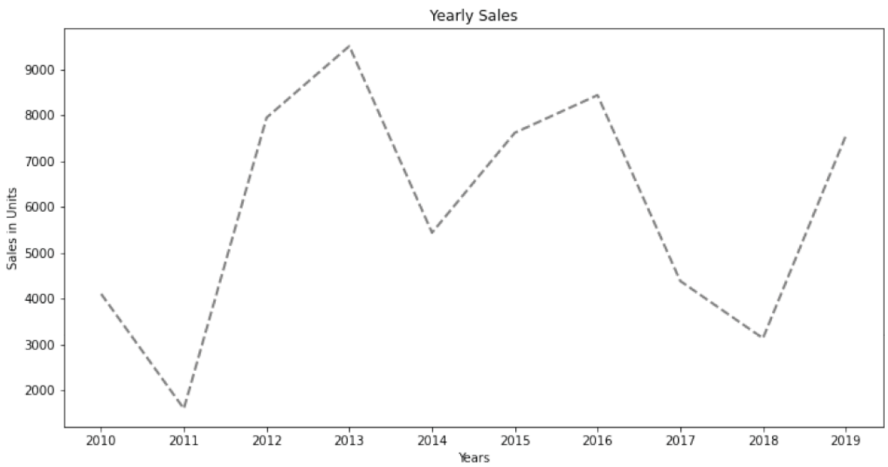




Yearly Sales
Sales in Units
Years
9000
8000
7000
6000
5000
4000
3000
2000
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019

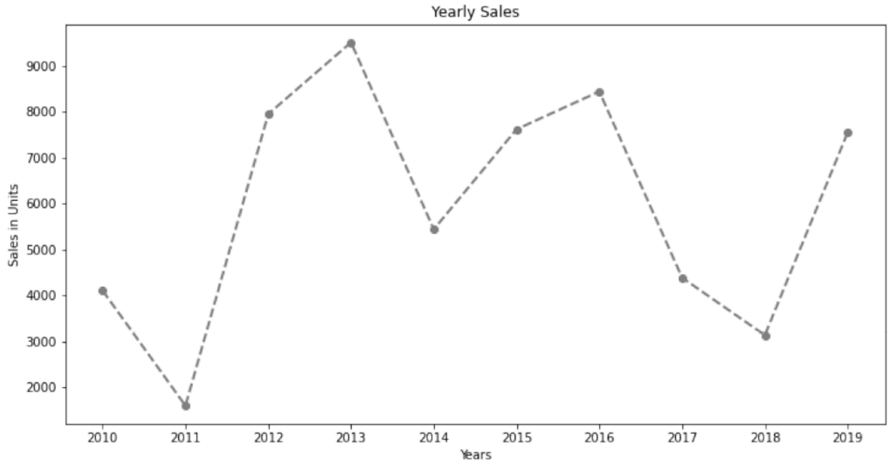


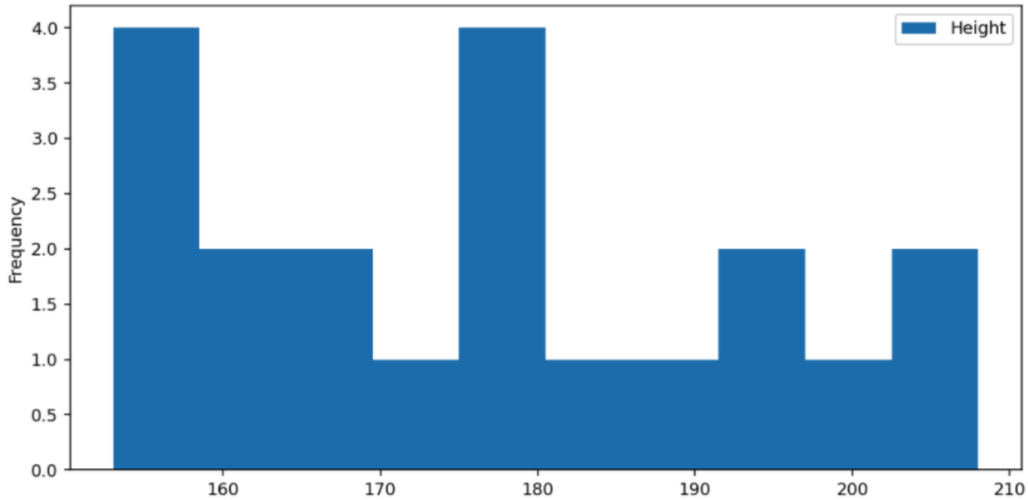
Yearly Sales
Sales in Units
Years
9000
8000
7000
6000
5000
4000
3000
2000
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019



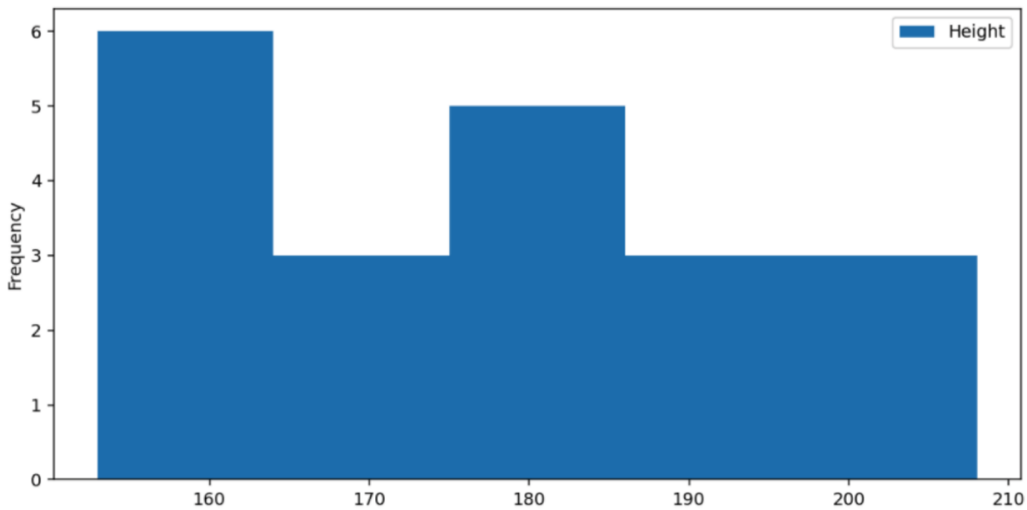
Yearly Sales
Sales in Units
Years
9000
8000
7000
6000
5000
4000
3000
2000
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019

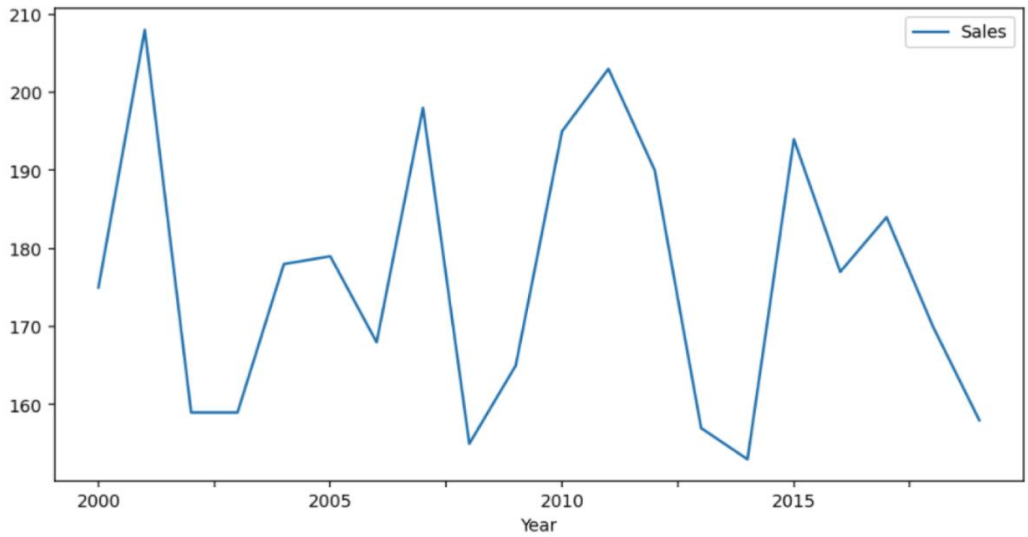
	Height
0	175
1	208
2	159
3	159
4	178
5	179
6	168
7	198
8	155
9	165
10	195
11	203
12	190
13	157
14	153
15	194
16	177
17	184
18	170
19	158



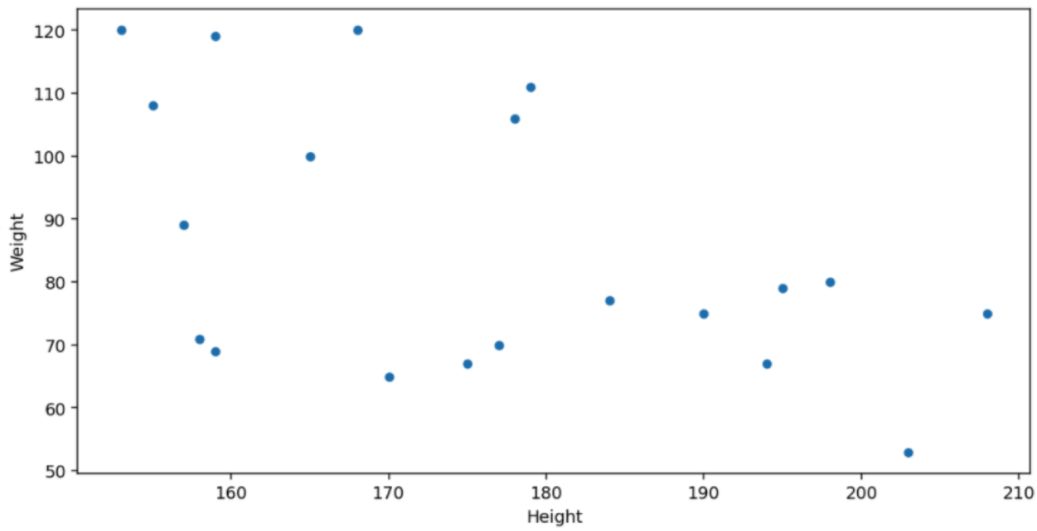


	Year	Sales
0	2000	175
1	2001	208
2	2002	159
3	2003	159
4	2004	178
5	2005	179
6	2006	168
7	2007	198
8	2008	155
9	2009	165
10	2010	195
11	2011	203
12	2012	190
13	2013	157
14	2014	153
15	2015	194
16	2016	177
17	2017	184
18	2018	170
19	2019	158

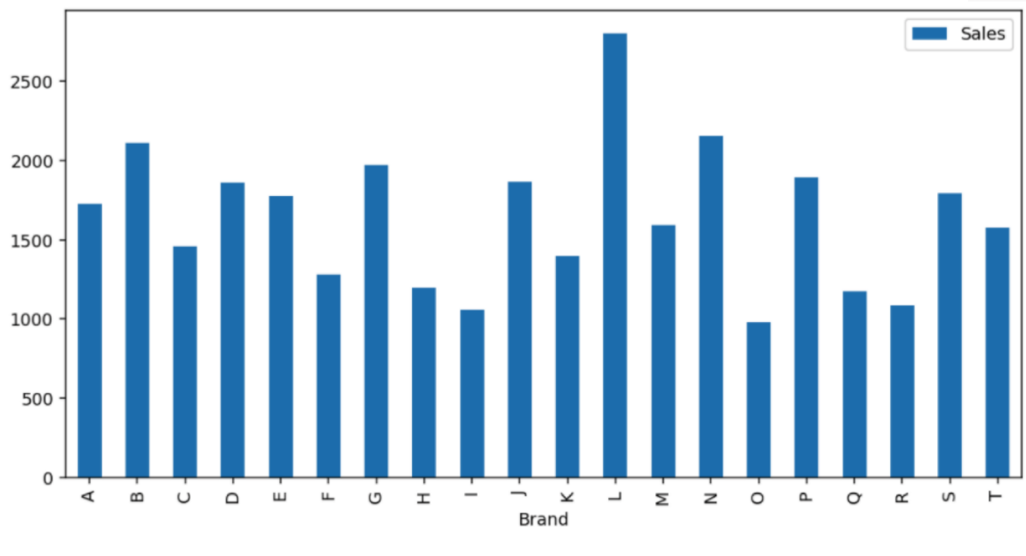




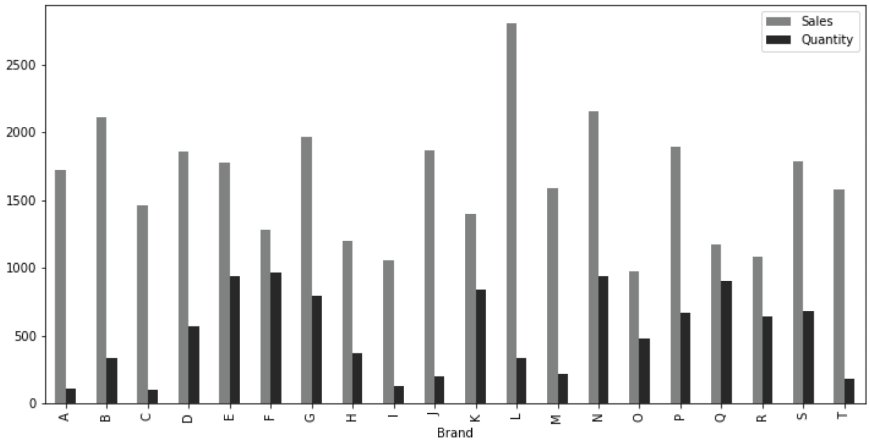
	Weight	Height
0	67	175
1	75	208
2	119	159
3	69	159
4	106	178
5	111	179
6	120	168
7	80	198
8	108	155
9	100	165
10	79	195
11	53	203
12	75	190
13	89	157
14	120	153
15	67	194
16	70	177
17	77	184
18	65	170
19	71	158



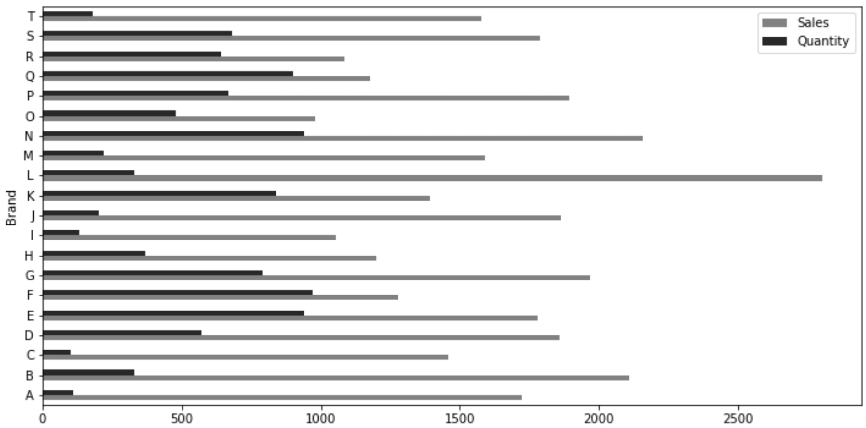
	Brand	Sales
0	A	1725
1	B	2108
2	C	1459
3	D	1859
4	E	1778
5	F	1279
6	G	1968
7	H	1198
8	I	1055
9	J	1865
10	K	1395
11	L	2803
12	M	1590
13	N	2157
14	O	978
15	P	1894
16	Q	1177
17	R	1084
18	S	1790
19	T	1578

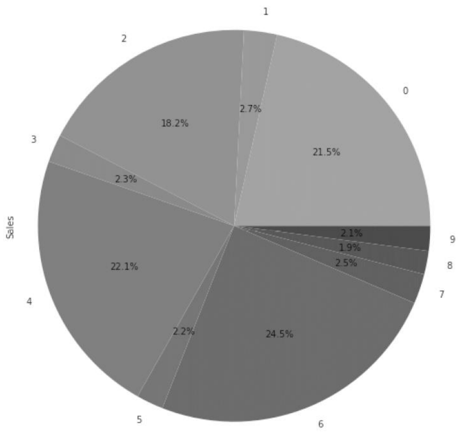


	Brand	Sales	Quantity
0	A	1725	110
1	B	2108	330
2	C	1459	100
3	D	1859	570
4	E	1778	940
5	F	1279	970
6	G	1968	790
7	H	1198	370
8	I	1055	130
9	J	1865	200
10	K	1395	840
11	L	2803	330
12	M	1590	220
13	N	2157	940
14	O	978	480
15	P	1894	670
16	Q	1177	900
17	R	1084	640
18	S	1790	680
19	T	1578	180

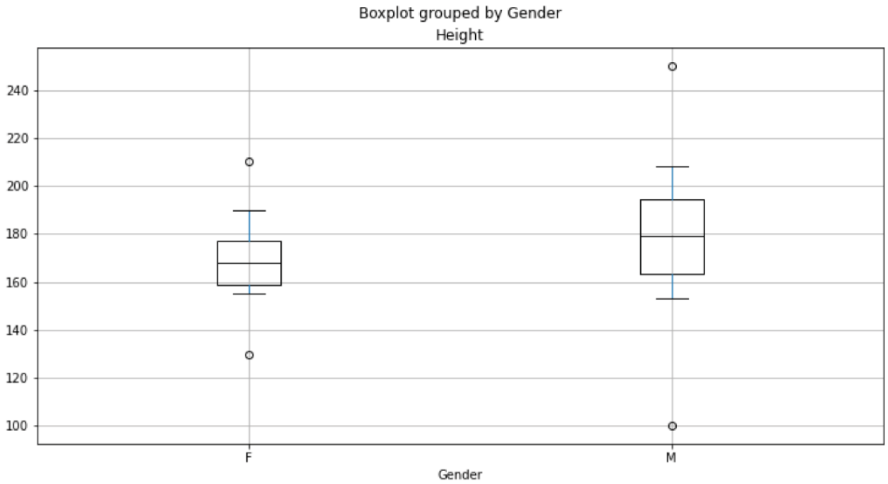


	Brand	Sales
0	1	1725
1	2	218
2	3	1459
3	4	185
4	5	1778
5	6	179
6	7	1968
7	8	198
8	9	155
9	10	165





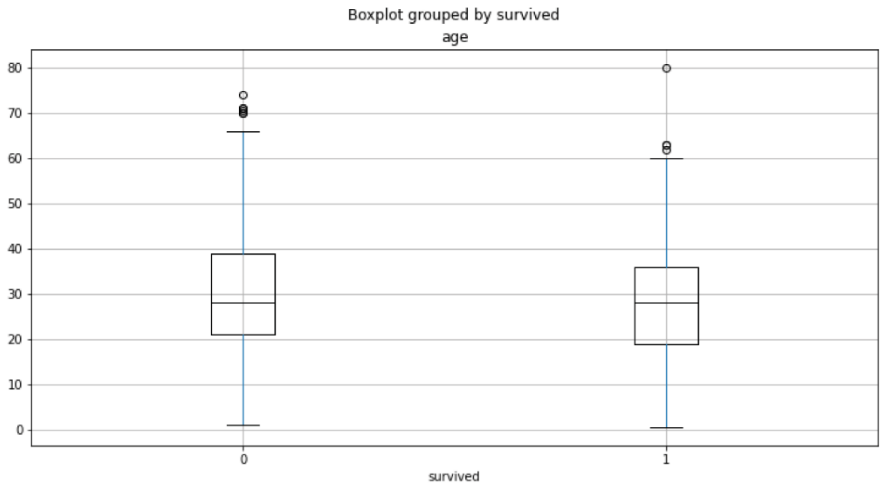
	Height	Gender
0	175	F
1	208	M
2	159	F
3	130	F
4	178	M
5	179	M
6	168	F
7	100	M
8	155	F
9	165	F
10	195	M
11	250	M
12	190	F
13	157	M
14	153	M
15	194	M
16	177	F
17	184	M
18	170	M
19	210	F

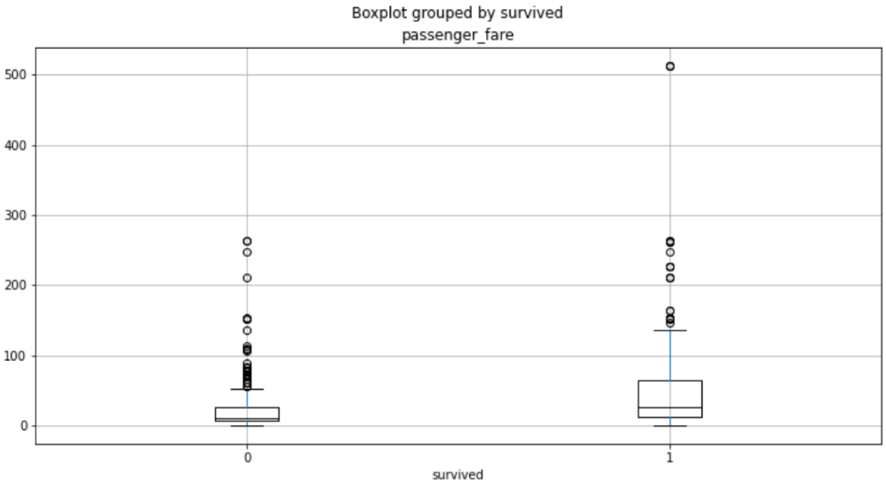


	survived	ticket_class	gender	age	number_sibling_spouse	number_parent_children	passenger_fare	port_of_embarkation	age_group
0	0	3	male	22.0	1	0	7.2500	S	18-59
1	1	1	female	38.0	1	0	71.2833	C	18-59
2	1	3	female	26.0	0	0	7.9250	S	18-59
3	1	1	female	35.0	1	0	53.1000	S	18-59
4	0	3	male	35.0	0	0	8.0500	S	18-59

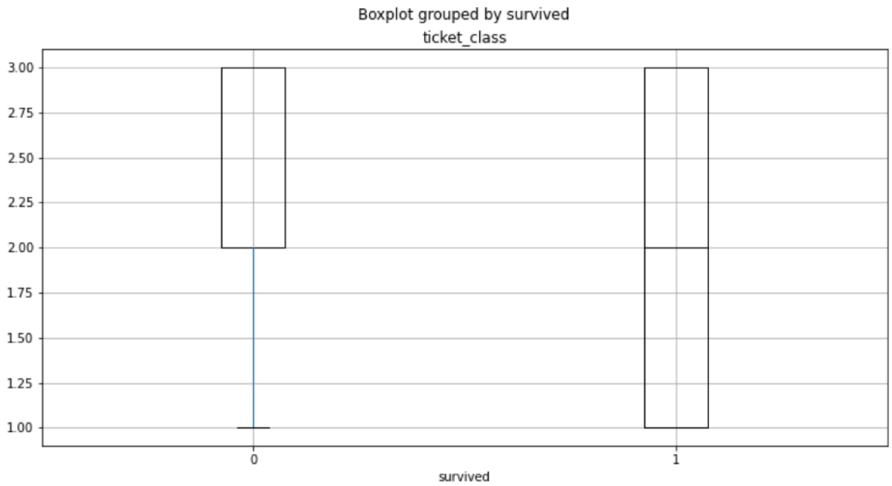
	survived	ticket_class	gender	age	number_sibling_spouse	number_parent_children	passenger_fare	port_of_embarkation	age_group
0	0	3	male	22.0	1	0	7.2500	S	18-59
1	1	1	female	38.0	1	0	71.2833	C	18-59
2	1	3	female	26.0	0	0	7.9250	S	18-59
3	1	1	female	35.0	1	0	53.1000	S	18-59
4	0	3	male	35.0	0	0	8.0500	S	18-59
...	...	...	...	...	...	...	...	...	...
885	0	3	female	39.0	0	5	29.1250	Q	18-59
886	0	2	male	27.0	0	0	13.0000	S	18-59
887	1	1	female	19.0	0	0	30.0000	S	18-59
889	1	1	male	26.0	0	0	30.0000	C	18-59
890	0	3	male	32.0	0	0	7.7500	Q	18-59

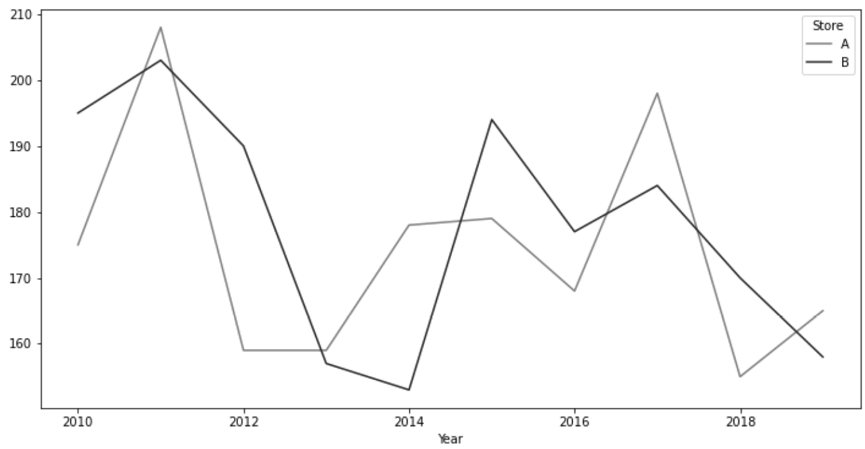
712 rows × 9 columns

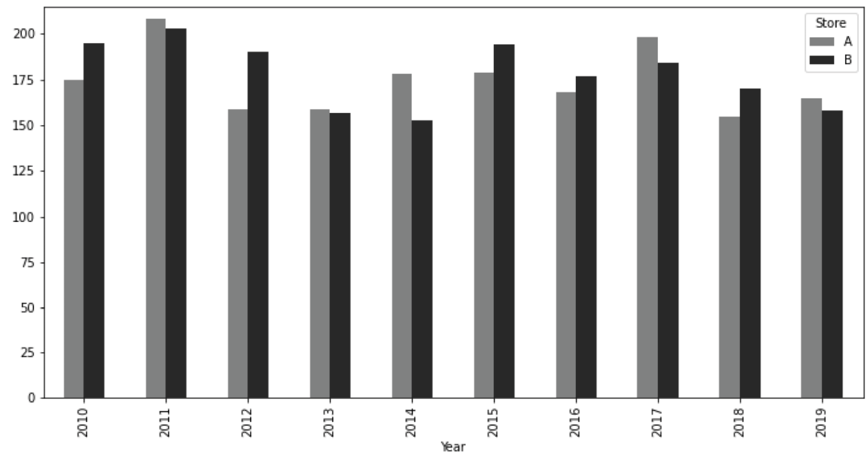




	Store	Year	Sales
0	A	2010	175
1	A	2011	208
2	A	2012	159
3	A	2013	159
4	A	2014	178
5	A	2015	179
6	A	2016	168
7	A	2017	198
8	A	2018	155
9	A	2019	165
10	B	2010	195
11	B	2011	203
12	B	2012	190
13	B	2013	157
14	B	2014	153
15	B	2015	194
16	B	2016	177
17	B	2017	184
18	B	2018	170
19	B	2019	158

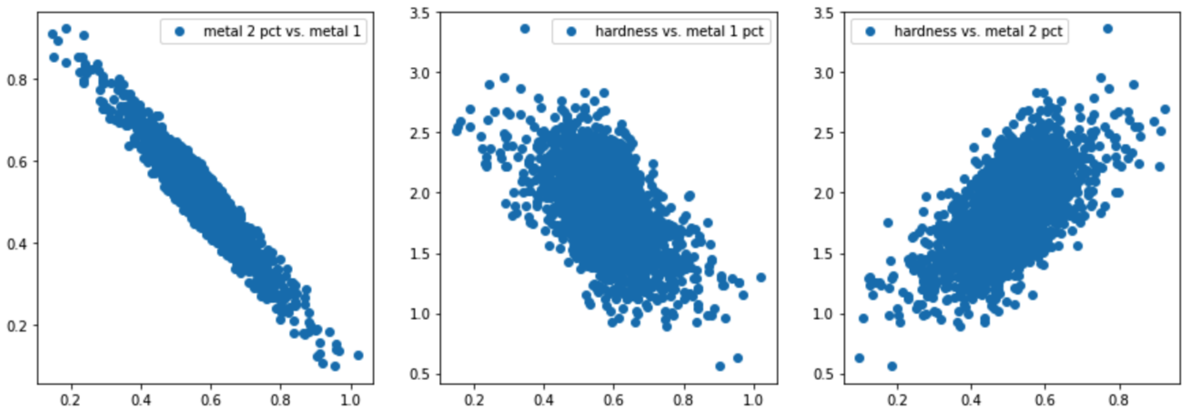



Store
A
B
210
200
190
180
170
160
2010
2012
2014
2016
2018
Year


Store
A
B
200
175
150
125
100
75
50
25
0
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
Year

Out[2]:

	metal_1	metal_2	alloy_hardness
0	0.958000	0.140659	1.254157
1	0.920147	0.107089	0.956846
2	0.590646	0.483316	1.952517
3	0.787427	0.239446	1.636522
4	0.223974	0.817454	2.367797
5	0.339729	0.694622	2.115060
6	0.242666	0.837370	2.899579
7	0.721072	0.365196	1.758518
8	0.666492	0.430698	1.591216
9	0.650387	0.414661	1.780010



```
                            OLS Regression Results
==============================================================================
Dep. Variable:          alloy_hardness   R-squared:                       0.394
Model:                             OLS   Adj. R-squared:                  0.394
Method:                  Least Squares   F-statistic:                     929.6
Date:                 Sun, 01 Aug 2021   Prob (F-statistic):          1.23e-311
Time:                         10:02:39   Log-Likelihood:                -44.409
No. Observations:                 2858   AIC:                             94.82
Df Residuals:                     2855   BIC:                             112.7
Df Model:                            2
Covariance Type:             nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.3434      0.147     -2.339      0.019      -0.631      -0.055
metal_1        1.1086      0.139      7.951      0.000       0.835       1.382
metal_2        3.0783      0.136     22.618      0.000       2.811       3.345
==============================================================================
Omnibus:                        1.075   Durbin-Watson:                   2.016
Prob(Omnibus):                  0.584   Jarque-Bera (JB):                1.023
Skew:                           0.044   Prob(JB):                        0.600
Kurtosis:                       3.031   Cond. No.                         66.1
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
const     -0.343381
metal_1    1.108639
metal_2    3.078313
dtype: float64
```

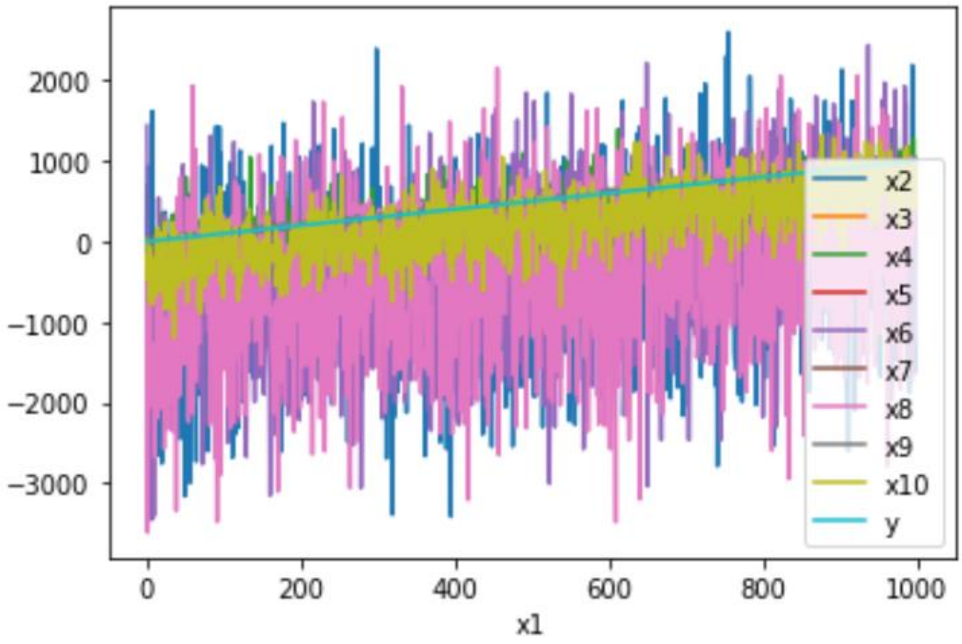
	x1	x2	x3	x4	x5	x6	x7 \
0	0	690.303674	-31.486707	-731.643758	17.270299	1436.411756	-10.636448
1	1	-685.241074	5.190095	-458.895861	30.668140	-716.580334	-1.656434
2	2	936.292932	39.294584	-712.144359	-22.953210	-122.183985	37.454739
3	3	-1798.095409	26.222489	-269.751619	15.487410	-464.936948	-12.999561
4	4	-2114.215496	34.656115	-480.576137	11.174598	-768.245414	40.964968

	x8	x9	x10	y
0	-492.026404	-46.120055	22.754113	0
1	-3610.645334	-30.322747	-472.866262	1
2	-762.459068	35.124581	-170.442837	2
3	-2052.517125	29.676836	-758.140719	3
4	-679.874801	-10.935731	-69.331760	4

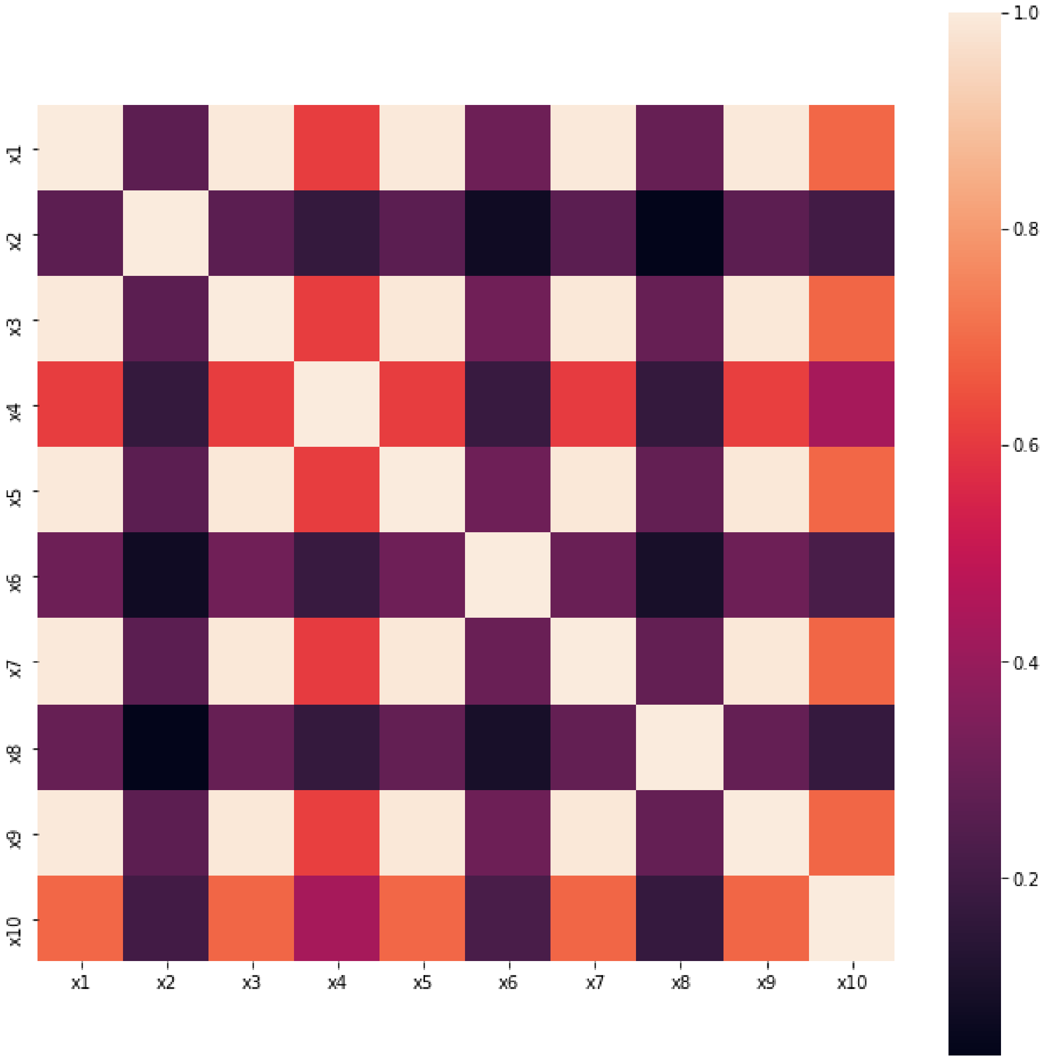
	x1	x2	x3	x4	x5	x6 \
995	995	-298.409060	1010.695516	814.386703	989.210703	782.825485
996	996	-286.163537	1041.661462	1279.540113	979.440118	-1432.060863
997	997	1018.026789	953.895802	805.609186	986.457232	752.152415
998	998	-1630.960898	953.705472	1000.300624	1000.994059	236.396803
999	999	-1273.687387	977.218707	540.179294	965.136736	-51.844077

	x7	x8	x9	x10	y
995	1013.443536	94.047877	972.315962	463.710712	995
996	996.322041	683.241456	966.951922	962.712279	996
997	1023.514885	378.872916	992.532875	797.471943	997
998	996.529063	1041.017110	1038.843755	988.338761	998
999	963.938164	914.585894	959.448032	430.768433	999

Out[13]: <AxesSubplot:xlabel='x1'>

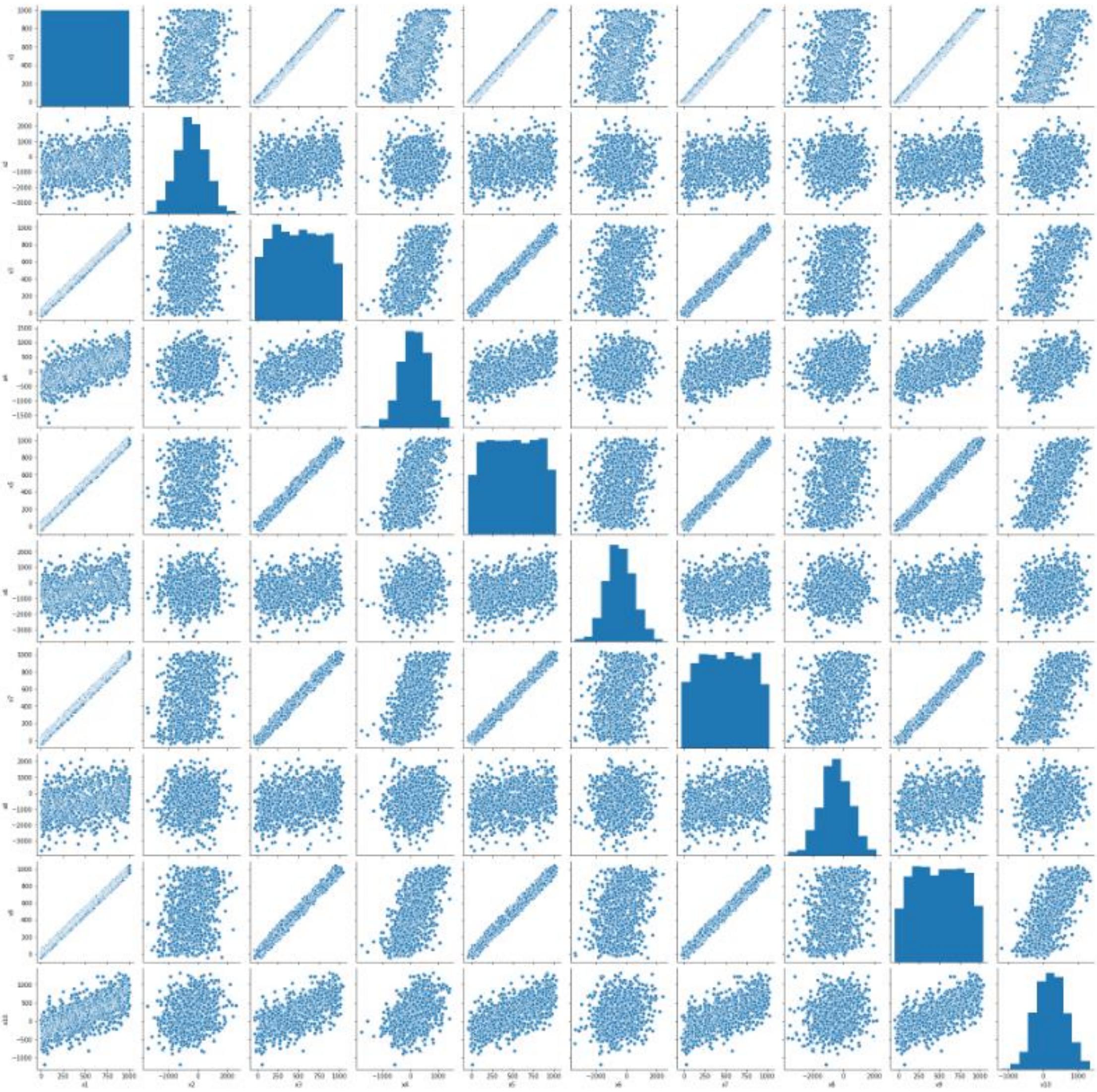


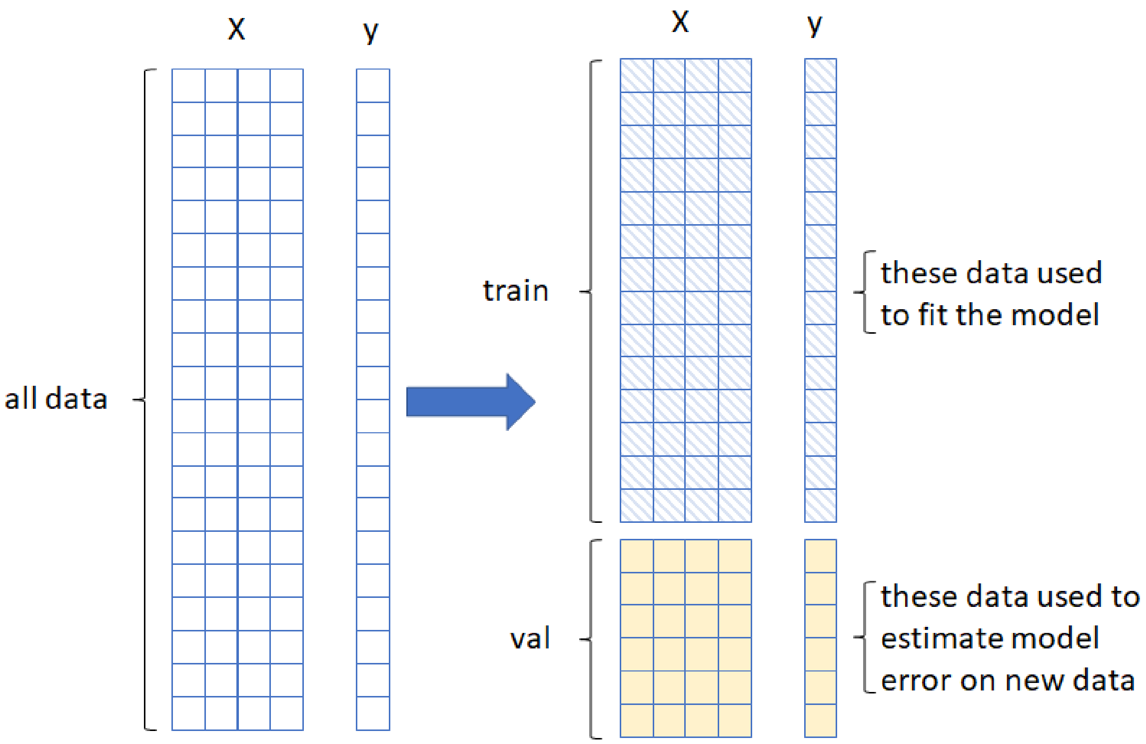
```
Out[72]: <matplotlib.axes._subplots.AxesSubplot at 0x220b4c03108>
```



Out[74]: <seaborn.axisgrid.PairGrid at 0x220b4e2c3c8>

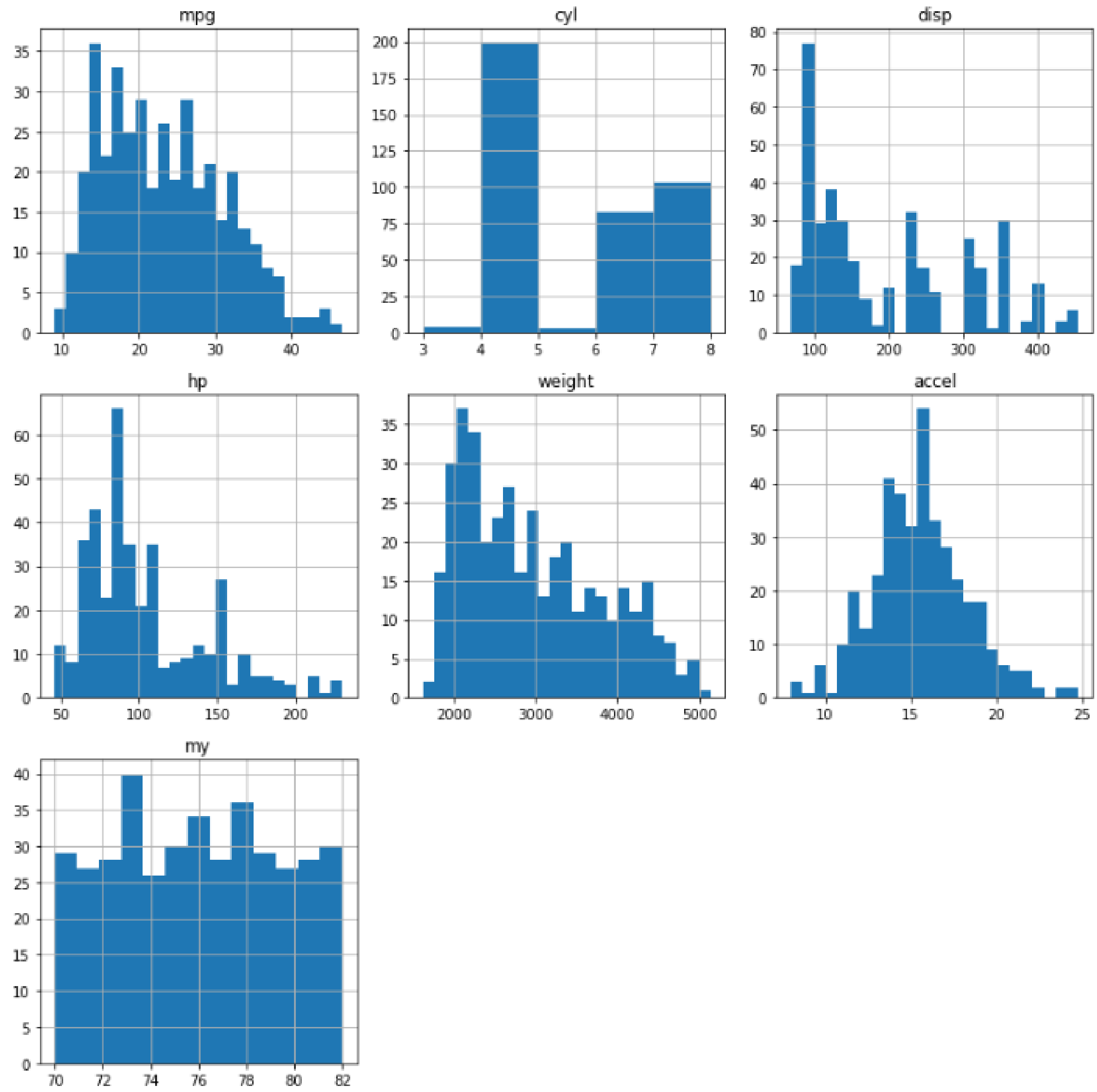
<Figure size 792x792 with 0 Axes>



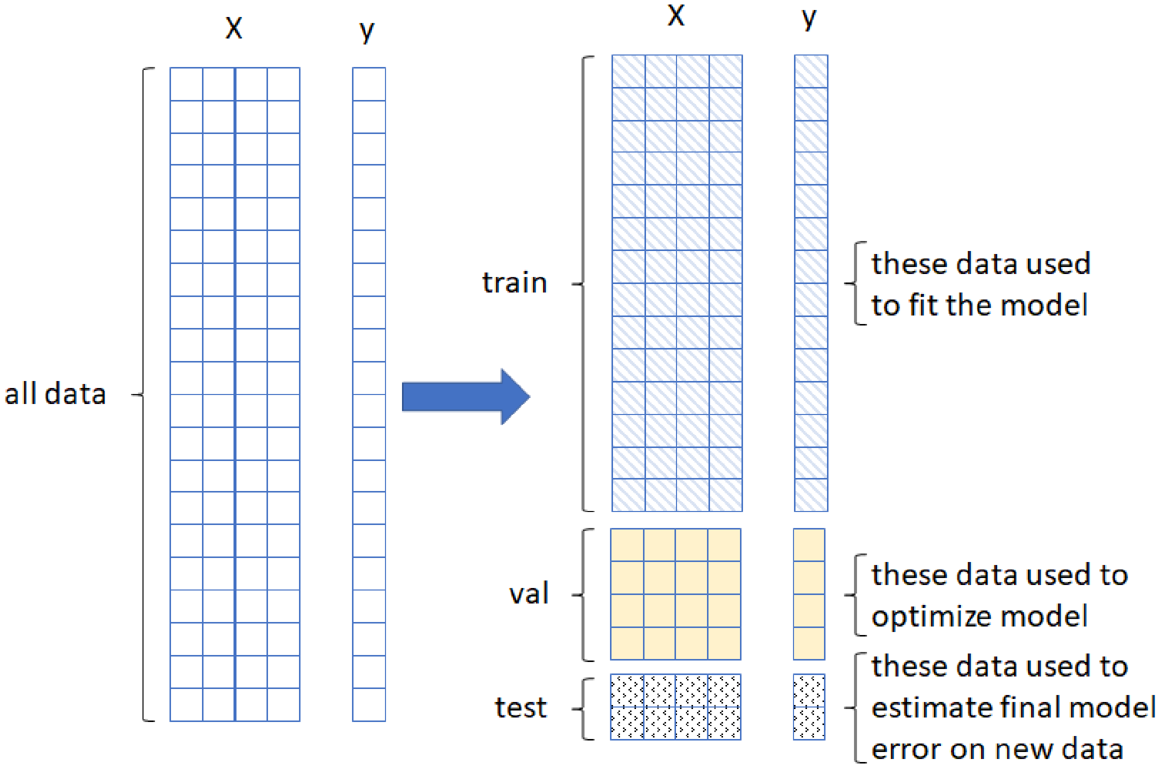


Out[3]:

	mpg	cyl	disp	hp	weight	accel	my	name
0	18.0	8	307.0	130	3504	12.0	70	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	ford torino




```
R2 score is  0.831869958782409
model coefficients:
 [[-3.53519873e-01 -4.91464180e-04 -1.15484755e-02 -6.08231188e-03
   2.60263994e-02  6.81342318e-01]]
intercept:  [-7.066461]
```



Out[2]:

	Date	TempHighF	TempAvgF	TempLowF	DewPointHighF	DewPointAvgF	DewPointLowF	HumidityHighPercent	HumidityAvgPercent	HumidityLowPercent	...
0	2013-12-21	74	60	45	67	49	43	93	75	57	...
1	2013-12-22	56	48	39	43	36	28	93	68	43	...
2	2013-12-23	58	45	32	31	27	23	76	52	27	...
3	2013-12-24	61	46	31	36	28	21	89	56	22	...
4	2013-12-25	58	50	41	44	40	36	86	71	56	...

5 rows × 21 columns

```
Out[12]: array(['Rain , Thunderstorm', ' ', 'Rain', 'Fog', 'Rain , Snow',
                'Fog , Rain', 'Thunderstorm', 'Fog , Rain , Thunderstorm',
                'Fog , Thunderstorm'], dtype=object)
```

Out[14]:

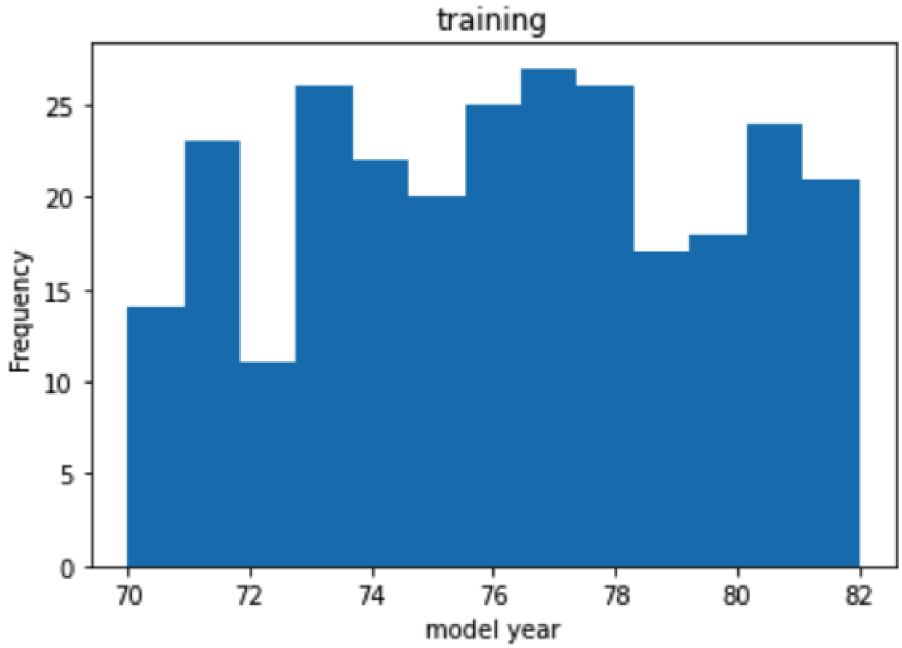
ressureLowInches	VisibilityHighMiles	VisibilityAvgMiles	VisibilityLowMiles	WindHighMPH	WindAvgMPH	WindGustMPH	PrecipitationSumInches	Events
29.59	10	7	2	20	4	31	0.46	Rain , Thunderstorm
29.87	10	10	5	16	6	25	0	None
30.41	10	10	10	8	3	12	0	None
30.3	10	10	7	12	4	20	0	None
30.27	10	10	7	10	2	16	T	None

Out[7]:

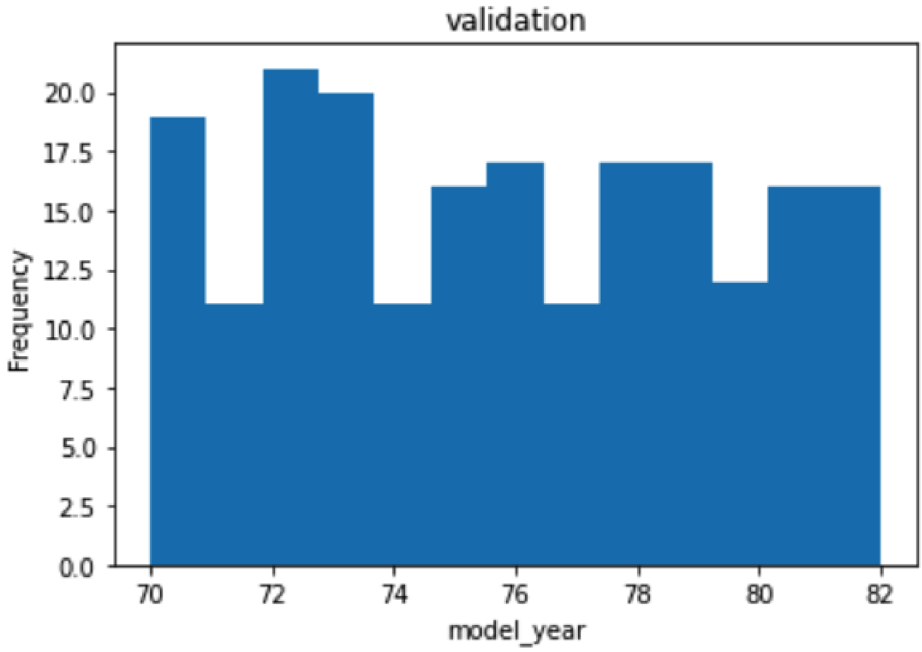
	TempHighF	TempAvgF	TempLowF	DewPointHighF	DewPointAvgF	DewPointLowF	HumidityHighPercent	HumidityAvgPercent	HumidityLowPercent	SeaL
677	81	66	51	64	54	49	96	66	35	
1046	91	81	71	73	71	64	100	72	44	
610	101	89	76	76	72	65	94	64	33	
49	65	51	37	42	36	29	85	63	40	
1284	91	81	71	74	72	67	100	75	50	

Out[17]:

	date	close
0	1986-01-02	209.59
1	1986-01-03	210.88
2	1986-01-06	210.65
3	1986-01-07	213.80
4	1986-01-08	207.97



training
Frequency
0
5
10
15
20
25
70
72
74
76
78
80
82
model year



validation
Frequency
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
70
72
74
76
78
80
82
model_year

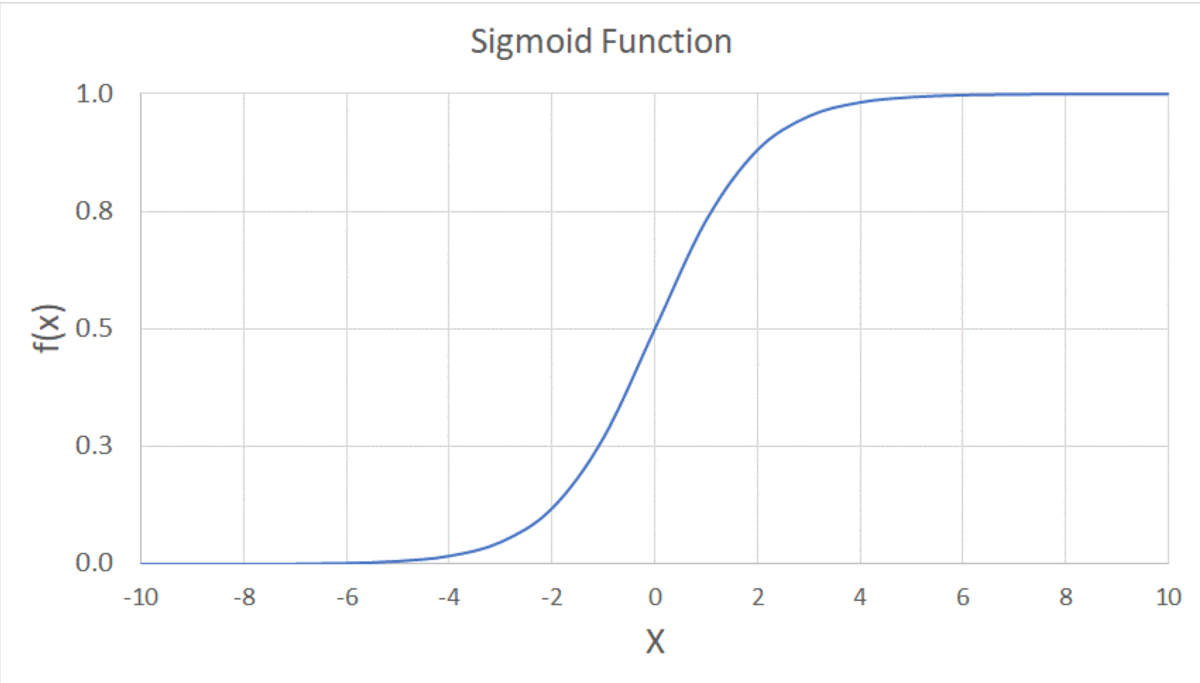
```
        Xmin    Xmax  Xrange
cyl      3.0     8.0     5.0
disp    71.0   455.0   384.0
hp      48.0   230.0   182.0
weight 1613.0 5140.0  3527.0
accel    9.5    23.7    14.2
my      70.0    82.0    12.0
```

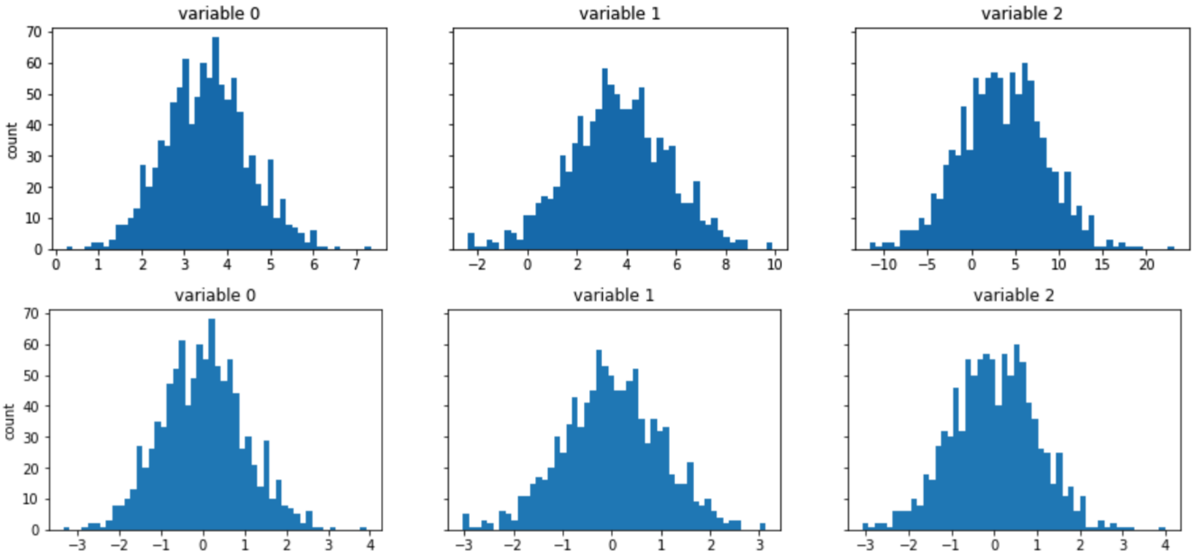
Out[50]:

	count	mean	std	min	25%	50%	75%	max
cyl	274.0	0.494161	0.329783	0.0	0.200000	0.300000	0.600000	1.0
disp	274.0	0.318041	0.260051	0.0	0.088542	0.208333	0.486979	1.0
hp	274.0	0.298187	0.196083	0.0	0.148352	0.258242	0.340659	1.0
weight	274.0	0.384055	0.237395	0.0	0.182733	0.339665	0.575631	1.0
accel	274.0	0.440552	0.183957	0.0	0.316901	0.443662	0.563380	1.0
my	274.0	0.519161	0.298829	0.0	0.250000	0.500000	0.750000	1.0

Out[52]:

	count	mean	std	min	25%	50%	75%	max
cyl	274.0	0.494161	0.329783	0.0	0.200000	0.300000	0.600000	1.0
disp	274.0	0.318041	0.260051	0.0	0.088542	0.208333	0.486979	1.0
hp	274.0	0.298187	0.196083	0.0	0.148352	0.258242	0.340659	1.0
weight	274.0	0.384055	0.237395	0.0	0.182733	0.339665	0.575631	1.0
accel	274.0	0.440552	0.183957	0.0	0.316901	0.443662	0.563380	1.0
my	274.0	0.519161	0.298829	0.0	0.250000	0.500000	0.750000	1.0





Out[87]:

	count	mean	std	min	25%	50%	75%	max
cyl	274.00	0.00	1.00	-1.50	-0.89	-0.59	0.32	1.54
disp	274.00	0.00	1.00	-1.23	-0.88	-0.42	0.65	2.63
hp	274.00	-0.00	1.00	-1.52	-0.77	-0.20	0.22	3.59
weight	274.00	0.00	1.00	-1.62	-0.85	-0.19	0.81	2.60
accel	274.00	-0.00	1.00	-2.40	-0.67	0.02	0.67	3.05
my	274.00	0.00	1.00	-1.74	-0.90	-0.06	0.77	1.61

```
R2 score is  0.831869958782409
model coefficients:
 [[-0.58185994 -0.0489877  -0.41137864 -5.08336838  0.06786155  2.438796  ]]
intercept:  [24.02262774]
the root mean square error is  3.2361376539382127
```

Out[32]:

	cyl	disp	hp	weight	accel	my
0	8.00	400.00	150.00	4997.00	14.00	73.00
1	4.00	98.00	65.00	2380.00	20.70	81.00
2	4.00	151.00	85.00	2855.00	17.60	78.00
3	6.00	232.00	100.00	2789.00	15.00	73.00
4	8.00	304.00	150.00	3892.00	12.50	72.00

Out[3]:

	count	mean	std	min	25%	50%	75%	max
TempHighF	1319.0	80.862775	14.766523	32.0	72.0	83.0	92.0	107.0
TempAvgF	1319.0	70.642911	14.045904	29.0	62.0	73.0	83.0	93.0
TempLowF	1319.0	59.902957	14.190648	19.0	49.0	63.0	73.0	81.0

```
                            count  mean   std   min   25%   50%   75%    max
TempHighF                 1305.00 80.79 14.71 32.00 72.00 83.00 92.00 107.00
TempAvgF                  1305.00 70.56 14.01 29.00 62.00 73.00 83.00  93.00
TempLowF                  1305.00 59.82 14.19 19.00 49.00 62.00 73.00  81.00
DewPointHighF             1305.00 61.52 13.58 13.00 53.00 66.00 73.00  80.00
DewPointAvgF              1305.00 56.64 14.86  8.00 46.00 61.00 69.00  76.00
DewPointLowF              1305.00 50.94 16.19  2.00 38.00 56.00 65.00  75.00
HumidityHighPercent       1305.00 87.83 11.05 37.00 85.00 90.00 94.00 100.00
HumidityAvgPercent        1305.00 66.66 12.50 27.00 59.00 67.00 74.00  97.00
HumidityLowPercent        1305.00 44.98 17.01 10.00 33.00 44.00 55.00  93.00
SeaLevelPressureHighInches 1305.00 30.11  0.18 29.63 29.99 30.08 30.21  30.83
SeaLevelPressureAvgInches 1305.00 30.02  0.17 29.55 29.91 30.00 30.10  30.74
SeaLevelPressureLowInches 1305.00 29.93  0.17 29.41 29.82 29.91 30.02  30.61
VisibilityHighMiles       1305.00  9.99  0.16  5.00 10.00 10.00 10.00  10.00
VisibilityAvgMiles        1305.00  9.16  1.46  2.00  9.00 10.00 10.00  10.00
VisibilityLowMiles        1305.00  6.84  3.68  0.00  3.00  9.00 10.00  10.00
WindHighMPH               1305.00 13.25  3.43  6.00 10.00 13.00 15.00  29.00
WindAvgMPH                1305.00  5.01  2.08  1.00  3.00  5.00  6.00  12.00
WindGustMPH               1305.00 21.38  5.89  9.00 17.00 21.00 25.00  57.00
PrecipitationSumInches    1305.00  0.12  0.43  0.00  0.00  0.00  0.00   5.20
```

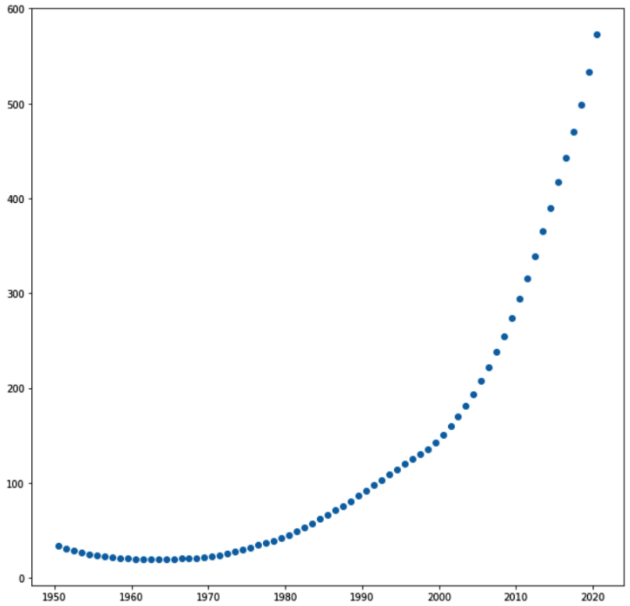
Out[20]:

	TempHighF	TempAvgF	TempLowF	DewPointHighF	DewPointAvgF	DewPointLowF	HumidityHighPercent	HumidityAvgPercent	HumidityLowPercent	SeaLevel
0	0.83	0.81	0.78	0.99	0.97	0.81	0.57	0.11	-0.23	
1	0.76	0.89	0.92	0.99	1.04	1.18	0.29	0.43	0.42	
2	-1.76	-1.76	-1.69	-2.24	-1.93	-1.73	-1.08	-1.01	-0.76	
3	0.35	0.17	0.01	-0.18	-0.11	-0.25	-0.35	-0.69	-0.76	
4	-0.80	-0.62	-0.35	-0.26	-0.38	-0.18	0.20	0.43	0.48	

Chapter 10: Data Modeling – Modeling Basics

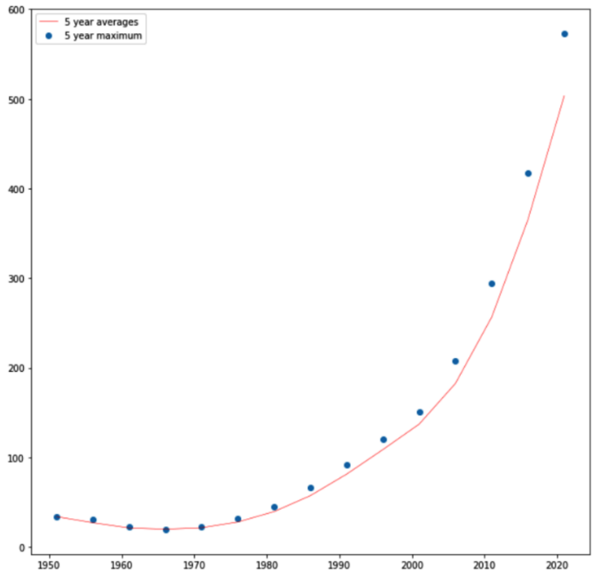
Classification	Regression	Clustering

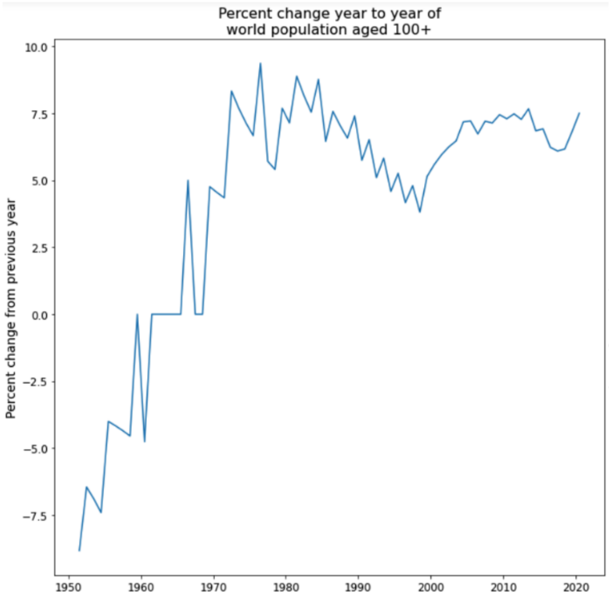
```
         date  population aged 100+ (000)
0  7/1/1950                           34
1  7/1/1951                           31
2  7/1/1952                           29
3  7/1/1953                           27
4  7/1/1954                           25
date                           object
population aged 100+ (000)      int64
dtype: object
```

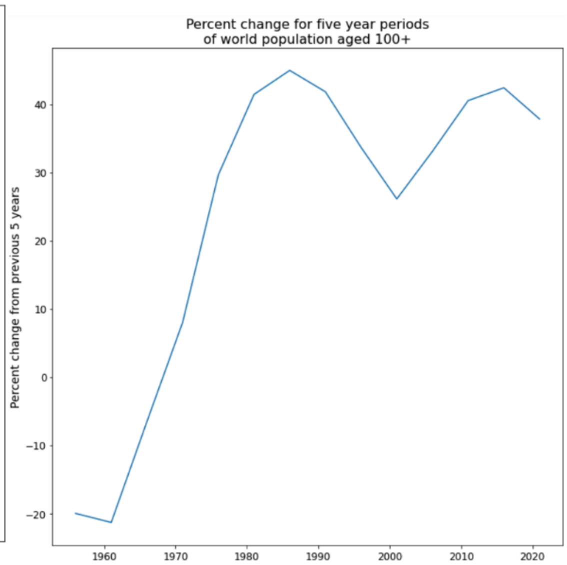


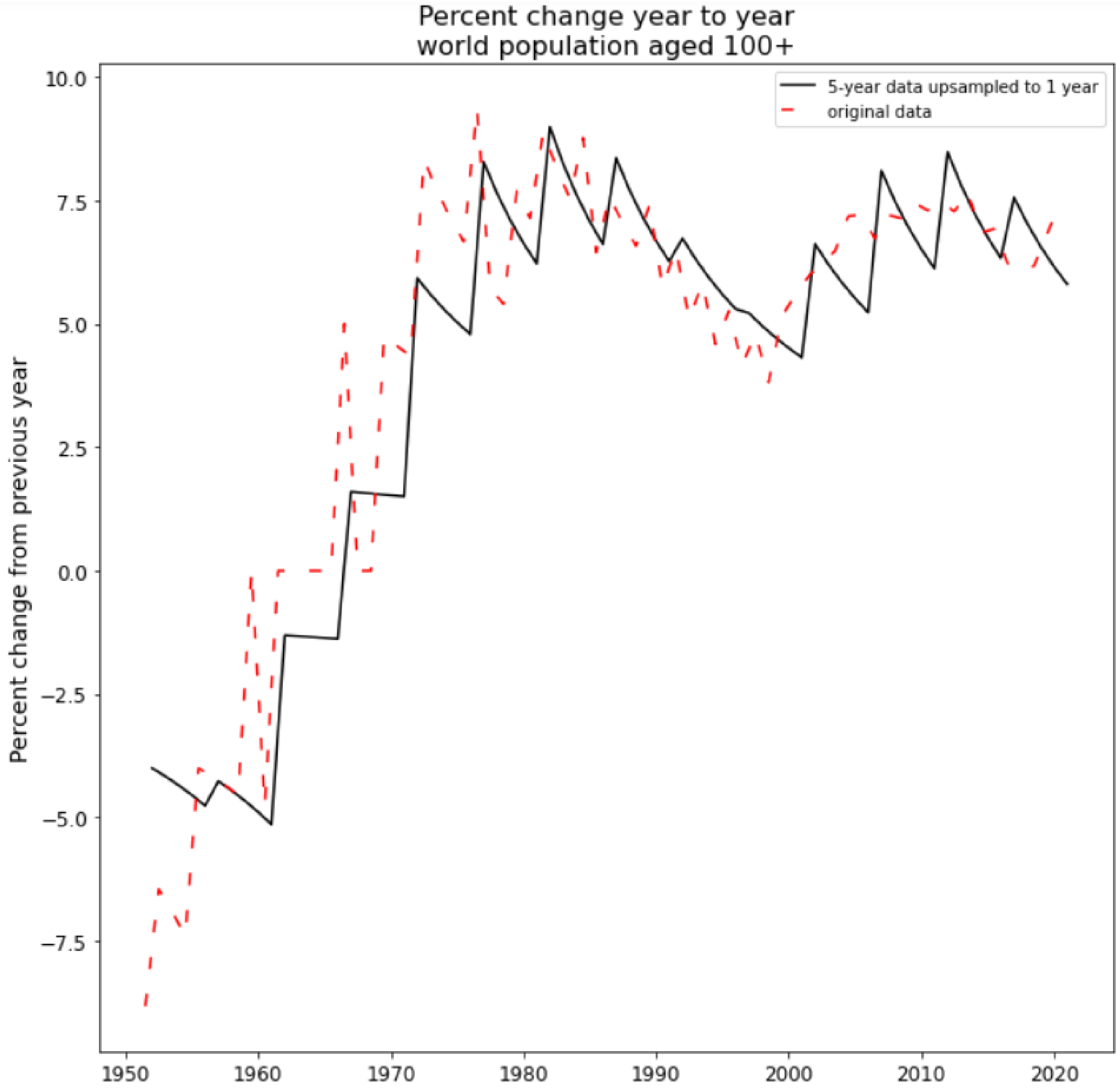
Out[16]:

date	population aged 100+ (000)
1950-12-31	34.0
1955-12-31	27.2
1960-12-31	21.4
1965-12-31	20.0
1970-12-31	21.6


5 year averages
5 year maximum


Percent change year to year of
world population aged 100+
Percent change from previous year


Percent change for five year periods
of world population aged 100+
Percent change from previous 5 years



Out[51]:

decade	population aged 100+ (000)
1950.0	29.200000
1960.0	21.000000
1970.0	24.000000
1980.0	47.000000
1990.0	92.333333
2000.0	156.272727
2010.0	299.222222
2020.0	489.166667

Out[56]:

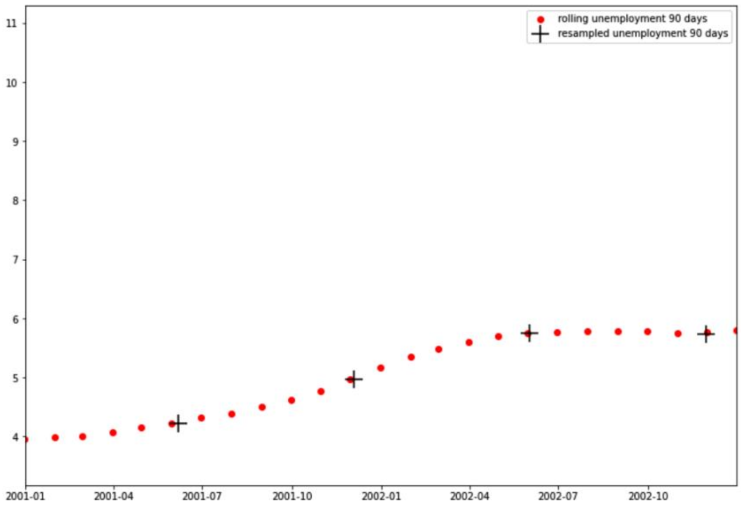
	date	population aged 100+ (000)	gender
0	1950-07-01	9	male
1	1950-07-01	25	female
2	1951-07-01	8	male
3	1951-07-01	23	female
4	1952-07-01	8	male
5	1952-07-01	21	female

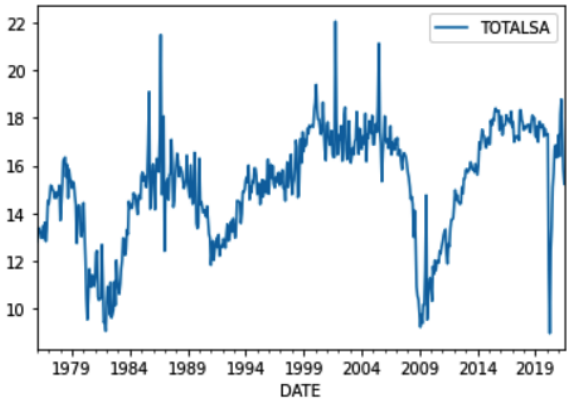
Out[83]:

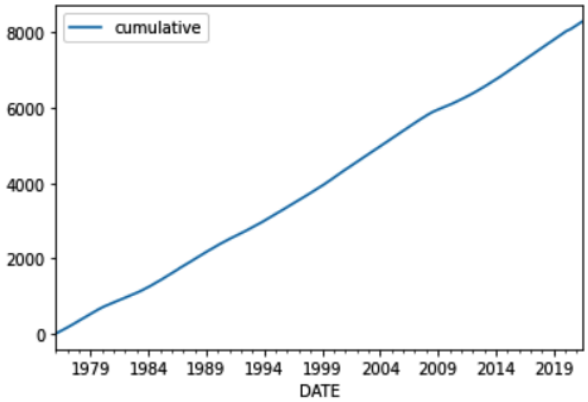
		population aged 100+ (000)
gender	**decade**	
female	**1950**	21.600000
	1960	15.818182
	1970	18.444444
	1980	36.545455
	1990	74.000000
	2000	127.545455
	2010	240.888889
	2020	386.166667
male	**1950**	7.600000
	1960	5.181818
	1970	5.777778
	1980	10.454545
	1990	18.333333
	2000	28.818182
	2010	58.222222
	2020	103.166667

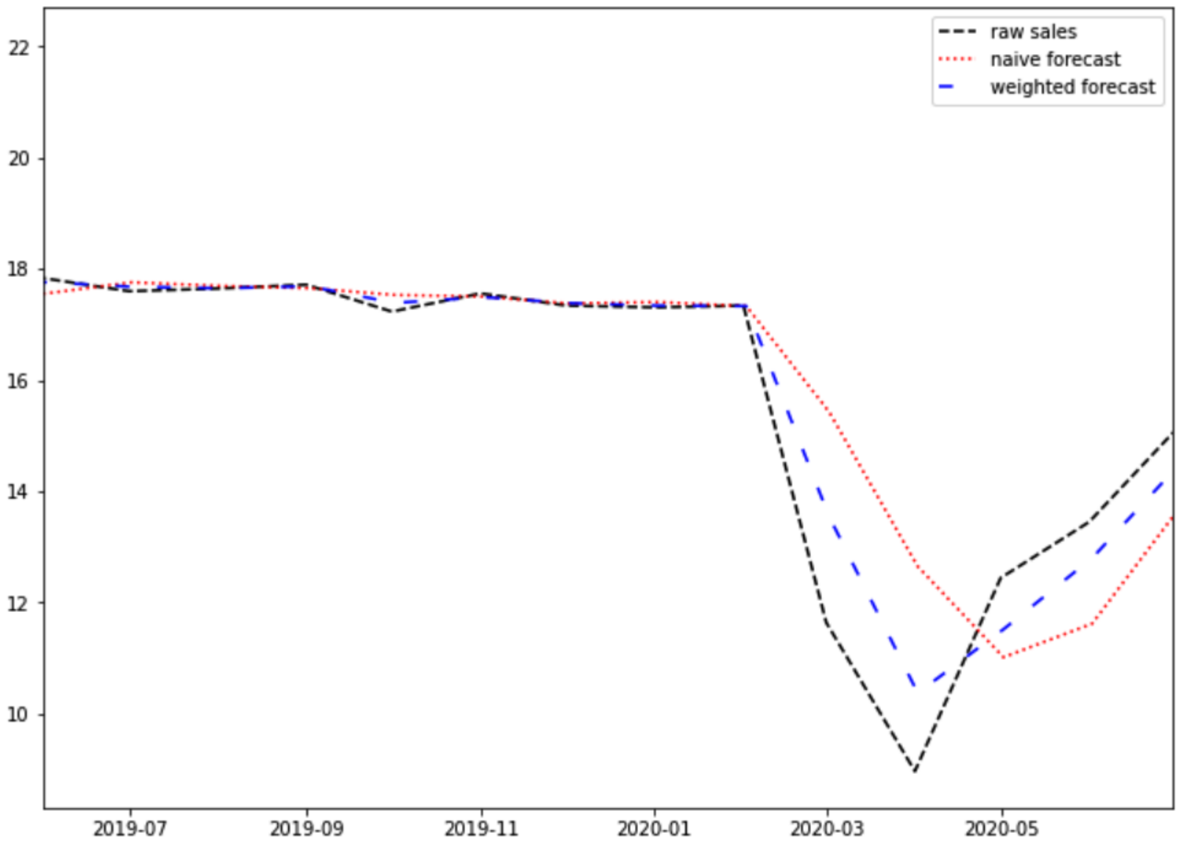
Out[25]:

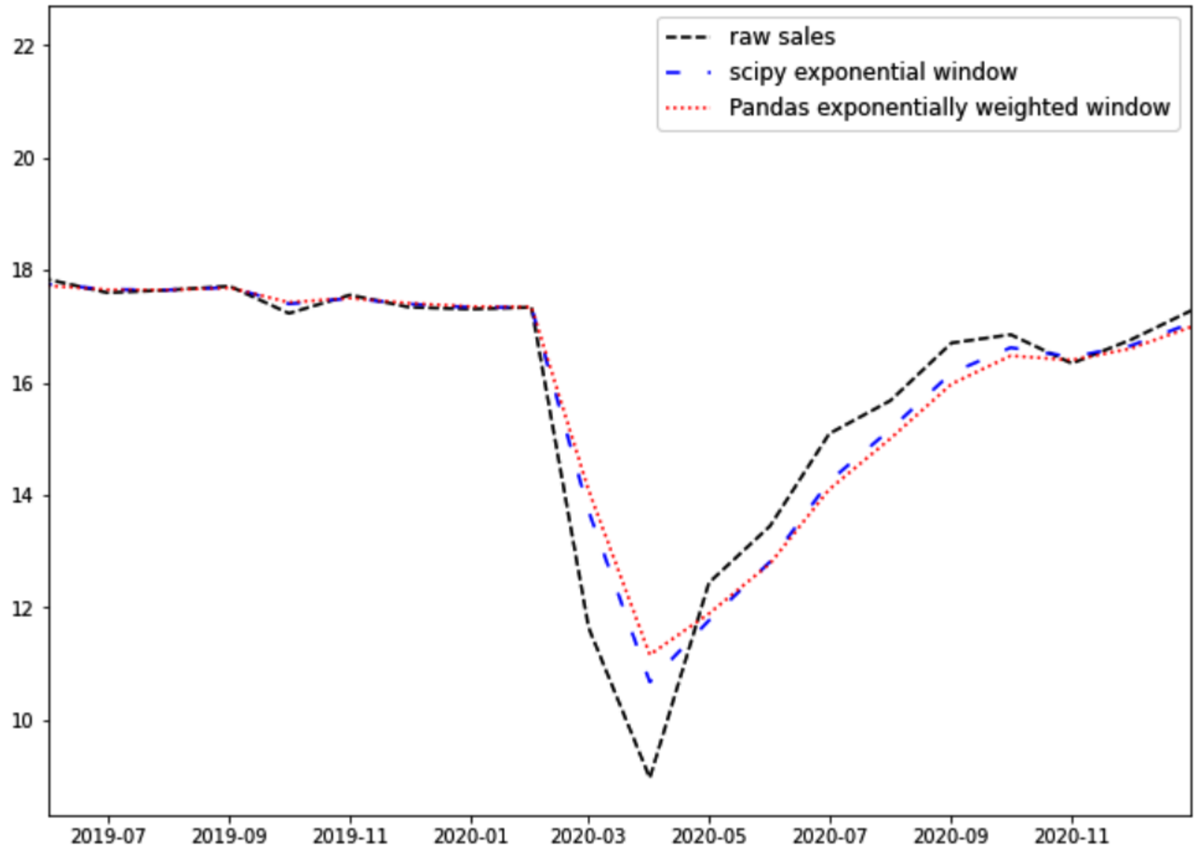
	mpg	cyl	disp	hp	weight	accel	my
mpg	1.002558	-0.779606	-0.807186	-0.780418	-0.834373	0.424411	0.582026
cyl	-0.779606	1.002558	0.953255	0.845139	0.899823	-0.505974	-0.346531
disp	-0.807186	0.953255	1.002558	0.899552	0.935381	-0.545191	-0.370801
hp	-0.780418	0.845139	0.899552	1.002558	0.866749	-0.690958	-0.417426
weight	-0.834373	0.899823	0.935381	0.866749	1.002558	-0.417905	-0.309910
accel	0.424411	-0.505974	-0.545191	-0.690958	-0.417905	1.002558	0.291059
my	0.582026	-0.346531	-0.370801	-0.417426	-0.309910	0.291059	1.002558

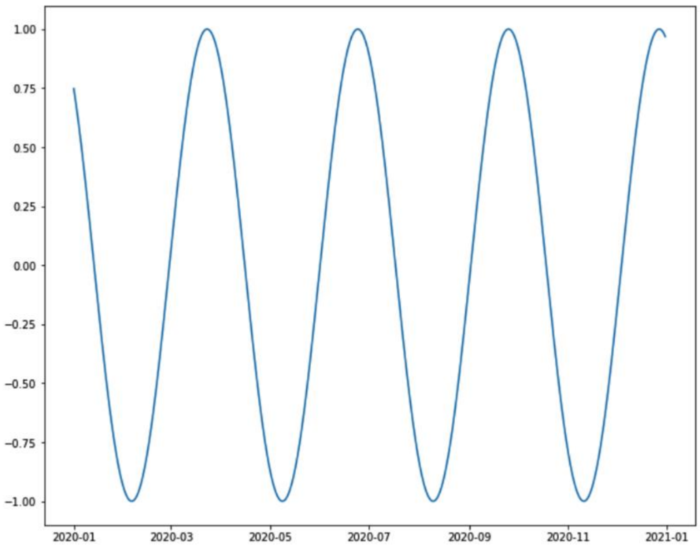

rolling unemployment 90 days
resampled unemployment 90 days
11
10
9
8
7
6
5
4
2001-01
2001-04
2001-07
2001-10
2002-01
2002-04
2002-07
2002-10

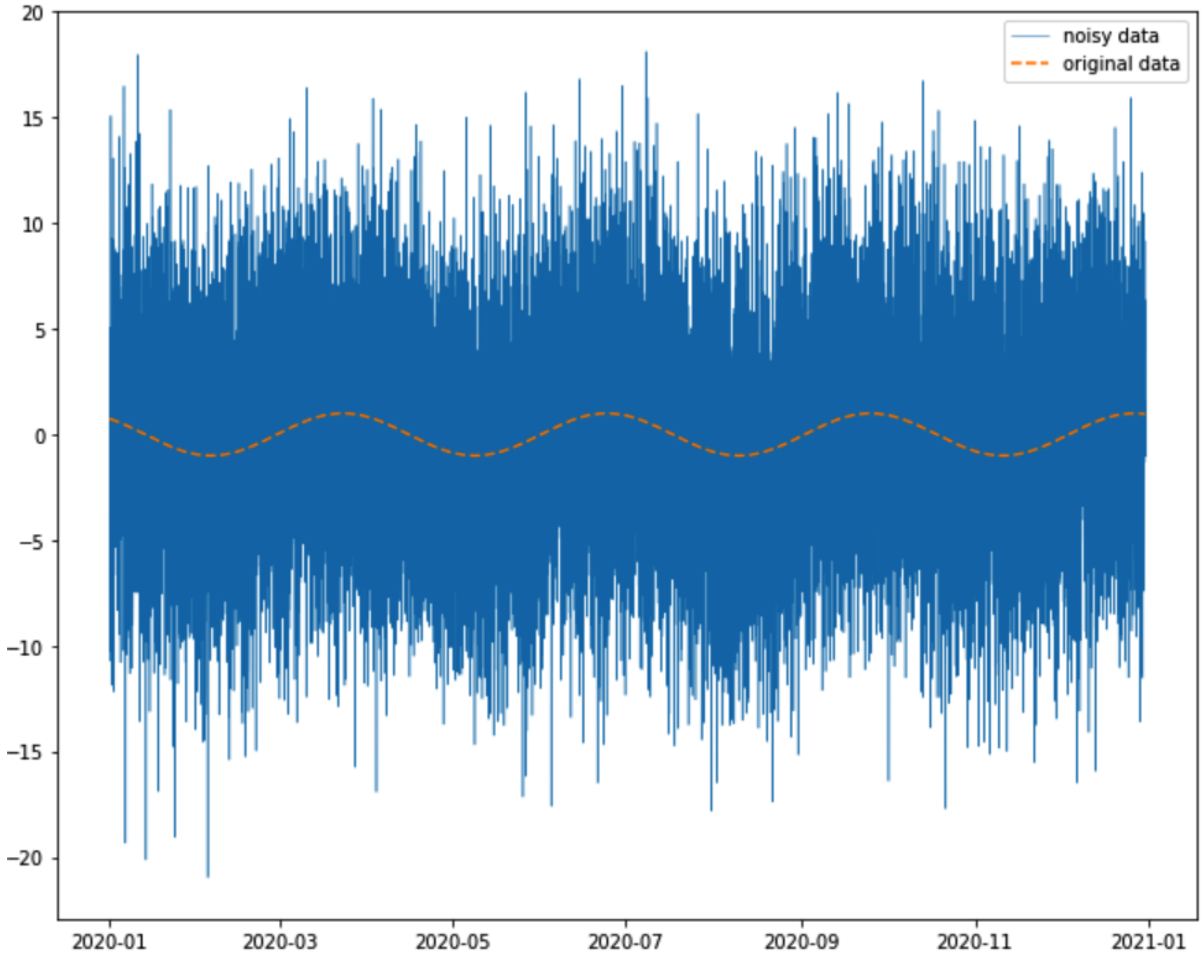

TOTALSA
22
20
18
16
14
12
10
1979
1984
1989
1994
1999
2004
2009
2014
2019
DATE


cumulative
8000
6000
4000
2000
0
1979
1984
1989
1994
1999
2004
2009
2014
2019
DATE

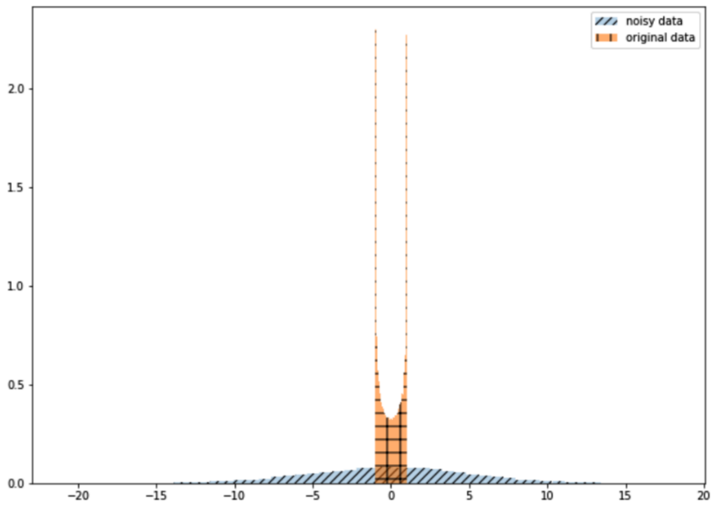

raw sales
naive forecast
weighted forecast
22
20
18
16
14
12
10
2019-07
2019-09
2019-11
2020-01
2020-03
2020-05


raw sales
scipy exponential window
Pandas exponentially weighted window
22
20
18
16
14
12
10
2019-07
2019-09
2019-11
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11

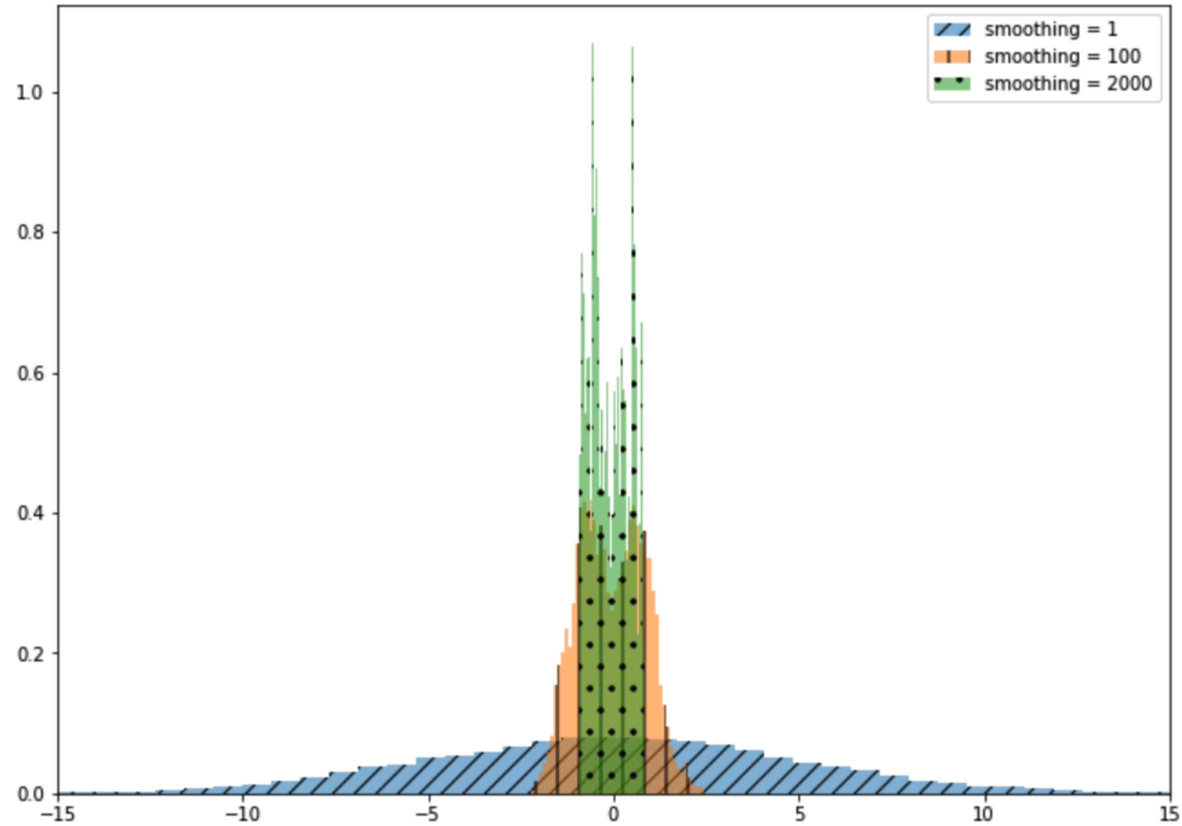

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01



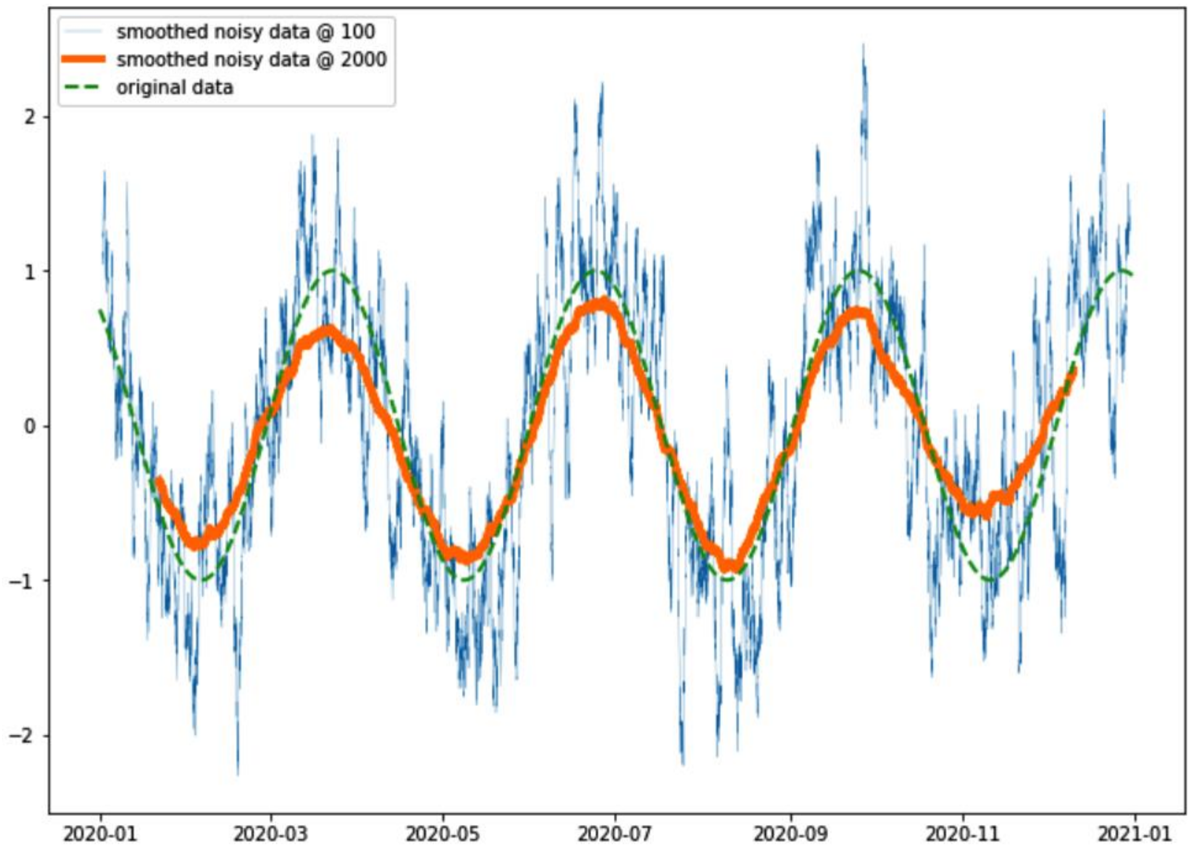
noisy data
original data



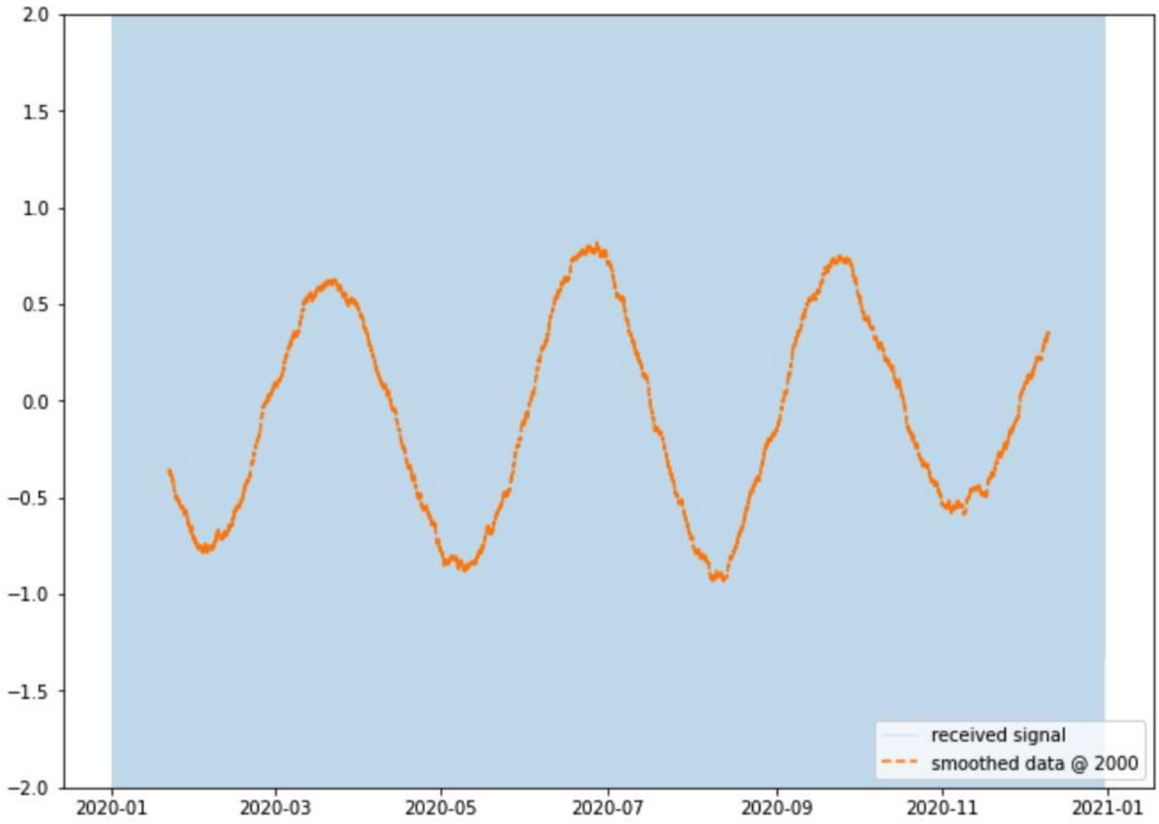
noisy data
original data



smoothing = 1
smoothing = 100
smoothing = 2000
1.0
0.8
0.6
0.4
0.2
0.0
−15
−10
−5
0
5
10
15



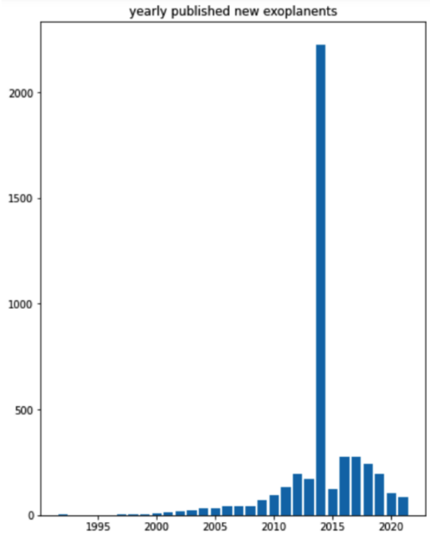
smoothed noisy data @ 100
smoothed noisy data @ 2000
original data
2
1
0
−1
−2
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01

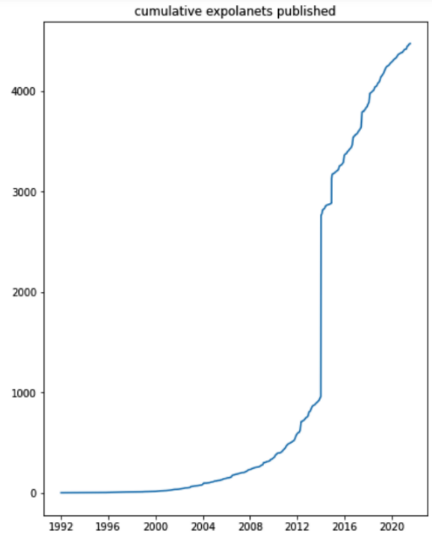


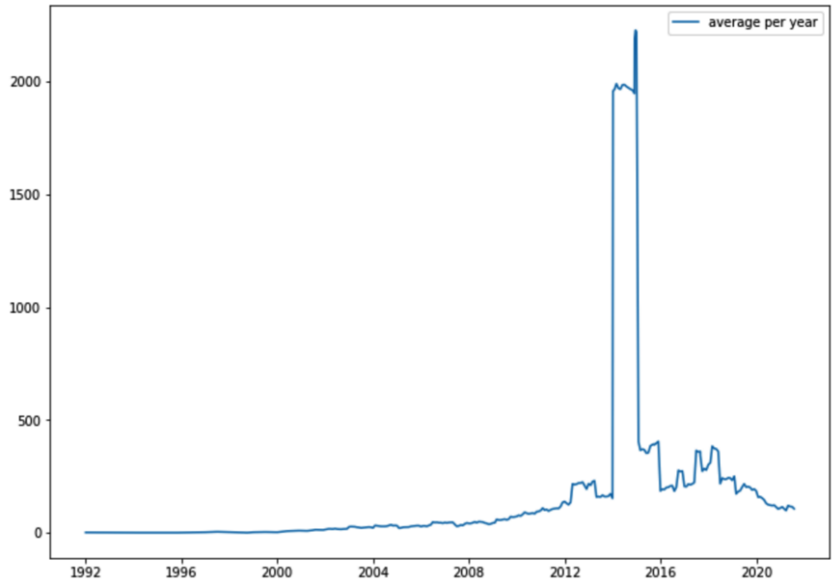
Out[500]:

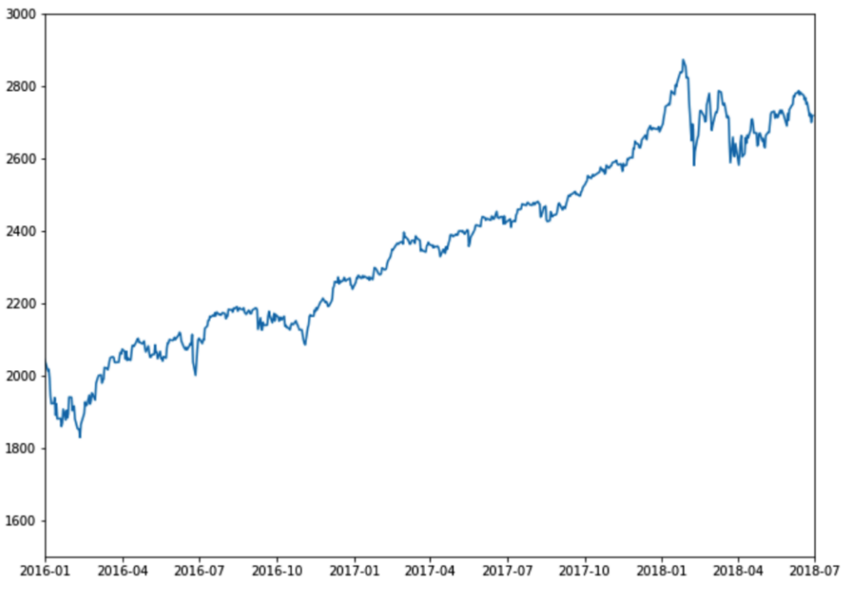
	date	num_recorded	cumulative_recorded
0	1/1/1992	2	2
1	4/1/1994	1	3
2	11/1/1995	1	4
3	1/1/1997	3	7
4	7/1/1997	2	9
...	...	...	...
234	4/1/2021	2	4418
235	5/1/2021	28	4446
236	6/1/2021	6	4452
237	7/1/2021	16	4468
238	8/1/2021	4	4472

239 rows × 3 columns

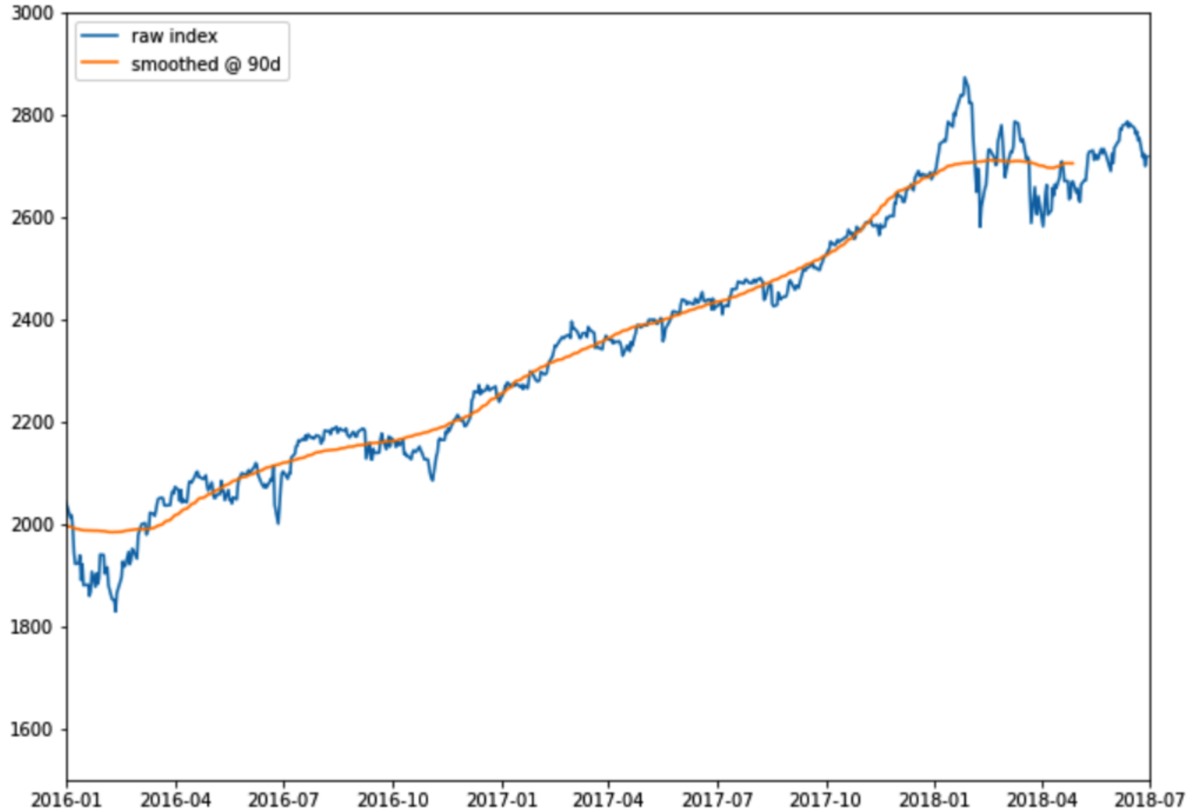

yearly published new exoplanents
0
500
1000
1500
2000
1995
2000
2005
2010
2015
2020

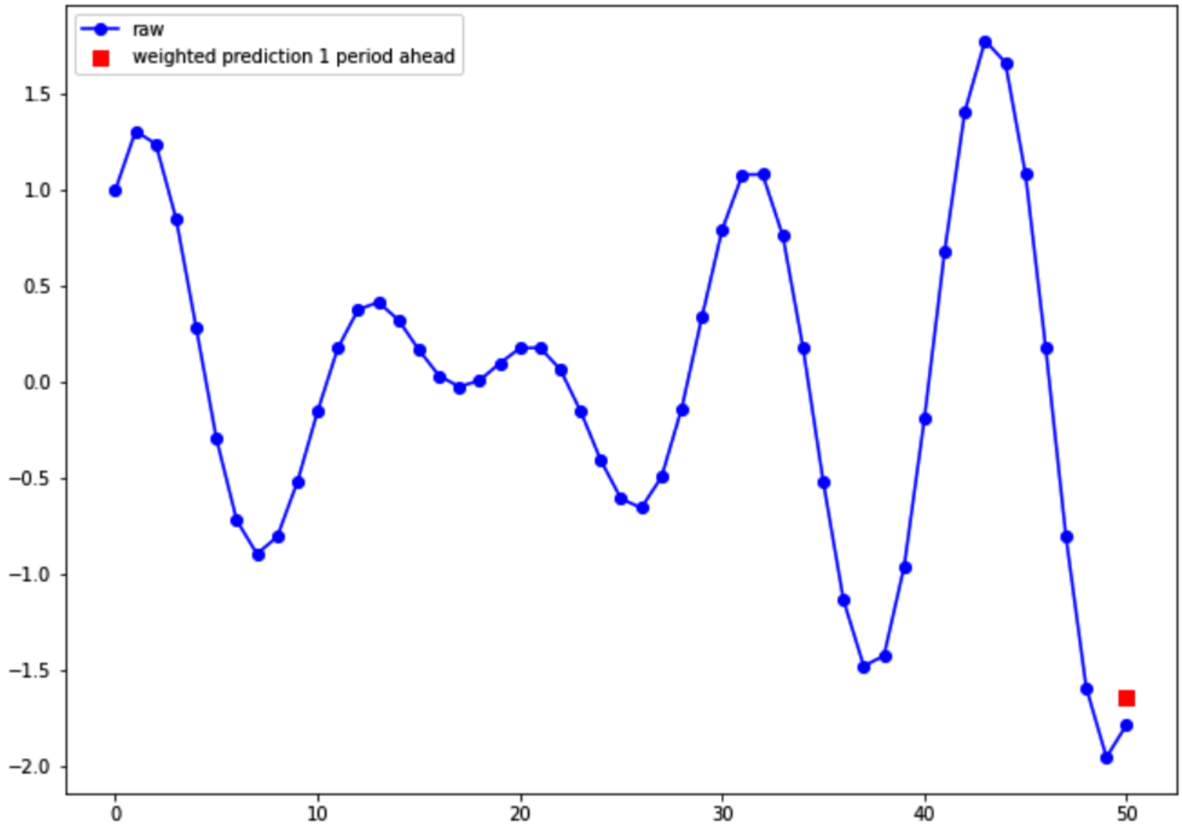

cumulative expolanets published
0
1000
2000
3000
4000
1992
1996
2000
2004
2008
2012
2016
2020


average per year
0
500
1000
1500
2000
1992
1996
2000
2004
2008
2012
2016
2020


1600
1800
2000
2200
2400
2600
2800
3000
2016-01
2016-04
2016-07
2016-10
2017-01
2017-04
2017-07
2017-10
2018-01
2018-04
2018-07

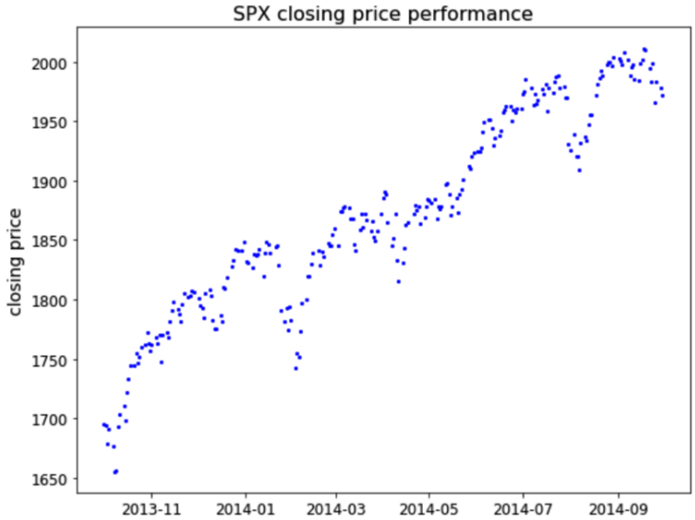

raw index
smoothed @ 90d
3000
2800
2600
2400
2200
2000
1800
1600
2016-01
2016-04
2016-07
2016-10
2017-01
2017-04
2017-07
2017-10
2018-01
2018-04
2018-07

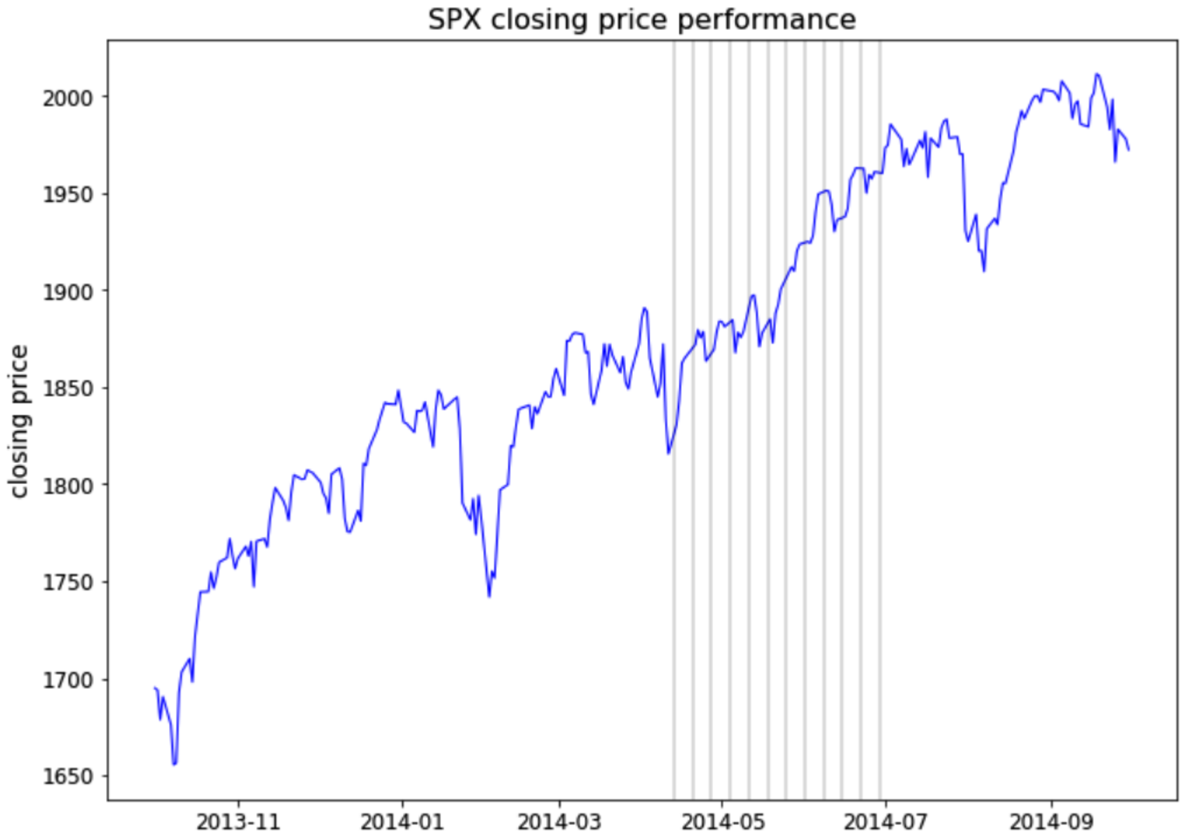

raw index
smoothed @ 90d
3000
2800
2600
2400
2200
2000
1800
1600
2016-01
2016-04
2016-07
2016-10
2017-01
2017-04
2017-07
2017-10
2018-01
2018-04
2018-07

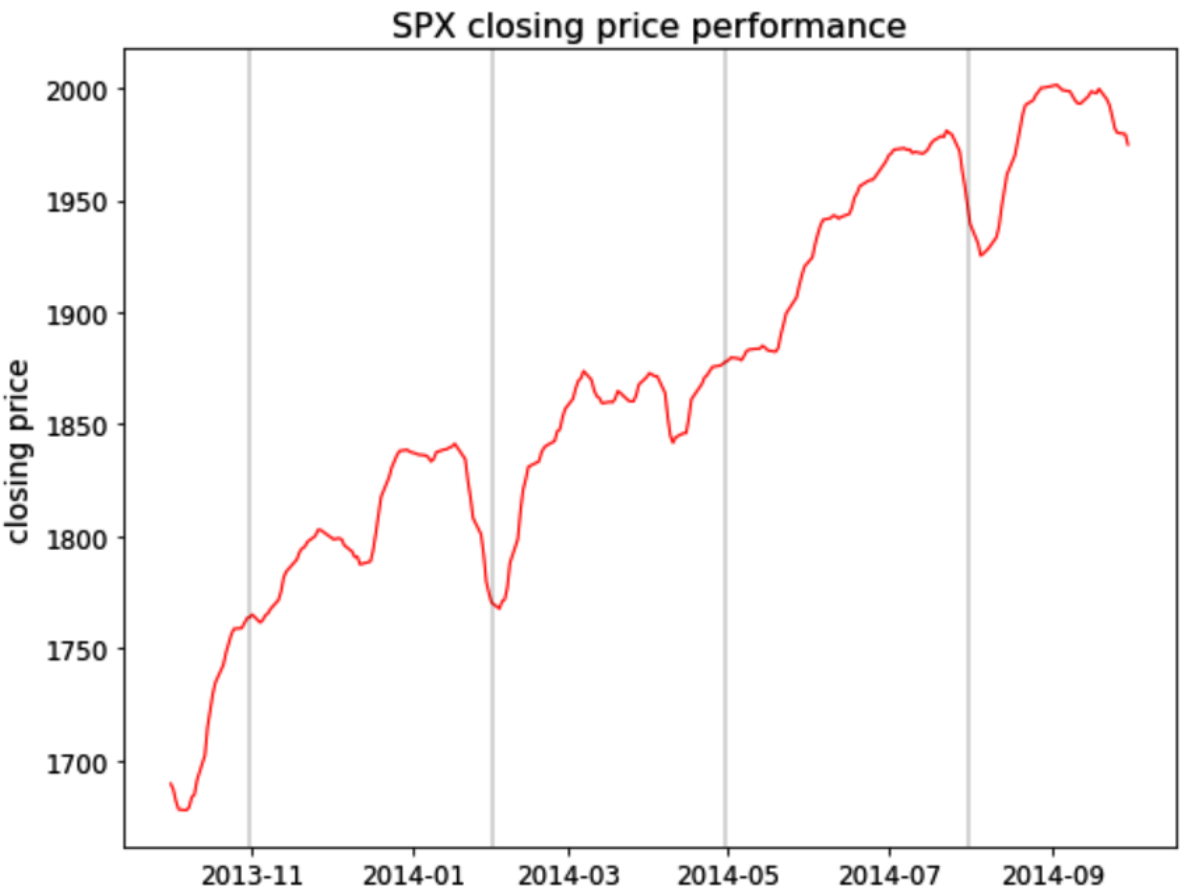


Out[3]:

	date	close
0	1986-01-02	209.59
1	1986-01-03	210.88
2	1986-01-06	210.65
3	1986-01-07	213.80
4	1986-01-08	207.97



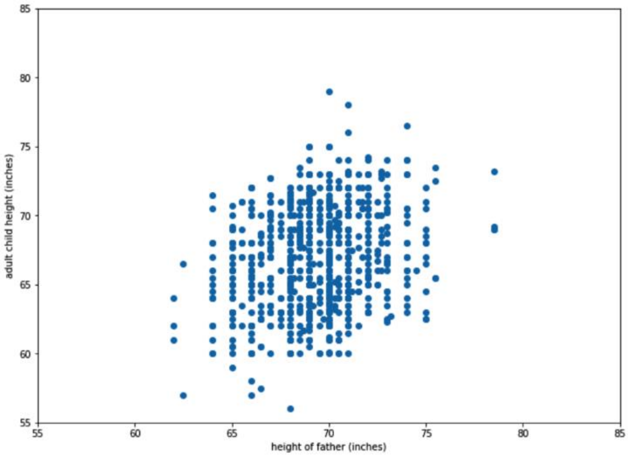

SPX closing price performance
closing price
2000
1950
1900
1850
1800
1750
1700
1650
2013-11
2014-01
2014-03
2014-05
2014-07
2014-09

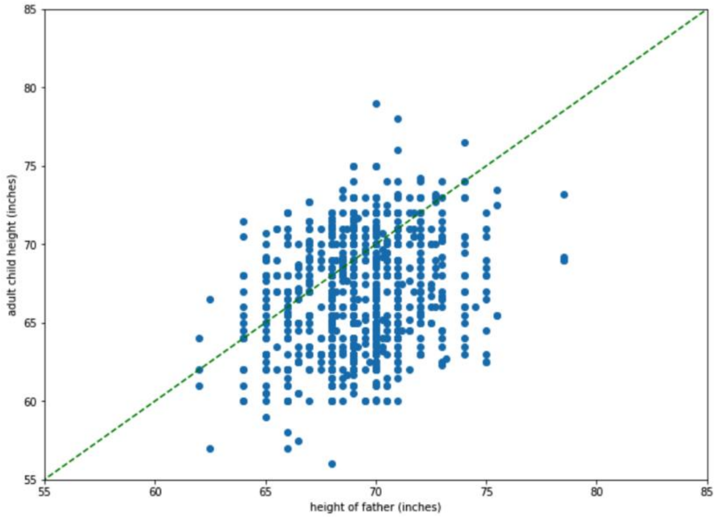

SPX closing price performance
closing price
2000
1950
1900
1850
1800
1750
1700
2013-11
2014-01
2014-03
2014-05
2014-07
2014-09

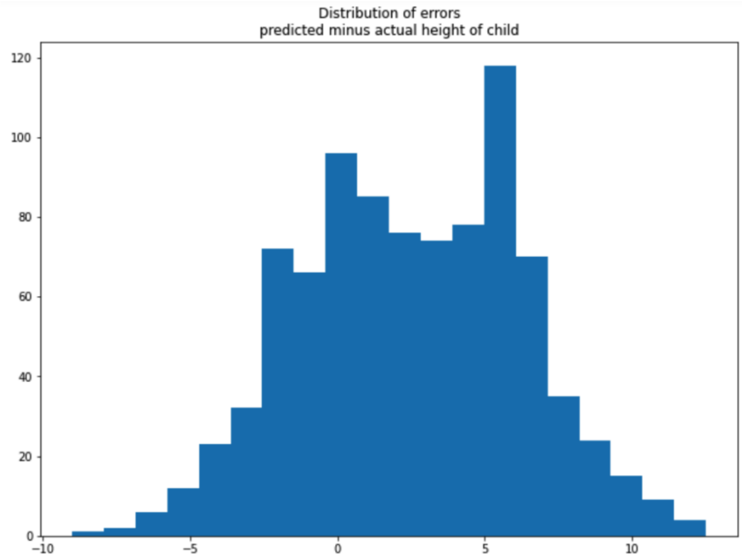
Chapter 11: Data Modeling – Regression Modeling

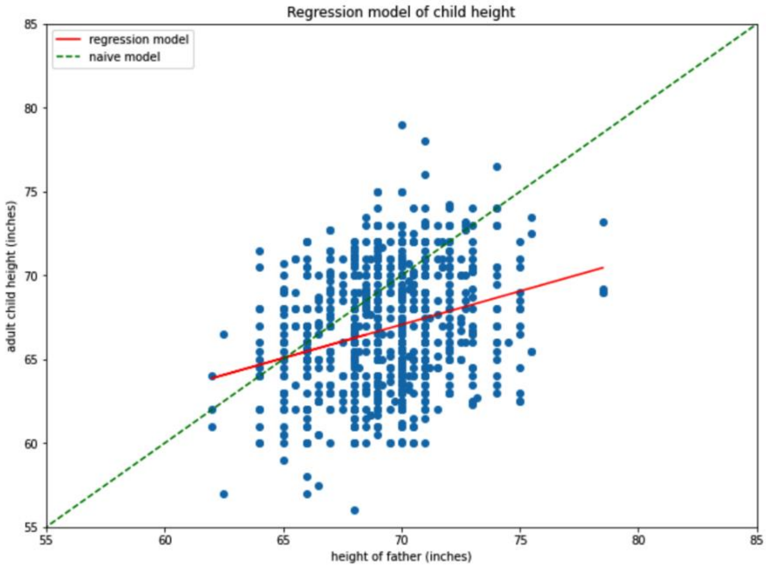
Out[17]:

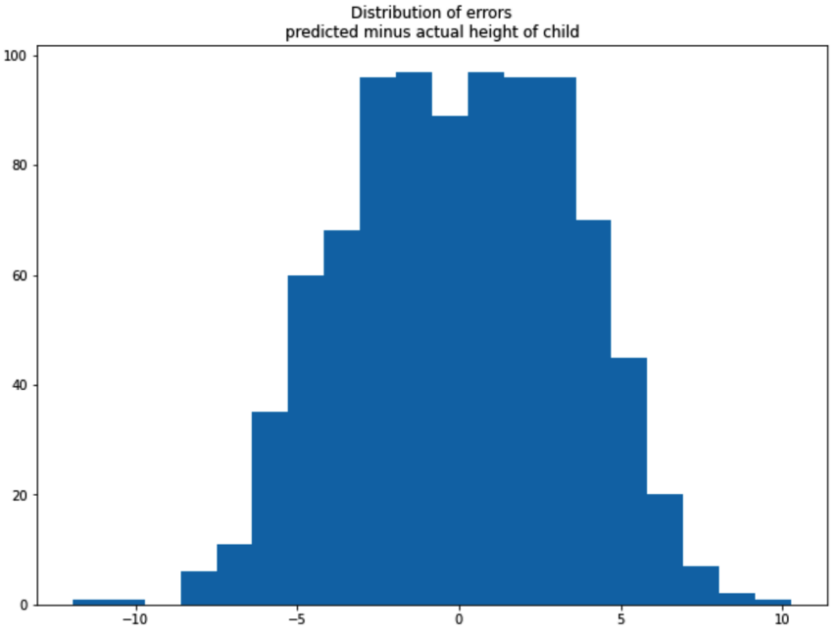
	ht_father	ht_child
0	78.5	73.2
1	78.5	69.2
2	78.5	69.0
3	78.5	69.0
4	75.5	73.5







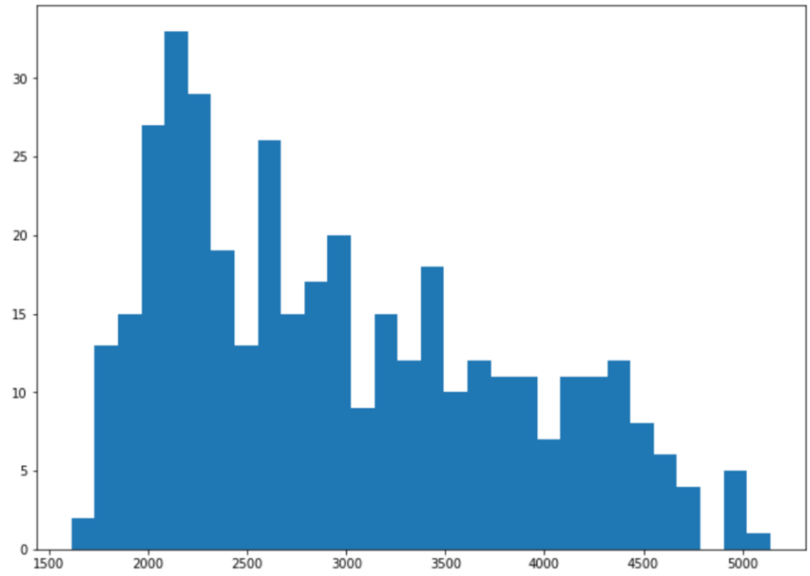




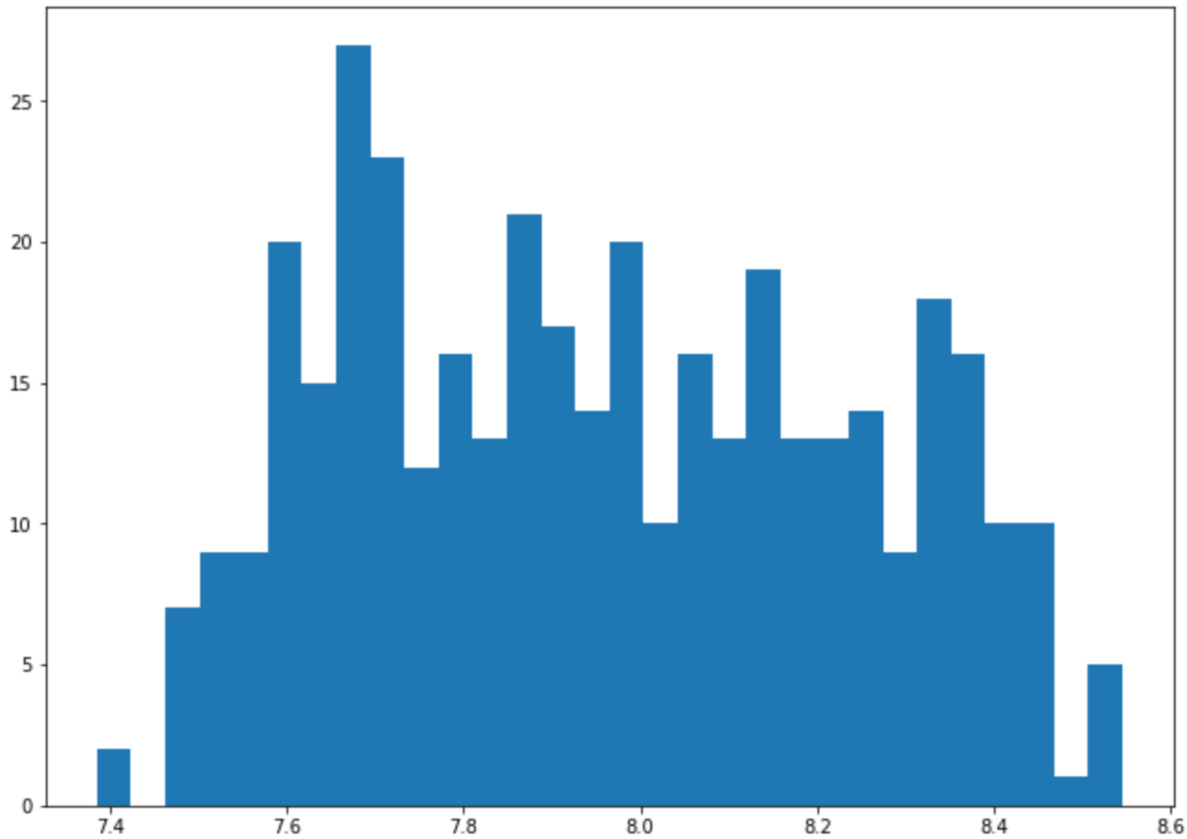
In [31]:

```
naive_SSE = np.sum((galton_heights.naive_pred - galton_heights.ht_child)**2)
OLS_SSE = np.sum((galton_heights.OLS_pred - galton_heights.ht_child)**2)
naive_RMSE = np.sqrt(naive_SSE / galton_heights.shape[0])
OLS_RMSE = np.sqrt(OLS_SSE / galton_heights.shape[0])
print('naive model gives:\n',
      'SSE = ', naive_SSE.round(3), '\n',
      'RMSE = ', naive_RMSE.round(3), '\n',
      'regression model gives:\n',
      'SSE = ', OLS_SSE.round(3), '\n',
      'RMSE = ', OLS_RMSE.round(3))
```

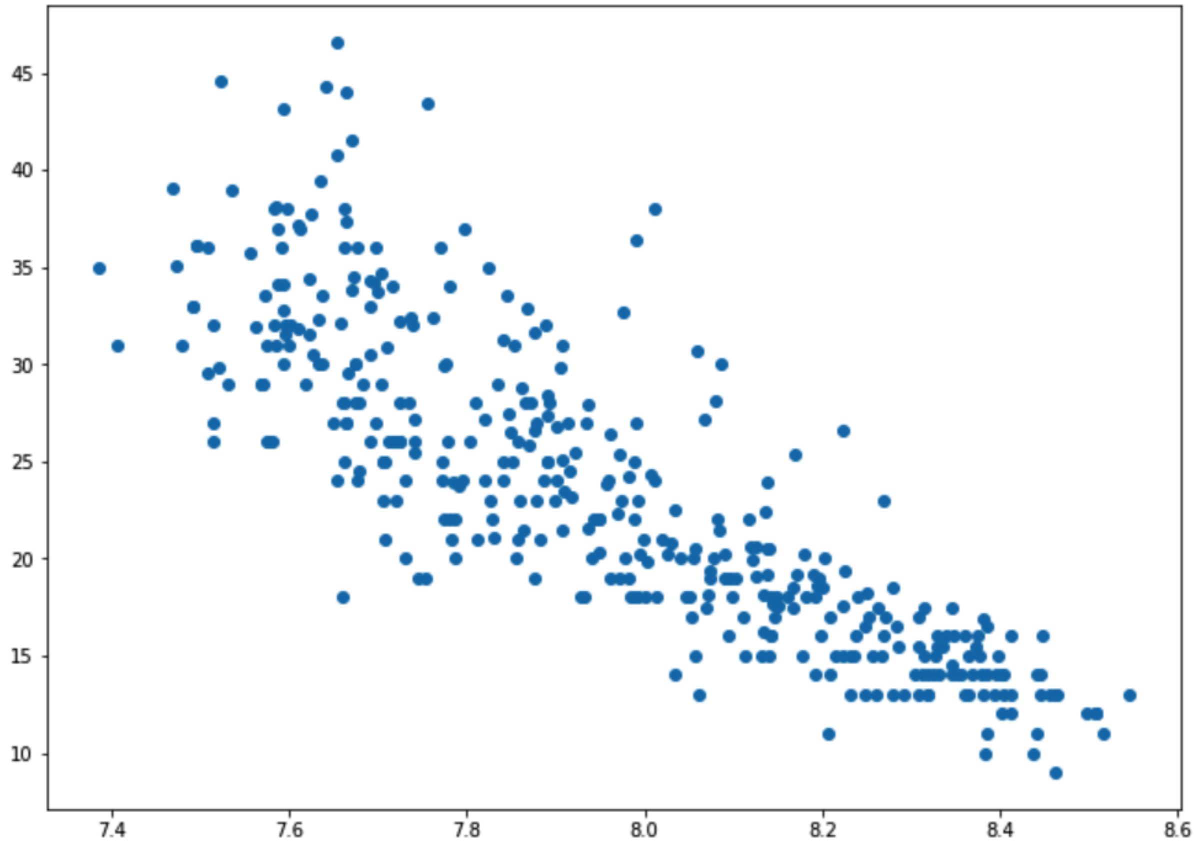
```
naive model gives:
 SSE =  18104.76
 RMSE =  4.49
 regression model gives:
 SSE =  10641.987
 RMSE =  3.442
```



30
25
20
15
10
5
0
1500
2000
2500
3000
3500
4000
4500
5000



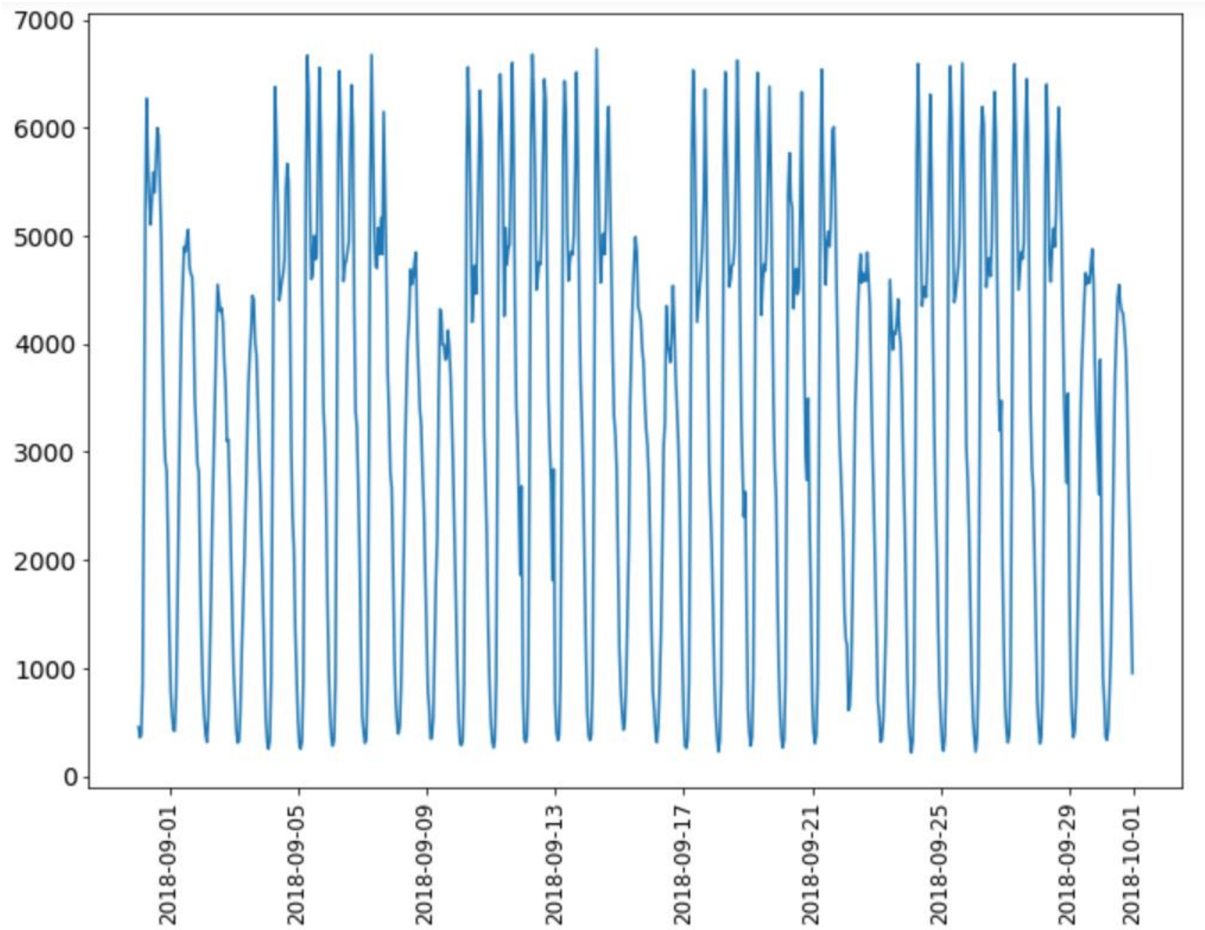
25
20
15
10
5
0
7.4
7.6
7.8
8.0
8.2
8.4
8.6



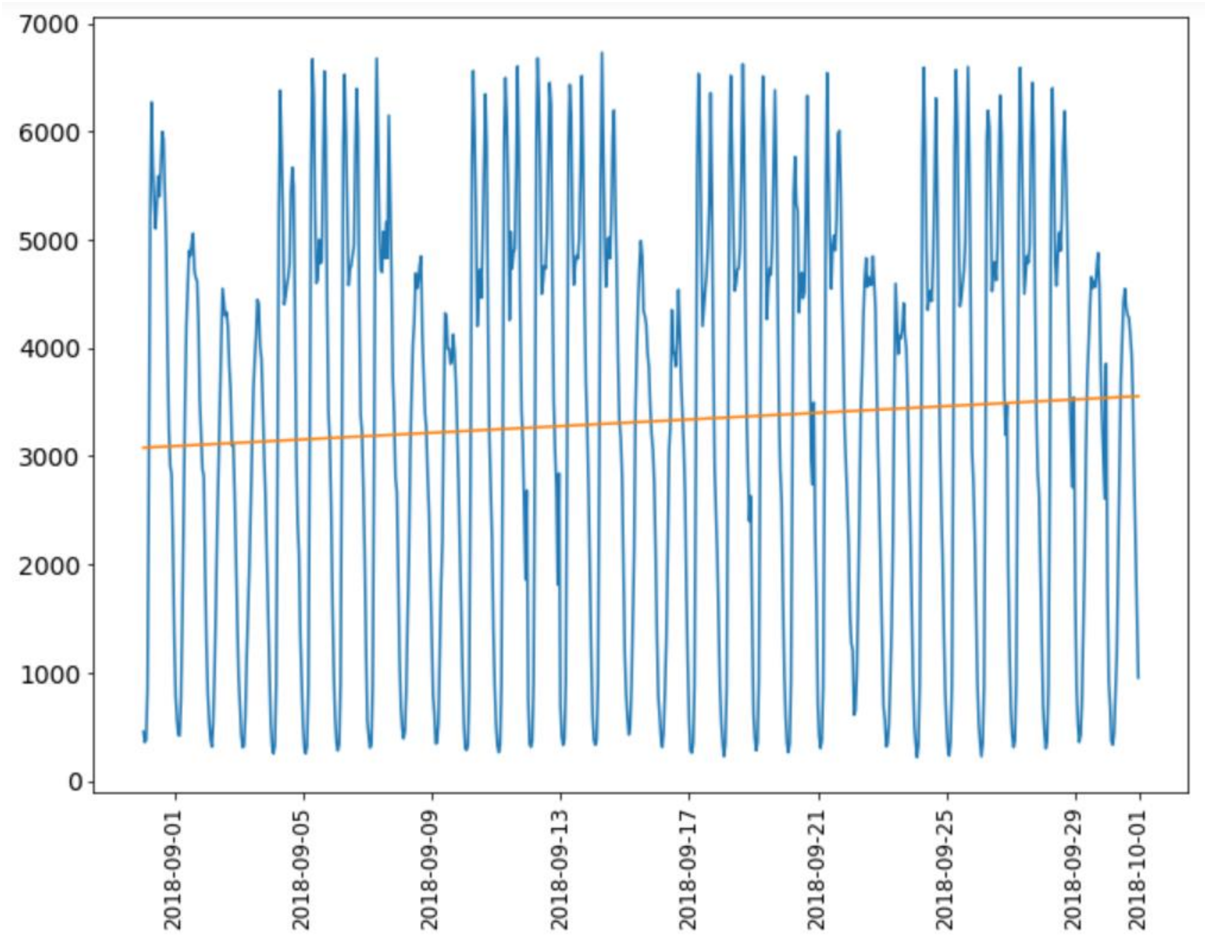

```
R2 score is  0.8259169101408546
model coefficients:
 [ 1.02477193 -1.50441625 -2.06014001 -3.61709086 -0.49864781  2.81397581]
intercept:  23.34270072992701
the root mean square error is  3.298247031404574
```

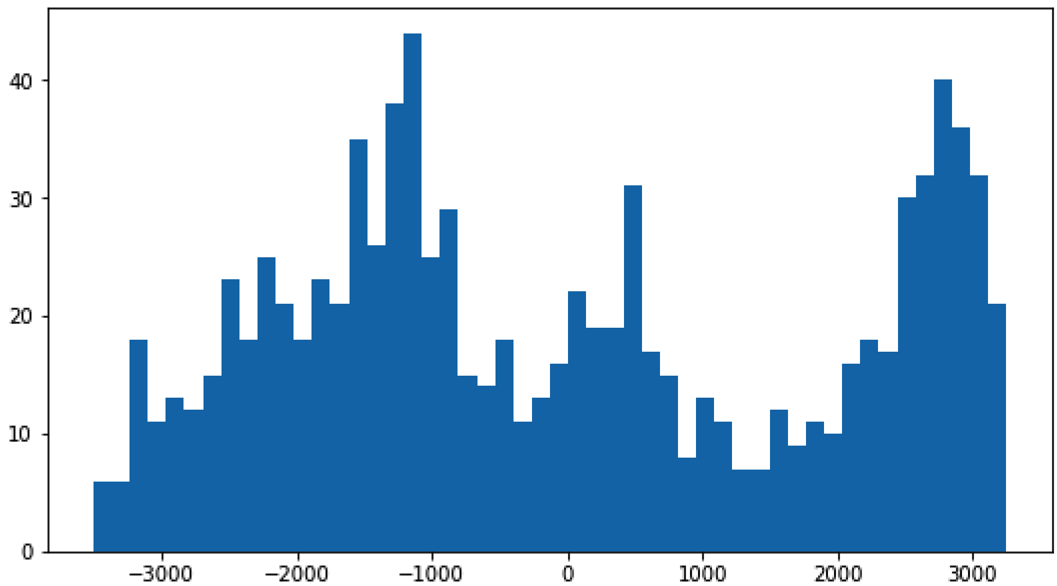
Out[2]:

	date_time	traffic_volume
0	10/2/2012 9:00	5545
1	10/2/2012 10:00	4516
2	10/2/2012 11:00	4767
3	10/2/2012 12:00	5026
4	10/2/2012 13:00	4918

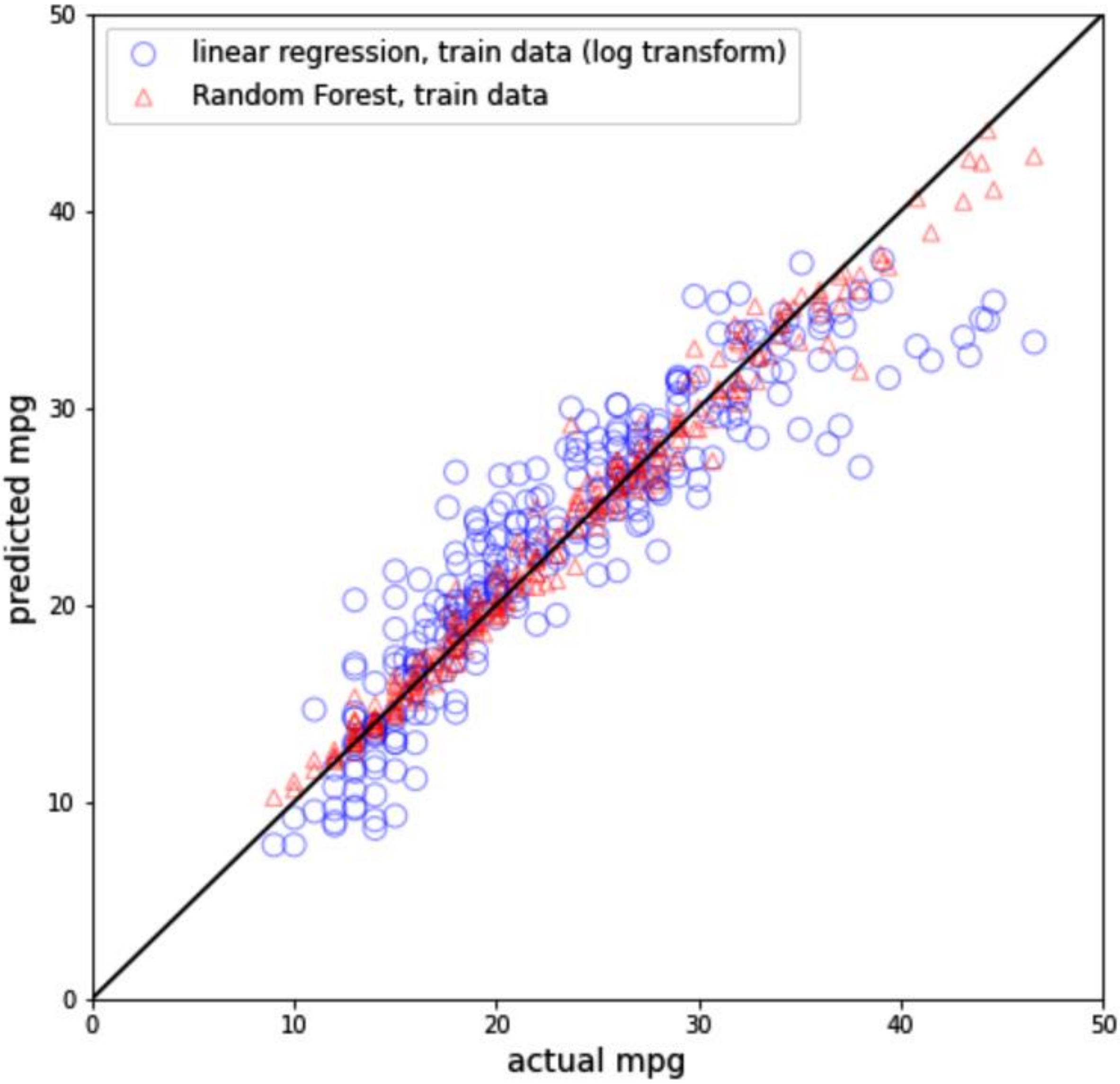


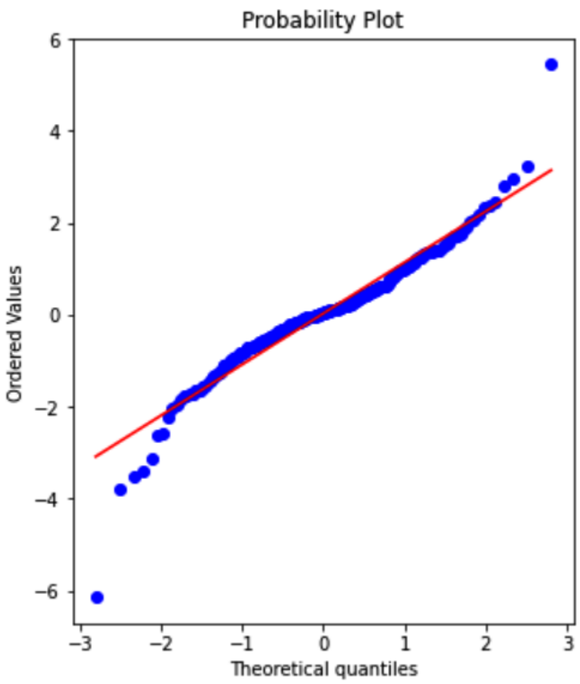

```
[3079.01901131]
[[1.78032483e-13]]
```

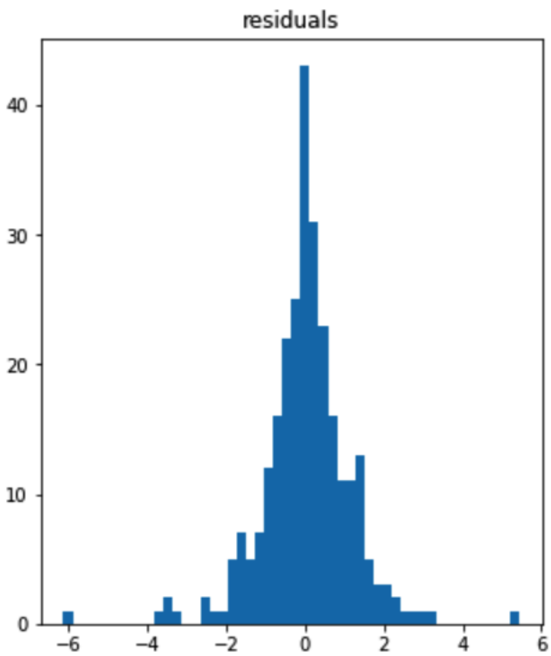




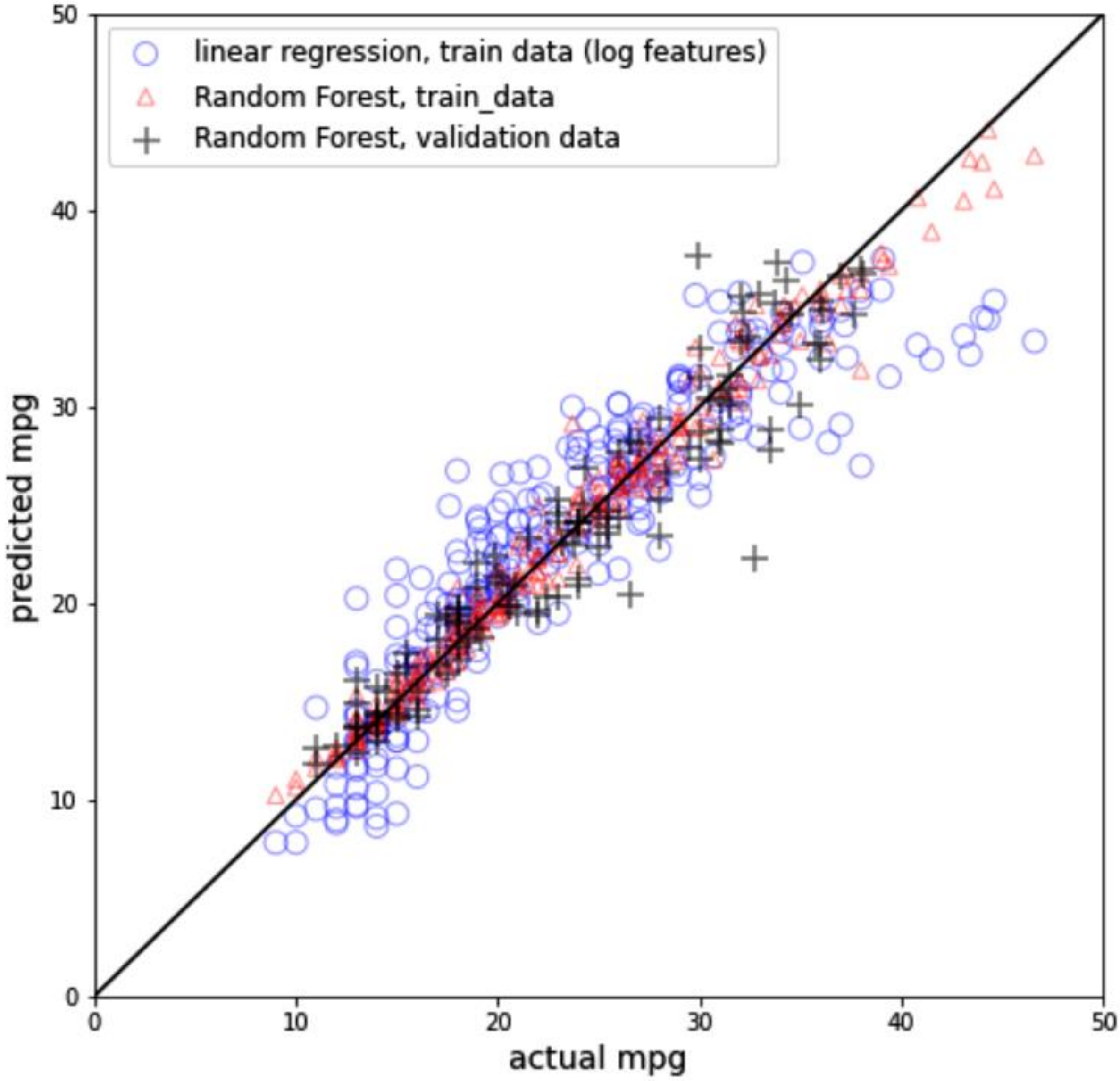

```
R2 score is  0.9790825512446402
the root mean square error is  1.143297473599363
```


linear regression, train data (log transform)
Random Forest, train data
predicted mpg
actual mpg
0
10
20
30
40
50

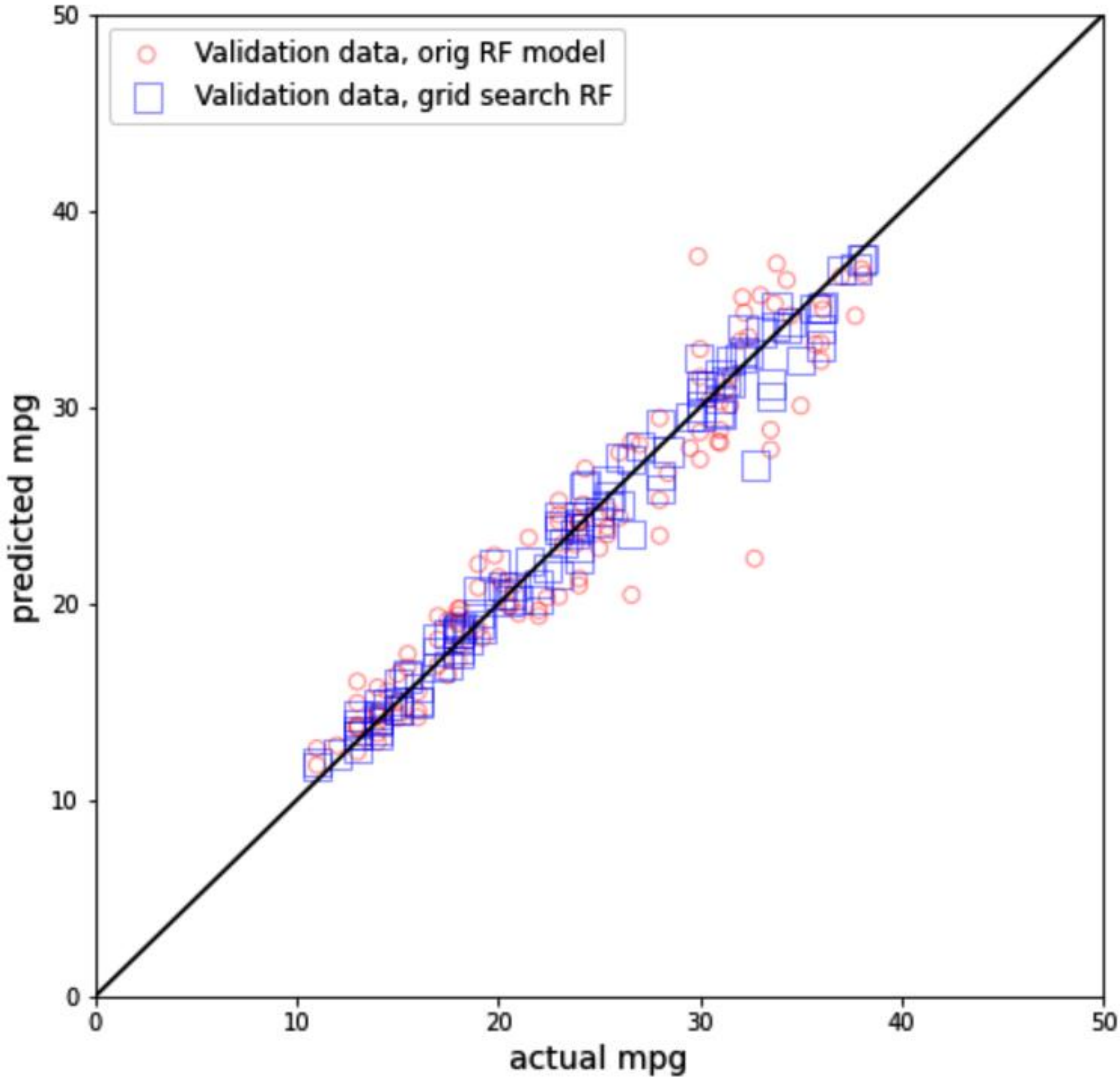

Probability Plot
Ordered Values
Theoretical quantiles


residuals

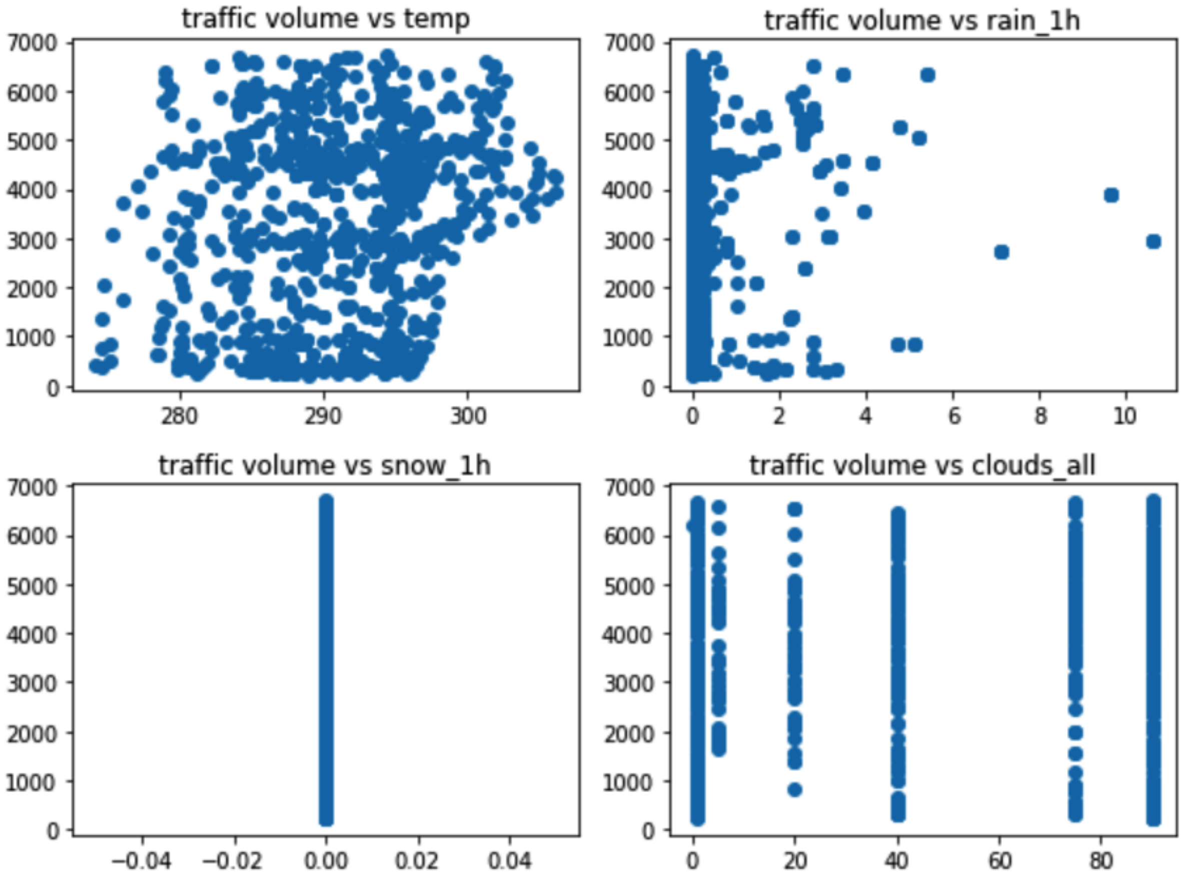
```
the validation RMSE is  2.2932374329163605
```




```
Fitting 1 folds for each of 486 candidates, totalling 486 fits
best model:
{'criterion': 'mse',
 'max_depth': 15,
 'max_features': 4,
 'min_samples_leaf': 2,
 'min_samples_split': 2,
 'n_estimators': 900}
1.1678655741624786
```




```
  holiday    temp  rain_1h  snow_1h  clouds_all  weather_main  \
0    None  294.76     0.25      0.0          75          Rain
1    None  294.61     0.25      0.0          75          Rain
2    None  294.54     0.25      0.0          90          Rain
3    None  294.54     0.25      0.0          90  Thunderstorm
4    None  294.04     1.40      0.0          90          Rain

      weather_description            date_time  traffic_volume
0              light rain  2018-08-31 00:00:00             764
1              light rain  2018-08-31 01:00:00             456
2              light rain  2018-08-31 02:00:00             358
3  proximity thunderstorm  2018-08-31 02:00:00             358
4           moderate rain  2018-08-31 03:00:00             378
(968, 9)
```

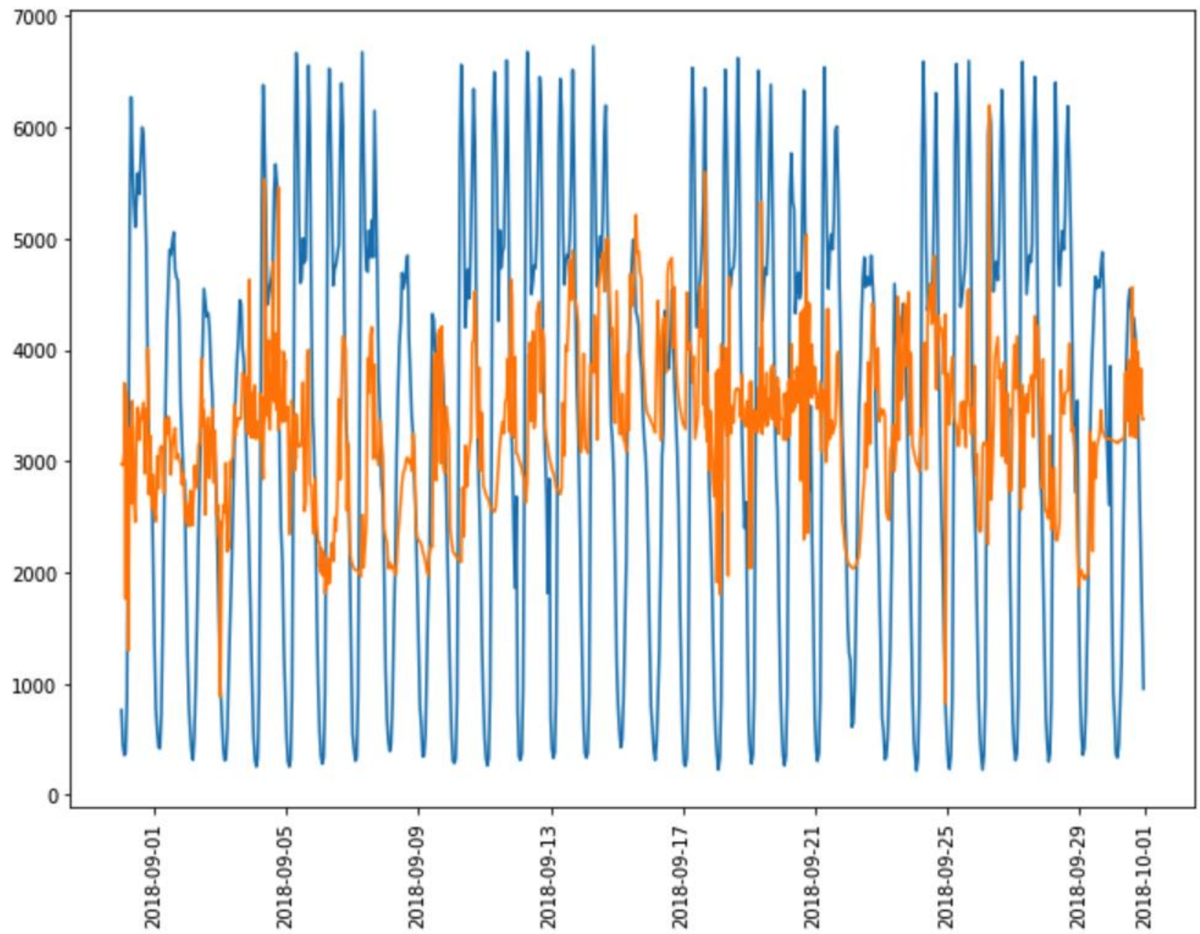


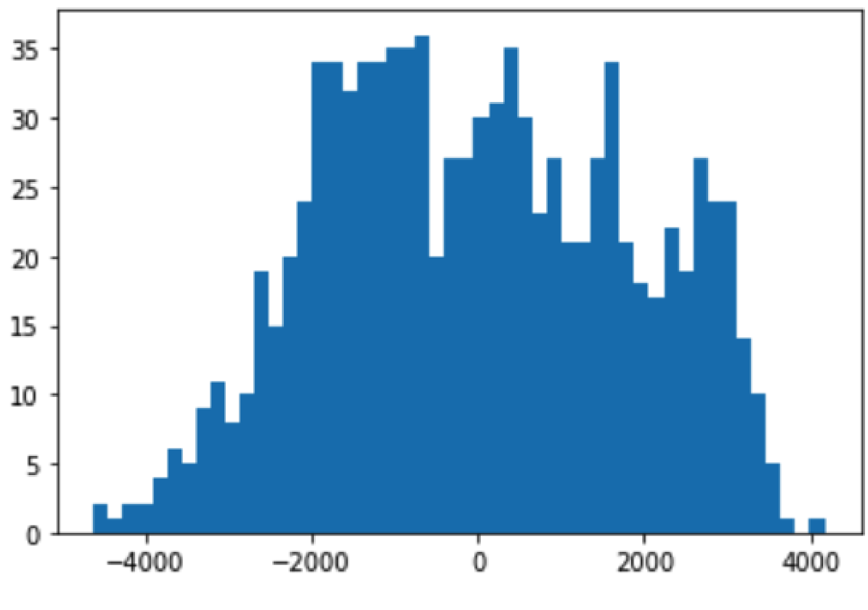
Out[46]:

	temp	rain_1h	clouds_all	int_time	holiday_Labor Day	holiday_None	weather_main_Clear	weather_main_Clouds	weather_main_Drizzle	weather_main_Fog	...
0	294.76	0.25	75	0.0	0	1	0	0	0	0	...
1	294.61	0.25	75	3600.0	0	1	0	0	0	0	...
2	294.54	0.25	90	7200.0	0	1	0	0	0	0	...
3	294.54	0.25	90	7200.0	0	1	0	0	0	0	...
4	294.04	1.40	90	10800.0	0	1	0	0	0	0	...

5 rows × 36 columns

```
[ 8.27936773e+01  1.23720714e+01  8.25196608e+00  4.17553926e-04
 -6.97245578e+02  6.97245577e+02  1.08786701e+03  2.46037578e+01
  3.06313660e+02 -2.57506459e+02 -2.98144206e+02 -2.09165922e+02
  4.80612430e+01 -7.02029087e+02  2.52845633e+03  2.38788198e+02
 -4.06271139e+02  7.09318270e+01 -2.57506459e+02 -2.98144206e+02
  1.19834748e+03  1.50177878e+02 -4.85762684e+02 -6.24466385e+02
 -2.09165922e+02 -2.45963831e+02 -3.88937455e+02  7.68313580e+02
  7.45203887e+02  2.46047307e+03  1.03821187e+02 -1.44058932e+03
 -1.16769699e+03 -2.59254554e+03 -6.03660704e+02  4.56197190e+02]
-22174.31595560447
0.12478430548635289
```

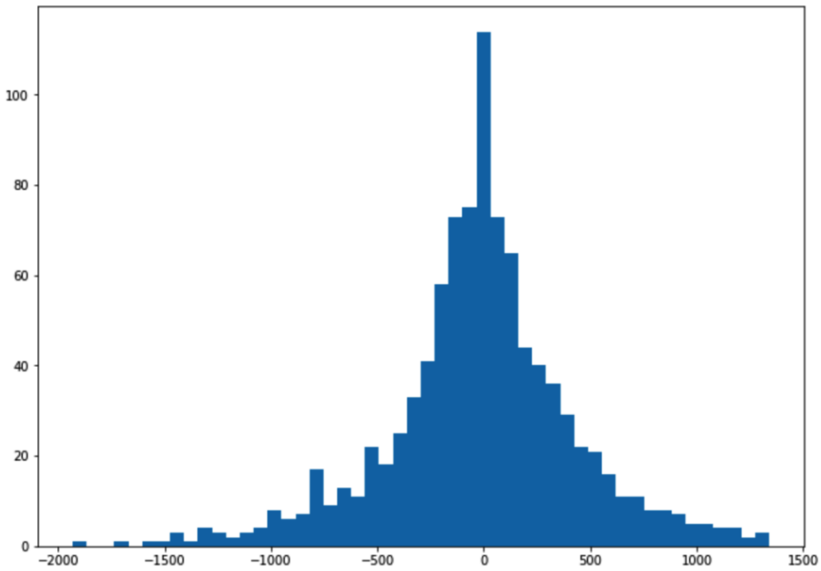

7000
6000
5000
4000
3000
2000
1000
0
2018-09-01
2018-09-05
2018-09-09
2018-09-13
2018-09-17
2018-09-21
2018-09-25
2018-09-29
2018-10-01

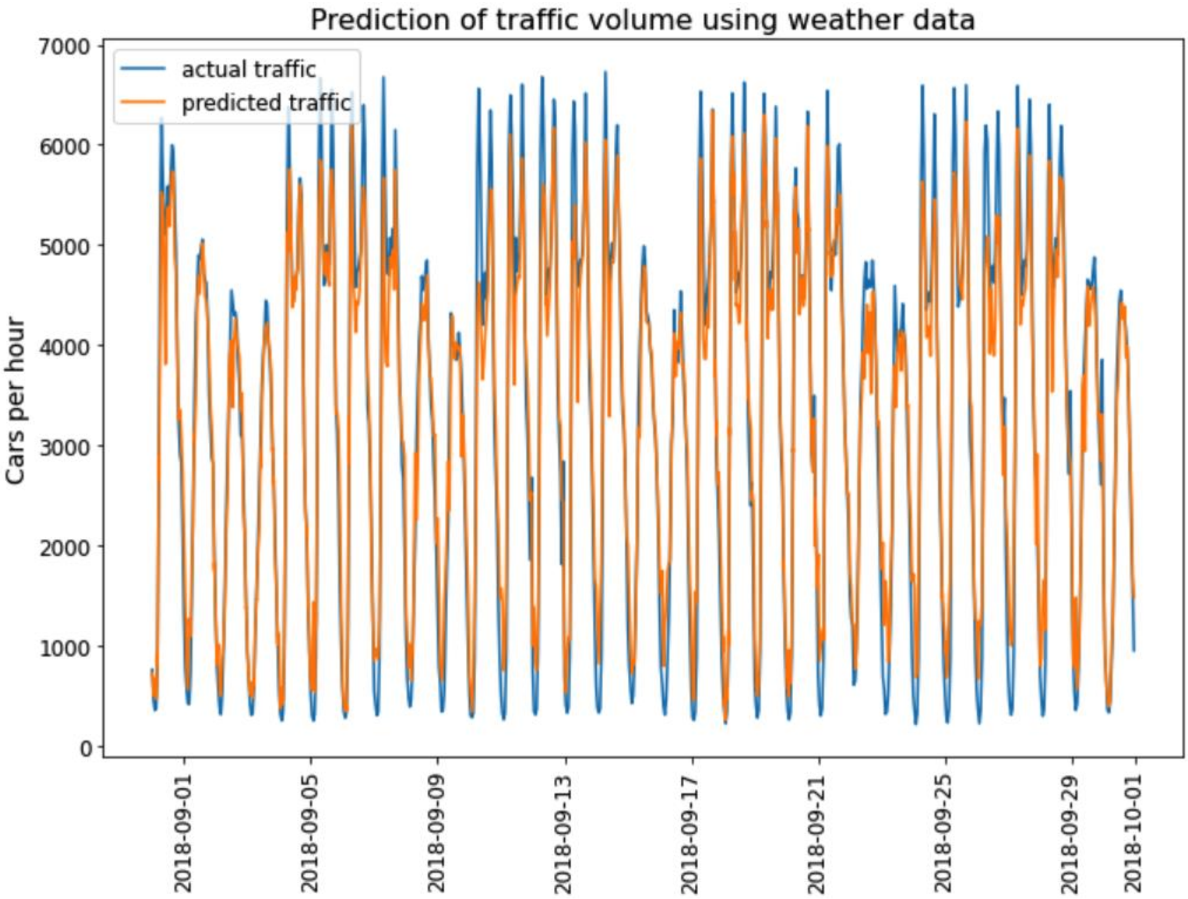

35
30
25
20
15
10
5
0
−4000
−2000
0
2000
4000

```
Out[52]: Index(['temp', 'rain_1h', 'clouds_all', 'int_time', 'holiday_Labor Day',
       'holiday_None', 'weather_main_Clear', 'weather_main_Clouds',
       'weather_main_Drizzle', 'weather_main_Fog', 'weather_main_Haze',
       'weather_main_Mist', 'weather_main_Rain', 'weather_main_Thunderstorm',
       'weather_description_Sky is Clear', 'weather_description_broken clouds',
       'weather_description_drizzle', 'weather_description_few clouds',
       'weather_description_fog', 'weather_description_haze',
       'weather_description_heavy intensity drizzle',
       'weather_description_heavy intensity rain',
       'weather_description_light intensity drizzle',
       'weather_description_light rain', 'weather_description_mist',
       'weather_description_moderate rain',
       'weather_description_overcast clouds',
       'weather_description_proximity shower rain',
       'weather_description_proximity thunderstorm',
       'weather_description_proximity thunderstorm with rain',
       'weather_description_scattered clouds',
       'weather_description_sky is clear', 'weather_description_thunderstorm',
       'weather_description_thunderstorm with heavy rain',
       'weather_description_thunderstorm with light drizzle',
       'weather_description_thunderstorm with light rain', 'sin_t', 'cos_t'],
      dtype='object')
```

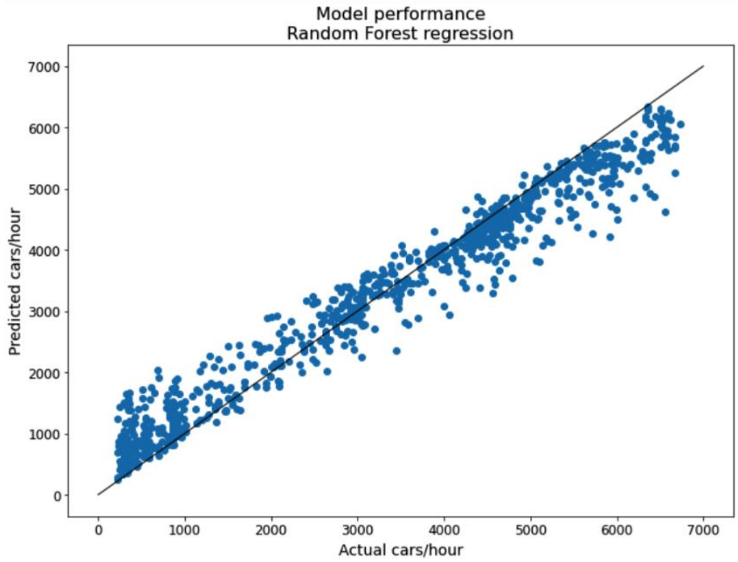
```
[ 8.28107060e+01  1.26203181e+01  8.24244476e+00  4.16119259e-04
 -6.98263168e+02  6.97891031e+02  1.08771818e+03  2.45839013e+01
  3.08964701e+02 -2.56895343e+02 -2.97414530e+02 -2.09558127e+02
  4.82579100e+01 -7.04325158e+02  2.52967857e+03  2.39476255e+02
 -4.08640308e+02  7.01252031e+01 -2.56859474e+02 -2.97365101e+02
  1.20821386e+03  1.49254877e+02 -4.89857091e+02 -6.24935205e+02
 -2.09245319e+02 -2.46088298e+02 -3.89358352e+02  7.67961945e+02
  7.47634289e+02  2.46226663e+03  1.02939561e+02 -1.44040378e+03
 -1.16767842e+03 -2.59893840e+03 -6.05853914e+02  4.58041745e+02
 -4.44455705e+07 -2.34378006e+01]
-22156.158917458004
0.12479213645488285
```

```
Out[56]: 0.9481262295271707
```

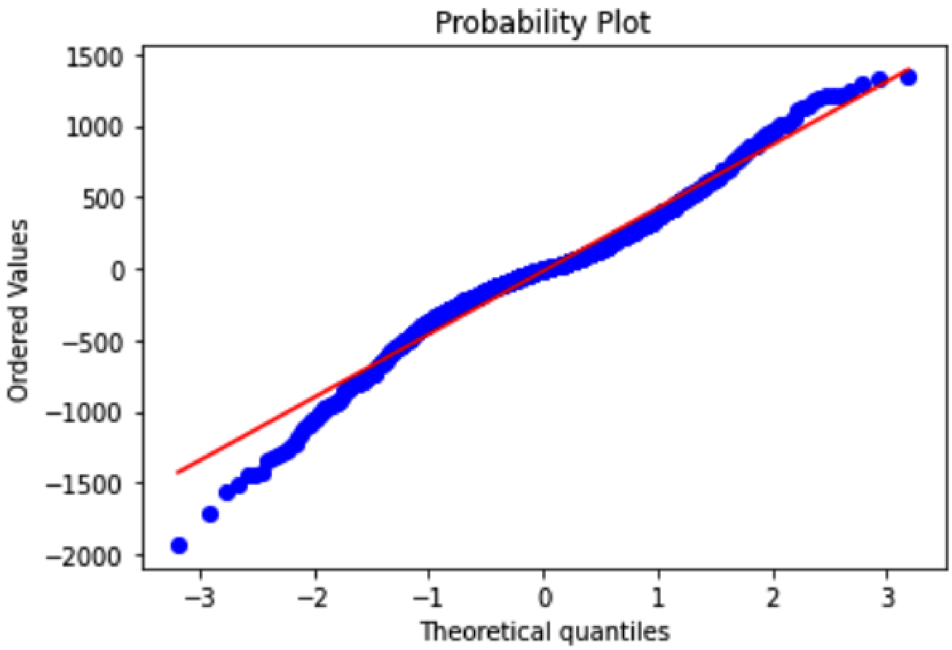




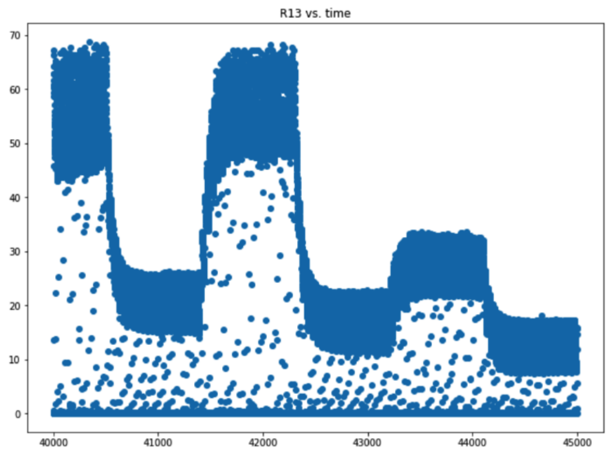
Prediction of traffic volume using weather data
actual traffic
predicted traffic
Cars per hour
0
1000
2000
3000
4000
5000
6000
7000
2018-09-01
2018-09-05
2018-09-09
2018-09-13
2018-09-17
2018-09-21
2018-09-25
2018-09-29
2018-10-01

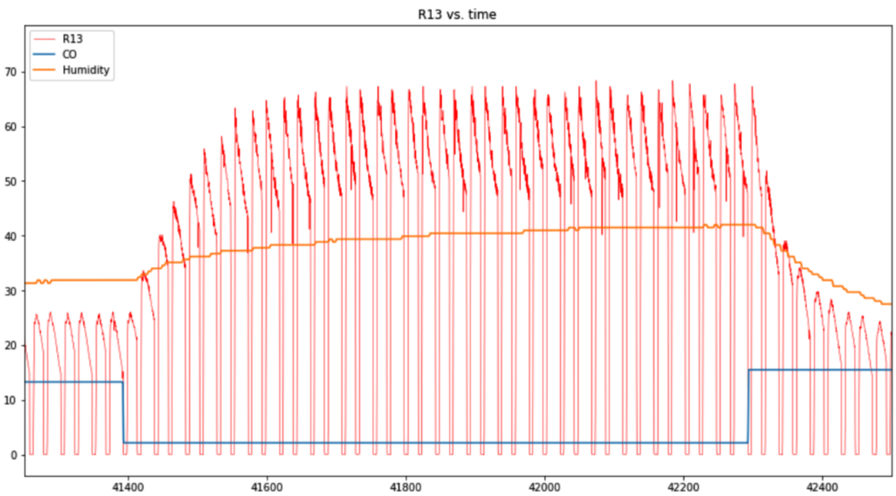


Model performance
Random Forest regression
Predicted cars/hour
Actual cars/hour
0
1000
2000
3000
4000
5000
6000
7000

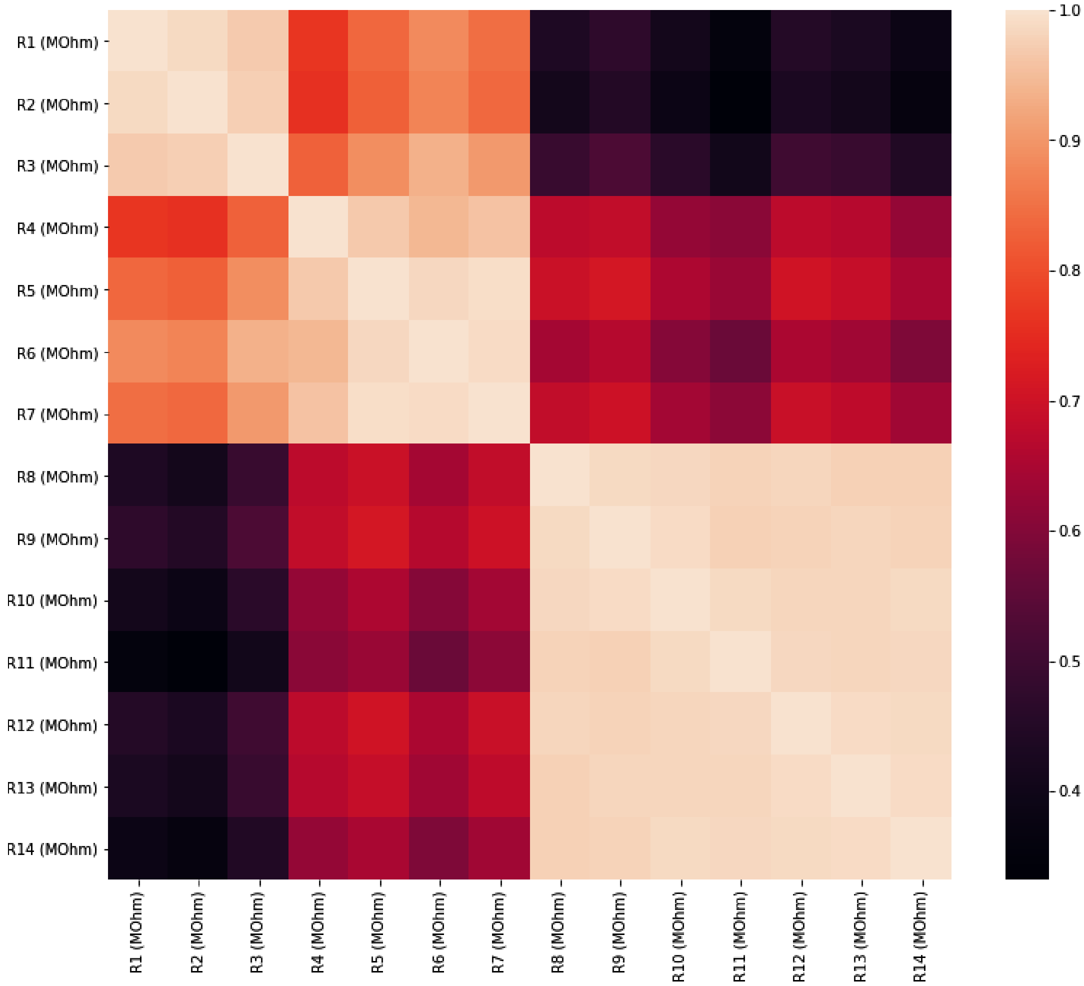


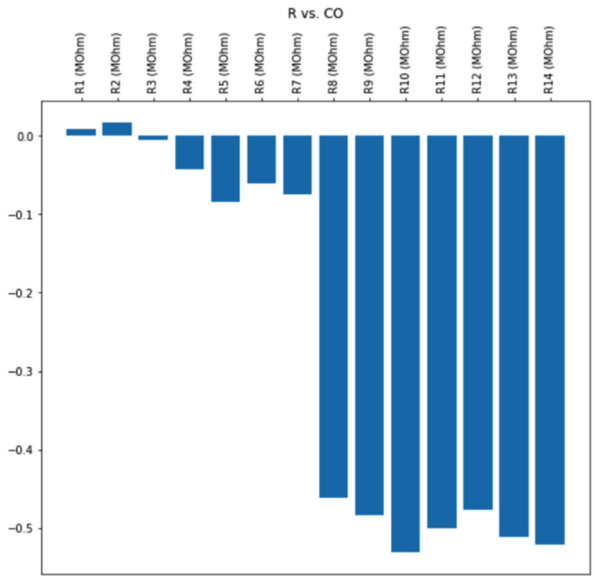
	Time (s)	CO (ppm)	Humidity (%r.h.)	Temperature (C)	Flow rate (mL/min)	Heater voltage (V)	R1 (MOhm)	R2 (MOhm)	R3 (MOhm)	R4 (MOhm)	R5 (MOhm)	R6 (MOhm)	R7 (MOhm)	R8 (MOhm)	R9 (MOhm)	R10 (MOhm)	F (MOh
0	0.000	0.0	49.21	26.38	247.2771	0.1994	0.5114	0.5863	0.5716	1.9386	1.1669	0.7103	0.5541	51.0146	40.8079	47.8748	4.6
1	0.311	0.0	49.21	26.38	243.3618	0.7158	0.0626	0.1586	0.1161	0.1347	0.1385	0.1545	0.1307	0.1935	0.1341	0.1773	0.14
2	0.620	0.0	49.21	26.38	242.4944	0.8840	0.0654	0.1496	0.1075	0.1076	0.1131	0.1363	0.1188	0.1195	0.1049	0.1289	0.1
3	0.930	0.0	49.21	26.38	241.6242	0.8932	0.0722	0.1444	0.1074	0.1032	0.1106	0.1306	0.1190	0.1125	0.1014	0.1232	0.1
4	1.238	0.0	49.21	26.38	240.8151	0.8974	0.0767	0.1417	0.1098	0.1025	0.1116	0.1284	0.1208	0.1111	0.1008	0.1226	0.1

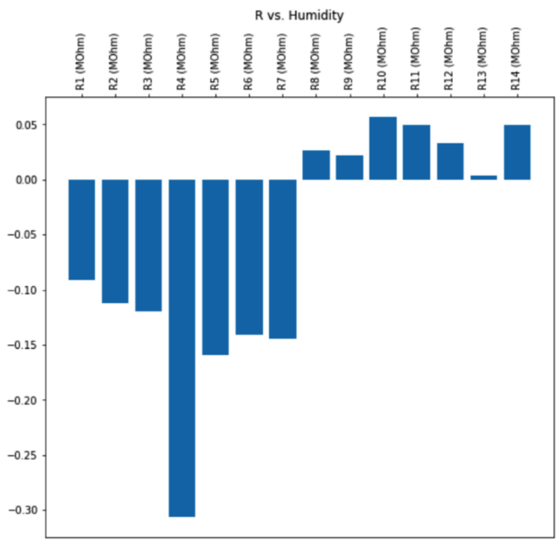


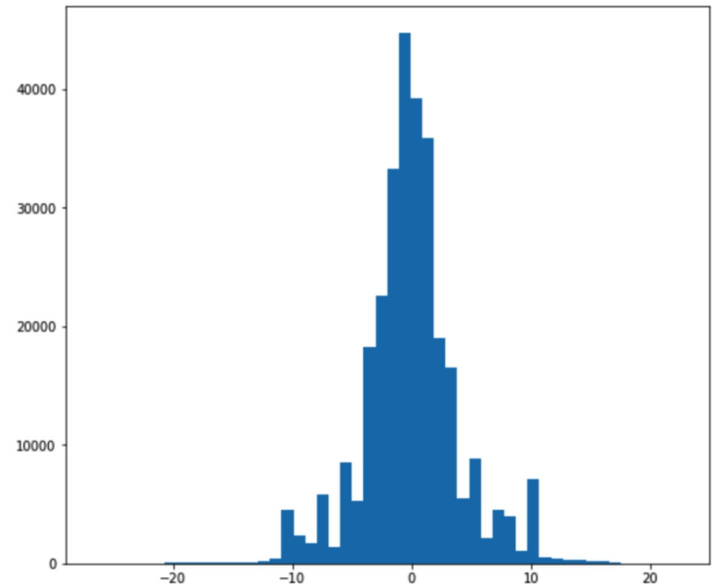


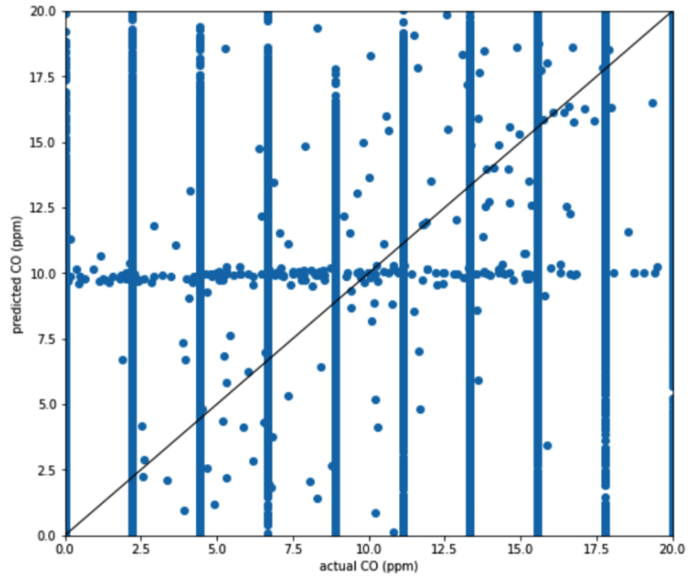
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x18d58f87908>

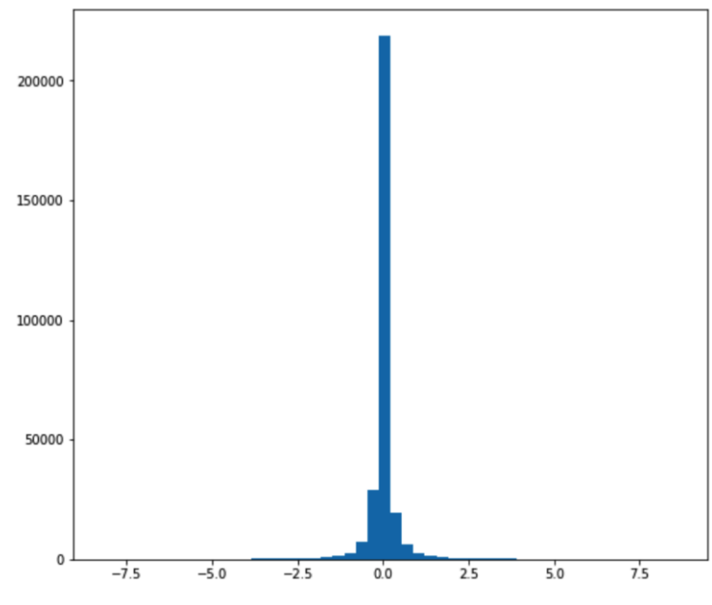


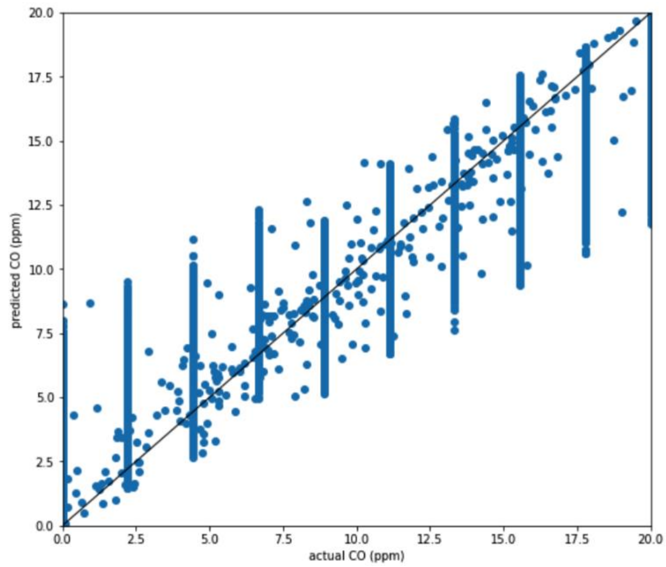


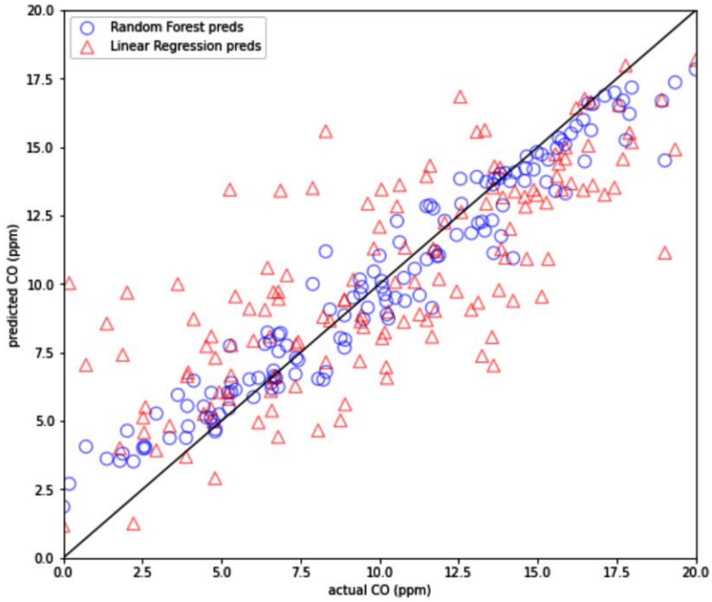



40000
30000
20000
10000
0
−20
−10
0
10
20


predicted CO (ppm)
actual CO (ppm)
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


200000
150000
100000
50000
0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5


predicted CO (ppm)
actual CO (ppm)


Random Forest preds
Linear Regression preds
predicted CO (ppm)
actual CO (ppm)

Chapter 12: Using Time in pandas

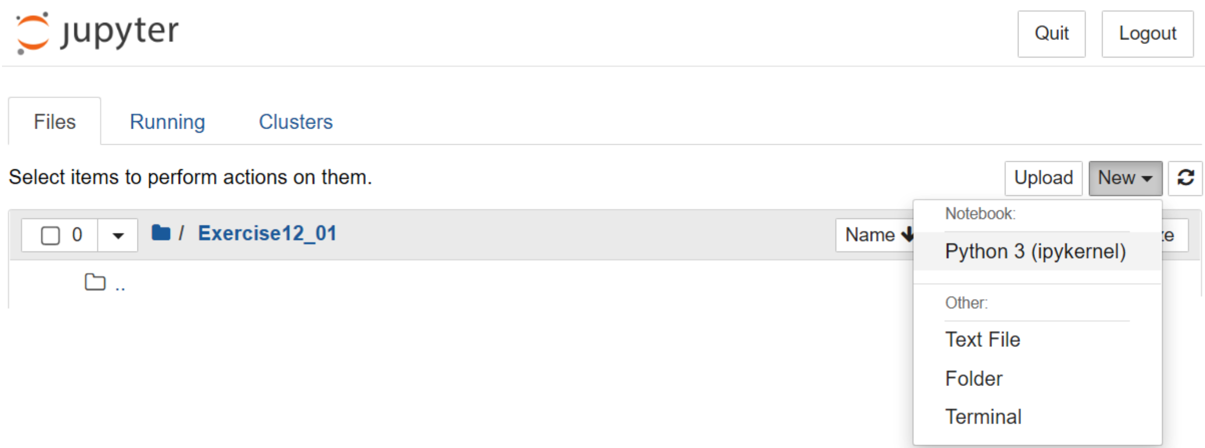
```
Out[3]: Timestamp('2020-12-25 00:00:00')
```

```
Out[9]: datetime.datetime(2020, 12, 25, 0, 0)
```

	TS	DT		TS	DT		TS	DT
0	asm8	-	25	is_month_end	-	50	time	time
1	astimezone	astimezone	26	is_month_start	-	51	timestamp	timestamp
2	ceil	-	27	is_quarter_end	-	52	timetuple	timetuple
3	combine	combine	28	is_quarter_start	-	53	timetz	timetz
4	ctime	ctime	29	is_year_end	-	54	to_datetime64	-
5	date	date	30	is_year_start	-	55	to_julian_date	-
6	day	day	31	isocalendar	isocalendar	56	to_numpy	-
7	day_name	-	32	isoformat	isoformat	57	to_period	-
8	day_of_week	-	33	isoweekday	isoweekday	58	to_pydatetime	-
9	day_of_year	-	34	max	max	59	today	today
10	dayofweek	-	35	microsecond	microsecond	60	toordinal	toordinal
11	dayofyear	-	36	min	min	61	tz	-
12	days_in_month	-	37	minute	minute	62	tz_convert	-
13	daysinmonth	-	38	month	month	63	tz_localize	-
14	dst	dst	39	month_name	-	64	tzinfo	tzinfo
15	floor	-	40	nanosecond	-	65	tzname	tzname
16	fold	fold	41	normalize	-	66	utcfromtimestamp	utcfromtimestamp
17	freq	-	42	now	now	67	utcnow	utcnow
18	freqstr	-	43	quarter	-	68	utcoffset	utcoffset
19	fromisocalendar	fromisocalendar	44	replace	replace	69	utctimetuple	utctimetuple
20	fromisoformat	fromisoformat	45	resolution	resolution	70	value	-
21	fromordinal	fromordinal	46	round	-	71	week	-
22	fromtimestamp	fromtimestamp	47	second	second	72	weekday	weekday
23	hour	hour	48	strftime	strftime	73	weekofyear	-
24	is_leap_year	-	49	strptime	strptime	74	year	year

```
3 PM 5 minutes 9 seconds 1234 microseconds 987 nanoseconds
```

```
Out[7]: Timestamp('2020-07-31 13:51:00')
```



Out[14]:

	datetime	power
0	2020-1-1 00:00	221.403465
1	2020-1-1 01:55	327.370592
2	2020-1-1 03:50	223.272440
3	2020-1-1 04:04	328.380592
4	2020-1-1 05:45	329.109239

```
2020-1-1 00:00
<class 'str'>
```

Out[9]:

	datetime	power
0	2020-01-01 00:00:00	221.403465
1	2020-01-01 01:55:00	327.370592
2	2020-01-01 03:50:00	223.272440
3	2020-01-01 04:04:00	328.380592
4	2020-01-01 05:45:00	329.109239
...	...	...
2114	2020-04-15 11:02:00	131.620792
2115	2020-04-15 11:16:00	8.703348
2116	2020-04-15 12:43:00	23.701833
2117	2020-04-15 12:57:00	110.785479
2118	2020-04-15 13:12:00	22.869297

2119 rows × 2 columns

Out[10]: `pandas._libs.tslibs.timestamps.Timestamp`

Out[11]:

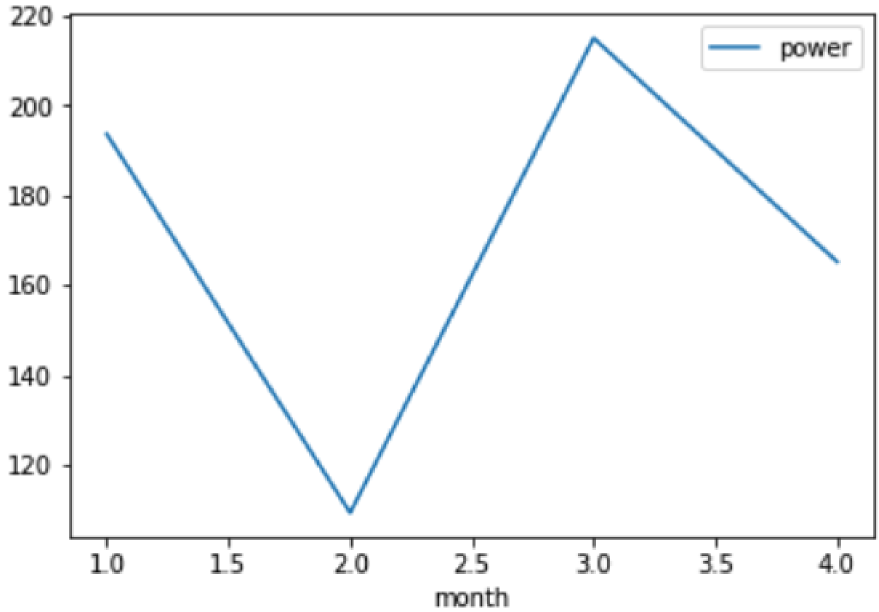
	datetime	power	month
0	2020-01-01 00:00:00	221.403465	1
1	2020-01-01 01:55:00	327.370592	1
2	2020-01-01 03:50:00	223.272440	1
3	2020-01-01 04:04:00	328.380592	1
4	2020-01-01 05:45:00	329.109239	1
...	...	...	...
2114	2020-04-15 11:02:00	131.620792	4
2115	2020-04-15 11:16:00	8.703348	4
2116	2020-04-15 12:43:00	23.701833	4
2117	2020-04-15 12:57:00	110.785479	4
2118	2020-04-15 13:12:00	22.869297	4

2119 rows × 3 columns

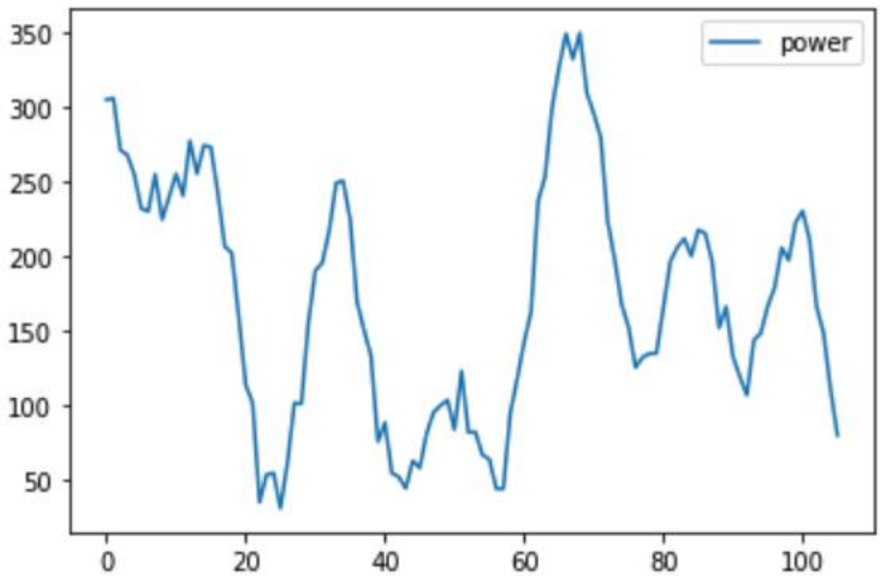
```
                datetime       power  month  day_of_month
0    2020-01-01 00:00:00  221.403466      1             1
1    2020-01-01 01:55:00  327.370592      1             1
2    2020-01-01 03:50:00  223.272440      1             1
3    2020-01-01 04:04:00  328.380592      1             1
4    2020-01-01 05:45:00  329.109239      1             1
...                  ...         ...    ...           ...
2114 2020-04-15 11:02:00  131.620792      4            15
2115 2020-04-15 11:16:00    8.703348      4            15
2116 2020-04-15 12:43:00   23.701833      4            15
2117 2020-04-15 12:57:00  110.785479      4            15
2118 2020-04-15 13:12:00   22.869297      4            15

[2119 rows x 4 columns]
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 25 26 27 28 29 30 31]
```

Out[9]: `array([2, 3, 4, 5, 6, 0, 1], dtype=int64)`

```
Out[10]: <AxesSubplot:xlabel='month'>
```




```
Out[11]: <AxesSubplot:>
```




```
Out[14]: pandas.core.indexes.datetimes.DatetimeIndex
```

```
DatetimeIndex(['2012-01-02', '2012-01-09', '2012-01-16', '2012-01-23',
               '2012-01-30', '2012-02-06', '2012-02-13', '2012-02-20',
               '2012-02-27', '2012-03-05',
               ...
               '2019-10-28', '2019-11-04', '2019-11-11', '2019-11-18',
               '2019-11-25', '2019-12-02', '2019-12-09', '2019-12-16',
               '2019-12-23', '2019-12-30'],
              dtype='datetime64[ns]', length=418, freq='W-MON')
```

```
Out[22]: pandas._libs.tslibs.timedeltas.Timedelta
```

Out[12]:

	date	value
0	2012-01-02	1037.650000
1	2012-01-09	1039.827529
2	2012-01-16	1043.808982
3	2012-01-23	1048.964772
4	2012-01-30	1055.070231

Out[13]:

	date	value	int_date
0	2012-01-02	1037.650000	0.0
1	2012-01-09	1039.827529	7.0
2	2012-01-16	1043.808982	14.0
3	2012-01-23	1048.964772	21.0
4	2012-01-30	1055.070231	28.0
...	...	...	...
413	2019-12-02	19313.980376	2891.0
414	2019-12-09	19380.399465	2898.0
415	2019-12-16	19446.898818	2905.0
416	2019-12-23	19513.478340	2912.0
417	2019-12-30	19580.137933	2919.0

418 rows × 3 columns

```
Out[51]: PeriodIndex(['2019-12-31/2020-01-06', '2020-01-07/2020-01-13',
                      '2020-01-14/2020-01-20', '2020-01-21/2020-01-27',
                      '2020-01-28/2020-02-03', '2020-02-04/2020-02-10',
                      '2020-02-11/2020-02-17', '2020-02-18/2020-02-24',
                      '2020-02-25/2020-03-02', '2020-03-03/2020-03-09',
                      '2020-03-10/2020-03-16', '2020-03-17/2020-03-23',
                      '2020-03-24/2020-03-30'],
                     dtype='period[W-MON]')
```

```
DatetimeIndex(['2019-12-31', '2020-01-07', '2020-01-14', '2020-01-21',
               '2020-01-28', '2020-02-04', '2020-02-11', '2020-02-18',
               '2020-02-25', '2020-03-03', '2020-03-10', '2020-03-17',
               '2020-03-24'],
              dtype='datetime64[ns]', freq='W-TUE')
 DatetimeIndex(['2020-01-06 23:59:59.999999999',
               '2020-01-13 23:59:59.999999999',
               '2020-01-20 23:59:59.999999999',
               '2020-01-27 23:59:59.999999999',
               '2020-02-03 23:59:59.999999999',
               '2020-02-10 23:59:59.999999999',
               '2020-02-17 23:59:59.999999999',
               '2020-02-24 23:59:59.999999999',
               '2020-03-02 23:59:59.999999999',
               '2020-03-09 23:59:59.999999999',
               '2020-03-16 23:59:59.999999999',
               '2020-03-23 23:59:59.999999999',
               '2020-03-30 23:59:59.999999999'],
              dtype='datetime64[ns]', freq=None)
```

```
Out[57]: PeriodIndex(['2020-01-14/2020-01-20', '2020-01-21/2020-01-27',
                      '2020-01-28/2020-02-03', '2020-02-04/2020-02-10',
                      '2020-02-11/2020-02-17', '2020-02-18/2020-02-24',
                      '2020-02-25/2020-03-02', '2020-03-03/2020-03-09',
                      '2020-03-10/2020-03-16', '2020-03-17/2020-03-23',
                      '2020-03-24/2020-03-30', '2020-03-31/2020-04-06',
                      '2020-04-07/2020-04-13'],
                     dtype='period[W-MON]')
```

```
Int64Index([31, 29, 30, 28], dtype='int64')
```

```
DatetimeIndex(['2012-12-31', '2013-09-30', '2014-03-31', '2014-06-30',
               '2018-12-31', '2019-09-30'],
              dtype='datetime64[ns]', freq=None)
```

```
2012-01-02      1
2012-01-09      2
2012-01-16      3
2012-01-23      4
2012-01-30      5
               ..
2019-12-02     49
2019-12-09     50
2019-12-16     51
2019-12-23     52
2019-12-30      1
Freq: W-MON, Name: week, Length: 418, dtype: UInt32
```

```
<class 'pandas._libs.tslibs.timedeltas.Timedelta'>
```

```
231 days 00:00:00  equals  19958400000000000 nanoseconds
```

```
19958400.0
```

Out[10]:

	datetime	power
0	2020-1-1 00:00	221.403466
1	2020-1-1 01:55	327.370592
2	2020-1-1 03:50	223.272440
3	2020-1-1 04:04	328.380592
4	2020-1-1 05:45	329.109239

Out[11]:

```
datetime     object
power       float64
dtype: object
```

```
Out[5]: datetime    datetime64[ns]
        power              float64
        dtype: object
```

Out[37]:

	datetime	power	sec_from_start
0	2020-01-01 00:00:00	221.403465	0.0
1	2020-01-01 01:55:00	327.370592	6900.0
2	2020-01-01 03:50:00	223.272440	13800.0
3	2020-01-01 04:04:00	328.380592	14640.0
4	2020-01-01 05:45:00	329.109239	20700.0
...	...	...	...
2114	2020-04-15 11:02:00	131.620792	9111720.0
2115	2020-04-15 11:16:00	8.703348	9112560.0
2116	2020-04-15 12:43:00	23.701833	9117780.0
2117	2020-04-15 12:57:00	110.785479	9118620.0
2118	2020-04-15 13:12:00	22.869297	9119520.0

2119 rows × 3 columns

Out[33]:

	datetime	power	sec_from_start	days_from_start
0	2020-01-01 00:00:00	221.403465	0.0	0.000000
1	2020-01-01 01:55:00	327.370592	6900.0	0.079861
2	2020-01-01 03:50:00	223.272440	13800.0	0.159722
3	2020-01-01 04:04:00	328.380592	14640.0	0.169444
4	2020-01-01 05:45:00	329.109239	20700.0	0.239583
...	...	...	...	...
2114	2020-04-15 11:02:00	131.620792	9111720.0	105.459722
2115	2020-04-15 11:16:00	8.703348	9112560.0	105.469444
2116	2020-04-15 12:43:00	23.701833	9117780.0	105.529861
2117	2020-04-15 12:57:00	110.785479	9118620.0	105.539583
2118	2020-04-15 13:12:00	22.869297	9119520.0	105.550000

2119 rows × 4 columns

string	usage	examples
%a	abbreviated weekday	Mon, Wed
%A	fulll weekday	Sunday, Monday
%w	numeric weekday, Sunday = 0	0, 1
%d	zero-padded day of month	07, 29
%b	abbreviated month	Jan, Mar
%B	full month	February, September
%m	zero-padded month	01, 07, 11
%f	zero-padded microsecond	012989, 000002
%Y	numeric year with century	2020, 1987
%H	zero-padded hour on 24-hour clock	00, 23
%I	zero-padded hour	01, 11
%p	AM or PM	AM, PM
%M	zero-padded minutes	23, 59
%S	zero-padded seconds	00, 13

```
2020-12-20 13:57:03.130000 2020-12-20 13:57:03
December 20, 2020 01:57:03 PM
```

```
2020-12-20 13:57:03
December 20, 2020 01:57:03 PM
```

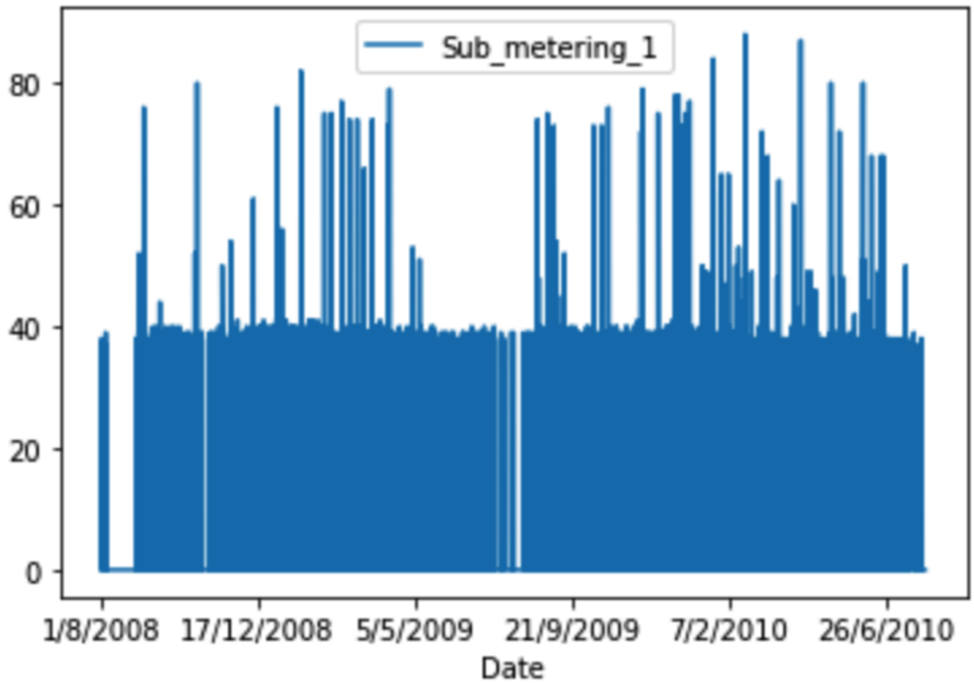
```
2020-12-20 13:57:03-07:00
December 21, 2020 05:57:03 AM
```

```
1677-09-21 00:12:43.145225
2262-04-11 23:47:16.854775807
```

Out[5]:

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	1/8/2008	00:00:00	0.500	0.226	239.750	2.400	0.000	0.000	1.0
1	1/8/2008	00:01:00	0.482	0.224	240.340	2.200	0.000	0.000	1.0
2	1/8/2008	00:02:00	0.502	0.234	241.680	2.400	0.000	0.000	0.0
3	1/8/2008	00:03:00	0.556	0.228	241.750	2.600	0.000	0.000	1.0
4	1/8/2008	00:04:00	0.854	0.342	241.550	4.000	0.000	1.000	7.0

Out[10]: <AxesSubplot:xlabel='Date'>



Out[13]:

	Date	Time	Kitchen_power_use
1074636	1/1/2009	00:00:00	0.0
1074637	1/1/2009	00:01:00	0.0
1074638	1/1/2009	00:02:00	0.0
1074639	1/1/2009	00:03:00	0.0
1074640	1/1/2009	00:04:00	0.0

Out[12]:

	Date	Time	Kitchen_power_use	timestamp
1074636	1/1/2009	00:00:00	0.0	2009-01-01 00:00:00
1074637	1/1/2009	00:01:00	0.0	2009-01-01 00:01:00
1074638	1/1/2009	00:02:00	0.0	2009-01-01 00:02:00
1074639	1/1/2009	00:03:00	0.0	2009-01-01 00:03:00
1074640	1/1/2009	00:04:00	0.0	2009-01-01 00:04:00

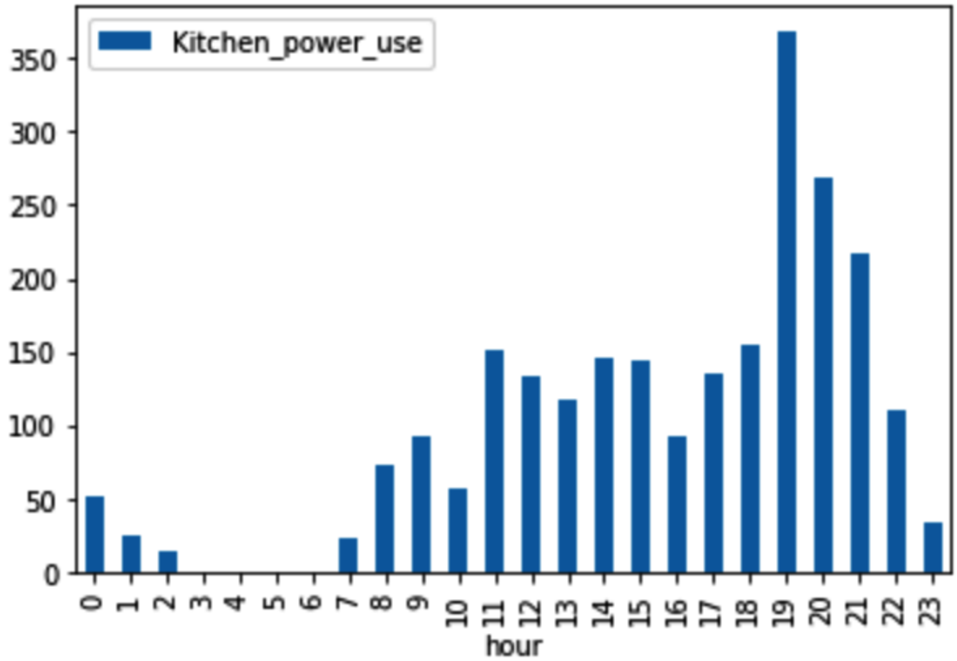
Out[34]:

	Date	Time	Kitchen_power_use	timestamp	hour	date
1074636	1/1/2009	00:00:00	0.0	2009-01-01 00:00:00	0	2009-01-01
1074637	1/1/2009	00:01:00	0.0	2009-01-01 00:01:00	0	2009-01-01
1074638	1/1/2009	00:02:00	0.0	2009-01-01 00:02:00	0	2009-01-01
1074639	1/1/2009	00:03:00	0.0	2009-01-01 00:03:00	0	2009-01-01
1074640	1/1/2009	00:04:00	0.0	2009-01-01 00:04:00	0	2009-01-01

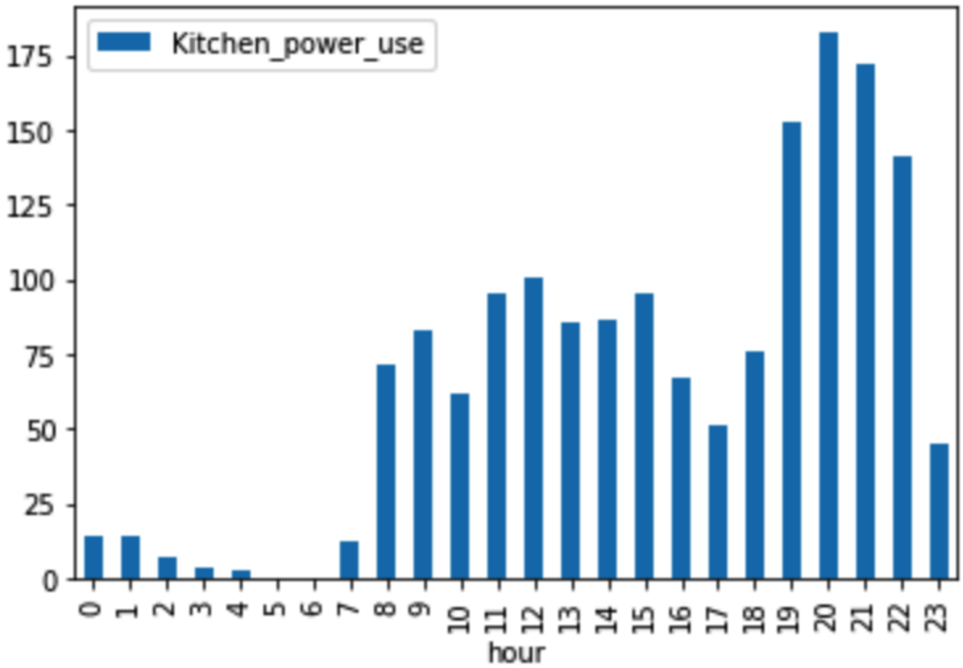
Out[55]:

	date	hour	Kitchen_power_use
20	2009-01-01	20	0.0
21	2009-01-01	21	0.0
22	2009-01-01	22	0.0
23	2009-01-01	23	0.0
24	2009-01-02	0	0.0
25	2009-01-02	1	0.0
26	2009-01-02	2	0.0
27	2009-01-02	3	0.0

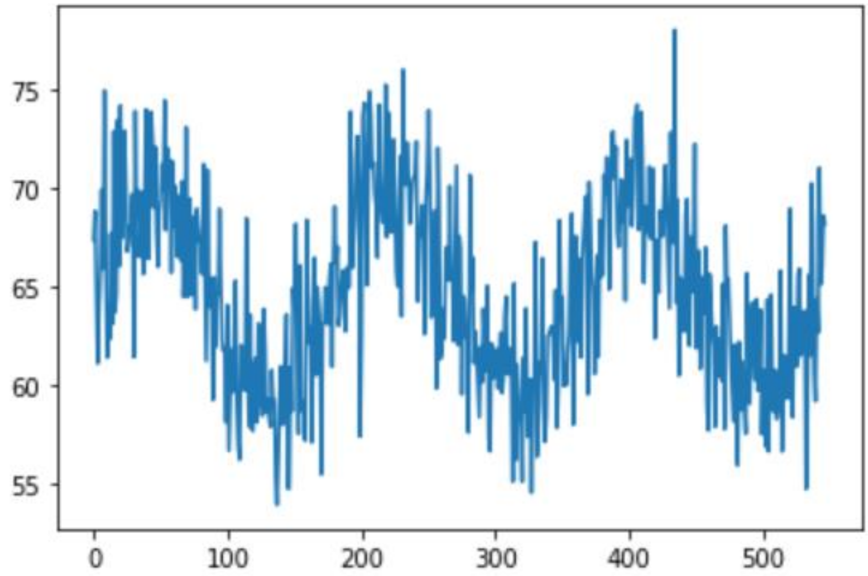
Out[50]: <AxesSubplot:xlabel='hour'>



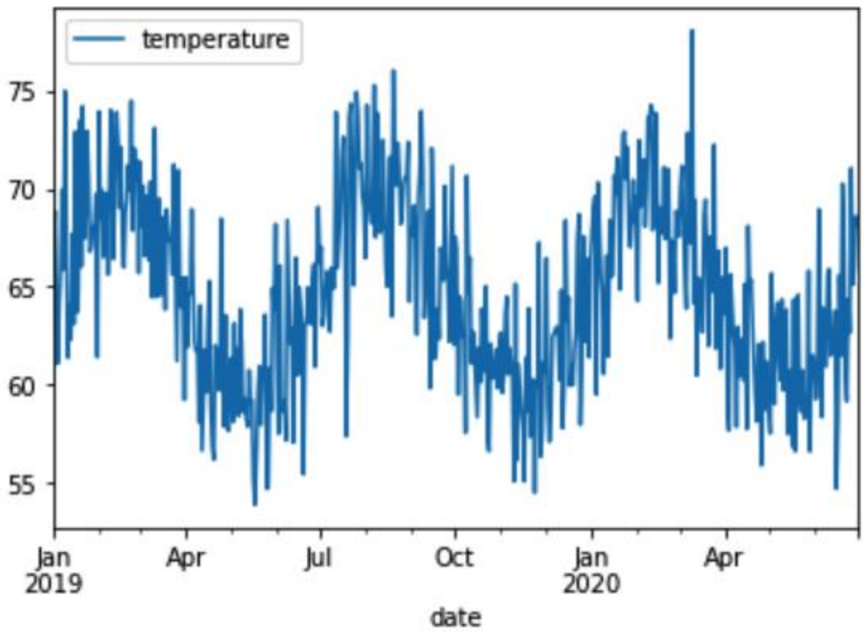
Out[56]: <AxesSubplot:xlabel='hour'>



Out[33]: <AxesSubplot:>



Out[34]: <AxesSubplot:xlabel='date'>




```
2019-01-01 00:00:00    2019-01-01 11:30:00
2019-01-02 00:00:00    2019-01-02 11:30:00
2019-01-03 00:00:00    2019-01-03 11:30:00
```

‘W’, ‘D’, ‘T’, ‘S’, ‘L’, ‘U’, or ‘N’
‘days’ or ‘day’
‘hours’, ‘hour’, ‘hr’, or ‘h’
‘minutes’, ‘minute’, ‘min’, or ‘m’
‘seconds’, ‘second’, or ‘sec’
‘milliseconds’, ‘millisecond’, ‘millis’, or ‘milli’
‘microseconds’, ‘microsecond’, ‘micros’, or ‘micro’
‘nanoseconds’, ‘nanosecond’, ‘nanos’, ‘nano’, or ‘ns’

```
Out[33]: DatetimeIndex(['2021-01-01', '2021-01-02', '2021-01-03', '2021-01-04',
                        '2021-01-05', '2021-01-06', '2021-01-07', '2021-01-08',
                        '2021-01-09', '2021-01-10', '2021-01-11', '2021-01-12',
                        '2021-01-13', '2021-01-14', '2021-01-15', '2021-01-16',
                        '2021-01-17', '2021-01-18', '2021-01-19', '2021-01-20',
                        '2021-01-21', '2021-01-22', '2021-01-23', '2021-01-24',
                        '2021-01-25', '2021-01-26', '2021-01-27', '2021-01-28',
                        '2021-01-29', '2021-01-30', '2021-01-31', '2021-02-01',
                        '2021-02-02', '2021-02-03', '2021-02-04', '2021-02-05',
                        '2021-02-06', '2021-02-07', '2021-02-08', '2021-02-09',
                        '2021-02-10', '2021-02-11', '2021-02-12', '2021-02-13',
                        '2021-02-14', '2021-02-15', '2021-02-16', '2021-02-17',
                        '2021-02-18', '2021-02-19', '2021-02-20', '2021-02-21',
                        '2021-02-22', '2021-02-23', '2021-02-24', '2021-02-25',
                        '2021-02-26', '2021-02-27', '2021-02-28', '2021-03-01',
                        '2021-03-02', '2021-03-03', '2021-03-04', '2021-03-05',
                        '2021-03-06', '2021-03-07', '2021-03-08', '2021-03-09',
                        '2021-03-10', '2021-03-11', '2021-03-12', '2021-03-13',
                        '2021-03-14', '2021-03-15', '2021-03-16', '2021-03-17',
                        '2021-03-18', '2021-03-19', '2021-03-20', '2021-03-21',
                        '2021-03-22', '2021-03-23', '2021-03-24', '2021-03-25',
                        '2021-03-26', '2021-03-27', '2021-03-28', '2021-03-29',
                        '2021-03-30', '2021-03-31', '2021-04-01'],
                       dtype='datetime64[ns]', freq='D')
```

```
Out[43]: DatetimeIndex(['2021-01-01', '2021-01-04', '2021-01-05', '2021-01-06',
                        '2021-01-07', '2021-01-08', '2021-01-11', '2021-01-12',
                        '2021-01-13', '2021-01-14', '2021-01-15', '2021-01-18',
                        '2021-01-19', '2021-01-20', '2021-01-21', '2021-01-22',
                        '2021-01-25', '2021-01-26', '2021-01-27', '2021-01-28',
                        '2021-01-29', '2021-02-01', '2021-02-02', '2021-02-03',
                        '2021-02-04', '2021-02-05', '2021-02-08', '2021-02-09',
                        '2021-02-10', '2021-02-11', '2021-02-12', '2021-02-15',
                        '2021-02-16', '2021-02-17', '2021-02-18', '2021-02-19',
                        '2021-02-22', '2021-02-23', '2021-02-24', '2021-02-25',
                        '2021-02-26', '2021-03-01', '2021-03-02', '2021-03-03',
                        '2021-03-04', '2021-03-05', '2021-03-08', '2021-03-09',
                        '2021-03-10', '2021-03-11', '2021-03-12', '2021-03-15',
                        '2021-03-16', '2021-03-17', '2021-03-18', '2021-03-19',
                        '2021-03-22', '2021-03-23', '2021-03-24', '2021-03-25',
                        '2021-03-26', '2021-03-29', '2021-03-30', '2021-03-31',
                        '2021-04-01'],
                       dtype='datetime64[ns]', freq='B')
```

```
Out[44]: DatetimeIndex(['2021-04-02', '2021-04-05', '2021-04-06', '2021-04-07',
                        '2021-04-08', '2021-04-09', '2021-04-12', '2021-04-13',
                        '2021-04-14', '2021-04-15', '2021-04-16', '2021-04-19',
                        '2021-04-20', '2021-04-21', '2021-04-22', '2021-04-23',
                        '2021-04-26', '2021-04-27', '2021-04-28', '2021-04-29',
                        '2021-04-30', '2021-05-03', '2021-05-04', '2021-05-05',
                        '2021-05-06', '2021-05-07', '2021-05-10', '2021-05-11',
                        '2021-05-12', '2021-05-13', '2021-05-14', '2021-05-17',
                        '2021-05-18', '2021-05-19', '2021-05-20', '2021-05-21',
                        '2021-05-24', '2021-05-25', '2021-05-26', '2021-05-27',
                        '2021-05-28', '2021-05-31', '2021-06-01', '2021-06-02',
                        '2021-06-03', '2021-06-04', '2021-06-07', '2021-06-08',
                        '2021-06-09', '2021-06-10', '2021-06-11', '2021-06-14',
                        '2021-06-15', '2021-06-16', '2021-06-17', '2021-06-18',
                        '2021-06-21', '2021-06-22', '2021-06-23', '2021-06-24',
                        '2021-06-25', '2021-06-28', '2021-06-29', '2021-06-30',
                        '2021-07-01'],
                       dtype='datetime64[ns]', freq=None)
```

```
Out[52]: DatetimeIndex(['2021-04-30', '2021-05-31', '2021-06-30', '2021-07-31'],
         dtype='datetime64[ns]', freq=None)
```

```
Out[53]: DatetimeIndex(['2021-06-30', '2021-07-31', '2021-08-31', '2021-09-30'],
         dtype='datetime64[ns]', freq=None)
```

Out[2]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	539993	22386	JUMBO BAG PINK POLKADOT	10	1/4/2011 10:00	1.95	13313.0	United Kingdom
1	539993	21499	BLUE POLKADOT WRAP	25	1/4/2011 10:00	0.42	13313.0	United Kingdom
2	539993	21498	RED RETROSPOT WRAP	25	1/4/2011 10:00	0.42	13313.0	United Kingdom
3	539993	22379	RECYCLING BAG RETROSPOT	5	1/4/2011 10:00	2.10	13313.0	United Kingdom
4	539993	20718	RED RETROSPOT SHOPPER BAG	10	1/4/2011 10:00	1.25	13313.0	United Kingdom

```
start:  2011-01-04 10:00:00
end:  2011-06-30 20:08:00
```

Out[4]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	539993	22386	JUMBO BAG PINK POLKADOT	10	2011-01-31	1.95	13313.0	United Kingdom
1	539993	21499	BLUE POLKADOT WRAP	25	2011-01-31	0.42	13313.0	United Kingdom
2	539993	21498	RED RETROSPOT WRAP	25	2011-01-31	0.42	13313.0	United Kingdom
3	539993	22379	RECYCLING BAG RETROSPOT	5	2011-01-31	2.10	13313.0	United Kingdom
4	539993	20718	RED RETROSPOT SHOPPER BAG	10	2011-01-31	1.25	13313.0	United Kingdom

Out[5]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country	Revenue
0	539993	22386	JUMBO BAG PINK POLKADOT	10	2011-01-31	1.95	13313.0	United Kingdom	19.5
1	539993	21499	BLUE POLKADOT WRAP	25	2011-01-31	0.42	13313.0	United Kingdom	10.5
2	539993	21498	RED RETROSPOT WRAP	25	2011-01-31	0.42	13313.0	United Kingdom	10.5
3	539993	22379	RECYCLING BAG RETROSPOT	5	2011-01-31	2.10	13313.0	United Kingdom	10.5
4	539993	20718	RED RETROSPOT SHOPPER BAG	10	2011-01-31	1.25	13313.0	United Kingdom	12.5

Out[7]:

	Revenue
InvoiceDate	
2011-01-31	560000.260
2011-02-28	498062.650
2011-03-31	683267.080
2011-04-30	493207.121
2011-05-31	723333.510
2011-06-30	691123.120

Chapter 13: Exploring Time Series

freq string	meaning	freq string	meaning
B	business day frequency	Q	quarter end frequency
C	custom business day frequency	BQ	business quarter end frequency
D	calendar day frequency	QS	quarter start frequency
W	weekly frequency	BQS	business quarter start frequency
M	month end frequency	A, Y	year end frequency
SM	semi-month end frequency	BA, BY	business year end frequency
	(15th and end of month)	AS, YS	year start frequency
BM	business month end frequency	BAS, BYS	business year start frequency
CBM	custom business month	BH	business hour frequency
MS	month start frequency	H	hourly frequency
	end frequency	T, min	minutely frequency
SMS	semi-month start frequency	S	secondly frequency
	(1st and 15th)	L, ms	milliseconds
BMS	business month start frequency	U, us	microseconds
CBMS	custom business month	N	nanoseconds
	start frequency		

```
Out[90]:  Period('2019-01-01/2019-01-07', 'W-MON')
```

```
Out[91]:  Period('2019-01-08/2019-01-14', 'W-MON')
```

```
DatetimeIndex(['2020-01-05', '2020-01-12', '2020-01-19', '2020-01-26',
               '2020-02-02', '2020-02-09'],
              dtype='datetime64[ns]', freq='W-SUN')
DatetimeIndex(['2020-01-11', '2020-01-18', '2020-01-25', '2020-02-01',
               '2020-02-08', '2020-02-15'],
              dtype='datetime64[ns]', freq=None)
```

```
Out[141]:  Float64Index([                   0.0,                  1e-09,
                                        2e-09, 3.0000000000000004e-09,
                                        4e-09,                  5e-09,
                          6.000000000000001e-09,  7.000000000000001e-09,
                                        8e-09,  9.000000000000001e-09,
              ...
                          9.991000000000001e-06,              9.992e-06,
                                    9.993e-06,  9.994000000000001e-06,
                                    9.995e-06,              9.996e-06,
                          9.997000000000001e-06,              9.998e-06,
                                    9.999e-06,                  1e-05],
             dtype='float64', length=10001)
```

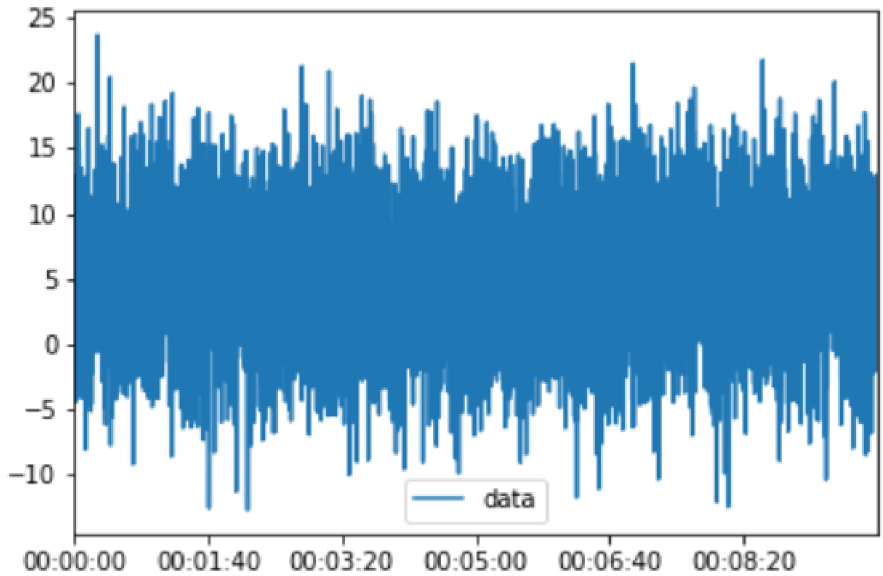
Out[125]:
```
TimedeltaIndex([       '0 days 00:00:00', '0 days 00:00:00.100000',
                '0 days 00:00:00.200000', '0 days 00:00:00.300000',
                '0 days 00:00:00.400000', '0 days 00:00:00.500000',
                '0 days 00:00:00.600000', '0 days 00:00:00.700000',
                '0 days 00:00:00.800000', '0 days 00:00:00.900000',
                ...
                       '0 days 00:09:59', '0 days 00:09:59.100000',
                '0 days 00:09:59.200000', '0 days 00:09:59.300000',
                '0 days 00:09:59.400000', '0 days 00:09:59.500000',
                '0 days 00:09:59.600000', '0 days 00:09:59.700000',
                '0 days 00:09:59.800000', '0 days 00:09:59.900000'],
               dtype='timedelta64[ns]', length=6000, freq=None)
```

Out[127]:

	data
0 days 00:00:00	10.692253
0 days 00:00:00.100000	11.023602
0 days 00:00:00.200000	-4.541415
0 days 00:00:00.300000	1.843395
0 days 00:00:00.400000	0.011138
...	...
0 days 00:09:59.500000	4.563188
0 days 00:09:59.600000	12.157702
0 days 00:09:59.700000	12.994389
0 days 00:09:59.800000	9.201197
0 days 00:09:59.900000	5.269805

6000 rows × 1 columns

Out[128]: <AxesSubplot:>

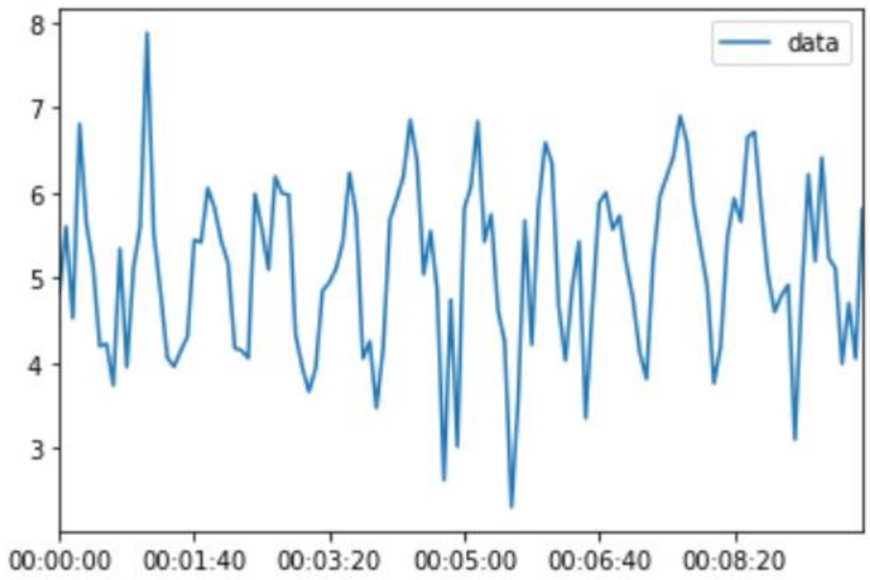


Out[192]:

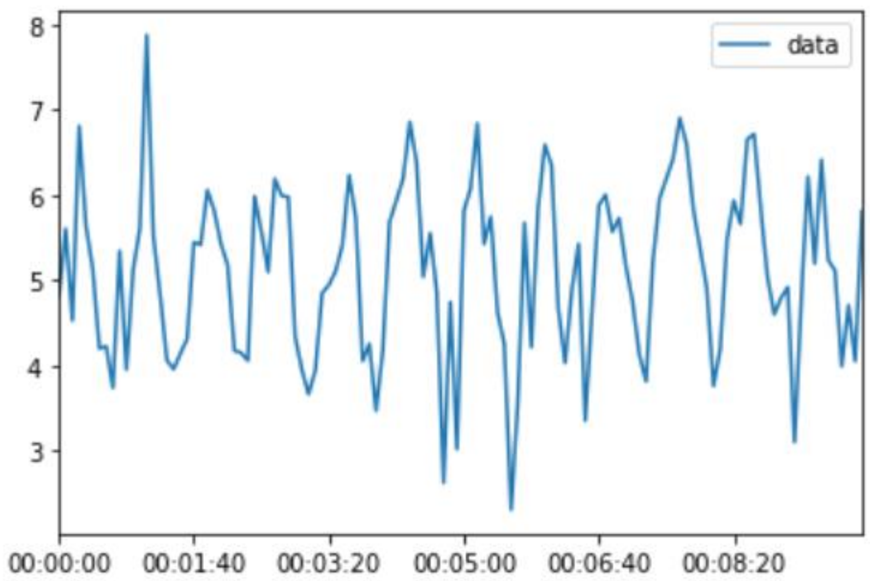
	data
0 days 00:00:00	4.595379
0 days 00:00:05	5.667126
0 days 00:00:10	6.863734
0 days 00:00:15	4.523539
0 days 00:00:20	4.402065
...	...
0 days 00:09:35	5.221181
0 days 00:09:40	3.371805
0 days 00:09:45	3.632050
0 days 00:09:50	5.272400
0 days 00:09:55	5.361699

120 rows × 1 columns

```
Out[130]: <AxesSubplot:>
```




```
Out[140]: <AxesSubplot:>
```



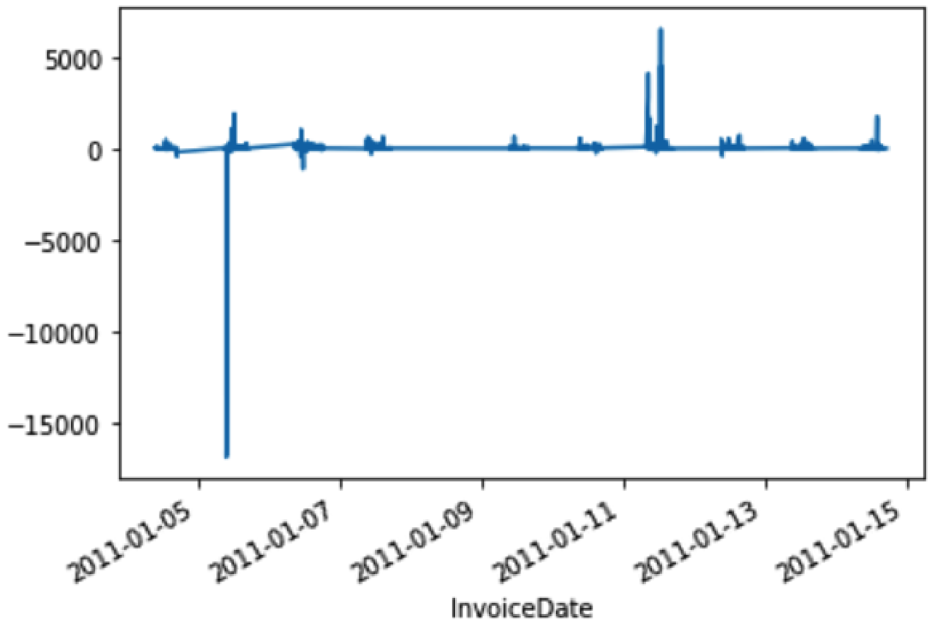
Out[2]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	539993	22386	JUMBO BAG PINK POLKADOT	10	1/4/2011 10:00	1.95	13313.0	United Kingdom
1	539993	21499	BLUE POLKADOT WRAP	25	1/4/2011 10:00	0.42	13313.0	United Kingdom
2	539993	21498	RED RETROSPOT WRAP	25	1/4/2011 10:00	0.42	13313.0	United Kingdom
3	539993	22379	RECYCLING BAG RETROSPOT	5	1/4/2011 10:00	2.10	13313.0	United Kingdom
4	539993	20718	RED RETROSPOT SHOPPER BAG	10	1/4/2011 10:00	1.25	13313.0	United Kingdom

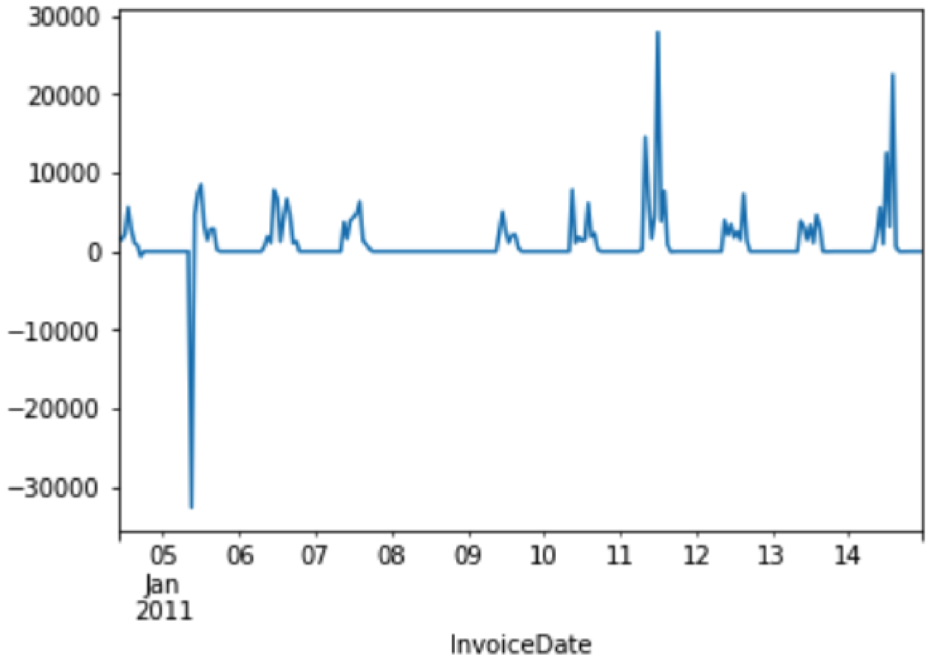
Out[27]:

	Invoice	StockCode	Description	Quantity	Price	Customer ID	Country
InvoiceDate							
2011-01-04 10:00:00	539993	22386	JUMBO BAG PINK POLKADOT	10	1.95	13313.0	United Kingdom
2011-01-04 10:00:00	539993	21499	BLUE POLKADOT WRAP	25	0.42	13313.0	United Kingdom
2011-01-04 10:00:00	539993	21498	RED RETROSPOT WRAP	25	0.42	13313.0	United Kingdom
2011-01-04 10:00:00	539993	22379	RECYCLING BAG RETROSPOT	5	2.10	13313.0	United Kingdom
2011-01-04 10:00:00	539993	20718	RED RETROSPOT SHOPPER BAG	10	1.25	13313.0	United Kingdom

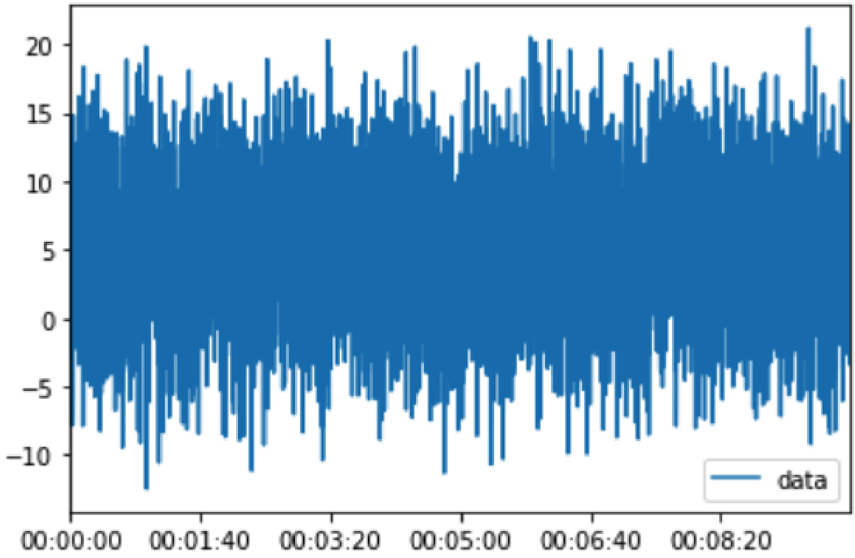
Out[17]: <AxesSubplot:xlabel='InvoiceDate'>



```
Out[29]: InvoiceDate
2011-01-04 10:00:00    1696.12
2011-01-04 11:00:00    1462.48
2011-01-04 12:00:00    2223.33
2011-01-04 13:00:00    5627.52
2011-01-04 14:00:00    2785.46
                         ...
2011-06-30 16:00:00    1321.58
2011-06-30 17:00:00    1539.94
2011-06-30 18:00:00    1144.65
2011-06-30 19:00:00     816.17
2011-06-30 20:00:00     203.86
Freq: H, Name: Revenue, Length: 4259, dtype: float64
Out[19]: <AxesSubplot:xlabel='InvoiceDate'>
```

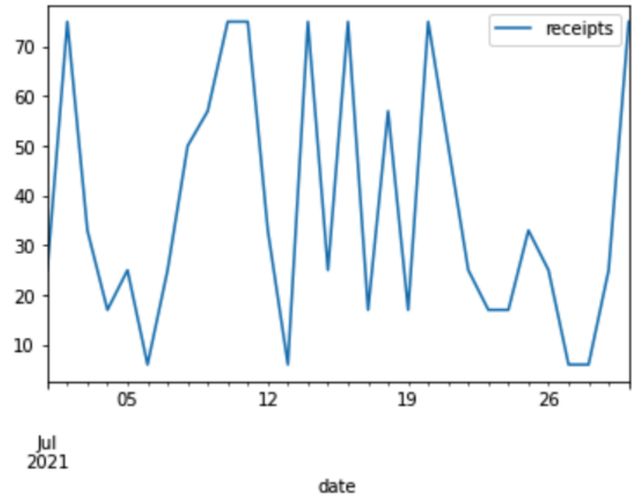


Out[198]: <matplotlib.axes._subplots.AxesSubplot at 0x1d4f91d5488>



date	receipts
2021-07-01	25
2021-07-02	75
2021-07-03	33
2021-07-04	17
2021-07-05	25
2021-07-06	6
2021-07-07	25
2021-07-08	50
2021-07-09	57
2021-07-10	75
2021-07-11	75
2021-07-12	33
2021-07-13	6
2021-07-14	75
2021-07-15	25
2021-07-16	75
2021-07-17	17
2021-07-18	57
2021-07-19	17
2021-07-20	75
2021-07-21	50
2021-07-22	25
2021-07-23	17
2021-07-24	17
2021-07-25	33
2021-07-26	25
2021-07-27	6
2021-07-28	6
2021-07-29	25
2021-07-30	75

Out[94]: <AxesSubplot:xlabel='date'>



7 days

First value = (25 + 75 + 33 + 17) / 4 = 37.5

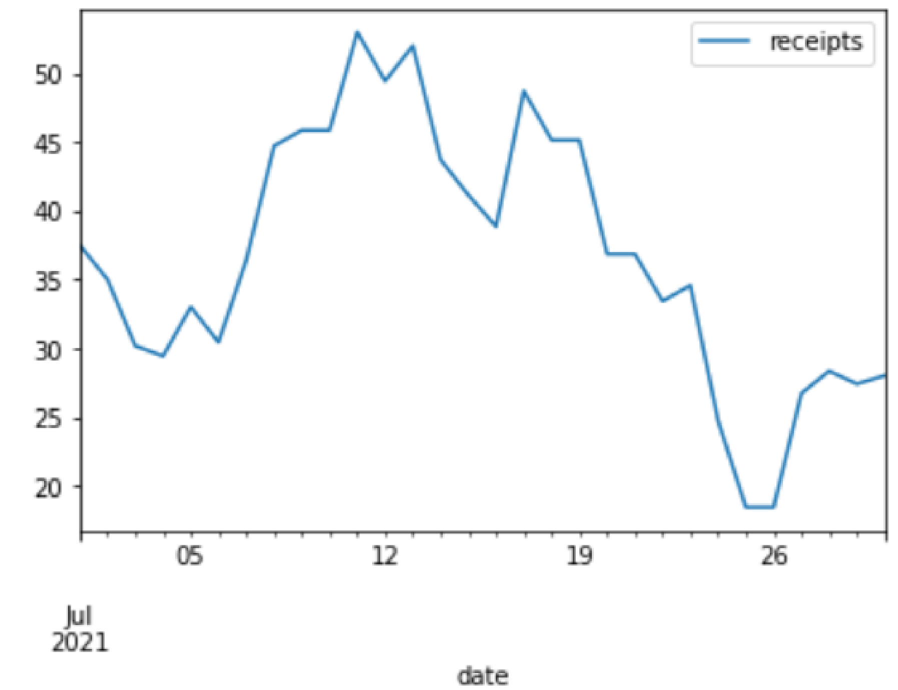
Second value = (25 + 75 + 33 + 17 + 25) / 5 = 35

Good week (7/12) = (57 + 75 + 75 + 33 + 6 + 75 + 25) / 7 = 49.6

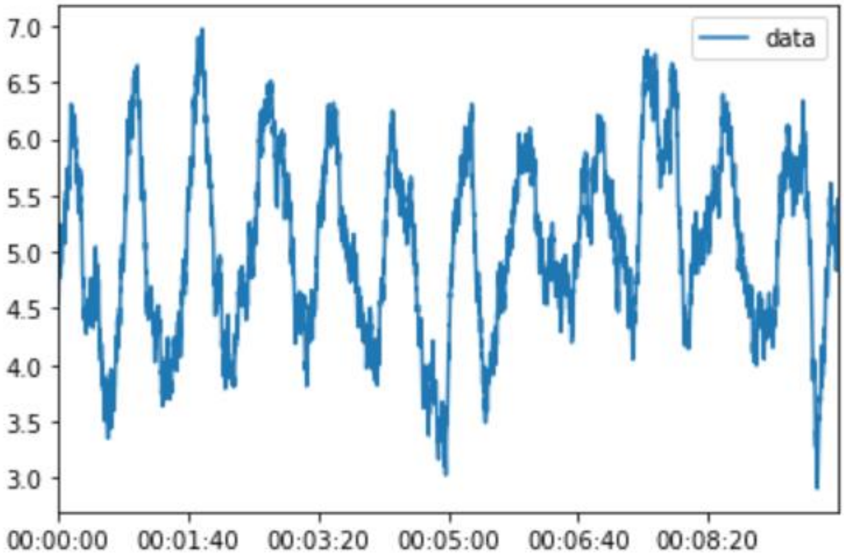
Bad week (7/25) = (25 + 17 + 17 + 33 + 25 + 6 + 6) / 7 = 18.4

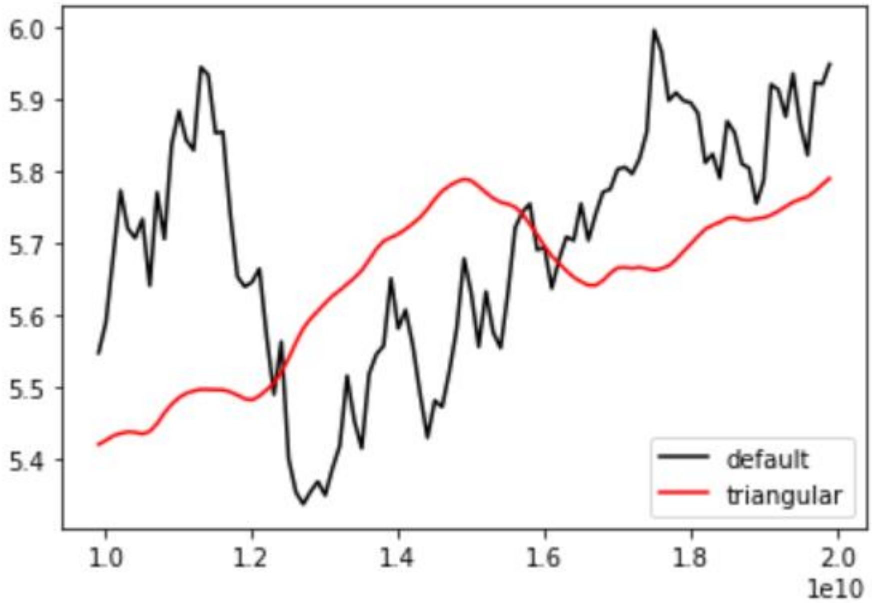
```
2021-07-01         25
2021-07-02         75
2021-07-03         33
2021-07-04         17
2021-07-05         25
2021-07-06          6
2021-07-07         25
2021-07-08         50
2021-07-09         57
2021-07-10         75
2021-07-11         75
2021-07-12         33
2021-07-13          6
2021-07-14         75
2021-07-15         25
2021-07-16         75
2021-07-17         17
2021-07-18         57
2021-07-19         17
2021-07-20         75
2021-07-21         50
2021-07-22         25
2021-07-23         17
2021-07-24         17
2021-07-25         33
2021-07-26         25
2021-07-27          6
2021-07-28          6
2021-07-29         25
2021-07-30         75
```

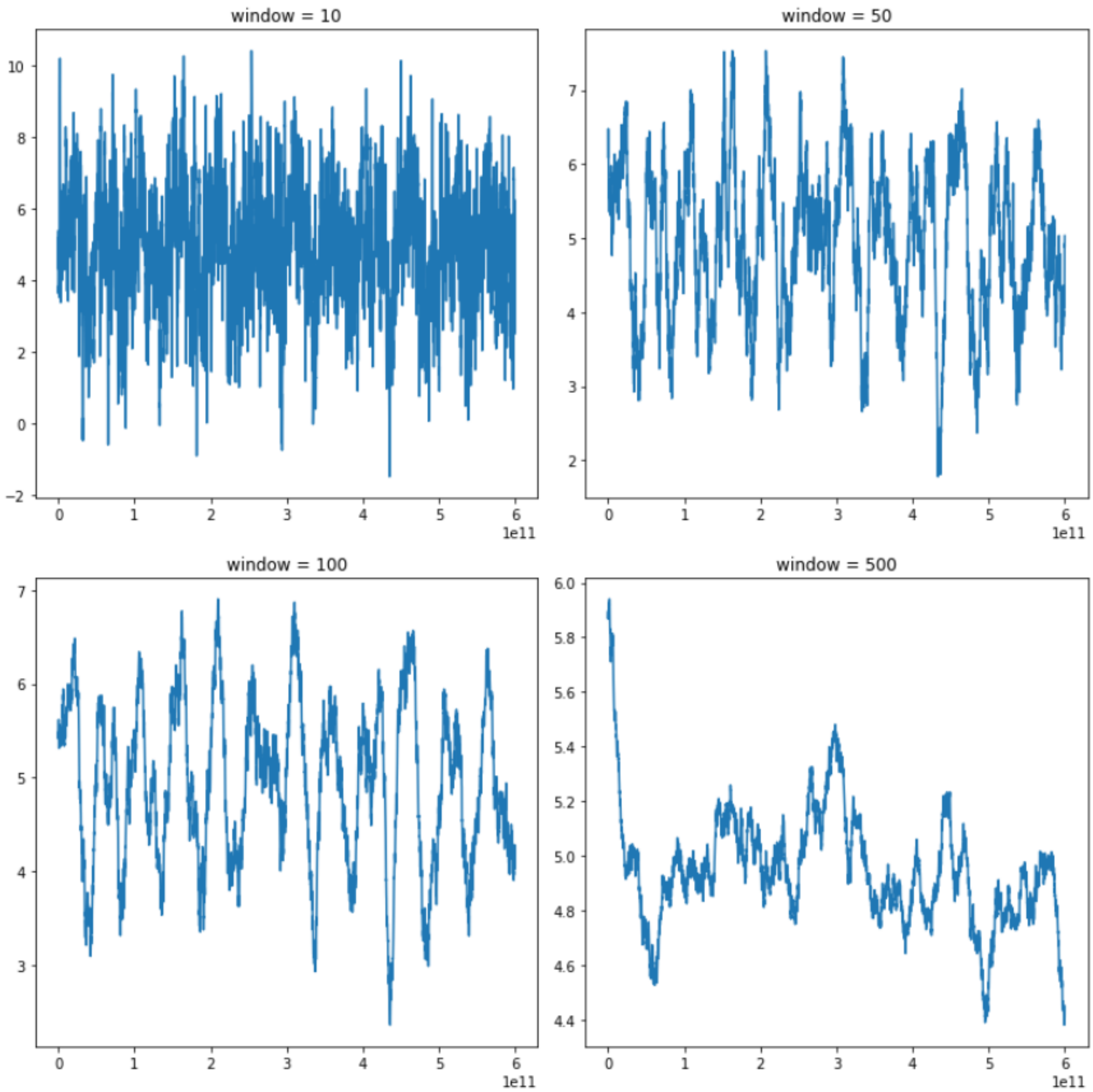
```
Out[95]: <AxesSubplot:xlabel='date'>
```



Out[204]: <matplotlib.axes._subplots.AxesSubplot at 0x1d4fa69cd88>







Out[91]:

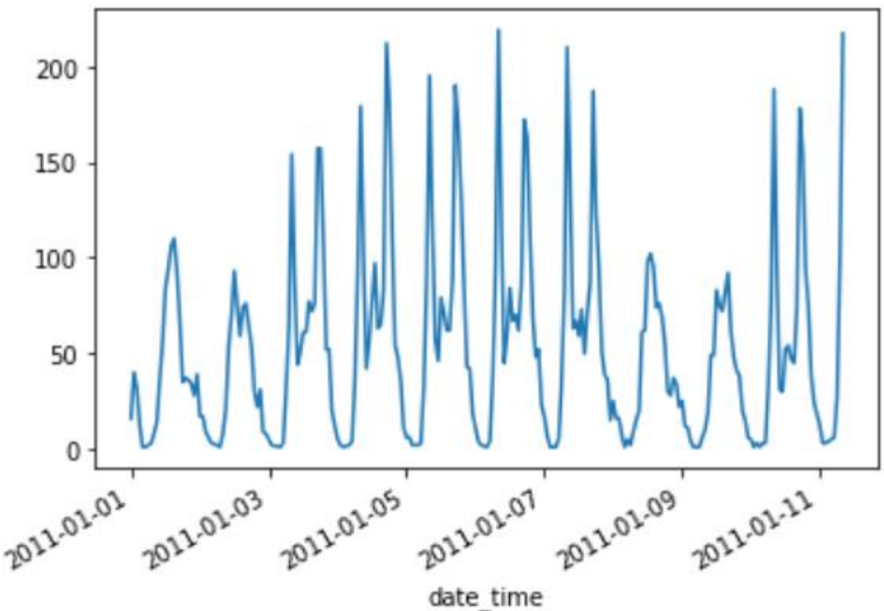
	date	hour	rentals
0	1/1/2011	0	16
1	1/1/2011	1	40
2	1/1/2011	2	32
3	1/1/2011	3	13
4	1/1/2011	4	1

Out[137]:

date_time	date	hour	rentals	date_time
2011-01-01 00:00:00	1/1/2011	0	16	1/1/2011 00:00:00
2011-01-01 01:00:00	1/1/2011	1	40	1/1/2011 01:00:00
2011-01-01 02:00:00	1/1/2011	2	32	1/1/2011 02:00:00
2011-01-01 03:00:00	1/1/2011	3	13	1/1/2011 03:00:00
2011-01-01 04:00:00	1/1/2011	4	1	1/1/2011 04:00:00
...	...	...	...	...
2012-12-31 19:00:00	12/31/2012	19	119	12/31/2012 19:00:00
2012-12-31 20:00:00	12/31/2012	20	89	12/31/2012 20:00:00
2012-12-31 21:00:00	12/31/2012	21	90	12/31/2012 21:00:00
2012-12-31 22:00:00	12/31/2012	22	61	12/31/2012 22:00:00
2012-12-31 23:00:00	12/31/2012	23	49	12/31/2012 23:00:00

17379 rows × 4 columns

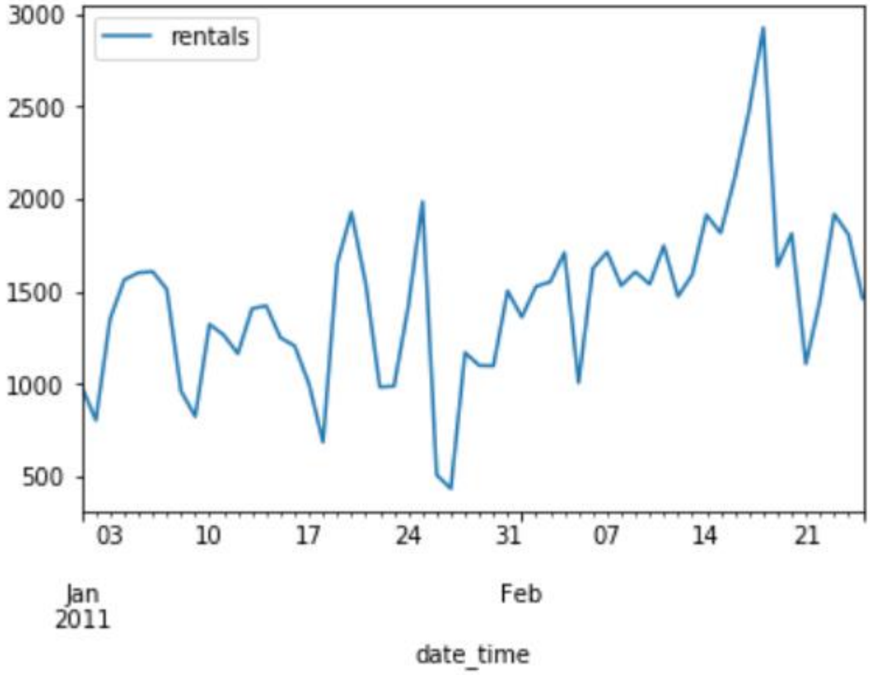
Out[94]: <matplotlib.axes._subplots.AxesSubplot at 0x25969241208>

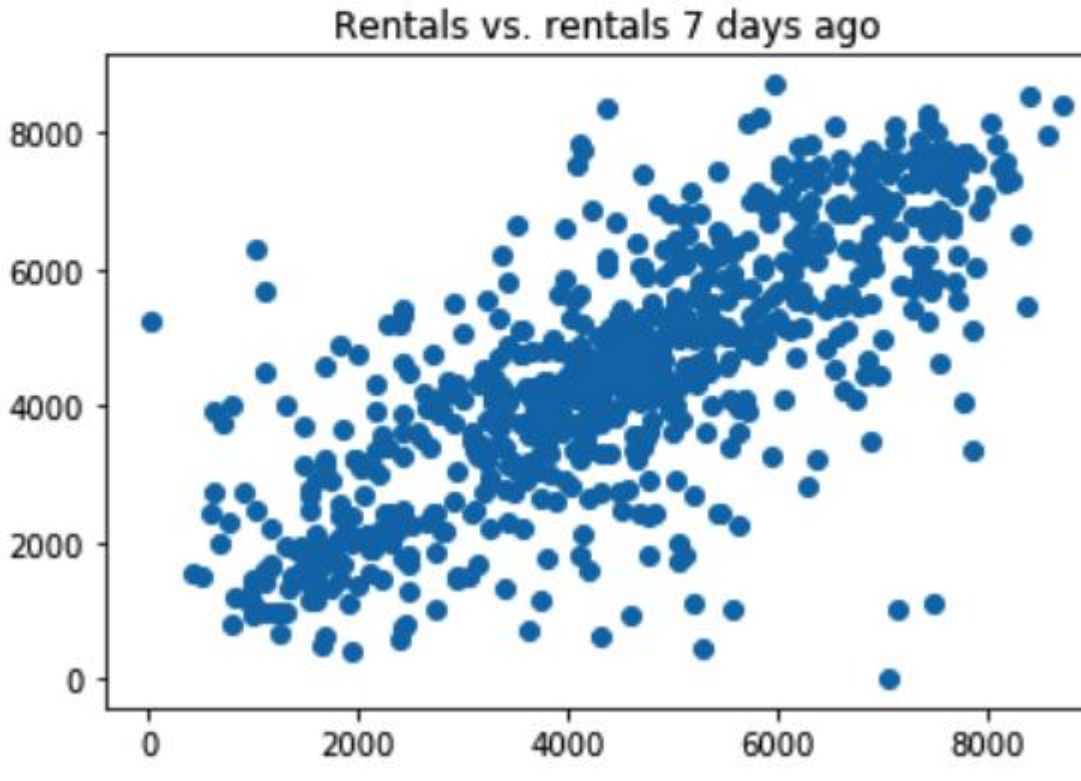


Out[95]:

	rentals
date_time	
2011-01-01	985
2011-01-02	801
2011-01-03	1349
2011-01-04	1562
2011-01-05	1600
2011-01-06	1606
2011-01-07	1510
2011-01-08	959
2011-01-09	822
2011-01-10	1321
2011-01-11	1263
2011-01-12	1162
2011-01-13	1406
2011-01-14	1421

Out[86]: <matplotlib.axes._subplots.AxesSubplot at 0x25968f5dac8>



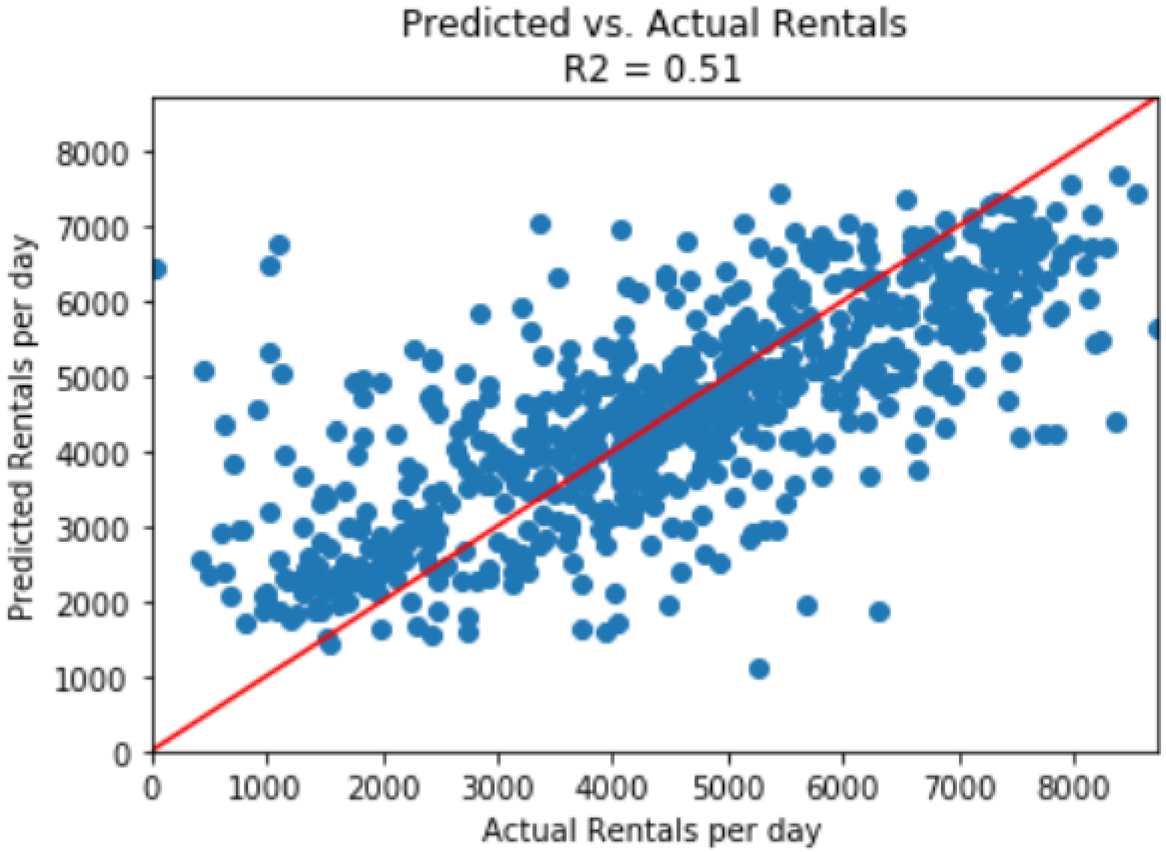


Out[109]:

date_time	rentals	lagged_rentals
2011-01-01	985	NaN
2011-01-02	801	NaN
2011-01-03	1349	NaN
2011-01-04	1562	NaN
2011-01-05	1600	NaN
...	...	...
2012-12-27	2114	4128.0
2012-12-28	3095	3623.0
2012-12-29	1341	1749.0
2012-12-30	1796	1787.0
2012-12-31	2729	920.0

731 rows × 2 columns

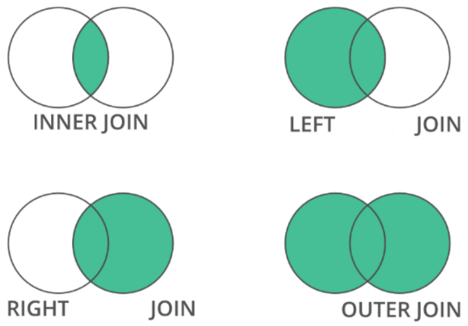
```
R2 is  0.5145071365683822  using:
            rentals  lagged_rentals
date_time
2011-01-08      959           985.0
2011-01-09      822           801.0
2011-01-10     1321          1349.0
2011-01-11     1263          1562.0
2011-01-12     1162          1600.0
```


Predicted vs. Actual Rentals
R2 = 0.51
Predicted Rentals per day
Actual Rentals per day
0
1000
2000
3000
4000
5000
6000
7000
8000

Chapter 14: Applying pandas Data Processing for Case Studies

	Jahr	Jan	Feb	Mrz	Apr	Mai	Jun	Jul	Aug	Sep	Okt	Nov	Dez
0	1951	48.0	49.0	98.0	61.0	23.0	34.0	44.0	146.0	64.0	89.0	47.0	72.0

	Jahr	variable	value
0	1951	Jan	48.0
1	1952	Jan	100.0
2	1953	Jan	100.0
3	1954	Jan	31.0
4	1955	Jan	21.0



	Jahr	Jan	Feb	Mrz	Apr	Mai	Jun	Jul	Aug	Sep	Okt	Nov	Dez
0	1951	48.0	49.0	98.0	61.0	23.0	34.0	44.0	146.0	64.0	89.0	47.0	72.0
1	1952	100.0	24.0	28.0	23.0	16.0	35.0	25.0	19.0	59.0	105.0	56.0	80.0
2	1953	100.0	102.0	50.0	86.0	15.0	16.0	2.0	31.0	113.0	91.0	124.0	127.0
3	1954	31.0	58.0	39.0	50.0	20.0	26.0	65.0	34.0	53.0	90.0	135.0	40.0
4	1955	21.0	10.0	70.0	75.0	3.0	2.0	53.0	105.0	126.0	30.0	103.0	74.0

	Jahr	Jan	Feb	Mrz	Apr	Mai	Jun	Jul	Aug	Sep	Okt	Nov	Dez
0	1968	26.4	26.4	28.1	28.1	29.8	29.8	29.8	29.8	29.8	29.8	28.1	28.1
1	1969	26.4	26.4	28.1	29.8	29.8	29.8	29.8	NaN	29.8	29.8	29.8	29.8
2	1970	28.1	28.1	28.1	29.9	29.8	29.9	28.1	29.8	29.8	29.8	29.8	NaN
3	1971	26.4	26.1	26.4	NaN	29.8	28.1	29.8	28.1	NaN	NaN	28.0	NaN
4	1972	NaN	NaN	NaN	NaN	NaN	NaN	NaN	28.1	NaN	NaN	NaN	28.0

	Jahr	Jan	Feb	Mrz	Apr	Mai	Jun	Jul	Aug	Sep	Okt	Nov	Dez
0	1978	NaN	NaN	257.0	NaN	NaN	NaN	NaN	170.0	NaN	NaN	209.0	245.0
1	1979	253.0	NaN	NaN	228.0	207.0	NaN	169.0	NaN	NaN	NaN	NaN	NaN
2	1980	NaN	230.0	249.0	232.0	NaN	NaN	213.0	195.0	195.0	195.0	NaN	197.0
3	1981	NaN	212.0	244.0	NaN	206.0	NaN	193.0	NaN	168.0	148.0	234.0	261.0
4	1982	199.0	174.0	NaN	NaN	212.0	153.0	214.0	NaN	180.0	197.0	271.0	268.0

	Jahr	variable	value
0	1951	Jan	48.0
1	1952	Jan	100.0
2	1953	Jan	100.0
3	1954	Jan	31.0
4	1955	Jan	21.0

	Jahr	variable	Precipitation
0	1951	Jan	48.0
1	1952	Jan	100.0
2	1953	Jan	100.0
3	1954	Jan	31.0
4	1955	Jan	21.0

	Jahr	variable	Vapour_Pressure
0	1968	Jan	26.4
1	1969	Jan	26.4
2	1970	Jan	28.1
3	1971	Jan	26.4
4	1972	Jan	NaN

	Jahr	variable	Sun_shine
0	1978	Jan	NaN
1	1979	Jan	253.0
2	1980	Jan	NaN
3	1981	Jan	NaN
4	1982	Jan	199.0

	Jahr	variable	Precipitation	Vapour_Pressure
0	1968	Jan	49.0	26.4
1	1969	Jan	19.0	26.4
2	1970	Jan	19.0	28.1
3	1971	Jan	51.0	26.4
4	1972	Jan	50.0	NaN

	Jahr	variable	Precipitation	Vapour_Pressure	Sun_shine
0	1978	Jan	62.0	28.1	NaN
1	1979	Jan	61.0	28.1	253.0
2	1980	Jan	60.0	NaN	NaN
3	1981	Jan	78.0	28.1	NaN
4	1982	Jan	59.0	28.0	199.0

	Jahr	variable	Precipitation	Vapour_Pressure	Sun_shine
0	1978	Jan	62.0	28.1	NaN
1	1979	Jan	61.0	28.1	253.0
2	1980	Jan	60.0	NaN	NaN
3	1981	Jan	78.0	28.1	NaN
4	1982	Jan	59.0	28.0	199.0
...	...	...	...	...	...
415	2012	Dec	42.0	29.8	220.0
416	2013	Dec	51.0	30.6	217.0
417	2014	Dec	70.0	30.2	234.0
418	2015	Dec	40.0	29.9	274.0
419	2016	Dec	63.0	29.9	229.0

	Year	variable	Precipitation	Vapour_Pressure	Sun_shine
0	1978	Jan	62.0	28.1	NaN
1	1979	Jan	61.0	28.1	253.0
2	1980	Jan	60.0	NaN	NaN
3	1981	Jan	78.0	28.1	NaN
4	1982	Jan	59.0	28.0	199.0

	Year	variable	Precipitation	Vapour_Pressure	Sun_shine	months
0	1978	Jan	62.0	28.1	NaN	Jan
1	1979	Jan	61.0	28.1	253.0	Jan
2	1980	Jan	60.0	NaN	NaN	Jan
3	1981	Jan	78.0	28.1	NaN	Jan
4	1982	Jan	59.0	28.0	199.0	Jan

	Year	variable	Precipitation	Vapour_Pressure	Sun_shine	months
0	1978	Jan	62.0	28.1	NaN	1
1	1979	Jan	61.0	28.1	253.0	1
2	1980	Jan	60.0	NaN	NaN	1
3	1981	Jan	78.0	28.1	NaN	1
4	1982	Jan	59.0	28.0	199.0	1
...	...	...	...	...	...	...
391	2010	Dec	70.0	NaN	219.0	12
392	2011	Dec	58.0	31.1	188.0	12
393	2012	Dec	42.0	29.8	220.0	12
394	2013	Dec	51.0	30.6	217.0	12
395	2014	Dec	70.0	30.2	234.0	12

	Year	variable	Precipitation	Vapour_Pressure	Sun_shine	months
0	1978	Jan	62.0	28.1	NaN	1
33	1978	Feb	25.0	NaN	NaN	2
66	1978	Mar	45.0	28.1	257.0	3
99	1978	Apr	32.0	26.4	NaN	4
132	1978	May	77.0	NaN	NaN	5
...	...	...	...	...	...	...
263	2014	Aug	36.0	31.4	216.0	8
296	2014	Sep	66.0	31.0	188.0	9
329	2014	Oct	82.0	31.7	186.0	10
362	2014	Nov	46.0	31.4	235.0	11
395	2014	Dec	70.0	30.2	234.0	12

	Year	variable	Precipitation	Vapour_Pressure	Sun_shine	months
0	1978	Jan	62.0	28.100000	257.0	1
33	1978	Feb	25.0	28.100000	257.0	2
66	1978	Mar	45.0	28.100000	257.0	3
99	1978	Apr	32.0	26.400000	239.6	4
132	1978	May	77.0	26.683333	222.2	5
...	...	...	...	...	...	...
263	2014	Aug	36.0	31.400000	216.0	8
296	2014	Sep	66.0	31.000000	188.0	9
329	2014	Oct	82.0	31.700000	186.0	10
362	2014	Nov	46.0	31.400000	235.0	11
395	2014	Dec	70.0	30.200000	234.0	12

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
0      32561 non-null int64
1      32561 non-null object
2      32561 non-null int64
3      32561 non-null object
4      32561 non-null int64
5      32561 non-null object
6      32561 non-null object
7      32561 non-null object
8      32561 non-null object
9      32561 non-null object
10     32561 non-null int64
11     32561 non-null int64
12     32561 non-null int64
13     32561 non-null object
14     32561 non-null object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

	age	workclass	education	marital-status	occupation	relationship	capital-gain	capital-loss	hours-per-week	native-country	earning
0	39	State-gov	Bachelors	Never-married	Adm-clerical	Not-in-family	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	Bachelors	Married-civ-spouse	Exec-managerial	Husband	0	0	13	United-States	<=50K
2	38	Private	HS-grad	Divorced	Handlers-cleaners	Not-in-family	0	0	40	United-States	<=50K
3	53	Private	11th	Married-civ-spouse	Handlers-cleaners	Husband	0	0	40	United-States	<=50K
4	28	Private	Bachelors	Married-civ-spouse	Prof-specialty	Wife	0	0	40	Cuba	<=50K

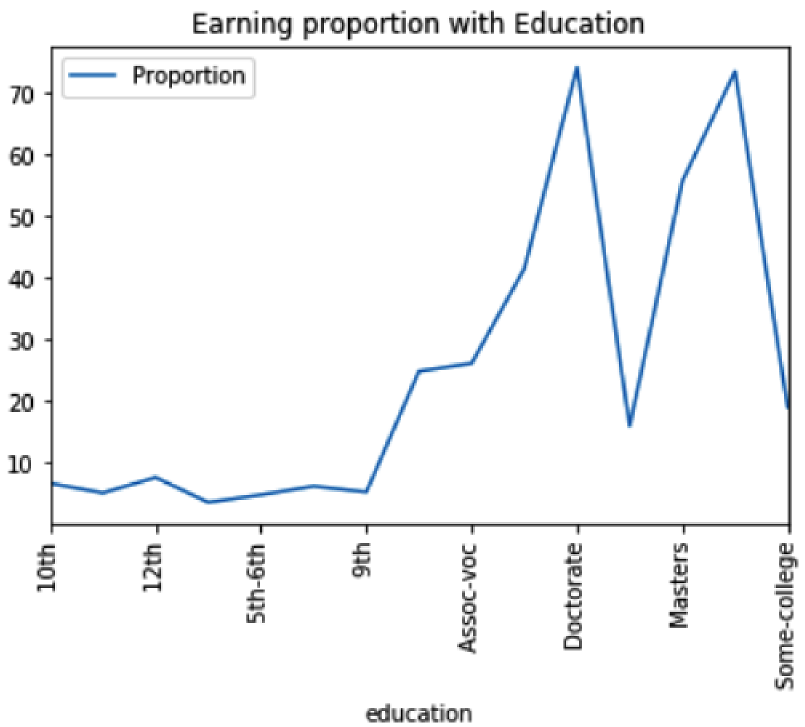
```
education
 10th              933
 11th             1175
 12th              433
 1st-4th           168
 5th-6th           333
 7th-8th           646
 9th               514
 Assoc-acdm       1067
 Assoc-voc        1382
 Bachelors        5355
 Doctorate         413
 HS-grad         10501
 Masters          1723
 Preschool          51
 Prof-school       576
 Some-college     7291
Name: earning, dtype: int64
```

```
education
 10th              62
 11th              60
 12th              33
 1st-4th            6
 5th-6th           16
 7th-8th           40
 9th               27
 Assoc-acdm       265
 Assoc-voc        361
 Bachelors       2221
 Doctorate        306
 HS-grad         1675
 Masters          959
 Prof-school      423
 Some-college    1387
Name: earning, dtype: int64
```

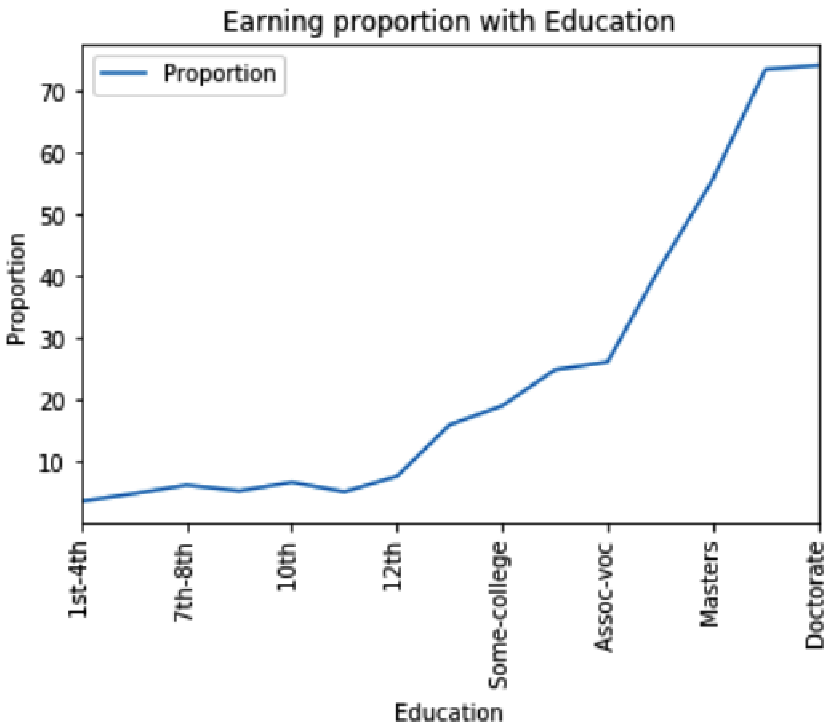
	earning
education	
10th	6.645230
11th	5.106383
12th	7.621247
1st-4th	3.571429
5th-6th	4.804805

	Proportion
education	
10th	6.645230
11th	5.106383
12th	7.621247
1st-4th	3.571429
5th-6th	4.804805

	Proportion
education	
10th	6.65
11th	5.11
12th	7.62
1st-4th	3.57
5th-6th	4.80



	Proportion
education	
1st-4th	3.57
5th-6th	4.80
7th-8th	6.19
9th	5.25
10th	6.65



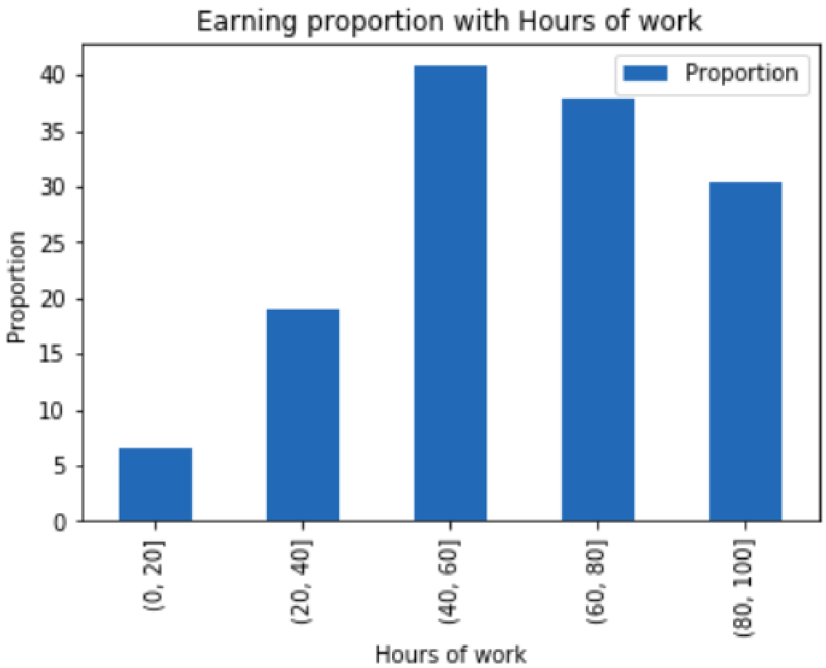

```
count    32561.000000
mean        40.437456
std         12.347429
min          1.000000
25%         40.000000
50%         40.000000
75%         45.000000
max         99.000000
Name: hours-per-week, dtype: float64
```

	age	workclass	education	marital-status	occupation	relationship	capital-gain	capital-loss	hours-per-week	native-country	earning	cut_hours
0	39	State-gov	Bachelors	Never-married	Adm-clerical	Not-in-family	2174	0	40	United-States	<=50K	(20, 40]
1	50	Self-emp-not-inc	Bachelors	Married-civ-spouse	Exec-managerial	Husband	0	0	13	United-States	<=50K	(0, 20]
2	38	Private	HS-grad	Divorced	Handlers-cleaners	Not-in-family	0	0	40	United-States	<=50K	(20, 40]
3	53	Private	11th	Married-civ-spouse	Handlers-cleaners	Husband	0	0	40	United-States	<=50K	(20, 40]
4	28	Private	Bachelors	Married-civ-spouse	Prof-specialty	Wife	0	0	40	Cuba	<=50K	(20, 40]

```
cut_hours
(0, 20]        2928
(20, 40]      20052
(40, 60]       8471
(60, 80]        902
(80, 100]       208
Name: earning, dtype: int64
```

cut_hours	earning
(0, 20]	6.659836
(20, 40]	18.900858
(40, 60]	40.750797
(60, 80]	37.804878
(80, 100]	30.288462

	Proportion
cut_hours	
(0, 20]	6.66
(20, 40]	18.90
(40, 60]	40.75
(60, 80]	37.80
(80, 100]	30.29



	id	id_android	speed	time	distance	rating	rating_bus	rating_weather	car_or_bus	linha
0	1	0	19.210586	0.138049	2.652	3	0	0	1	NaN
1	2	0	30.848229	0.171485	5.290	3	0	0	1	NaN
2	3	1	13.560101	0.067699	0.918	3	0	0	2	NaN
3	4	1	19.766679	0.389544	7.700	3	0	0	2	NaN
4	8	0	25.807401	0.154801	3.995	2	0	0	1	NaN

	id	latitude	longitude	track_id	time
0	1	-10.939341	-37.062742	1	2014-09-13 07:24:32
1	2	-10.939341	-37.062742	1	2014-09-13 07:24:37
2	3	-10.939324	-37.062765	1	2014-09-13 07:24:42
3	4	-10.939211	-37.062843	1	2014-09-13 07:24:47
4	5	-10.938939	-37.062879	1	2014-09-13 07:24:53

	id_x	id_android	speed	time_x	distance	rating	rating_bus	rating_weather	car_or_bus	linha	id_y	latitude	longitude	track_id	time_y
0	1	0	19.210586	0.138049	2.652	3	0	0	1	NaN	1	-10.939341	-37.062742	1	2014-09-13 07:24:32
1	1	0	19.210586	0.138049	2.652	3	0	0	1	NaN	2	-10.939341	-37.062742	1	2014-09-13 07:24:37
2	1	0	19.210586	0.138049	2.652	3	0	0	1	NaN	3	-10.939324	-37.062765	1	2014-09-13 07:24:42
3	1	0	19.210586	0.138049	2.652	3	0	0	1	NaN	4	-10.939211	-37.062843	1	2014-09-13 07:24:47
4	1	0	19.210586	0.138049	2.652	3	0	0	1	NaN	5	-10.938939	-37.062879	1	2014-09-13 07:24:53

	id_x	id_android	speed	time_x	distance	rating	rating_bus	rating_weather	car_or_bus	linha	id_y	latitude	longitude	track_id	time_y
0	1	0	19.210586	0.138049	2.652	3	0	0	1	NaN	1	-10.939341	-37.062742	1	2014-09-13 07:24:32
90	2	0	30.848229	0.171485	5.290	3	0	0	1	NaN	91	-10.939439	-37.062428	2	2014-09-13 13:37:54
203	3	1	13.560101	0.067699	0.918	3	0	0	2	NaN	204	-10.903162	-37.048294	3	2014-09-17 05:09:23
226	4	1	19.766679	0.389544	7.700	3	0	0	2	NaN	227	-10.908893	-37.052372	4	2014-09-17 05:09:23
355	8	0	25.807401	0.154801	3.995	2	0	0	1	NaN	564	-10.943777	-37.052344	8	2014-09-26 15:26:53

l_x	id_android	speed	time_x	distance	rating	rating_bus	rating_weather	car_or_bus	linha	id_y	latitude	longitude	track_id	time_y	Suburb
1	0	19.210586	0.138049	2.652	3	0	0	1	NaN	1	-10.939341	-37.062742	1	2014-09-13 07:24:32	Grageru
2	0	30.848229	0.171485	5.290	3	0	0	1	NaN	91	-10.939439	-37.062428	2	2014-09-13 13:37:54	Grageru
3	1	13.560101	0.067699	0.918	3	0	0	2	NaN	204	-10.903162	-37.048294	3	2014-09-17 05:09:23	Industrial
4	1	19.766679	0.389544	7.700	3	0	0	2	NaN	227	-10.908893	-37.052372	4	2014-09-17 05:09:23	Centro
8	0	25.807401	0.154801	3.995	2	0	0	1	NaN	564	-10.943777	-37.052344	8	2014-09-26 15:26:53	Jardins

rating_bus	rating_weather	car_or_bus	linha	...	latitude	longitude	track_id	time_y	Suburb	Parse_date	Weekday	Day	Month	StartHour
0	0	1	NaN	...	-10.939341	-37.062742	1	2014-09-13 07:24:32	Grageru	2014-09-13 07:24:32	5	Saturday	September	07
0	0	1	NaN	...	-10.939439	-37.062428	2	2014-09-13 13:37:54	Grageru	2014-09-13 13:37:54	5	Saturday	September	13
0	0	2	NaN	...	-10.903162	-37.048294	3	2014-09-17 05:09:23	Industrial	2014-09-17 05:09:23	2	Wednesday	September	05
0	0	2	NaN	...	-10.908893	-37.052372	4	2014-09-17 05:09:23	Centro	2014-09-17 05:09:23	2	Wednesday	September	05
0	0	1	NaN	...	-10.943777	-37.052344	8	2014-09-26 15:26:53	Jardins	2014-09-26 15:26:53	4	Friday	September	15

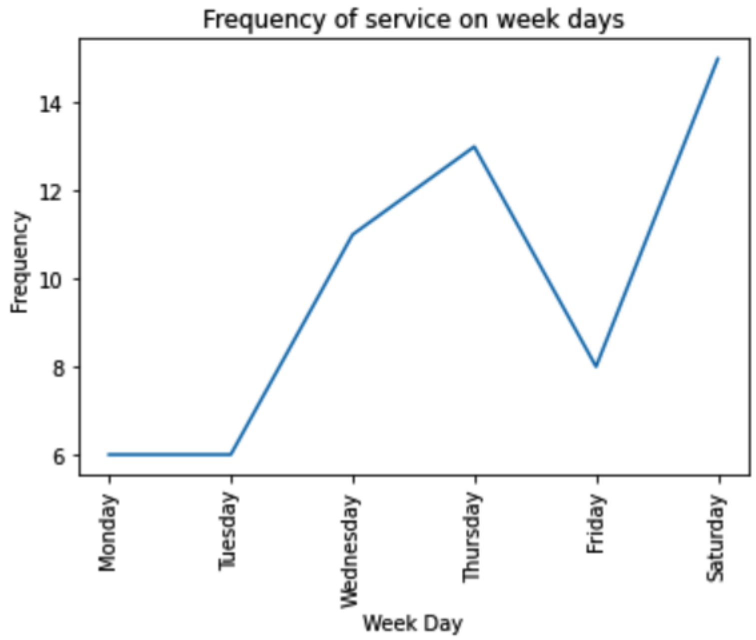
```
Suburb
Industrial       49
São José         12
Coroa do Meio    11
Jabutiana         9
Centro            9
Name: Suburb, dtype: int64
```

rating_bus	rating_weather	car_or_bus	linha	...	longitude	track_id	time_y	Suburb	Parse_date	Weekday	Day	Month	StartHour	cut_hours
0	0	1	NaN	...	-37.062742	1	2014-09-13 07:24:32	Grageru	2014-09-13 07:24:32	5	Saturday	September	7	(6, 10]
0	0	1	NaN	...	-37.062428	2	2014-09-13 13:37:54	Grageru	2014-09-13 13:37:54	5	Saturday	September	13	(10, 15]
0	0	2	NaN	...	-37.048294	3	2014-09-17 05:09:23	Industrial	2014-09-17 05:09:23	2	Wednesday	September	5	(0, 6]
0	0	2	NaN	...	-37.052372	4	2014-09-17 05:09:23	Centro	2014-09-17 05:09:23	2	Wednesday	September	5	(0, 6]
0	0	1	NaN	...	-37.052344	8	2014-09-26 15:26:53	Jardins	2014-09-26 15:26:53	4	Friday	September	15	(10, 15]

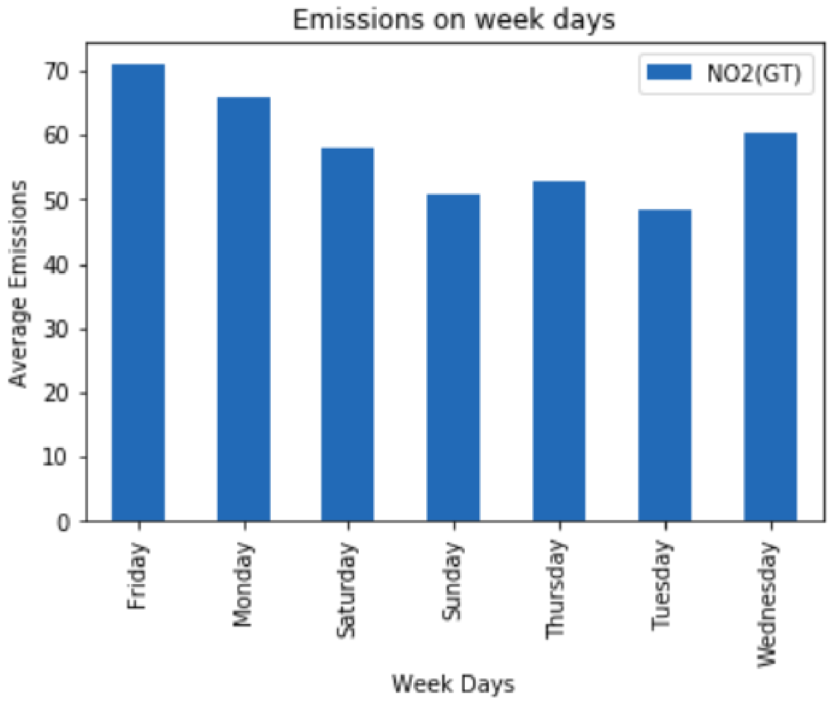
```
cut_hours
(0, 6]       22
(6, 10]      59
(10, 15]     49
(15, 20]     28
(20, 23]      4
```

```
Day
Friday        8
Monday        6
Saturday     15
Thursday     13
Tuesday       6
Wednesday    11
```

```
Day
Monday        6
Tuesday       6
Wednesday    11
Thursday     13
Friday        8
Saturday     15
```



Frequency of service on week days
Frequency
14
12
10
8
6
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Week Day



Emissions on week days
NO2(GT)
Average Emissions
70
60
50
40
30
20
10
0
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
Week Days

Chapter 15: Appendix

	Months	Grocery_sales	Stationary_sales
0	Jan	16	57
1	Jan	44	139
2	Jan	15	85
3	Jan	59	8
4	Jan	36	106

	Months	Grocery_sales	Stationary_sales
0	Jan	36	84
1	Jan	51	63
2	Jan	17	71
3	Jan	48	65
4	Jan	57	66

	Months	Grocery_sales	Stationary_sales
1	Jan	44	139
2	Jan	15	85
4	Jan	36	106
5	Jan	27	136
6	Jan	74	116
7	Jan	63	142
8	Jan	65	129
9	Jan	12	138
10	Feb	34	112
11	Feb	73	100
12	Feb	45	135
13	Feb	31	13

Out[2]:

	date	GDP
0	2017-03-31	19190.4
1	2017-06-30	19356.6
2	2017-09-30	19611.7
3	2017-12-31	19918.9
4	2018-03-31	20163.2

Out[4]:

	GDP
date	
2017-03-31	19190.4
2017-06-30	19356.6
2017-09-30	19611.7
2017-12-31	19918.9
2018-03-31	20163.2

Out[3]:

```
date       object
GDP       float64
dtype: object
```

Out[3]:

	name
0	Customers
1	Orders

Out[3]:

	index	Customer_Number	date	item	qty	price	amount
0	0	25058	10/19/2020	354161666	62	91.50	5673.14
1	1	25058	11/10/2020	1129038342	38	79.79	3032.14
2	2	26069	11/23/2020	421919566	40	55.67	2226.76
3	3	26069	12/22/2020	1156861472	54	80.30	4336.03
4	4	26858	11/30/2020	936049686	64	45.37	2903.99
5	5	26858	12/9/2020	458515506	54	15.55	839.51
6	6	26858	11/6/2020	937462037	83	44.92	3728.20

Out[8]:

	index	Customer_Number	Company	City	State
0	5	35549	Certain Construction	Honolulu	HI

	buying	maint	doors	persons	lug_boot	safety	class
0	vhigh	vhigh	2	2.0	small	low	unacc
1	vhigh	vhigh	2	2.0	small	med	unacc
2	vhigh	vhigh	2	NaN	small	high	unacc
3	vhigh	vhigh	2	2.0	med	low	unacc
4	vhigh	vhigh	2	2.0	med	med	unacc
5	NaN	vhigh	2	2.0	med	high	NaN
6	vhigh	vhigh	2	2.0	big	low	unacc
7	vhigh	vhigh	2	2.0	big	NaN	unacc
8	vhigh	vhigh	2	2.0	big	high	unacc
9	vhigh	NaN	2	4.0	small	low	unacc

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1728 entries, 0 to 1727
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   buying    1727 non-null   object 
 1   maint     1727 non-null   object 
 2   doors     1728 non-null   int64  
 3   persons   1151 non-null   float64
 4   lug_boot  1728 non-null   object 
 5   safety    1727 non-null   object 
 6   class     1727 non-null   object 
dtypes: float64(1), int64(1), object(5)
memory usage: 94.6+ KB
```

	buying	maint	doors	persons	lug_boot	safety	class
0	vhigh	vhigh	2	2.0	small	low	unacc
1	vhigh	vhigh	2	2.0	small	med	unacc
2	vhigh	vhigh	2	3.0	small	high	unacc
3	vhigh	vhigh	2	2.0	med	low	unacc
4	vhigh	vhigh	2	2.0	med	med	unacc
5	Unknown	vhigh	2	2.0	med	high	Unknown
6	vhigh	vhigh	2	2.0	big	low	unacc
7	vhigh	vhigh	2	2.0	big	Unknown	unacc
8	vhigh	vhigh	2	2.0	big	high	unacc
9	vhigh	Unknown	2	4.0	small	low	unacc

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1728 entries, 0 to 1727
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype   
---  ------    --------------  -----   
 0   buying    1728 non-null   category
 1   maint     1728 non-null   category
 2   doors     1728 non-null   int64   
 3   persons   1728 non-null   int32   
 4   lug_boot  1728 non-null   category
 5   safety    1728 non-null   category
 6   class     1728 non-null   category
dtypes: category(5), int32(1), int64(1)
memory usage: 29.8 KB
```

Out[3]:

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring numbe
0	p	x	s	n	t	p	f	c	n	k	...	s	w	w	p	w	(
1	e	x	s	y	t	a	f	c	b	k	...	s	w	w	p	w	(
2	e	b	s	w	t	l	f	c	b	n	...	s	w	w	p	w	(
3	p	x	y	w	t	p	f	c	n	n	...	s	w	w	p	w	(
4	e	x	s	g	f	n	f	w	b	k	...	s	w	w	p	w	(

5 rows × 23 columns

Out[12]:

```
Index(['class', 'cap-shape', 'cap-surface', 'cap-color', 'bruises', 'odor',
       'gill-attachment', 'gill-spacing', 'gill-size', 'gill-color',
       'stalk-shape', 'stalk-root', 'stalk-surface-above-ring',
       'stalk-surface-below-ring', 'stalk-color-above-ring',
       'stalk-color-below-ring', 'veil-type', 'veil-color', 'ring-number',
       'ring-type', 'spore-print-color', 'population', 'habitat'],
      dtype='object')
```

Out[19]:

class	population	habitat	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	stalk-surface-above-ring	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type
p	**s**	**u**	x	s	n	t	p	f	c	n	k	e	e	s	s	w	w	p
e	**n**	**g**	x	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p
		m	b	s	w	t	l	f	c	b	n	e	c	s	s	w	w	p
p	**s**	**u**	x	y	w	t	p	f	c	n	n	e	e	s	s	w	w	p
e	**a**	**g**	x	s	g	f	n	f	w	b	k	t	e	s	s	w	w	p
	n	**g**	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p
		m	b	s	w	t	a	f	c	b	g	e	c	s	s	w	w	p
	s	**m**	b	y	w	t	l	f	c	b	n	e	c	s	s	w	w	p
p	**v**	**g**	x	y	w	t	p	f	c	n	p	e	e	s	s	w	w	p
e	**s**	**m**	b	s	y	t	a	f	c	b	g	e	c	s	s	w	w	p

Out[20]:

population	habitat	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	stalk-surface-above-ring	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color
n	**g**	x	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w
	m	b	s	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w
a	**g**	x	s	g	f	n	f	w	b	k	t	e	s	s	w	w	p	w
n	**g**	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w
	m	b	s	w	t	a	f	c	b	g	e	c	s	s	w	w	p	w
...	**...**	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
v	**l**	x	s	n	f	n	a	c	b	y	e	?	s	s	o	o	p	o
c	**l**	k	s	n	f	n	a	c	b	y	e	?	s	s	o	o	p	o
v	**l**	x	s	n	f	n	a	c	b	y	e	?	s	s	o	o	p	n
c	**l**	f	s	n	f	n	a	c	b	n	e	?	s	s	o	o	p	o
	l	x	s	n	f	n	a	c	b	y	e	?	s	s	o	o	p	o

4208 rows × 20 columns

```
Out[13]: City
         New York                8190209
         Los Angeles             3795512
         Chicago                 2697477
         Houston                 2100280
         Phoenix                 1449038
         Philadelphia            1528283
         San Antonio             1332299
         San Diego               1305906
         Dallas                  1200350
         San Jose                 954940
         Austin                   806164
         Jacksonville             823114
         Fort Worth               748441
         Columbus                 790943
         Charlotte                738444
         San Francisco            805505
         Indianapolis             821579
         Seattle                  610630
         Denver                   603359
         District of Columbia     605226
         Name: 2010, dtype: int64
```

```
Out[20]: City
New York                  8336817
Los Angeles               3979576
Chicago                   2693976
Houston                   2320268
Phoenix                   1680992
Philadelphia              1584064
San Antonio               1547253
San Diego                 1423851
Dallas                    1343573
San Jose                  1021795
Austin                     978908
Jacksonville               911507
Fort Worth                 909585
Columbus                   898553
Charlotte                  885708
San Francisco              881549
Indianapolis               876384
Seattle                    753675
Denver                     727211
District of Columbia       705749
Name: 2019, dtype: int64
top 3 changed 2.2 %
 vs. all changed 8.0 %
```

Out[4]:

	Sex	Length	Diameter	Height	Whole weight	.Shucked weight	Viscera weight	Shell weight	Rings
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7

Out[5]:

Sex	Rings	Length	Diameter	Height	Whole weight	.Shucked weight	Viscera weight	Shell weight
M	15	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150
	7	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070
F	9	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210
M	10	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155
I	7	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055
	8	0.425	0.300	0.095	0.3515	0.1410	0.0775	0.120
F	20	0.530	0.415	0.150	0.7775	0.2370	0.1415	0.330
	16	0.545	0.425	0.125	0.7680	0.2940	0.1495	0.260
M	9	0.475	0.370	0.125	0.5095	0.2165	0.1125	0.165
F	19	0.550	0.440	0.150	0.8945	0.3145	0.1510	0.320

```
for oysters with 16 or more rings

males weigh 0.458 vs. females weigh 0.449
males are 0.603 long  vs. females are 0.603 long
males are 0.478 in diameter  vs. females are 0.479 in diameter
males are 0.176 in height  vs. females are 0.174 in height
```

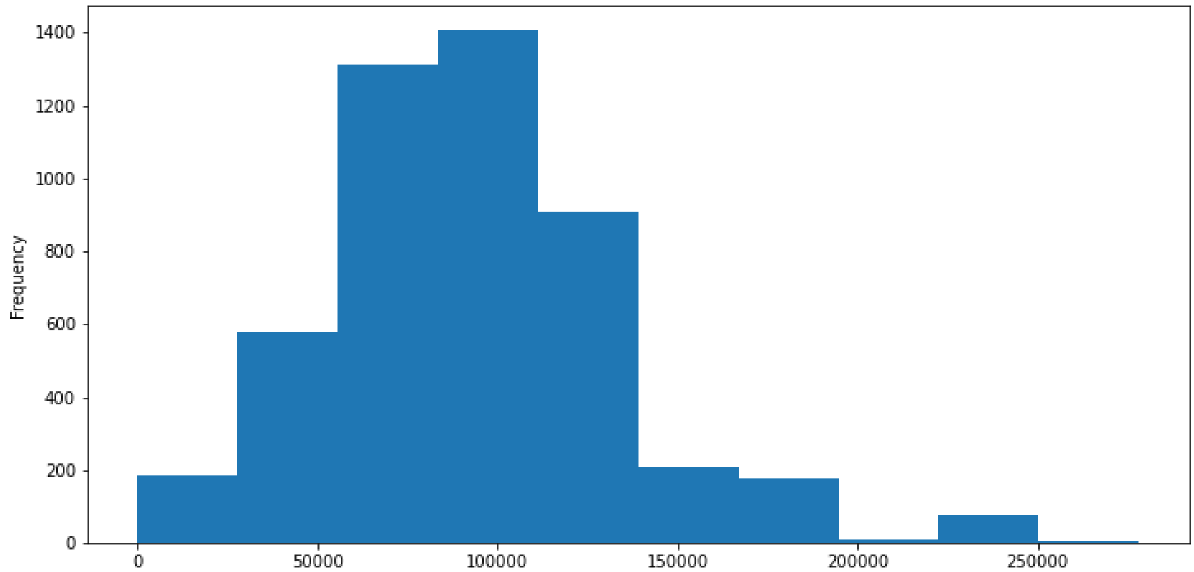
	school	sex	age	address	health	absences	G1	G2	G3
0	GP	F	18	U	3	6	5	6	6
1	GP	F	17	U	3	4	5	5	6
2	GP	F	15	U	3	10	7	8	10
3	GP	F	15	U	5	2	15	14	15
4	GP	F	16	U	5	4	6	10	10
5	GP	M	16	U	5	10	15	15	15
6	GP	M	16	U	3	0	12	12	11
7	GP	F	17	U	1	6	6	5	6
8	GP	M	15	U	1	0	16	18	19
9	GP	M	15	U	5	0	14	15	15

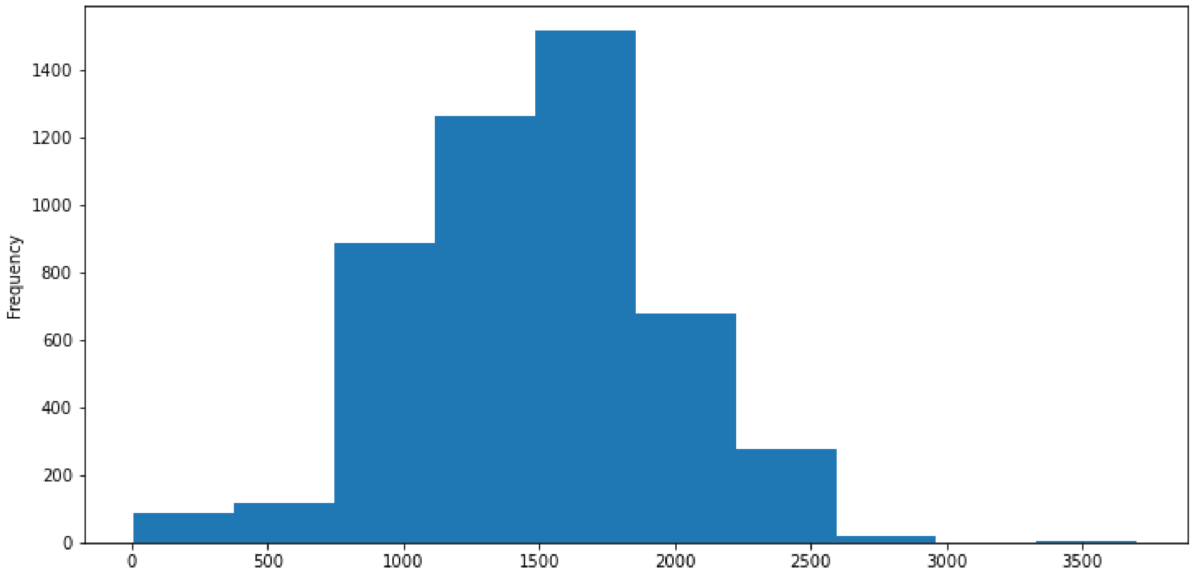
school	G1	G2	G3	absences	age	health
GP	10.939828	10.782235	10.489971	5.965616	16.521490	3.575931
MS	10.673913	10.195652	9.847826	3.760870	18.021739	3.391304

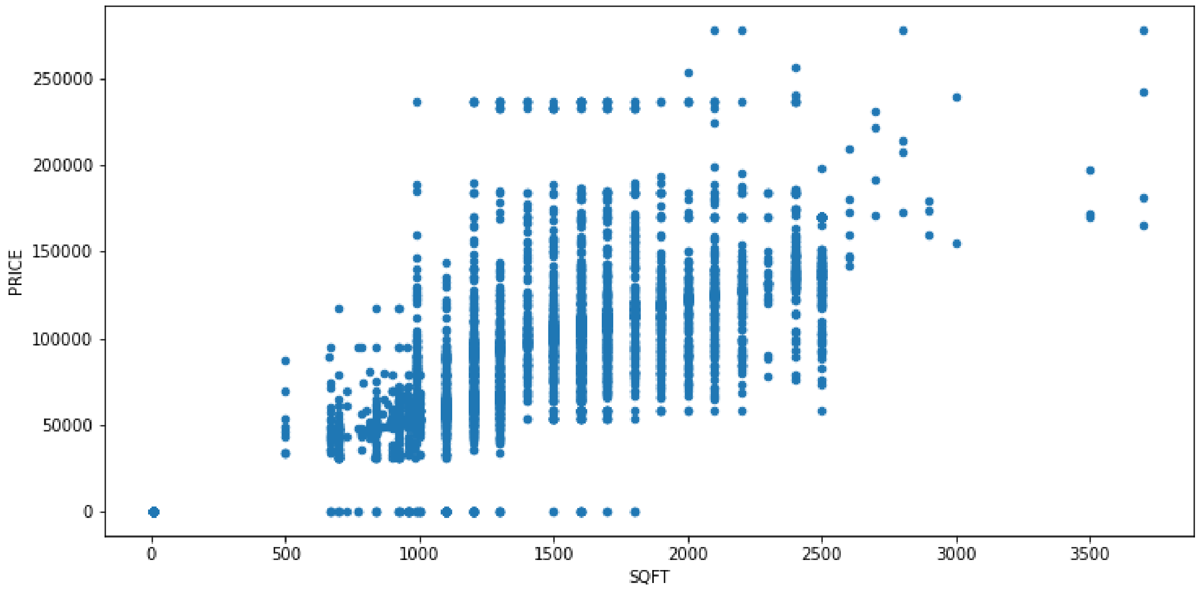
school	age	G1	G2	G3	absences	health
GP	15	11.231707	11.365854	11.256098	3.341463	3.585366
	16	10.942308	11.182692	11.028846	5.451923	3.701923
	17	10.802326	10.383721	10.232558	6.709302	3.639535
	18	10.614035	9.964912	9.157895	7.333333	3.350877
	19	11.222222	10.055556	9.055556	12.777778	3.277778
	20	17.000000	18.000000	18.000000	0.000000	5.000000
	22	6.000000	8.000000	8.000000	16.000000	1.000000
MS	17	11.583333	11.166667	10.583333	4.666667	2.500000
	18	10.960000	10.520000	10.440000	3.120000	3.640000
	19	7.333333	6.833333	5.666667	3.500000	4.166667
	20	12.000000	11.500000	12.000000	7.500000	3.500000
	21	10.000000	8.000000	7.000000	3.000000	3.000000

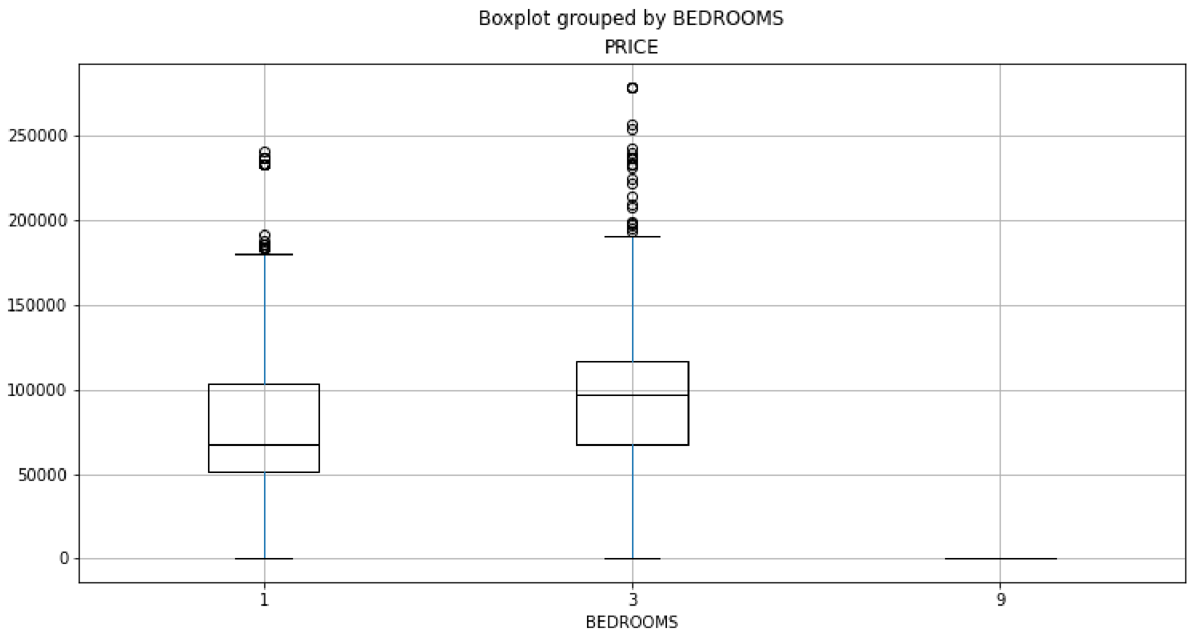
school	sex	age	mean	sum
GP	F	15	3.894737	148.0
		16	5.888889	318.0
		17	7.120000	356.0
		18	8.137931	236.0
		19	13.083333	157.0
	M	15	2.863636	126.0
		16	4.980000	249.0
		17	6.138889	221.0
		18	6.500000	182.0
		19	12.166667	73.0
		20	0.000000	0.0
		22	16.000000	16.0
MS	F	17	5.625000	45.0
		18	1.785714	25.0
		19	2.000000	4.0
		20	4.000000	4.0
	M	17	2.750000	11.0
		18	4.818182	53.0
		19	4.250000	17.0
		20	11.000000	11.0
		21	3.000000	3.0

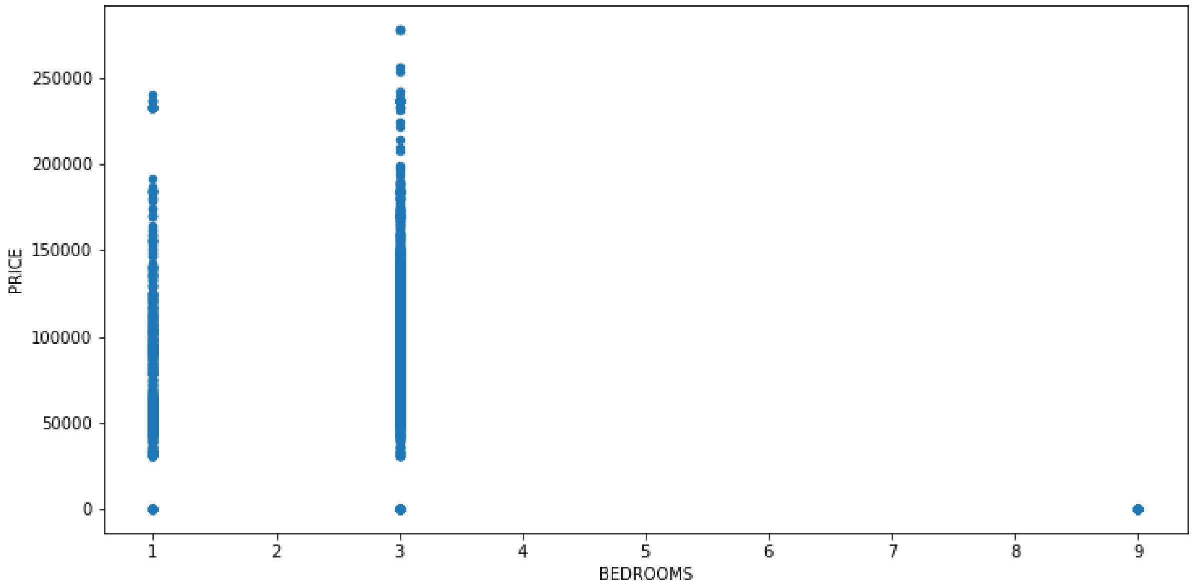
	REGION	SQFT	BEDROOMS	PRICE
0	3	960	1	52000
1	3	1300	3	39900
2	4	1200	3	60000
3	4	730	1	9
4	4	500	1	87000
5	4	1100	3	56000
6	1	1000	1	9
7	3	700	1	42600
8	3	700	1	46300
9	3	1200	3	61000

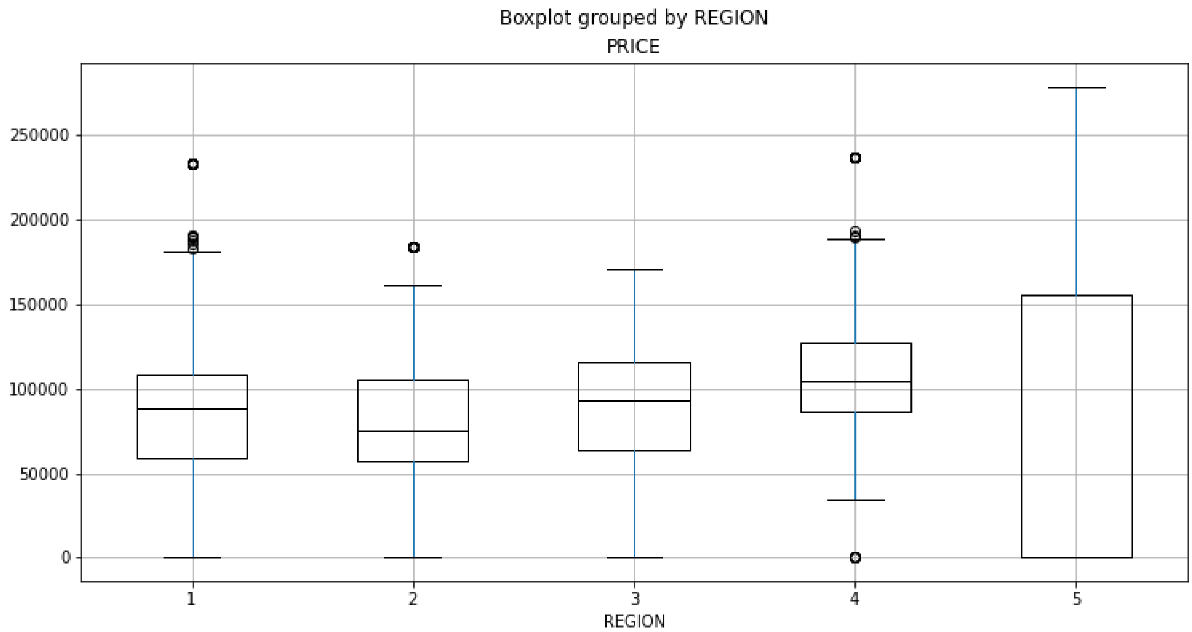


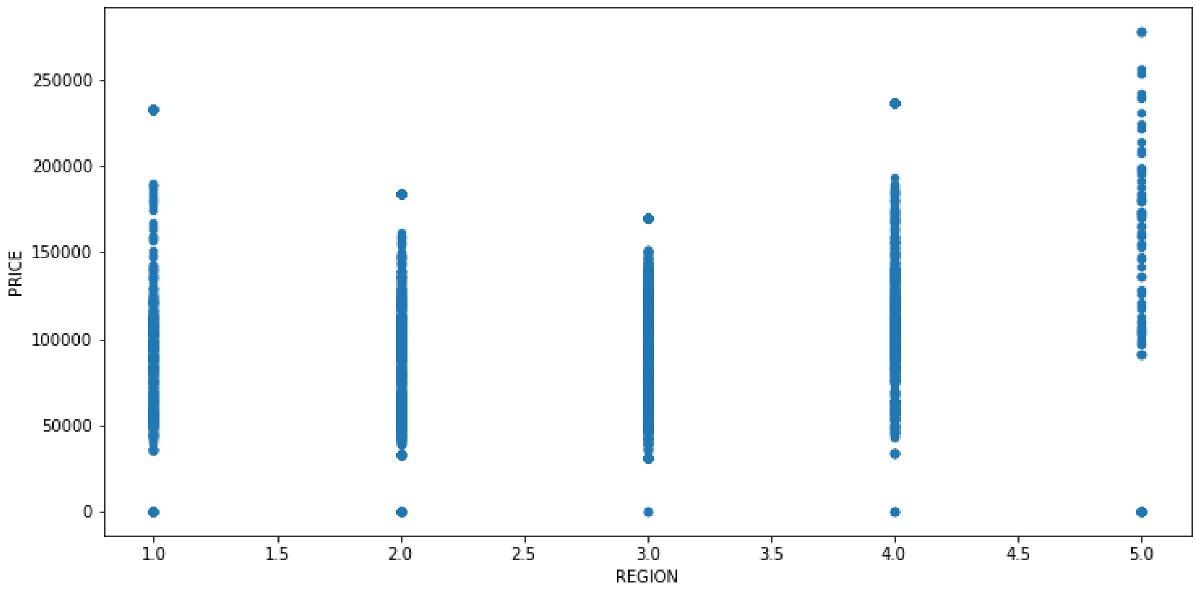

Frequency
0
200
400
600
800
1000
1200
1400
0
500
1000
1500
2000
2500
3000
3500

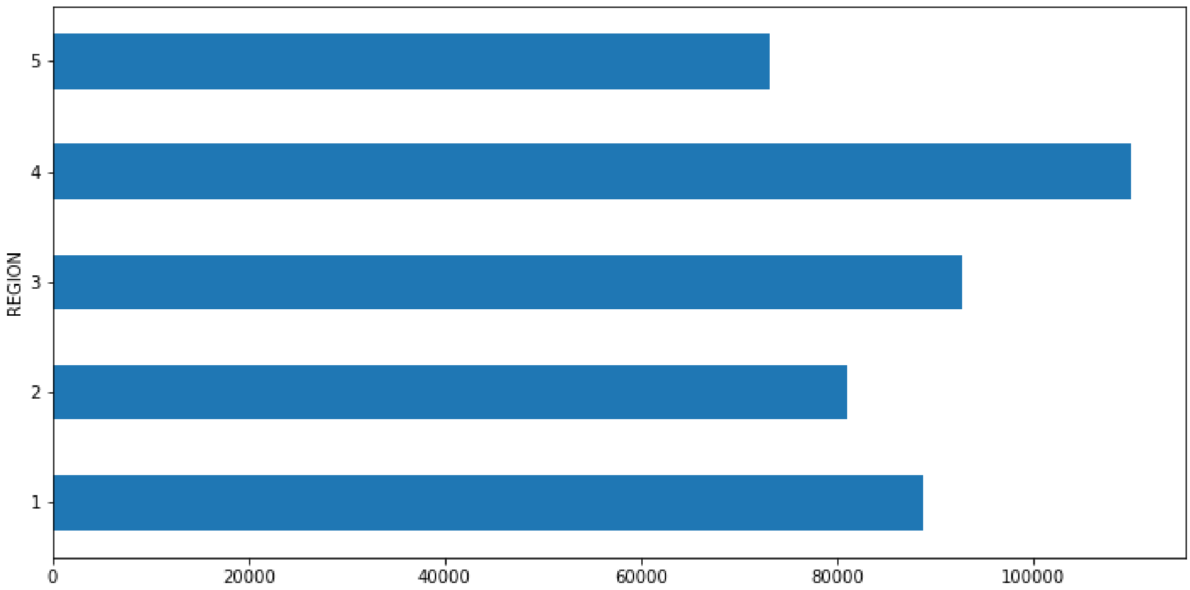

PRICE
SQFT
0
50000
100000
150000
200000
250000
0
500
1000
1500
2000
2500
3000
3500

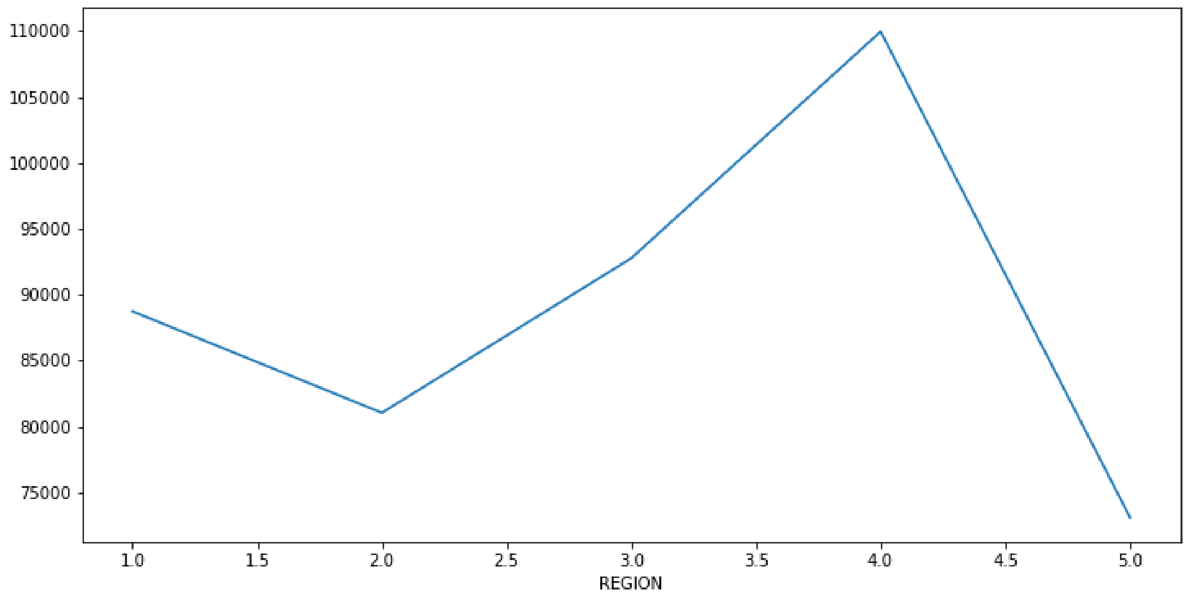

Boxplot grouped by BEDROOMS
PRICE
0
50000
100000
150000
200000
250000
1
3
9
BEDROOMS


PRICE
250000
200000
150000
100000
50000
0
1
2
3
4
5
6
7
8
9
BEDROOMS


Boxplot grouped by REGION
PRICE
250000
200000
150000
100000
50000
0
1
2
3
4
5
REGION


PRICE
250000
200000
150000
100000
50000
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
REGION


REGION
5
4
3
2
1
0
20000
40000
60000
80000
100000


110000
105000
100000
95000
90000
85000
80000
75000
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
REGION

```
Out[5]:
```

	AT	V	AP	RH	EP
0	8.34	40.77	1010.84	90.01	480.48
1	23.64	58.49	1011.40	74.20	445.75
2	29.74	56.90	1007.15	41.91	438.76
3	19.07	49.69	1007.22	76.79	453.09
4	11.80	40.66	1017.13	97.20	464.43

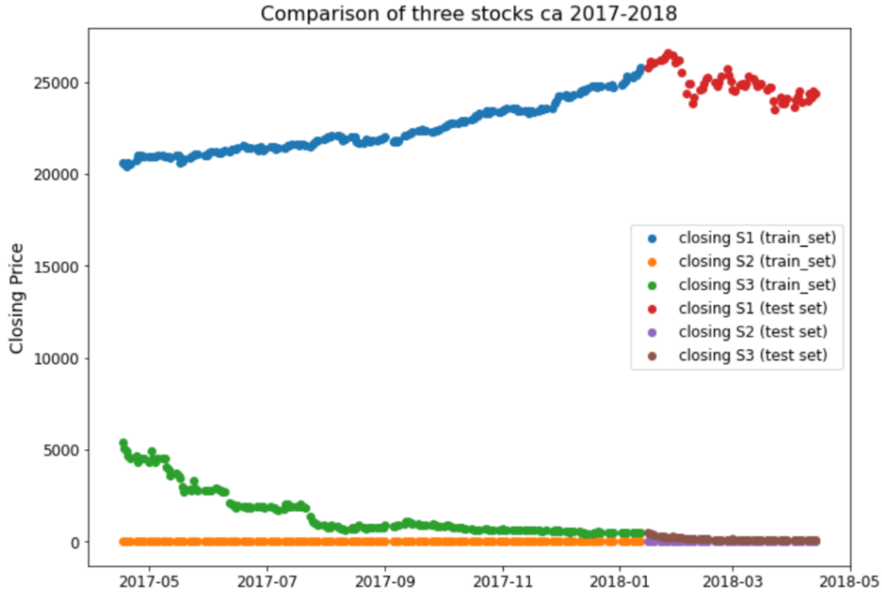
```
train is  (7654, 5)  rows, cols train is  (7654, 5)  rows, cols
 val is  (956, 5)  rows, cols     val is  (957, 5)  rows, cols
 test is  (958, 5) rows, cols     test is  (957, 5) rows, cols
              train score:  0.9287072840354756
              validation score:  0.9238845251967255
              test score:  0.9333918854821254
              train RMSE:  20.732519659228675
              validation RMSE:  22.82059184376622
              test RMSE:  19.0233909525747
```

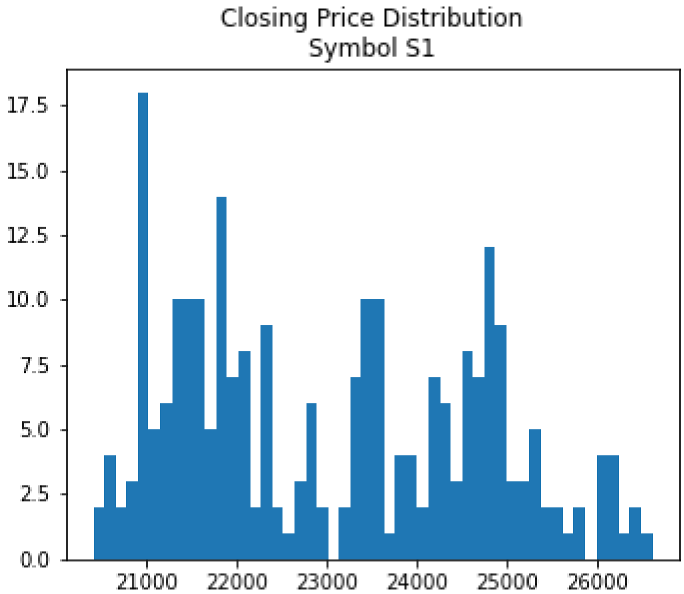
```
Out[5]:
```

	Date	Close	Volume	symbol	sentiment
0	2017-04-17	20636.919922	229240000	S1	NEUTRAL
1	2017-04-17	20.000000	88300	S2	NEUTRAL
2	2017-04-17	5400.000000	0	S3	NEUTRAL
3	2017-04-18	20523.279297	263180000	S1	NEUTRAL
4	2017-04-18	20.150000	60500	S2	NEUTRAL

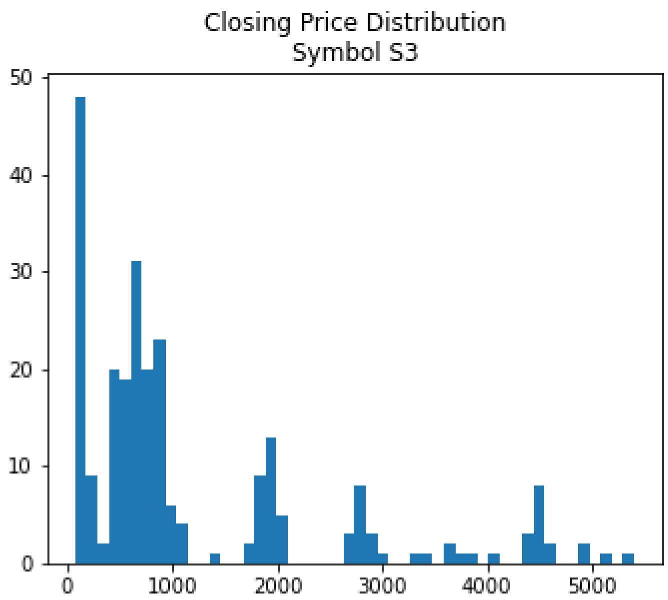
```
Out[6]: Date              object
        Close            float64
        Volume             int64
        symbol            object
        sentiment         object
        dtype: object
```

```
Out[7]: count                        753
        unique                       251
        top          2017-08-29 00:00:00
        freq                           3
        first        2017-04-17 00:00:00
        last         2018-04-13 00:00:00
        Name: Date, dtype: object
```

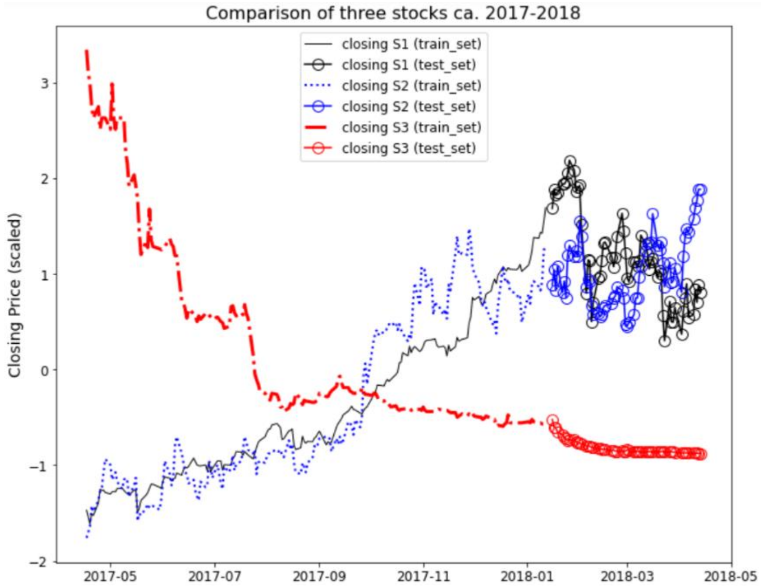

Comparison of three stocks ca 2017-2018
Closing Price
25000
20000
15000
10000
5000
0
2017-05
2017-07
2017-09
2017-11
2018-01
2018-03
2018-05
closing S1 (train_set)
closing S2 (train_set)
closing S3 (train_set)
closing S1 (test set)
closing S2 (test set)
closing S3 (test set)

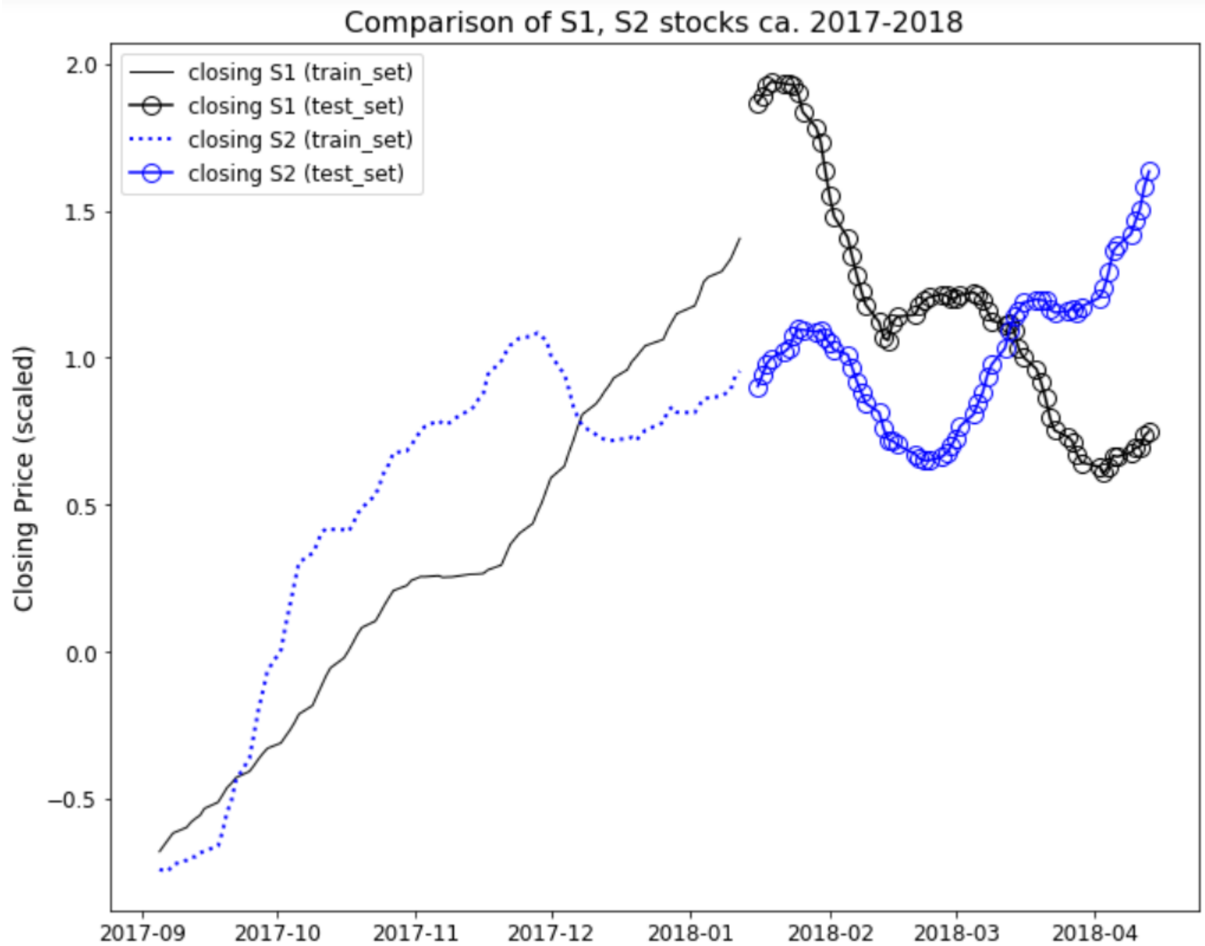

Closing Price Distribution
Symbol S1
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
21000
22000
23000
24000
25000
26000


Closing Price Distribution
Symbol S2
16
14
12
10
8
6
4
2
0
20
22
24
26
28



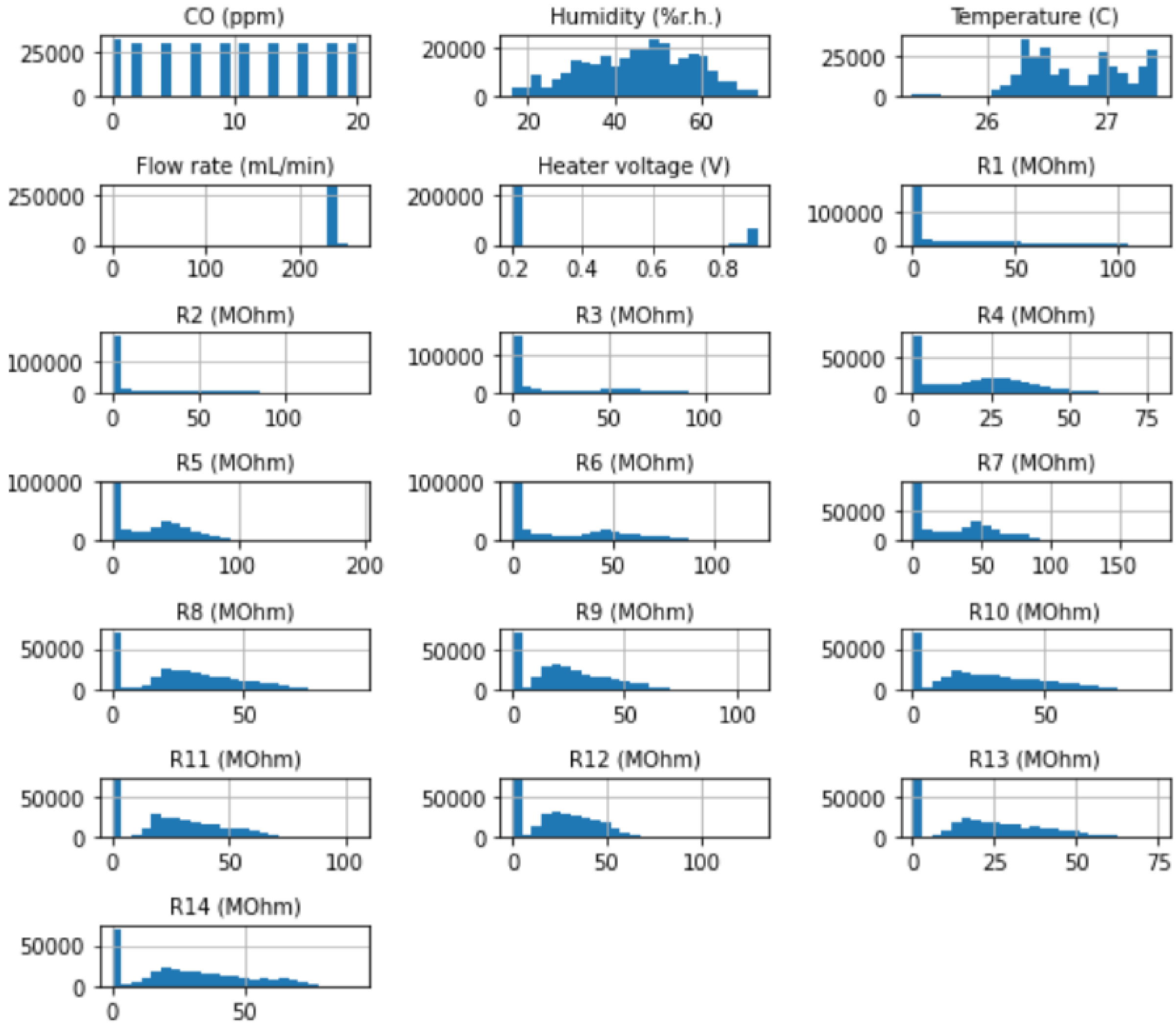

```
Out[41]: [          Date     Close    Volume symbol sentiment
          0  2017-04-17 -1.469506 -1.175399     S1   NEUTRAL
          3  2017-04-18 -1.538998 -0.840327     S1   NEUTRAL
          6  2017-04-19 -1.611638 -0.528257     S1   NEUTRAL
          9  2017-04-20 -1.505101 -0.354008     S1       POS
          14 2017-04-21 -1.524028  0.210303     S1   NEUTRAL,
                    Date     Close    Volume symbol sentiment
          1  2017-04-17 -1.757829  0.198494     S2   NEUTRAL
          4  2017-04-18 -1.699092 -0.359611     S2   NEUTRAL
          7  2017-04-19 -1.640355  0.351069     S2   NEUTRAL
          10 2017-04-20 -1.424984 -0.443929     S2       POS
          12 2017-04-21 -1.483721 -0.259233     S2   NEUTRAL,
                    Date     Close    Volume symbol sentiment
          2  2017-04-17  3.342186 -0.211226     S3   NEUTRAL
          5  2017-04-18  3.104449 -0.211226     S3   NEUTRAL
          8  2017-04-19  2.985580 -0.211226     S3   NEUTRAL
          11 2017-04-20  2.747843 -0.211226     S3       NEG
          13 2017-04-21  2.628974 -0.211226     S3   NEUTRAL]
```

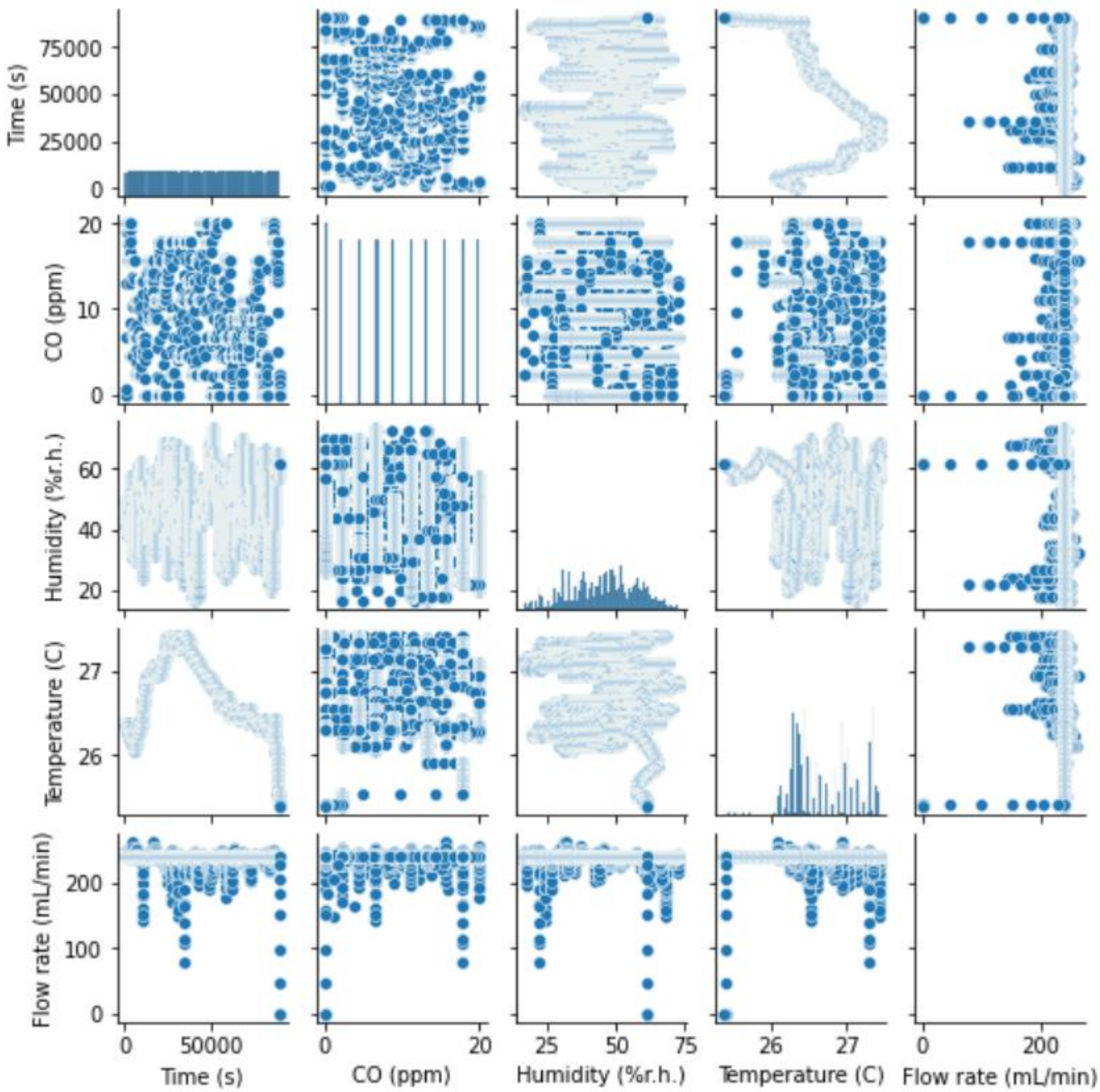


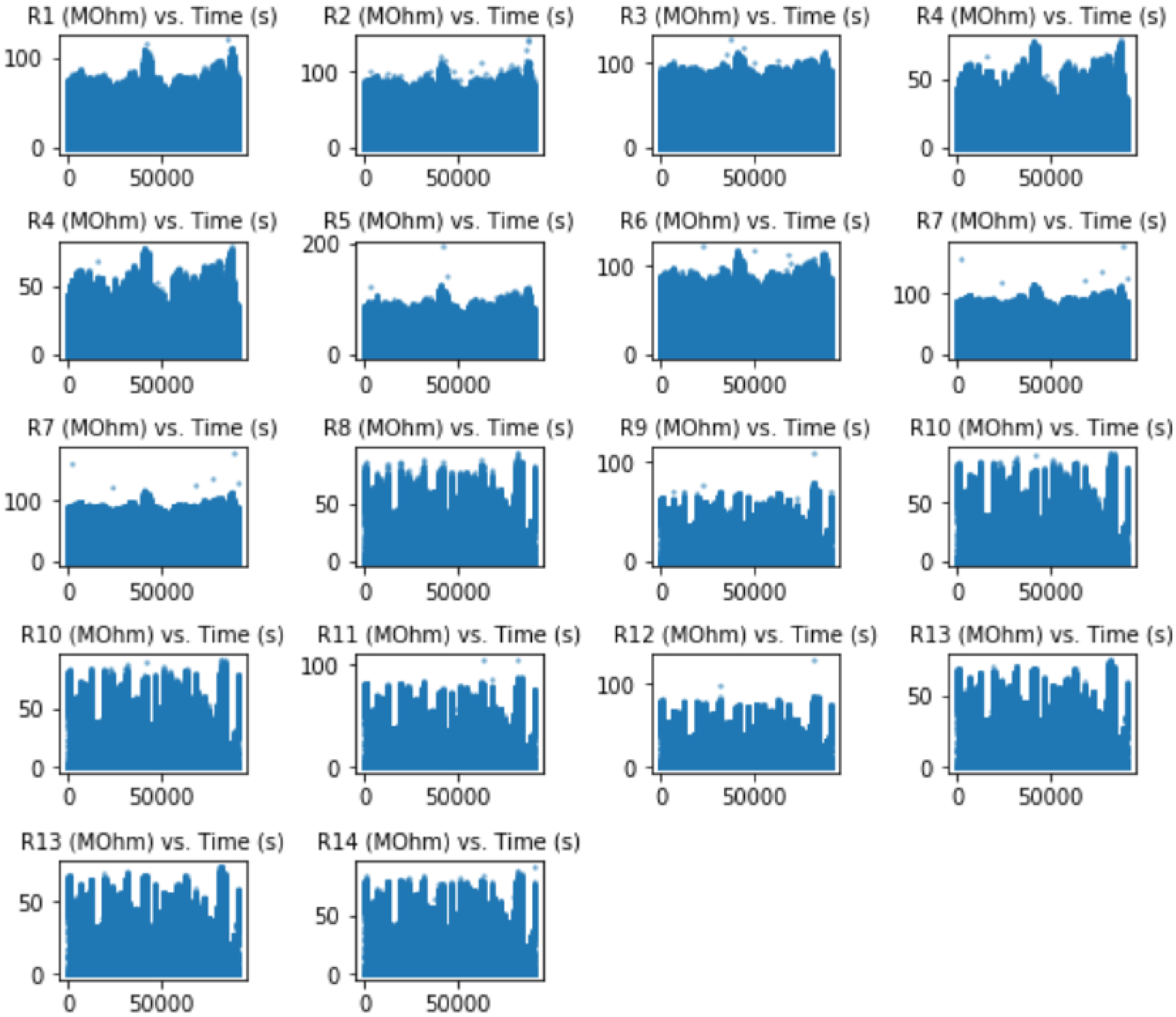


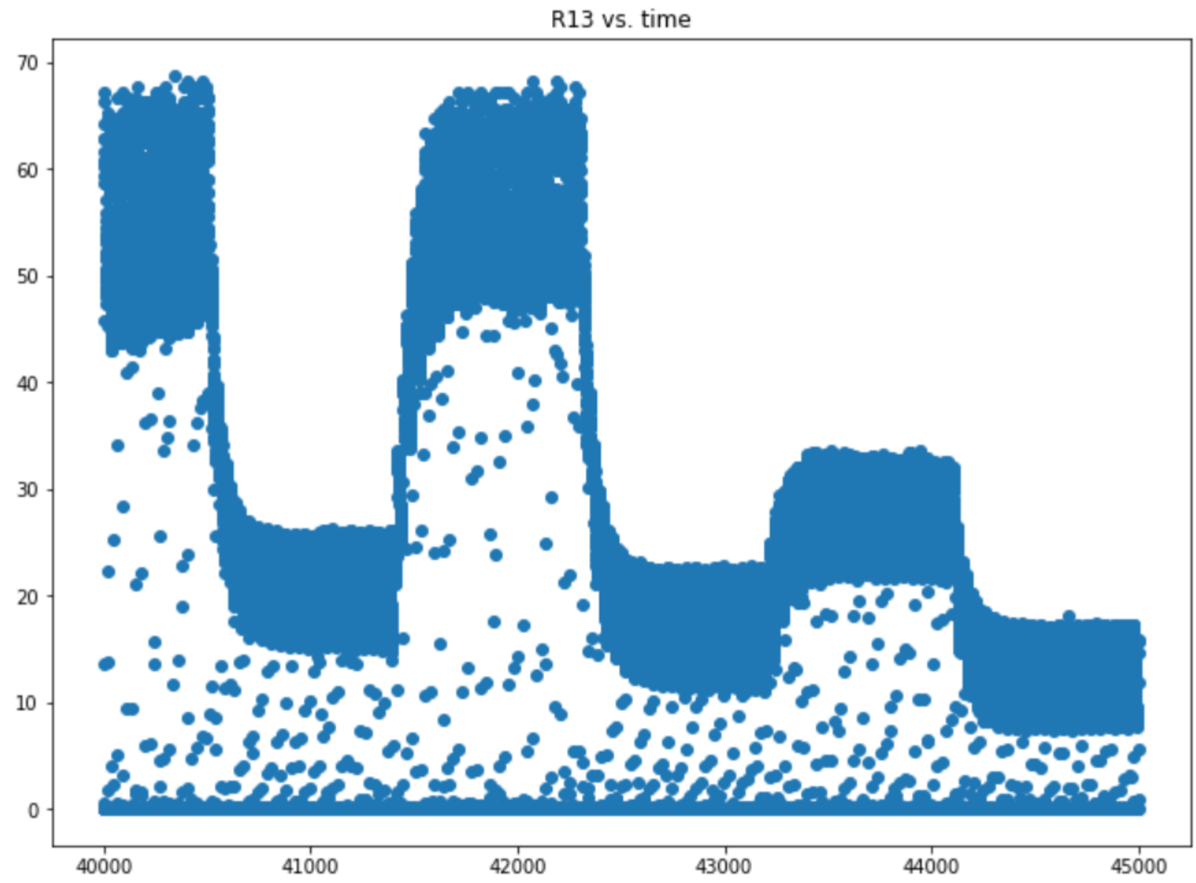
	Time (s)	CO (ppm)	Humidity (%r.h.)	Temperature (C)	Flow rate (mL/min)	Heater voltage (V)	R1 (MOhm)	R2 (MOhm)	R3 (MOhm)	R4 (MOhm)	R5 (MOhm)	R6 (MOhm)	R7 (MOhm)	R8 (MOhm)	R9 (MOhm)	R10 (MOhm)	R (MOh
0	0.000	0.0	49.21	26.38	247.2771	0.1994	0.5114	0.5863	0.5716	1.9386	1.1669	0.7103	0.5541	51.0146	40.8079	47.8748	4.6
1	0.311	0.0	49.21	26.38	243.3618	0.7158	0.0626	0.1586	0.1161	0.1347	0.1385	0.1545	0.1307	0.1935	0.1341	0.1773	0.1
2	0.620	0.0	49.21	26.38	242.4944	0.8840	0.0654	0.1496	0.1075	0.1076	0.1131	0.1363	0.1188	0.1195	0.1049	0.1289	0.1
3	0.930	0.0	49.21	26.38	241.6242	0.8932	0.0722	0.1444	0.1074	0.1032	0.1106	0.1306	0.1190	0.1125	0.1014	0.1232	0.1
4	1.238	0.0	49.21	26.38	240.8151	0.8974	0.0767	0.1417	0.1098	0.1025	0.1116	0.1284	0.1208	0.1111	0.1008	0.1226	0.1

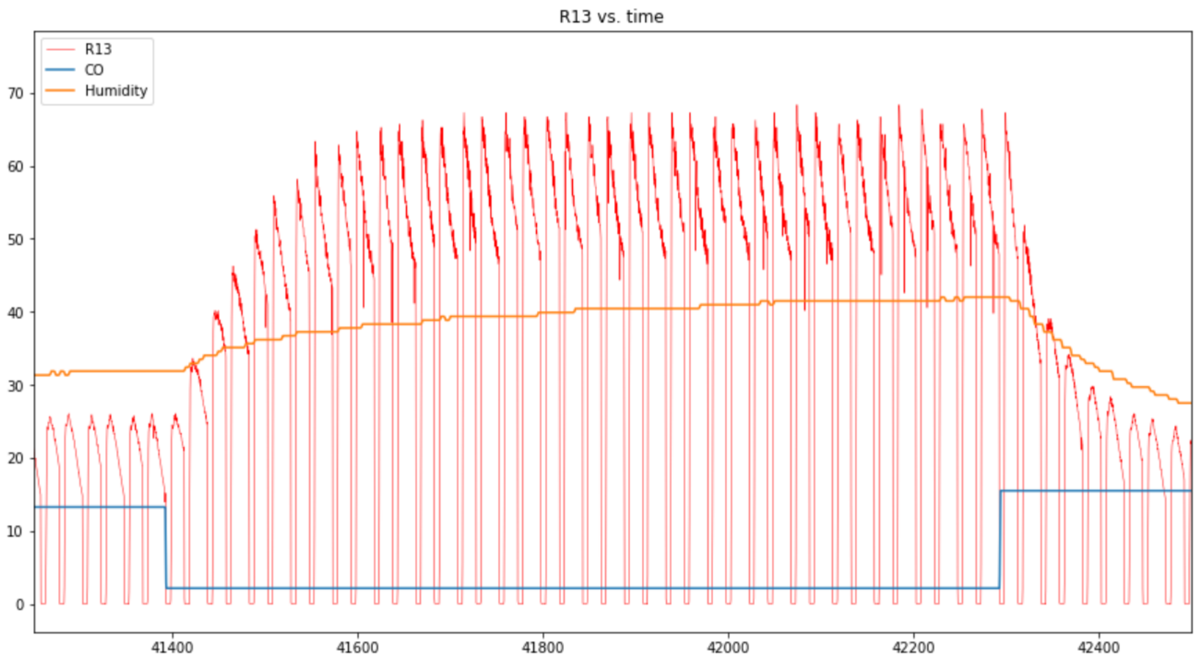
	count	mean	std	min	25%	50%	75%	max
Time (s)	295700.0	45435.140266	26245.705362	0.0000	22696.21350	45430.5430	68165.08150	90901.7260
CO (ppm)	295700.0	9.900266	6.426957	0.0000	4.44000	8.8900	15.56000	20.0000
Humidity (%r.h.)	295700.0	45.607506	12.445601	16.4300	36.14000	46.7000	55.37000	72.9800
Temperature (C)	295700.0	26.720057	0.418020	25.3800	26.38000	26.6600	27.06000	27.4200
Flow rate (mL/min)	295700.0	239.943680	1.697848	0.0000	239.90420	239.9716	240.03660	262.3167
Heater voltage (V)	295700.0	0.355212	0.288572	0.1990	0.20000	0.2000	0.20700	0.9010
R1 (MOhm)	295700.0	15.198374	22.583110	0.0324	0.40480	1.7121	25.85040	119.5851
R2 (MOhm)	295700.0	17.440031	26.665302	0.0555	0.48140	1.3664	29.05830	142.5199
R3 (MOhm)	295700.0	22.151461	28.585001	0.0541	0.57940	4.0667	44.88580	127.2483
R4 (MOhm)	295700.0	19.759571	16.412620	0.0394	1.94360	19.9434	31.75500	78.4601
R5 (MOhm)	295700.0	31.360319	27.068315	0.0480	1.72010	32.3170	51.48750	194.6753
R6 (MOhm)	295700.0	28.601243	27.198270	0.0493	1.50860	22.5929	49.60550	122.0913
R7 (MOhm)	295700.0	31.640992	27.612186	0.0517	1.80335	31.2996	52.41740	177.9975
R8 (MOhm)	295700.0	26.658295	19.523869	0.0334	11.69870	26.4721	40.41290	93.4149
R9 (MOhm)	295700.0	23.000006	17.919762	0.0291	8.44600	21.5685	35.50410	109.1693
R10 (MOhm)	295700.0	25.417975	20.410103	0.0368	7.56070	23.1211	39.88530	92.5828
R11 (MOhm)	295700.0	27.205435	20.348773	0.0309	10.29880	26.6826	41.73510	105.0967
R12 (MOhm)	295700.0	25.201259	18.560530	0.0327	9.45670	25.2860	38.99700	129.9261
R13 (MOhm)	295700.0	22.026591	17.036098	0.0331	7.59640	20.8730	34.05870	74.7083
R14 (MOhm)	295700.0	28.258380	21.982871	0.0316	9.47520	26.3557	44.15375	92.5210


CO (ppm)
Humidity (%r.h.)
Temperature (C)
Flow rate (mL/min)
Heater voltage (V)
R1 (MOhm)
R2 (MOhm)
R3 (MOhm)
R4 (MOhm)
R5 (MOhm)
R6 (MOhm)
R7 (MOhm)
R8 (MOhm)
R9 (MOhm)
R10 (MOhm)
R11 (MOhm)
R12 (MOhm)
R13 (MOhm)
R14 (MOhm)

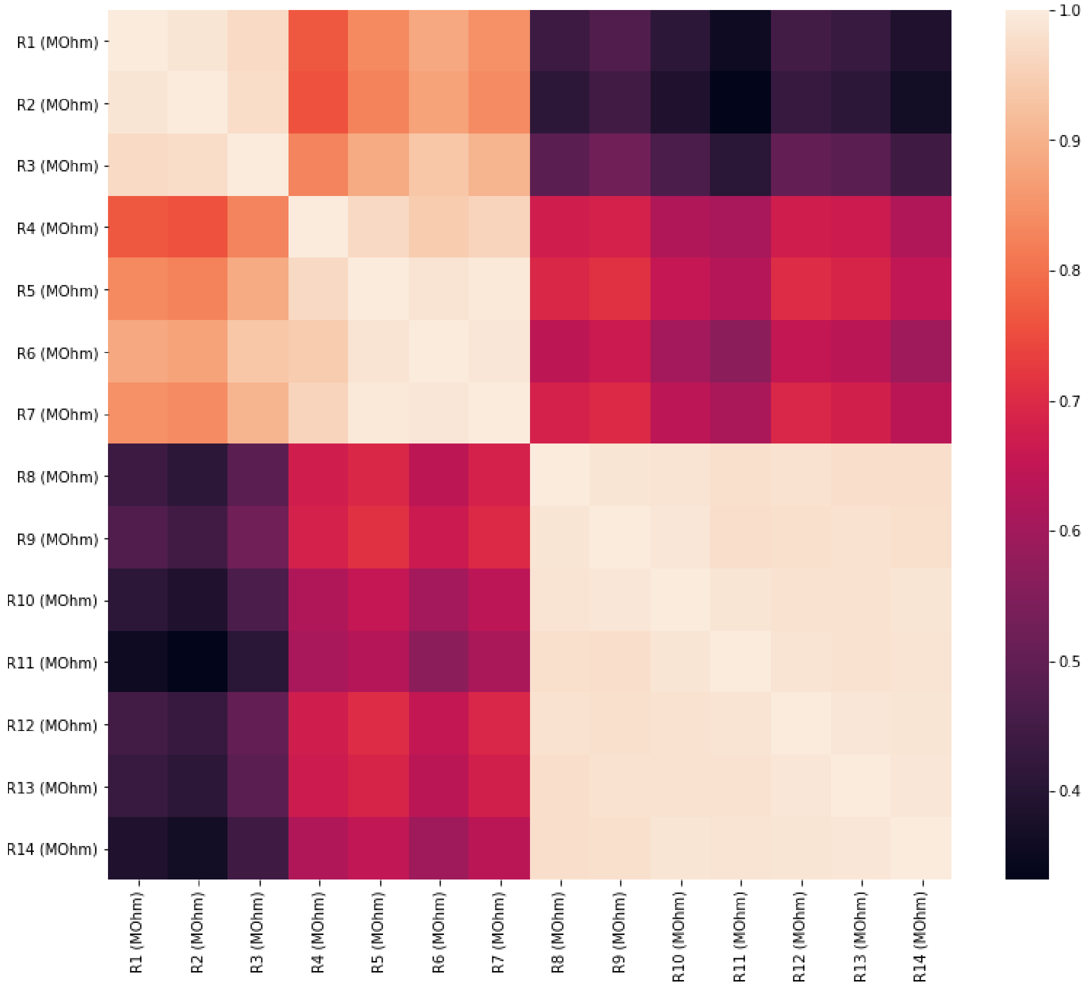

Time (s)
CO (ppm)
Humidity (%r.h.)
Temperature (C)
Flow rate (mL/min)
75000
50000
25000
0
20
10
0
60
40
20
27
26
200
100
0
50000
20
25
50
75
26
27
0
200

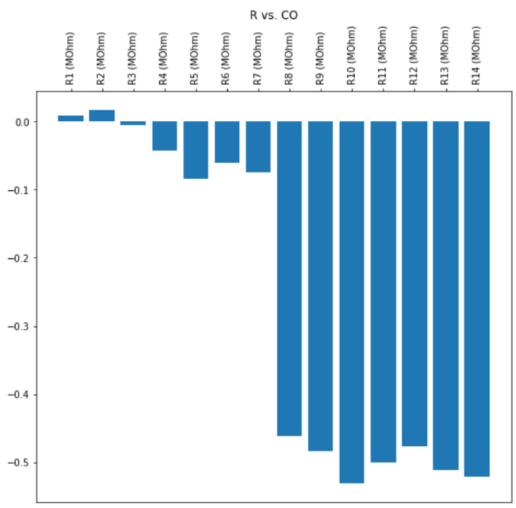

R1 (MOhm) vs. Time (s)
R2 (MOhm) vs. Time (s)
R3 (MOhm) vs. Time (s)
R4 (MOhm) vs. Time (s)
R4 (MOhm) vs. Time (s)
R5 (MOhm) vs. Time (s)
R6 (MOhm) vs. Time (s)
R7 (MOhm) vs. Time (s)
R7 (MOhm) vs. Time (s)
R8 (MOhm) vs. Time (s)
R9 (MOhm) vs. Time (s)
R10 (MOhm) vs. Time (s)
R10 (MOhm) vs. Time (s)
R11 (MOhm) vs. Time (s)
R12 (MOhm) vs. Time (s)
R13 (MOhm) vs. Time (s)
R13 (MOhm) vs. Time (s)
R14 (MOhm) vs. Time (s)

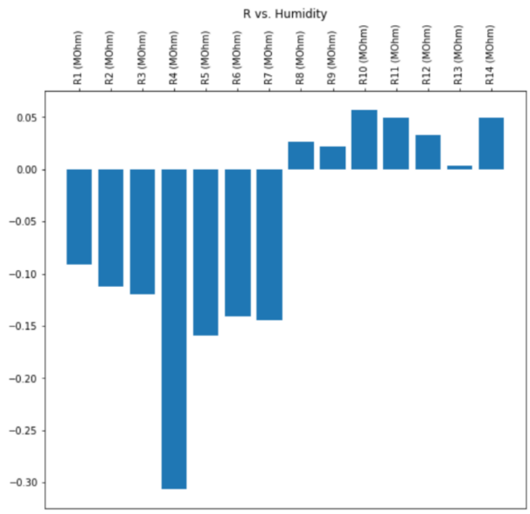

R13 vs. time
70
60
50
40
30
20
10
0
40000
41000
42000
43000
44000
45000

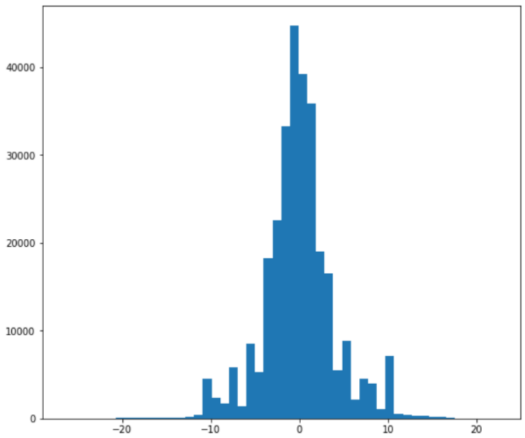

R13 vs. time
R13
CO
Humidity
70
60
50
40
30
20
10
0
41400
41600
41800
42000
42200
42400

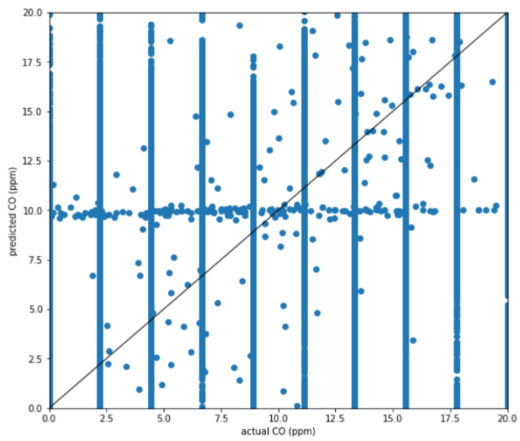
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x18d58f87908>

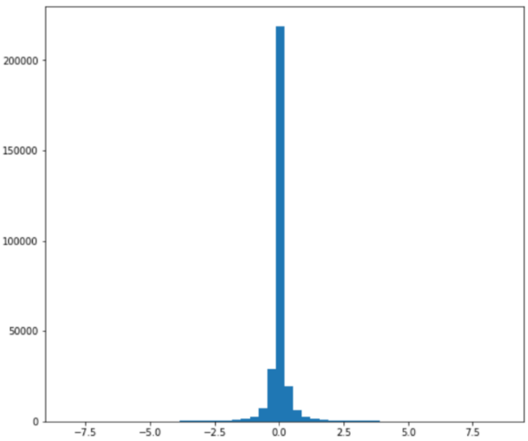


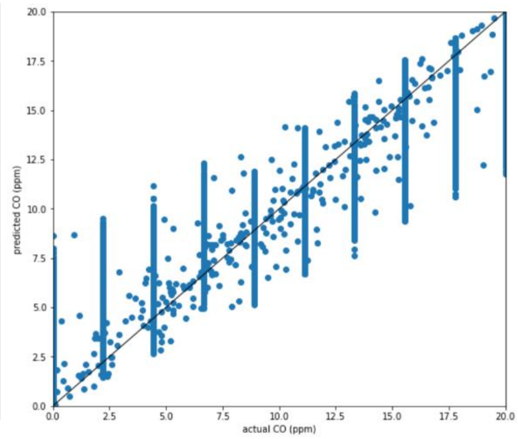


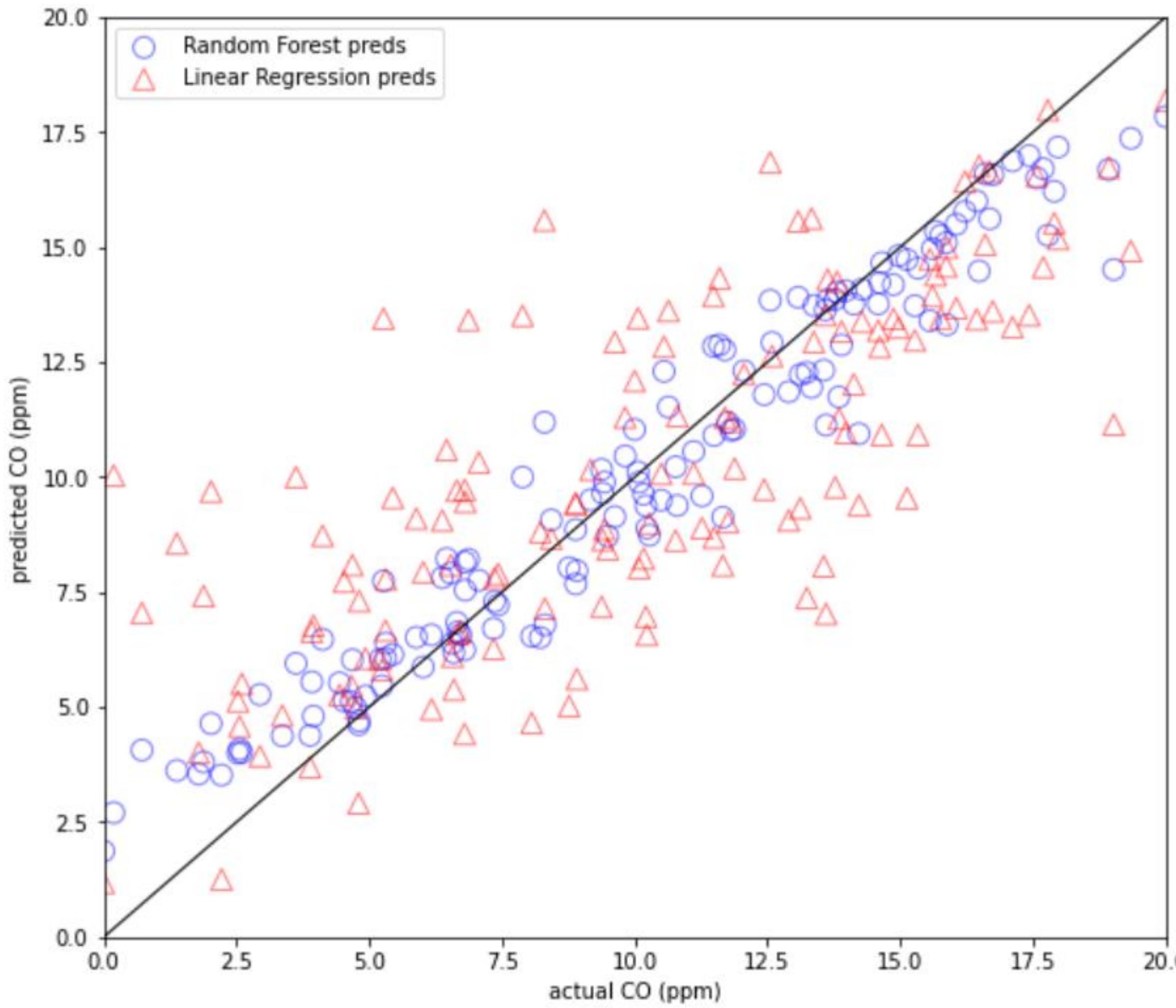



40000
30000
20000
10000
0
−20
−10
0
10
20


predicted CO (ppm)
actual CO (ppm)
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0


200000
150000
100000
50000
0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5


predicted CO (ppm)
actual CO (ppm)
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0

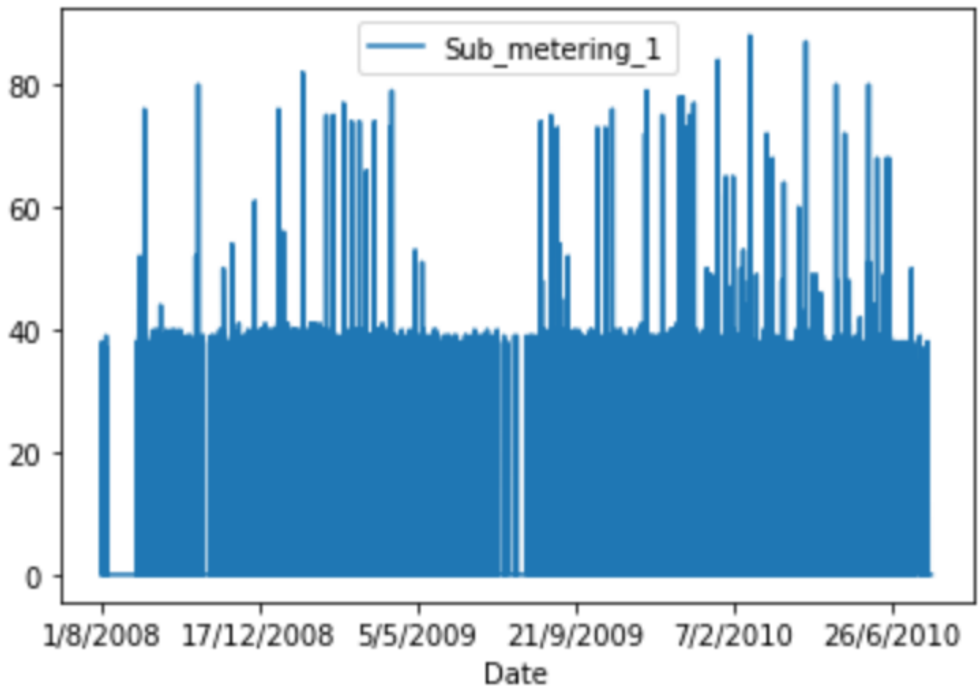


Out[5]:

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	1/8/2008	00:00:00	0.500	0.226	239.750	2.400	0.000	0.000	1.0
1	1/8/2008	00:01:00	0.482	0.224	240.340	2.200	0.000	0.000	1.0
2	1/8/2008	00:02:00	0.502	0.234	241.680	2.400	0.000	0.000	0.0
3	1/8/2008	00:03:00	0.556	0.228	241.750	2.600	0.000	0.000	1.0
4	1/8/2008	00:04:00	0.854	0.342	241.550	4.000	0.000	1.000	7.0

```
Out[4]: Date                      object
        Time                      object
        Global_active_power       object
        Global_reactive_power     object
        Voltage                   object
        Global_intensity          object
        Sub_metering_1            object
        Sub_metering_2            object
        Sub_metering_3           float64
        dtype: object
information for column Global_active_power:
 count     1049760
unique       3852
top             ?
freq         9570
Name: Global_active_power, dtype: object
information for column Global_reactive_power:
 count     1049760
unique        510
top         0.000
freq       230359
Name: Global_reactive_power, dtype: object
information for column Voltage:
 count     1049760
unique       2738
top             ?
freq         9570
Name: Voltage, dtype: object
```

Out[10]: <AxesSubplot:xlabel='Date'>



Out[13]:

	Date	Time	Kitchen_power_use
1074636	1/1/2009	00:00:00	0.0
1074637	1/1/2009	00:01:00	0.0
1074638	1/1/2009	00:02:00	0.0
1074639	1/1/2009	00:03:00	0.0
1074640	1/1/2009	00:04:00	0.0

Out[12]:

	Date	Time	Kitchen_power_use	timestamp
1074636	1/1/2009	00:00:00	0.0	2009-01-01 00:00:00
1074637	1/1/2009	00:01:00	0.0	2009-01-01 00:01:00
1074638	1/1/2009	00:02:00	0.0	2009-01-01 00:02:00
1074639	1/1/2009	00:03:00	0.0	2009-01-01 00:03:00
1074640	1/1/2009	00:04:00	0.0	2009-01-01 00:04:00

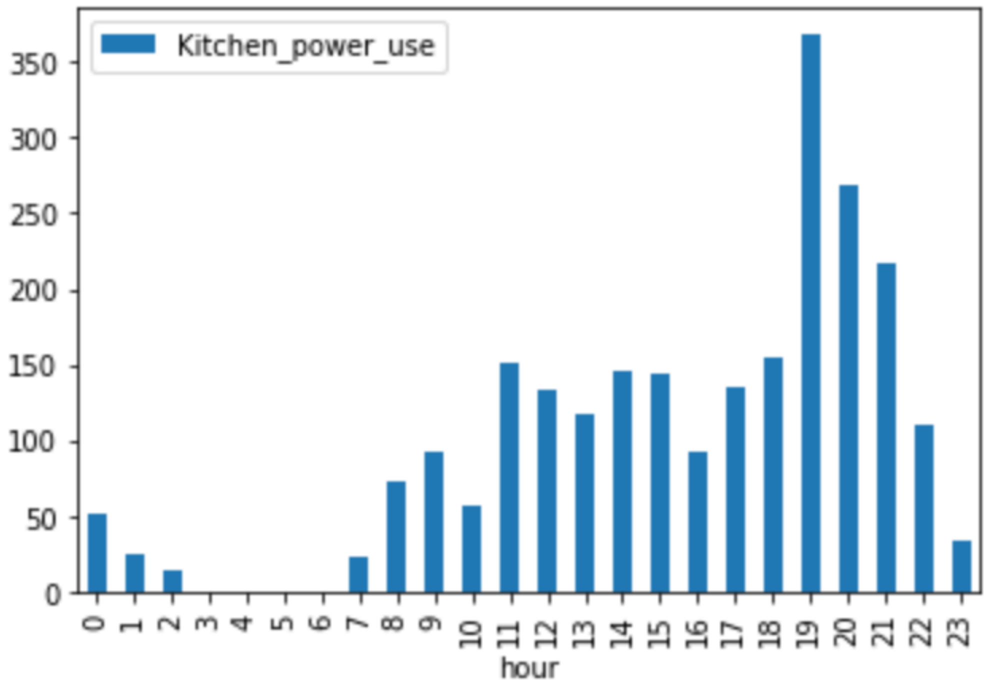
Out[34]:

	Date	Time	Kitchen_power_use	timestamp	hour	date
1074636	1/1/2009	00:00:00	0.0	2009-01-01 00:00:00	0	2009-01-01
1074637	1/1/2009	00:01:00	0.0	2009-01-01 00:01:00	0	2009-01-01
1074638	1/1/2009	00:02:00	0.0	2009-01-01 00:02:00	0	2009-01-01
1074639	1/1/2009	00:03:00	0.0	2009-01-01 00:03:00	0	2009-01-01
1074640	1/1/2009	00:04:00	0.0	2009-01-01 00:04:00	0	2009-01-01

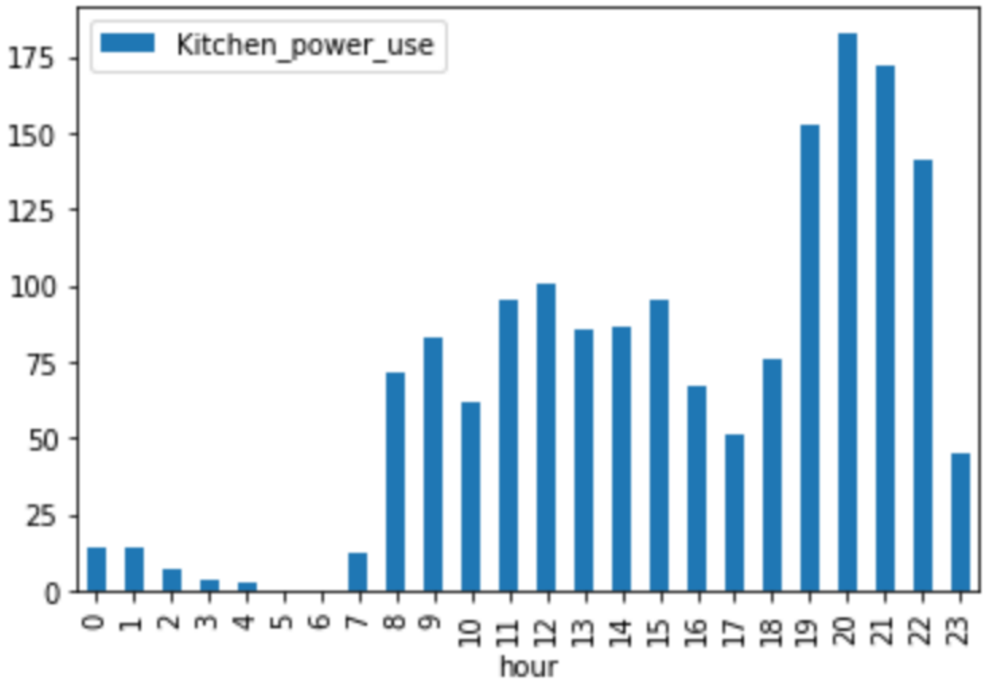
Out[55]:

	date	hour	Kitchen_power_use
20	2009-01-01	20	0.0
21	2009-01-01	21	0.0
22	2009-01-01	22	0.0
23	2009-01-01	23	0.0
24	2009-01-02	0	0.0
25	2009-01-02	1	0.0
26	2009-01-02	2	0.0
27	2009-01-02	3	0.0

Out[50]: <AxesSubplot:xlabel='hour'>



Out[56]: <AxesSubplot:xlabel='hour'>



Out[91]:

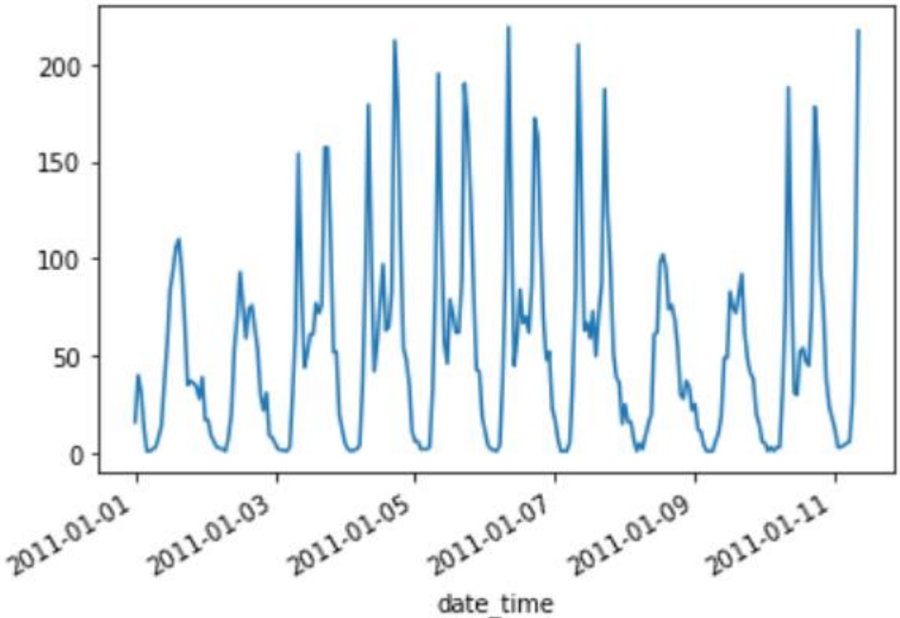
	date	hour	rentals
0	1/1/2011	0	16
1	1/1/2011	1	40
2	1/1/2011	2	32
3	1/1/2011	3	13
4	1/1/2011	4	1

Out[137]:

	date	hour	rentals	date_time
date_time				
2011-01-01 00:00:00	1/1/2011	0	16	1/1/2011 00:00:00
2011-01-01 01:00:00	1/1/2011	1	40	1/1/2011 01:00:00
2011-01-01 02:00:00	1/1/2011	2	32	1/1/2011 02:00:00
2011-01-01 03:00:00	1/1/2011	3	13	1/1/2011 03:00:00
2011-01-01 04:00:00	1/1/2011	4	1	1/1/2011 04:00:00
...	...	...	...	...
2012-12-31 19:00:00	12/31/2012	19	119	12/31/2012 19:00:00
2012-12-31 20:00:00	12/31/2012	20	89	12/31/2012 20:00:00
2012-12-31 21:00:00	12/31/2012	21	90	12/31/2012 21:00:00
2012-12-31 22:00:00	12/31/2012	22	61	12/31/2012 22:00:00
2012-12-31 23:00:00	12/31/2012	23	49	12/31/2012 23:00:00

17379 rows × 4 columns

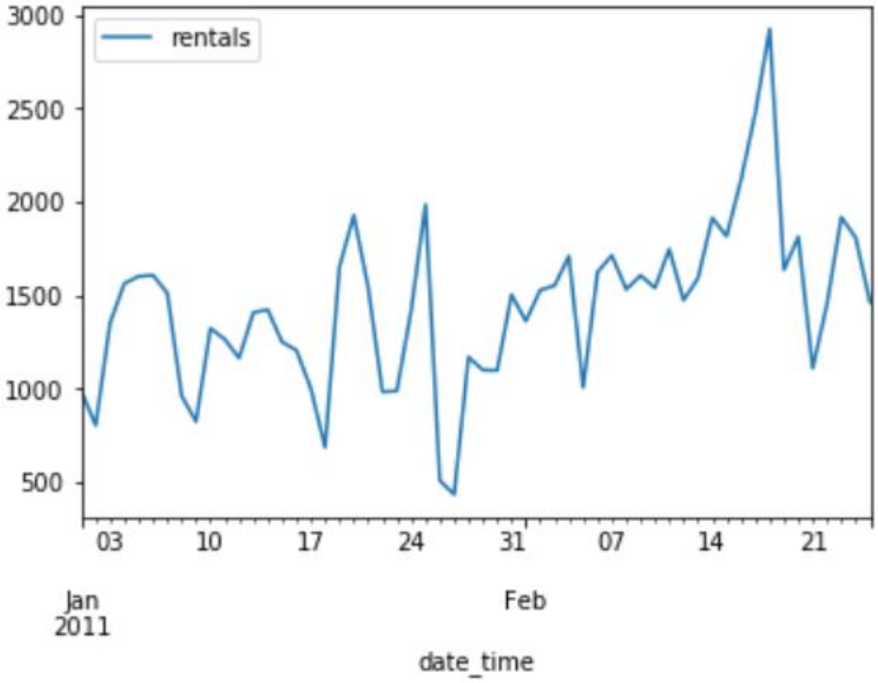
```
Out[94]: <matplotlib.axes._subplots.AxesSubplot at 0x25969241208>
```

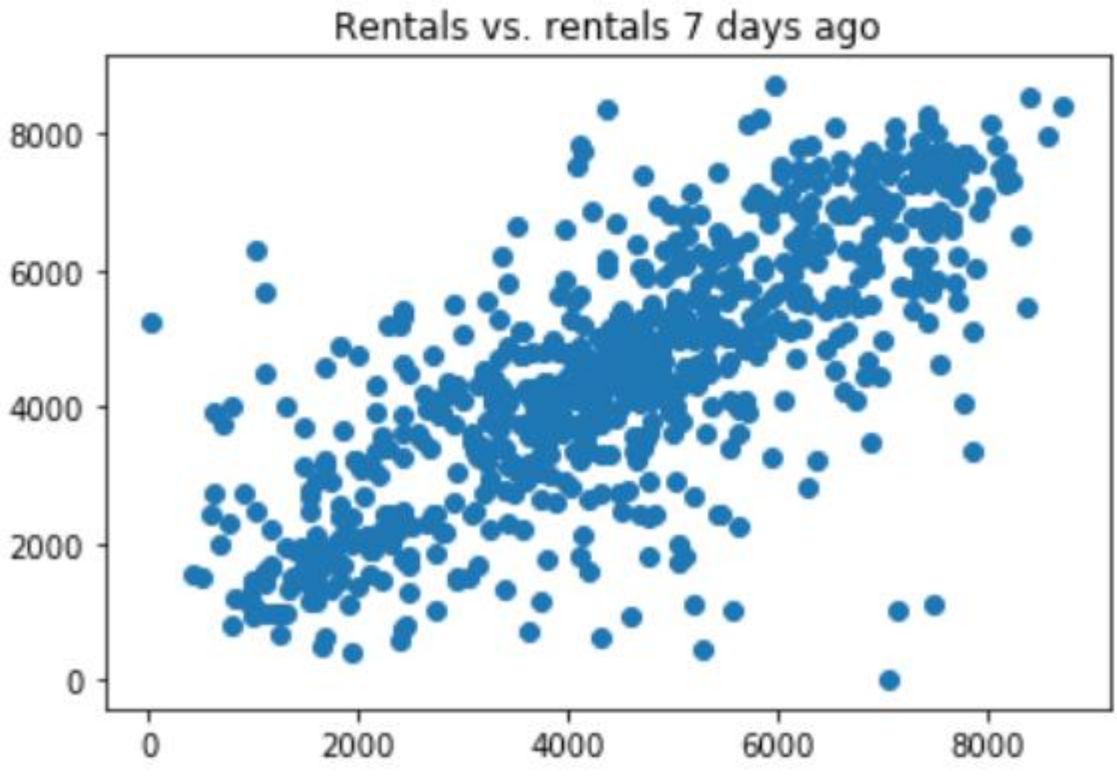


Out[95]:

	rentals
date_time	
2011-01-01	985
2011-01-02	801
2011-01-03	1349
2011-01-04	1562
2011-01-05	1600
2011-01-06	1606
2011-01-07	1510
2011-01-08	959
2011-01-09	822
2011-01-10	1321
2011-01-11	1263
2011-01-12	1162
2011-01-13	1406
2011-01-14	1421

```
Out[86]: <matplotlib.axes._subplots.AxesSubplot at 0x25968f5dac8>
```



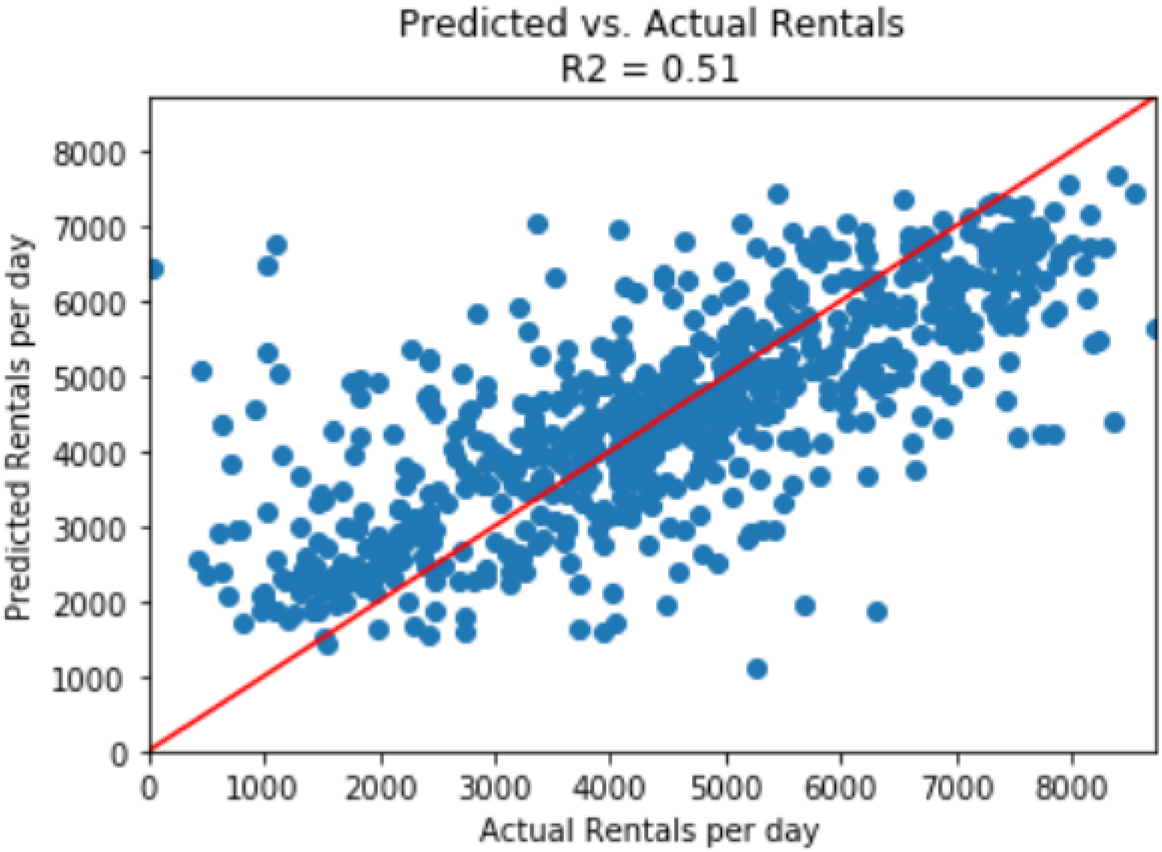


Out[109]:

date_time	rentals	lagged_rentals
2011-01-01	985	NaN
2011-01-02	801	NaN
2011-01-03	1349	NaN
2011-01-04	1562	NaN
2011-01-05	1600	NaN
...	...	...
2012-12-27	2114	4128.0
2012-12-28	3095	3623.0
2012-12-29	1341	1749.0
2012-12-30	1796	1787.0
2012-12-31	2729	920.0

731 rows × 2 columns

```
R2 is  0.5145071365683822  using:
            rentals  lagged_rentals
date_time
2011-01-08      959           985.0
2011-01-09      822           801.0
2011-01-10     1321          1349.0
2011-01-11     1263          1562.0
2011-01-12     1162          1600.0
```



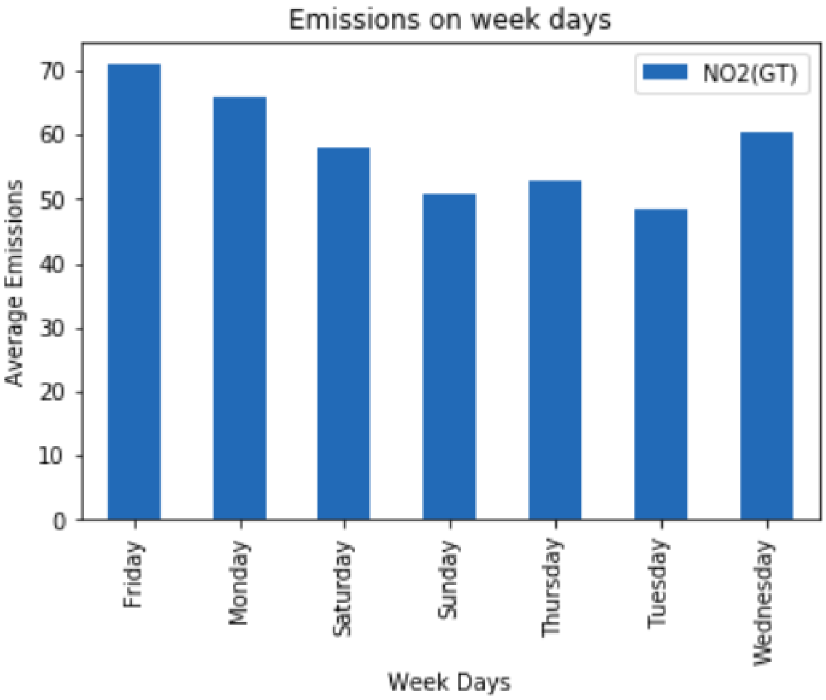
PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH	Unnamed: 15	Unnamed: 16
1360.0	150.0	11,9	1046.0	166.0	1056.0	113.0	1692.0	1268.0	13,6	48,9	0,7578	NaN	NaN
1292.0	112.0	9,4	955.0	103.0	1174.0	92.0	1559.0	972.0	13,3	47,7	0,7255	NaN	NaN
1402.0	88.0	9,0	939.0	131.0	1140.0	114.0	1555.0	1074.0	11,9	54,0	0,7502	NaN	NaN
1376.0	80.0	9,2	948.0	172.0	1092.0	122.0	1584.0	1203.0	11,0	60,0	0,7867	NaN	NaN
1272.0	51.0	6,5	836.0	131.0	1205.0	116.0	1490.0	1110.0	11,2	59,6	0,7888	NaN	NaN

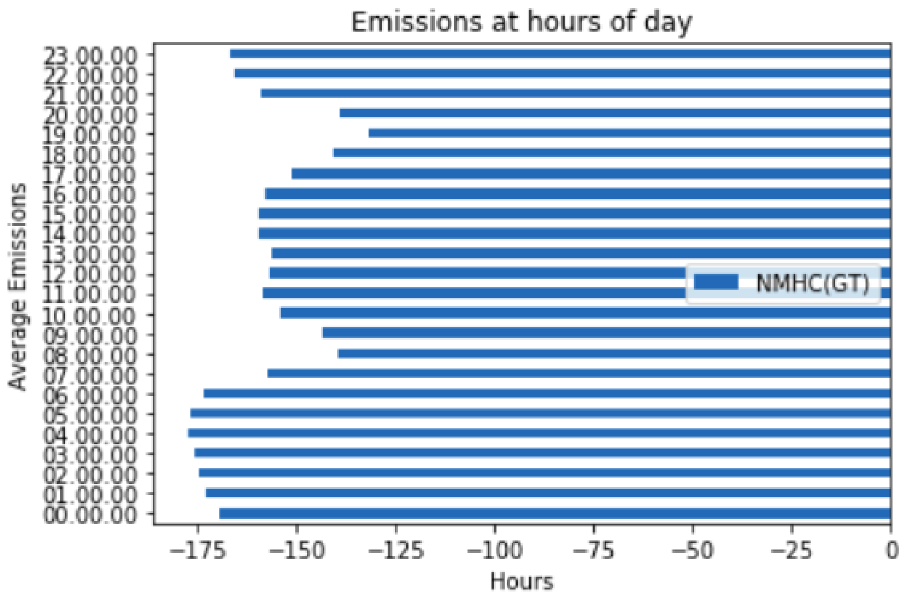
Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH
10/03/2004	18.00.00	2,6	1360.0	150.0	11,9	1046.0	166.0	1056.0	113.0	1692.0	1268.0	13,6	48,9	0,7578
10/03/2004	19.00.00	2	1292.0	112.0	9,4	955.0	103.0	1174.0	92.0	1559.0	972.0	13,3	47,7	0,7255
10/03/2004	20.00.00	2,2	1402.0	88.0	9,0	939.0	131.0	1140.0	114.0	1555.0	1074.0	11,9	54,0	0,7502
10/03/2004	21.00.00	2,2	1376.0	80.0	9,2	948.0	172.0	1092.0	122.0	1584.0	1203.0	11,0	60,0	0,7867
10/03/2004	22.00.00	1,6	1272.0	51.0	6,5	836.0	131.0	1205.0	116.0	1490.0	1110.0	11,2	59,6	0,7888

```
Int64Index: 9357 entries, 0 to 9356
Data columns (total 15 columns):
Date             9357 non-null object
Time             9357 non-null object
CO(GT)           9357 non-null object
PT08.S1(CO)      9357 non-null float64
NMHC(GT)         9357 non-null float64
C6H6(GT)         9357 non-null object
PT08.S2(NMHC)    9357 non-null float64
NOx(GT)          9357 non-null float64
PT08.S3(NOx)     9357 non-null float64
NO2(GT)          9357 non-null float64
PT08.S4(NO2)     9357 non-null float64
PT08.S5(O3)      9357 non-null float64
T                9357 non-null object
RH               9357 non-null object
AH               9357 non-null object
dtypes: float64(8), object(7)
```

MHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH	Parse_date	Weekday	Day	Month
150.0	11,9	1046.0	166.0	1056.0	113.0	1692.0	1268.0	13,6	48,9	0,7578	2004-10-03	6	Sunday	October
112.0	9,4	955.0	103.0	1174.0	92.0	1559.0	972.0	13,3	47,7	0,7255	2004-10-03	6	Sunday	October
88.0	9,0	939.0	131.0	1140.0	114.0	1555.0	1074.0	11,9	54,0	0,7502	2004-10-03	6	Sunday	October
80.0	9,2	948.0	172.0	1092.0	122.0	1584.0	1203.0	11,0	60,0	0,7867	2004-10-03	6	Sunday	October
51.0	6,5	836.0	131.0	1205.0	116.0	1490.0	1110.0	11,2	59,6	0,7888	2004-10-03	6	Sunday	October

	NO2(GT)
Day	
Friday	70.924851
Monday	65.771155
Saturday	58.110340
Sunday	50.844978
Thursday	52.760417
Tuesday	48.289394
Wednesday	60.279545





	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T
0	10/03/2004	18.00.00	2.6	1360.0	150.0	11,9	1046.0	166.0	1056.0	113.0	1692.0	1268.0	13,6
1	10/03/2004	19.00.00	2.0	1292.0	112.0	9,4	955.0	103.0	1174.0	92.0	1559.0	972.0	13,3
2	10/03/2004	20.00.00	2.2	1402.0	88.0	9,0	939.0	131.0	1140.0	114.0	1555.0	1074.0	11,9
3	10/03/2004	21.00.00	2.2	1376.0	80.0	9,2	948.0	172.0	1092.0	122.0	1584.0	1203.0	11,0
4	10/03/2004	22.00.00	1.6	1272.0	51.0	6,5	836.0	131.0	1205.0	116.0	1490.0	1110.0	11,2

Month	CO(GT)
April	-72.784898
August	-61.484274
December	-37.215495
February	-18.839943
January	-17.982552
July	-53.468952
June	-9.092500
March	-19.758170
May	-38.836290
November	-7.513172
October	-53.451467
September	-28.124722