Data analysis with R

for microbial oceanographers

Daniel Vaulot

2018/11/23 (updated: 2018-11-25)

Station Biologique de Roscoff, CNRS-Sorbonne Université

Tidy data

Initialize

Read the data

Manipulate columns

Manipulating rows

Joining tables

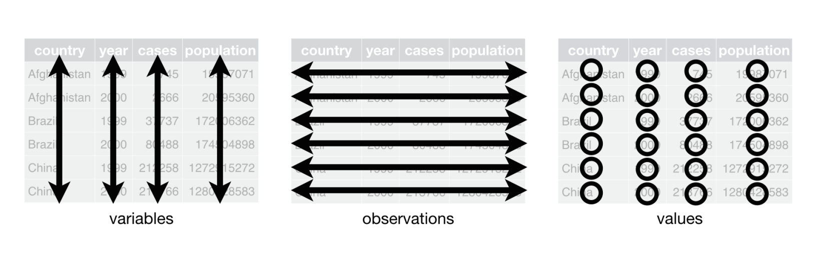
Displaying the data

Final words

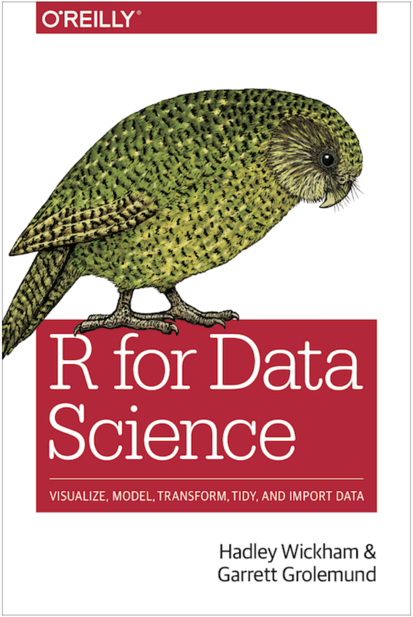
Tidy data

Concept of tidy data

1. Each variable must have its own column.
2. Each observation must have its own row.
3. Each value must have its own cell.



Suggested reading



O'REILLY

R for Data Science

VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA

Hadley Wickham &
Garrett Grolemund

Initialize

Load necessary libraries

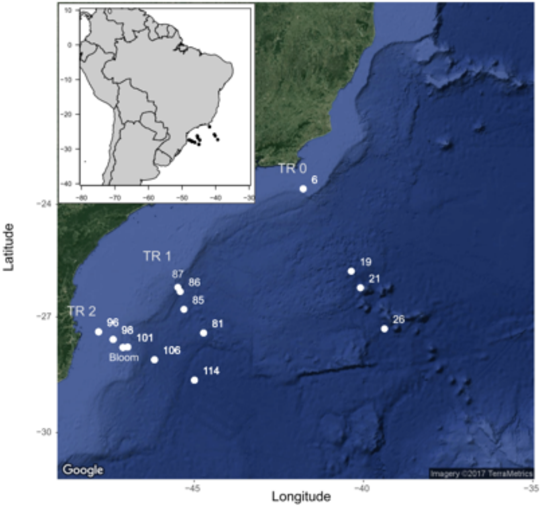
```r
library("readxl") # Import the data from Excel file
library("readr")  # Import the data from Excel file

library("dplyr")  # filter and reformat data frames
library("tidyr")  # make data tidy

library("ggplot2") # graphics
```

Read the data

CARBOM cruise



Read the data - read_excel

```r
samples <- readxl::read_excel("../data/CARBOM data.xlsx",
                              sheet = "Samples_boat")
```

sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	bottom_depth	level
10	1	81	140	-27.42	-44.72	3278	1232	2632	Deep
11	1	85	110	-26.80	-45.30	16312	1615	2100	Deep
120	2	96	5	-27.39	-47.82	1150	75	111	Surf
121	2		30	-27.39	-47.82	1737	218	111	Deep
122	2		50	-27.39	-47.82	853	234	111	Deep
125	2	98	5	-27.59	-47.39	3086	1300	265	Surf
126	2		50	-27.59	-47.39	1217	782	265	Deep
127	2		85	-27.59	-47.39	3420	226	265	Deep
13	1	86	105	-26.33	-45.41	6366	1007	1739	Deep
140	2	101	5	-27.79	-46.96	500	366	625	Surf

- Showing only the first 10 rows
- There are missing values in the column **station** because only recorded when changed

Filling missing values - fill

```
samples <- tidyr::fill(samples, station)
```

sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	bottom_depth	level
10	1	81	140	-27.42	-44.72	3278	1232	2632	Deep
11	1	85	110	-26.80	-45.30	16312	1615	2100	Deep
120	2	96	5	-27.39	-47.82	1150	75	111	Surf
121	2	96	30	-27.39	-47.82	1737	218	111	Deep
122	2	96	50	-27.39	-47.82	853	234	111	Deep
125	2	98	5	-27.59	-47.39	3086	1300	265	Surf
126	2	98	50	-27.59	-47.39	1217	782	265	Deep
127	2	98	85	-27.59	-47.39	3420	226	265	Deep
13	1	86	105	-26.33	-45.41	6366	1007	1739	Deep
140	2	101	5	-27.79	-46.96	500	366	625	Surf

All missing values have been filled in.

- To read text files
    - **readr::read_tsv()** : for tab delimited files
    - **readr::read_csv()** : for comma delimited files

Manipulate columns

dplyr

@allison_horst

List the columns

```
colnames(samples)
```

```
##  [1] "sample_number"     "transect"          "station"
##  [4] "depth"             "latitude"          "longitude"
##  [7] "picoeuks"          "nanoeuks"          "bottom_depth"
## [10] "level"             "transect_distance" "date"
## [13] "time"              "phosphates"        "silicates"
## [16] "ammonia"           "nitrates"          "nitrites"
## [19] "temperature"       "fluorescence"      "salinity"
```

Select specific columns - select

```r
samples_select <- dplyr::select(samples, sample_number, transect,
                                   station, depth, latitude, longitude,
                                   picoeuks, nanoeuks, level)
```

sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level
10	1	81	140	-27.42	-44.72	3278	1232	Deep
11	1	85	110	-26.80	-45.30	16312	1615	Deep
120	2	96	5	-27.39	-47.82	1150	75	Surf
121	2	96	30	-27.39	-47.82	1737	218	Deep
122	2	96	50	-27.39	-47.82	853	234	Deep
125	2	98	5	-27.59	-47.39	3086	1300	Surf
126	2	98	50	-27.59	-47.39	1217	782	Deep

Alternate syntax

```r
# Unselect some columns
  samples_select <- dplyr::select (samples, -bottom_depth, -transect_distance)
# Select a range of columns
  samples_select <- dplyr::select(samples, sample_number:nanoeuks, level)
```

- Note that column names are not “quoted”

Using the pipe operator - %>%

```
samples_select <- samples %>% dplyr::select(sample_number:nanoeuks, level)
```

sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level
10	1	81	140	-27.42	-44.72	3278	1232	Deep
11	1	85	110	-26.80	-45.30	16312	1615	Deep
120	2	96	5	-27.39	-47.82	1150	75	Surf
121	2	96	30	-27.39	-47.82	1737	218	Deep
122	2	96	50	-27.39	-47.82	853	234	Deep
125	2	98	5	-27.59	-47.39	3086	1300	Surf
126	2	98	50	-27.59	-47.39	1217	782	Deep
127	2	98	85	-27.59	-47.39	3420	226	Deep
13	1	86	105	-26.33	-45.41	6366	1007	Deep
140	2	101	5	-27.79	-46.96	500	366	Surf

- It is cleaner to write on 2 lines

```
samples_select <- samples %>%
  dplyr::select(sample_number:nanoeuks, level)
```

Creating new variables - mutate

```
samples_select <- samples_select %>%
  dplyr::mutate(pico_pct = picoeuks/(picoeuks+nanoeuks)*100)
```

sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level	pico_pct
10	1	81	140	-27.42	-44.72	3278	1232	Deep	72.68293
11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124
120	2	96	5	-27.39	-47.82	1150	75	Surf	93.87755
121	2	96	30	-27.39	-47.82	1737	218	Deep	88.84910
122	2	96	50	-27.39	-47.82	853	234	Deep	78.47286
125	2	98	5	-27.59	-47.39	3086	1300	Surf	70.36024
126	2	98	50	-27.59	-47.39	1217	782	Deep	60.88044
127	2	98	85	-27.59	-47.39	3420	226	Deep	93.80143
13	1	86	105	-26.33	-45.41	6366	1007	Deep	86.34206
140	2	101	5	-27.79	-46.96	500	366	Surf	57.73672

- You can also use **transmute()** but then it will drop all the other columns -> It is much much better to do all derivative operations in R than in Excel, because you can easily track and correct errors.

Using the pipe operator you can chain operations

```r
samples_select <- samples  %>%
  dplyr::select(sample_number:nanoeuks, level) %>%
  dplyr::mutate(pico_pct = picoeuks/(picoeuks+nanoeuks)*100)
```

sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level	pico_pct
10	1	81	140	-27.42	-44.72	3278	1232	Deep	72.68293
11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124
120	2	96	5	-27.39	-47.82	1150	75	Surf	93.87755
121	2	96	30	-27.39	-47.82	1737	218	Deep	88.84910
122	2	96	50	-27.39	-47.82	853	234	Deep	78.47286
125	2	98	5	-27.59	-47.39	3086	1300	Surf	70.36024
126	2	98	50	-27.59	-47.39	1217	782	Deep	60.88044
127	2	98	85	-27.59	-47.39	3420	226	Deep	93.80143
13	1	86	105	-26.33	-45.41	6366	1007	Deep	86.34206
140	2	101	5	-27.79	-46.96	500	366	Surf	57.73672

Creating labels with mutate

```
samples_select <- samples_select %>%
  dplyr::mutate(sample_label = paste0("TR",transect,"_St",station))
```

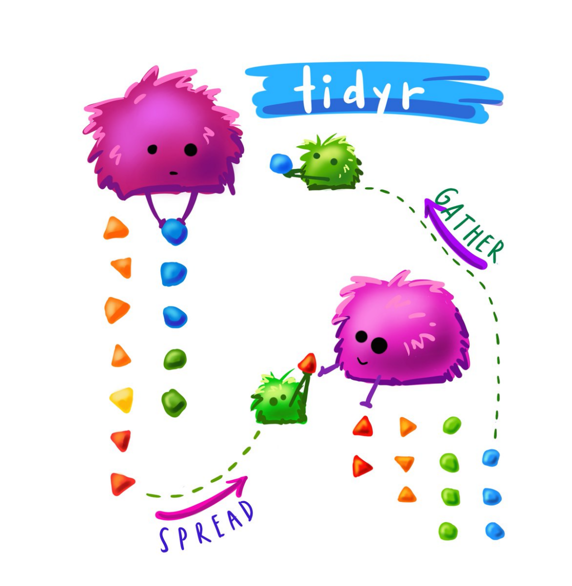
sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level	pico_pct	sample_label
10	1	81	140	-27.42	-44.72	3278	1232	Deep	72.68293	TR1_St81
11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124	TR1_St85
120	2	96	5	-27.39	-47.82	1150	75	Surf	93.87755	TR2_St96
121	2	96	30	-27.39	-47.82	1737	218	Deep	88.84910	TR2_St96
122	2	96	50	-27.39	-47.82	853	234	Deep	78.47286	TR2_St96
125	2	98	5	-27.59	-47.39	3086	1300	Surf	70.36024	TR2_St98
126	2	98	50	-27.59	-47.39	1217	782	Deep	60.88044	TR2_St98
127	2	98	85	-27.59	-47.39	3420	226	Deep	93.80143	TR2_St98
13	1	86	105	-26.33	-45.41	6366	1007	Deep	86.34206	TR1_St86
140	2	101	5	-27.79	-46.96	500	366	Surf	57.73672	TR2_St101

Rename specific columns - rename

```
samples_select <- samples_select %>%
  dplyr::rename(pico_percent = pico_pct)
```

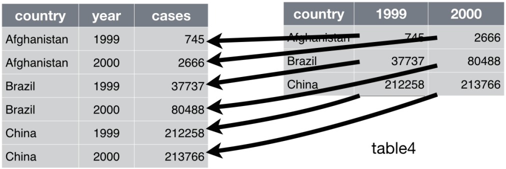
sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level	pico_percent
10	1	81	140	-27.42	-44.72	3278	1232	Deep	72.68293
11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124
120	2	96	5	-27.39	-47.82	1150	75	Surf	93.87755
121	2	96	30	-27.39	-47.82	1737	218	Deep	88.84910
122	2	96	50	-27.39	-47.82	853	234	Deep	78.47286
125	2	98	5	-27.59	-47.39	3086	1300	Surf	70.36024
126	2	98	50	-27.59	-47.39	1217	782	Deep	60.88044
127	2	98	85	-27.59	-47.39	3420	226	Deep	93.80143
13	1	86	105	-26.33	-45.41	6366	1007	Deep	86.34206
140	2	101	5	-27.79	-46.96	500	366	Surf	57.73672

Change table from wide to long form and conversely



@allison_horst

Go from wide to long - gather



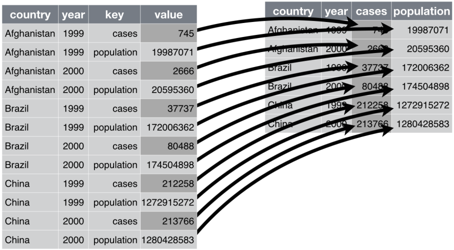
country	year	cases
Afghanistan	1999	745
Afghanistan	2000	2666
Brazil	1999	37737
Brazil	2000	80488
China	1999	212258
China	2000	213766

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

```
samples_long <- samples_select  %>%
  tidyr::gather(key="population", value="cell_ml", picoeuks, nanoeuks)
```

sample_number	transect	station	depth	latitude	longitude	level	pico_percent	population	cell_ml
10	1	81	140	-27.42	-44.72	Deep	72.68293	picoeuks	3278
11	1	85	110	-26.80	-45.30	Deep	90.99124	picoeuks	16312
120	2	96	5	-27.39	-47.82	Surf	93.87755	picoeuks	1150
121	2	96	30	-27.39	-47.82	Deep	88.84910	picoeuks	1737
122	2	96	50	-27.39	-47.82	Deep	78.47286	picoeuks	853
125	2	98	5	-27.59	-47.39	Surf	70.36024	picoeuks	3086

Go from long to wide - spread



country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

table2

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

```r
samples_wide <- samples_long  %>%
  tidyr::spread(key="population", value="cell_ml")
```

sample_number	transect	station	depth	latitude	longitude	level	pico_percent	nanoeuks	picoeuks
1	0	6	45	-23.58	-41.78	Deep	61.22759	4845	7651
10	1	81	140	-27.42	-44.72	Deep	72.68293	1232	3278
11	1	85	110	-26.80	-45.30	Deep	90.99124	1615	16312
120	2	96	5	-27.39	-47.82	Surf	93.87755	75	1150
121	2	96	30	-27.39	-47.82	Deep	88.84910	218	1737
122	2	96	50	-27.39	-47.82	Deep	78.47286	234	853

Manipulating rows

Order rows - arrange

```
samples_select <- samples_select %>%
  dplyr::arrange(transect, station)
```

sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level	pico_percent
3	0	19	5	-25.79	-40.36	1005	898	Surf	52.81135
5	0	21	5	-26.23	-40.09	793	660	Surf	54.57674
7	0	26	5	-27.31	-39.38	907	856	Surf	51.44640
1	0	6	45	-23.58	-41.78	7651	4845	Deep	61.22759
2	0	6	45	-23.58	-41.78	7343	3258	Deep	69.26705
10	1	81	140	-27.42	-44.72	3278	1232	Deep	72.68293
9	1	81	140	-27.42	-44.72	3181	1235	Deep	72.03351
11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124
13	1	86	105	-26.33	-45.41	6366	1007	Deep	86.34206
15	1	87	105	-26.22	-45.48	6189	622	Deep	90.86771

- ! Station 6 is not ordered numerically. It is because **station** is a character column.

Order rows - transform to numeric

```
samples_select <- samples_select %>%
  dplyr::mutate(station = as.numeric(station)) %>%
  arrange(transect, station)
```

```
## Warning in evalq(as.numeric(station), <environment>): NAs introduits lors
## de la conversion automatique
```

sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level	pico_percent
1	0	6	45	-23.58	-41.78	7651	4845	Deep	61.22759
2	0	6	45	-23.58	-41.78	7343	3258	Deep	69.26705
3	0	19	5	-25.79	-40.36	1005	898	Surf	52.81135
5	0	21	5	-26.23	-40.09	793	660	Surf	54.57674
7	0	26	5	-27.31	-39.38	907	856	Surf	51.44640
10	1	81	140	-27.42	-44.72	3278	1232	Deep	72.68293
9	1	81	140	-27.42	-44.72	3181	1235	Deep	72.03351
11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124
13	1	86	105	-26.33	-45.41	6366	1007	Deep	86.34206
15	1	87	105	-26.22	-45.48	6189	622	Deep	90.86771

- ! One station was named “Bloom” and then could not be converted to numerical (-> NA)

Summarize rows - group_by / summarize

- Group by transect and station
- Compute mean of the percent picoplankton

```
samples_mean <- samples_select %>%
  dplyr::group_by(transect, station, level) %>%
  dplyr::summarise(n_samples = n(),
            mean_pico_percent = mean(pico_percent, na.rm=TRUE))
```

transect	station	level	n_samples	mean_pico_percent
0	6	Deep	2	65.24732
0	19	Surf	1	52.81135
0	21	Surf	1	54.57674
0	26	Surf	1	51.44640
1	81	Deep	2	72.35822
1	85	Deep	1	90.99124
1	86	Deep	1	86.34206
1	87	Deep	1	90.86771

Filtering rows - filter

- Get only the surface samples

```
samples_surf <- samples_select %>%
  dplyr::filter(level == "Surf" )
```

sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level	pico_percent
3	0	19	5	-25.79	-40.36	1005	898	Surf	52.81135
5	0	21	5	-26.23	-40.09	793	660	Surf	54.57674
7	0	26	5	-27.31	-39.38	907	856	Surf	51.44640
120	2	96	5	-27.39	-47.82	1150	75	Surf	93.87755
125	2	98	5	-27.59	-47.39	3086	1300	Surf	70.36024
140	2	101	5	-27.79	-46.96	500	366	Surf	57.73672
155	2	106	5	-28.12	-46.17	355	18	Surf	95.17426
165	2	114	5	-28.65	-44.99	728	226	Surf	76.31027
Trichod.1	2		0	-27.80	-47.10	1002	194	Surf	83.77926
Trichod.2	2		0	-27.80	-47.10	744	206	Surf	78.31579

- ! Use the logical operators == != > >= < <= is.na()

Joining tables

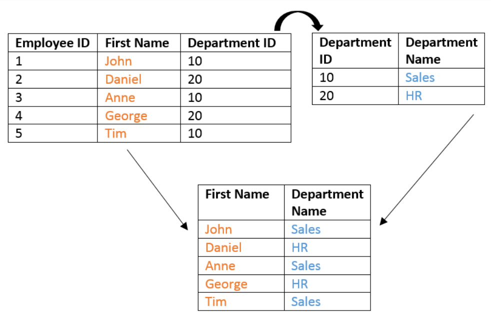
Joining tables

Very often you have tables that contain a common field and that you need to **join** together. A common example in oceanography

- Station - Longitude, Latitude
- CTD - Several CTD per station
- CTD profile - Parameter values at different depth
- Bottles
- Samples

In order to join 2 tables, they must have a common field. It is called the **KEY**.

For example it can be station number or sample_number



Employee ID	First Name	Department ID
1	John	10
2	Daniel	20
3	Anne	10
4	George	20
5	Tim	10

Department ID	Department Name
10	Sales
20	HR

First Name	Department Name
John	Sales
Daniel	HR
Anne	Sales
George	HR
Tim	Sales

Reading table with medtabarcoding samples

```
metabarcodes <- readxl::read_excel("../data/CARBOM data.xlsx",
                                   sheet = "Samples_metabarcodes")
```

Table metabarcodes

For metabarcoding, each sample has been split into 2 fractions by sorting : pico- and nano-.

sample	fraction	Select_18S_nifH	total_18S	total_16S	total_nifH	sample_number
X1n	Nano	Yes	95054	9139	163	1
X1p	Pico	Yes	19466	15987	137117	1
X10n	Nano	Yes	53230	8772	36	10
X10p	Pico	Yes	47390	4448	6241	10
X11n	Nano	No	24007	6193	3772	11

Tables samples

sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level	pico_percent
1	0	6	45	-23.58	-41.78	7651	4845	Deep	61.22759
10	1	81	140	-27.42	-44.72	3278	1232	Deep	72.68293
11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124
120	2	96	5	-27.39	-47.82	1150	75	Surf	93.87755
121	2	96	30	-27.39	-47.82	1737	218	Deep	88.84910

- The two tables have a common field called **sample_number** (KEY).

Joining metabarcode and sample tables.

```
metabarcodes_join <- left_join(metabarcodes, samples_select)
```

```
## Joining, by = "sample_number"
```

sample	fraction	Select_18S_nifH	total_18S	total_16S	total_nifH	sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level	pico_percent
X1n	Nano	Yes	95054	9139	163	1	0	6	45	-23.58	-41.78	7651	4845	Deep	61.22759
X1p	Pico	Yes	19466	15987	137117	1	0	6	45	-23.58	-41.78	7651	4845	Deep	61.22759
X10n	Nano	Yes	53230	8772	36	10	1	81	140	-27.42	-44.72	3278	1232	Deep	72.68293
X10p	Pico	Yes	47390	4448	6241	10	1	81	140	-27.42	-44.72	3278	1232	Deep	72.68293
X11n	Nano	No	24007	6193	3772	11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124
X11p	Pico	Yes	31899	14	10201	11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124
X120n	Nano	Yes	70455	5292	93	120	2	96	5	-27.39	-47.82	1150	75	Surf	93.87755
X120p	Pico	Yes	76182	53272	23147	120	2	96	5	-27.39	-47.82	1150	75	Surf	93.87755
X121n	Nano	Yes	52401	5958	26838	121	2	96	30	-27.39	-47.82	1737	218	Deep	88.84910
X121p	Pico	Yes	71785	10993	23706	121	2	96	30	-27.39	-47.82	1737	218	Deep	88.84910

Joining metabarcode and sample tables.

- If the **KEY** do not have the same name in the two tables it is possible to specify the name of the two columns used for joining.

```
metabarcodes <- metabarcodes %>%
  rename(sample_code = sample_number)
```

sample	fraction	Select_18S_nifH	total_18S	total_16S	total_nifH	sample_code
X1n	Nano	Yes	95054	9139	163	1
X1p	Pico	Yes	19466	15987	137117	1
X10n	Nano	Yes	53230	8772	36	10
X10p	Pico	Yes	47390	4448	6241	10
X11n	Nano	No	24007	6193	3772	11

```
metabarcodes_join <- left_join(metabarcodes, samples_select,
                               by= c("sample_code" = "sample_number"))
```

sample	fraction	Select_18S_nifH	total_18S	total_16S	total_nifH	sample_code	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level	pico_percent
X1n	Nano	Yes	95054	9139	163	1	0	6	45	-23.58	-41.78	7651	4845	Deep	61.22759
X1p	Pico	Yes	19466	15987	137117	1	0	6	45	-23.58	-41.78	7651	4845	Deep	61.22759
X10n	Nano	Yes	53230	8772	36	10	1	81	140	-27.42	-44.72	3278	1232	Deep	72.68293
X10p	Pico	Yes	47390	4448	6241	10	1	81	140	-27.42	-44.72	3278	1232	Deep	72.68293
X11n	Nano	No	24007	6193	3772	11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124
X11p	Pico	Yes	31899	14	10201	11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124
X120n	Nano	Yes	70455	5292	93	120	2	96	5	-27.39	-47.82	1150	75	Surf	93.87755
X120p	Pico	Yes	76182	53272	23147	120	2	96	5	-27.39	-47.82	1150	75	Surf	93.87755
X121n	Nano	Yes	52401	5958	26838	121	2	96	30	-27.39	-47.82	1737	218	Deep	88.84910
X121p	Pico	Yes	71785	10993	23706	121	2	96	30	-27.39	-47.82	1737	218	Deep	88.84910

Joining with missing data

```r
samples_select <- samples_select %>%
  filter(sample_number != "10")
```

sample_number	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level	pico_percent
1	0	6	45	-23.58	-41.78	7651	4845	Deep	61.22759
11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124
120	2	96	5	-27.39	-47.82	1150	75	Surf	93.87755
121	2	96	30	-27.39	-47.82	1737	218	Deep	88.84910
122	2	96	50	-27.39	-47.82	853	234	Deep	78.47286

```r
metabarcodes_join <- left_join(metabarcodes, samples_select,
                               by= c("sample_code" = "sample_number"))
```

sample	fraction	Select_18S_nifH	total_18S	total_16S	total_nifH	sample_code	transect	station	depth	latitude	longitude	picoeuks	nanoeuks	level	pico_percent
X1n	Nano	Yes	95054	9139	163	1	0	6	45	-23.58	-41.78	7651	4845	Deep	61.22759
X1p	Pico	Yes	19466	15987	137117	1	0	6	45	-23.58	-41.78	7651	4845	Deep	61.22759
X10n	Nano	Yes	53230	8772	36	10									
X10p	Pico	Yes	47390	4448	6241	10									
X11n	Nano	No	24007	6193	3772	11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124
X11p	Pico	Yes	31899	14	10201	11	1	85	110	-26.80	-45.30	16312	1615	Deep	90.99124
X120n	Nano	Yes	70455	5292	93	120	2	96	5	-27.39	-47.82	1150	75	Surf	93.87755
X120p	Pico	Yes	76182	53272	23147	120	2	96	5	-27.39	-47.82	1150	75	Surf	93.87755
X121n	Nano	Yes	52401	5958	26838	121	2	96	30	-27.39	-47.82	1737	218	Deep	88.84910
X121p	Pico	Yes	71785	10993	23706	121	2	96	30	-27.39	-47.82	1737	218	Deep	88.84910

Displaying the data

ggplot2

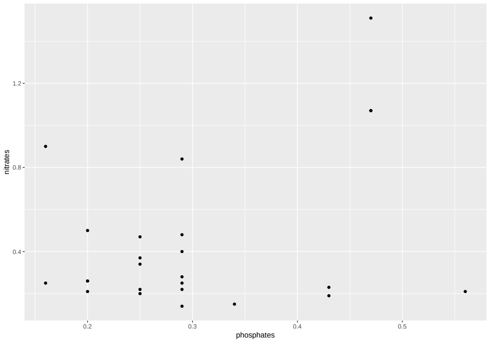
@allison_horst

A simple plot

- Choose the data set
- Choose the geometric representation
- Choose the **aesthetics** : x,y, color, shape etc. . .

```
ggplot(samples) +
 geom_point(mapping = aes(x=phosphates, y=nitrates))
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

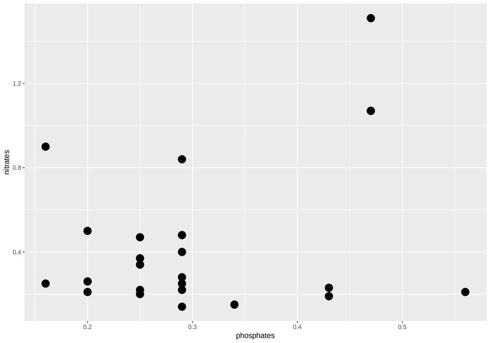


- All functions are from **ggplot2** package unless specified

Make dot size bigger

- Add: **size=5**

```
ggplot(samples) +
 geom_point(mapping = aes(x=phosphates, y=nitrates), size=5)
```

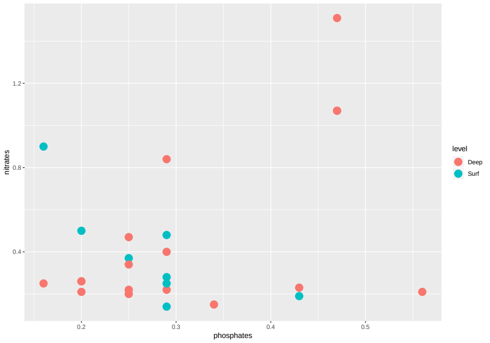


Color according to depth level

- Add: **color=level**

```
ggplot(samples) +
 geom_point(mapping = aes(x=phosphates, y=nitrates,color=level), size=5)
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

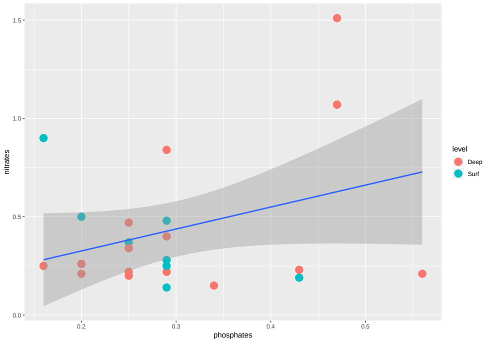


- The mapping aesthetics must be an argument of the aes function:
- geom_point(mapping = aes(x=phosphates, y=nitrates), **color=level**, size=5) will generate an error. . .

Adding a regression line

- Add: **geom_smooth()**

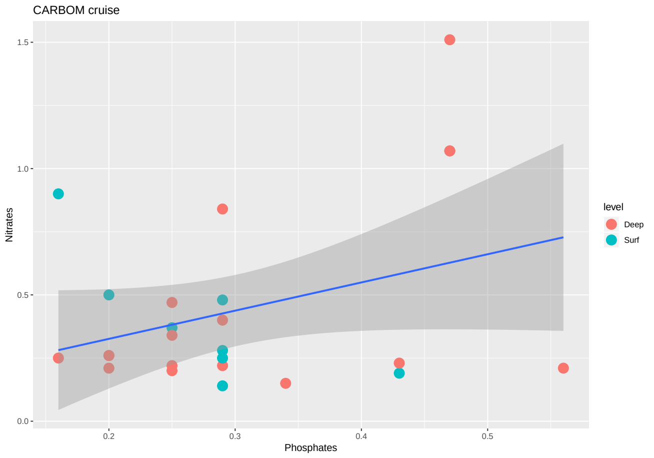
```
ggplot(samples) +
 geom_point(mapping = aes(x=phosphates, y=nitrates,color=level), size=5) +
 geom_smooth(mapping = aes(x=phosphates, y=nitrates), method="lm")
```



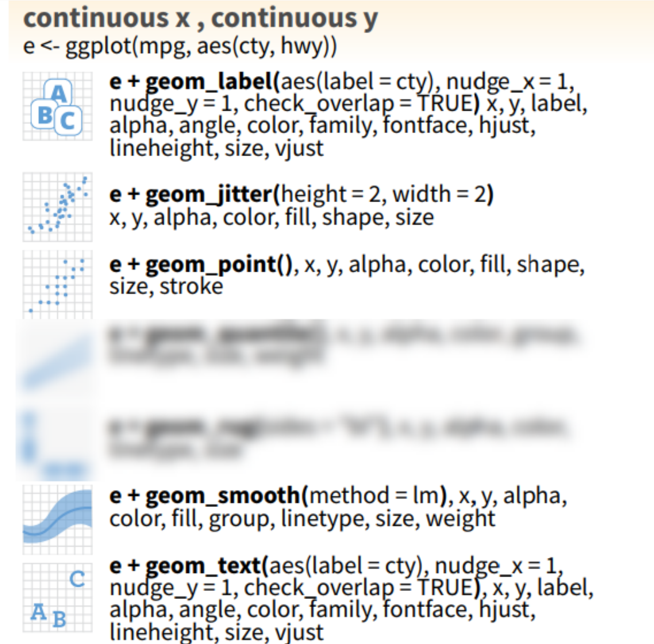
- You can choose the type of smoothing “lm” is for linear model

Finalizing the graph

```
ggplot(samples) +
 geom_point(mapping = aes(x=phosphates, y=nitrates,color=level), size=5) +
 geom_smooth(mapping = aes(x=phosphates, y=nitrates), method="lm") +
 xlab("Phosphates") + ylab("Nitrates") + ggtitle("CARBOM cruise")
```



A lot of geom - Continuous x and y



continuous x , continuous y
e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE**)** x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_jitter(height = 2, width = 2**)** x, y, alpha, color, fill, shape, size

e + geom_point(), x, y, alpha, color, fill, shape, size, stroke

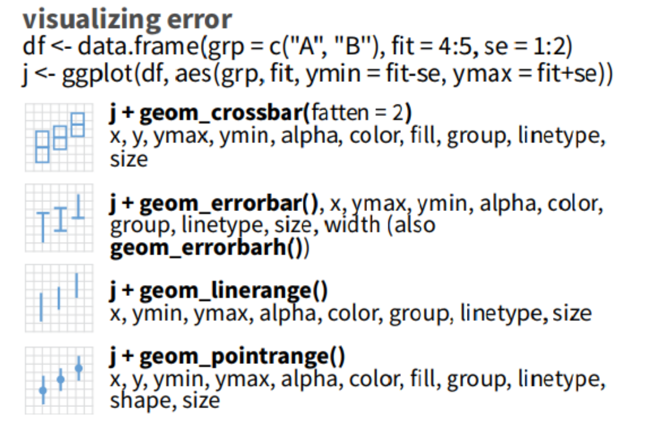
e + geom_quantile(), x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl"**)**, x, y, alpha, color, linetype, size

e + geom_smooth(method = lm**)**, x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE**)**, x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

A lot of geom - Plotting error



visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))

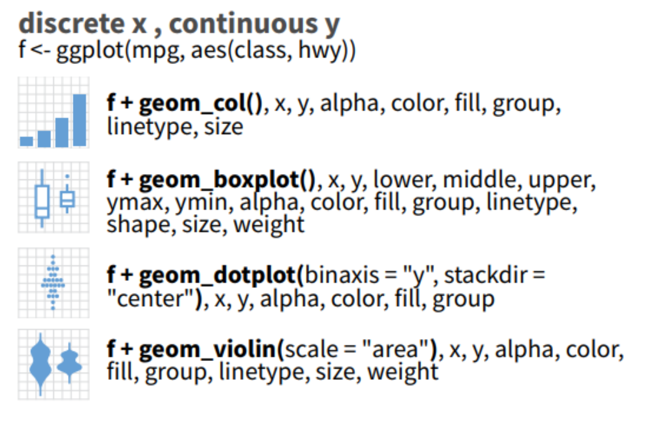
j + geom_crossbar(fatten = 2**)**
x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar(), x, ymax, ymin, alpha, color, group, linetype, size, width (also **geom_errorbarh()**)

j + geom_linerange()
x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange()
x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

A lot of geom - Discrete x - Continuous y



discrete x , continuous y
f <- ggplot(mpg, aes(class, hwy))

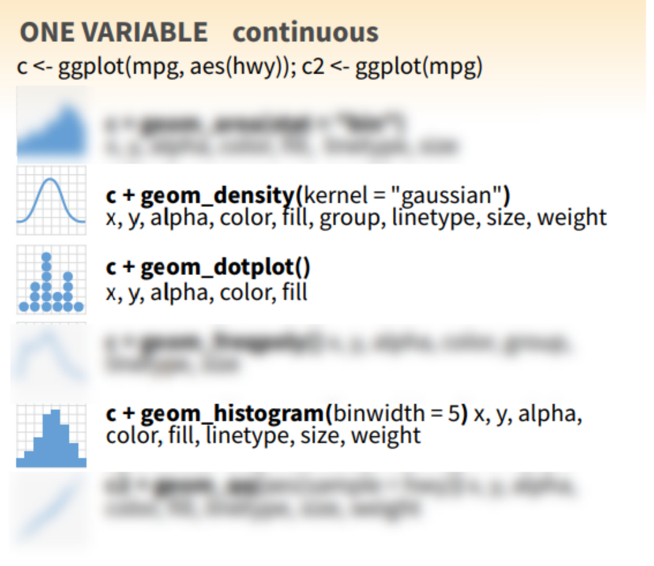
f + geom_col(), x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot(), x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = "center"**)**, x, y, alpha, color, fill, group

f + geom_violin(scale = "area"**)**, x, y, alpha, color, fill, group, linetype, size, weight

A lot of geom - Continuous x



ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

[illegible]

c + geom_density(kernel = "gaussian"**)**
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

[illegible]

c + geom_histogram(binwidth = 5**)** x, y, alpha,
color, fill, linetype, size, weight

[illegible]

Final words

Other useful packages

stringr - manipulate strings

- str_c: concatenate strings (cf paste and paste0)
- str_detect: to find a specific string
- str_replace: to replace a string

lubridate - manipulate date

tibble - manipulate data frame

- e.g. row names -> column or reverse

Useful links

- R for data science: https://r4ds.had.co.nz/
- R graph gallery: https://www.r-graph-gallery.com/
- Dplyr manipulating tables: https://suzan.rbind.io/2018/01/dplyr-tutorial-1/