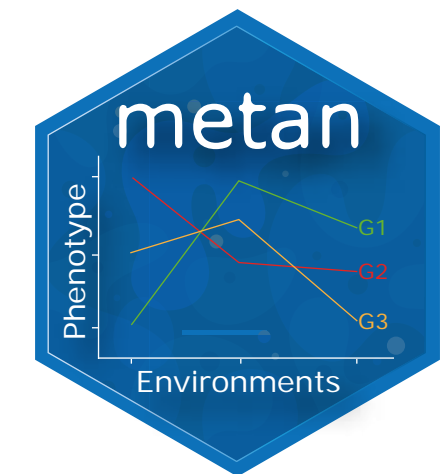


Multi-environment trial analysis with R package metan: CHEAT SHEET



DATA MANIPULATION

Handling with strings Translate all non-numeric strings of a data frame to lower, upper, or title case, remove blank spaces, remove, replace and tidy up strings.

```
messy_text <- c("ENV 1", "Env 1", "env_1")
all_upper_case(messy_text) # "ENV 1" "ENV 1" "ENV 1"
all_lower_case(messy_text) # "env 1" "env 1" "env 1"
all_title_case(messy_text) # "Env 1" "Env 1" "Env 1"
extract_string(messy_text) # "ENV" "Env" "env"
remove_space(messy_text) # "ENV1" "Env1" "env_1"
remove_strings(messy_text) # 1 1 1
tidy_strings(messy_text) # "ENV_1" "ENV_1" "ENV_1"
```

All of them works with data frames and the forward-pipe operator %>%

```
df <- data.frame(var = messy_text)
all_upper_case(df) tidy_strings(df, sep="") %>% remove_space(df) %>%
  all_title_case() all_lower_case()
var      var      var
1 ENV_1 1 Env1    1 env1
2 ENV_1 2 Env1    2 env1
3 ENV_1 3 Env1    3 env1
```

Handling with numbers Round a whole data frame to significant figures, find text fragments in numeric variables and extract numbers of a string.

```
df <- data.frame(var = c(1, 20.12372))
df[3, 1] <- "2m.503"
find_text_in_num(df, var) extract_number(df, var, drop = TRUE) %>%
  round_number()
# [1] 2 20 503
# [1] 1 20.12372 2.503
```

Handling with columns Add, select, remove, concatenate, and reorder columns easily.

```
df <- data.frame(env = "E", gen = "G", ear = 2.12, spiklet = 3.02)
add_one_or_more_columns_to_an_existing_data_frame_if_specified_before_or_after_columns_does_not_exist_columns_are_appended_at_the_end_of_the_data_is_possible_to_add_a_column_based_on_existing_data
```

```
add_cols(df, mult = ear * spiklet, mult_2 = mult ^ 2, ,after = "gen")
env gen mult mult_2 ear spiklet
1 E G 6.40 41.0 2.12 3.02
```

Select variables based on their names or length of names

```
select(df, env) select(df, contains("k")) select(df, width_of(3))
env spiklet
1 3.02
env gen ear
1 E G 2.12
```

Select numeric and non-numeric columns

```
select_numeric_cols(df) select_non_numeric_cols(df)
ear spiklet
1 2.12 3.02
1 E G 2.12 3.02
```

Select first or last variable, possibly with an offset.

```
select_first_col(df) select_last_col(df) select_last_col(df, offset = 1)
env spiklet
1 3.02
env gen ear
1 E G 2.12 3.02
```

Remove columns

```
remove_cols(df, ear) remove_cols(df, width_greater_than(3))
env gen spiklet new_var
1 E G 3.02 2.12
```

Concatenate columns

```
concatenate(df, env, gen) concatenate(df, env, gen, pull = TRUE)
env gen ear spiklet new_var
1 E G 2.12 3.02 E G 2.12
```

Reorder columns

```
reorder_cols(df, env, gen, ,after = "spiklet") column_to_first(df, ear")
ear spiklet env gen
1 3.02 2.12 E G
```

column_to_last(df, env, ear) column_to_last(df, starts_with("e"))

```
gen spiklet env ear
1 G 3.02 E 2.12
```

Handling with NA values Check for NA values, remove rows or columns with NA and replace NA quickly.

```
df_na <- data.frame(a = c(1, 3, NA), b = c(NA, 3, 2))
remove_cols_na(df_na) remove_rows_na(df_na) replace_na(df_na)
a b
1 1 0
2 2 3
3 3 2
```

Handling with matrices make upper, lower or symmetric matrices quick.

```
mat <- matrix(1:9, nrow = 3, ncol = 3)
make_sym(mat) make_lower_tri(mat) make_upper_tri(mat, diag = 0)
[1] [2] [3] [1] [2] [3]
[1] NA NA NA [1] 0 4 7
[2] 2 NA 6 [2] 2 NA NA [2] NA 0 8
[3] 3 6 NA [3] 3 6 NA [3] NA NA 0
```

Handling with 'long' and 'wide' data Convert a 'long' data format into a two-way table and vice-versa

```
df <- data.frame(expand.grid(ENV = c("E1", "E2", "E3"),
  GEN = c("G1", "G2"))) %>% add_cols(Y = 1:6)
mat <- make_mat(df, GEN, ENV, Y) make_long(mat)
E1 E2 E3 1 1 0
G1 1 2 3 2 1 0
G2 4 5 6 3 2 3
G3 2 4 5 6 3 2
```

CHECK DATA

Inspect data graphically multi-environment trial data are expected to be balanced (all genotypes in all environments), with factor columns for environment, genotype and replication and numeric variables with no missing values or outliers.

```
inspect(data_ge) inspect(data_ge, ENV, GY, HM, plot = TRUE)
Variable Class Missing Levels
1 ENV factr No 14
2 GEN factr No 10
3 REP factr No 3
4 GY num~ No -
5 HM num~ No -
# ... with 5 more variables:
# Valid n<int>, Min<dbl>,
# Median<dbl>, Max<dbl>,
# Outlier<dbl>
```

Find outliers find possible outliers in a data set

```
find_outliers(data_ge2, EL) find_outliers(data_ge2, EL, plots = TRUE)
Number of possible outliers: 1
Lines: 137
Proportion: 0.6%
Mean of the outliers: 11.5
Maximum of the outliers: 11.5 | Line 137
Minimum of the outliers: 11.5 | Line 137
With outliers: mean = 15.163 | CV = 8.284%
Without outliers: mean = 15.187 | CV = 8.065%
```

DESCRIPTIVE STATISTICS

Common statistics type the function you want, the dataset, and it will take care of the details!

```
av_dev() - avg. absolute deviation
ci_mean() - confidence interval
cv() - coefficient of variation
freq_table() - frequency table
hmean() - harmonic mean
gmean() - geometric mean
kurt() - kurtosis like used in SAS
range_data() - range of the data
sd_amo() - sample stand. dev.
sd_pop() - population stand. dev.
sem() - standard error of the mean
skew() - skewness like used in SAS
sum_dev() - sum of the absolute dev.
sum_sq_dev() - sum of the squared dev.
var_amo() - sample variance
var_pop() - population variance
valid_n() - valid (not NA) length of a data
```

```
Summarise a whole dataset, possibly with variable selection
cv(data_ge) cv(data_ge, GY) freq_table(data_ge2, NR)
GY HM 1 34.6 9.09 GY 1 34.6
1 12.4 1 0.00641 0.00641
2 13.2 3 0.0192 0.0256
# ... with 18 more rows
```

Summarise a numeric vector

```
set.seed(1)
vctr <- rnorm(100, 400, 30)
mean(vctr) # 403.27 sd_amo(vctr) # 26.94 skew(vctr) # -0.0733
gmean(vctr) # 402.37 hmean(vctr) # 401.46 kurt(vctr) # 0.0705
hmean(vctr) # 401.46
```

The wrapper function desc_stat() computes the most used measures of central tendency, position, and dispersion at once!

```
Common statistics for all numeric variables of a data set
desc_stat(data_ge) desc_stat(data_ge, stats = "mean")
variable cv max mean median min sd.amo se ci variable mean
1 GY 34.6 5.09 2.67 2.61 0.671 0.924 0.0451 0.0886 1 GY 2.67
2 HM 9.09 58 48.1 48 38 4.37 0.213 0.419 2 HM 48.1
```

```
desc_stat(data_ge, stats = c("mean, gmean, hmean"), hist = TRUE)
variable mean gmean hmean
1 GY 2.67 2.50 2.32
2 HM 48.1 47.9 47.7
```

Statistics by levels of a factor

```
desc_stat(data_ge, by = ENV, stats = c("mean, se"))
ENV variable mean se
1 E1 GY 2.52 0.0814
2 E1 HM 47.4 0.315
3 E10 GY 2.18 0.0715
4 E10 HM 44.3 0.603
5 E11 GY 1.37 0.0678
# ... with 23 more rows
```

ONE-WAY ANOVA

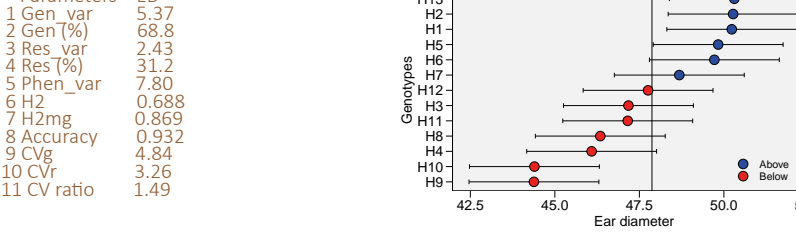
Fixed-effect models Computes individual ANOVA in multi-environment trials and analyzes genotype trials using fixed-effect models.

```
anova_ind(data, env, gen, rep, resp, block = NULL) - Computes within-environment analysis of variance and Post-hoc test.
gafem(data, gen, rep, resp, prob = 0.05, block = NULL) - One-way analysis of variance of genotypes conducted in RCBD and alpha-lattice designs.
```

Mixed-effect models Variance components, genetic parameters, random effects and BLUP prediction.

gamem(data, gen, rep, resp, prob = 0.05, block = NULL) - Analysis of genotypes in single trials using mixed-effect models

```
mod <- gafem(data_g, GEN, REP, ED)
gmd(mod, "genpar")
Parameters ED
1 Gen_var 5.37
2 Gen(%) 68.8
3 Res_var 2.43
4 Res(%) 31.2
5 Phen_var 7.80
6 H2 0.698
7 H2mg 0.869
8 Accuracy 0.932
9 CVg 4.84
10 CVg 3.26
11 CV ratio 1.49
```



TWO-WAY ANOVA

Fixed-effect models Computes a joint-ANOVA of in multi-environment trials using fixed-effect models

```
anova_joint(data, env, gen, rep, resp, block = NULL, verbose = TRUE)
Computes a joint-analysis of variance using fixed-effect models
mod <- anova_joint(data_ge, ENV, GEN, REP, GY) gmd(mod, "details")
Joint ANOVA table Parameters GY
Source Df Sum Sq Mean Sq F value Pr(>F)
1 ENV 13 279.6 21.506 62.3 0.0e+00
2 REP(ENV) 28 9.7 0.345 3.6 3.6e-08
3 GEN 9 13.0 1.444 14.9 2.2e-19
4 GEN:ENV 117 31.2 0.267 2.8 1.0e-11
5 Residuals 252 24.4 0.097 NA NA
6 CV(%) 11.6 NA NA NA NA
7 MSR/MSR- 6.7 NA NA NA NA
8 OVRmean 2.7 NA NA NA NA
10 MaxGEN "G8 (3)"
```

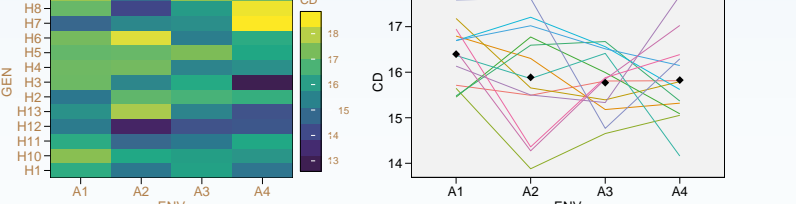
Mixed-effect models Variance components, genetic parameters, random effects and BLUP prediction in multi-environment trials

```
gamem(data, env, gen, rep, resp, block = NULL, random = "gen",
  prob = 0.05, verbose = TRUE) - Genotype analysis in multi-environment trials
using mixed-effect or random-effect models.
mod <- gamem_met(data_ge, ENV, GEN, REP, GY, random = "all")
gmd(mod, "vcomp") gmd(mod, "pval_lrt") gmd(mod, "ranef")
Group GY model GY GEN GY
1 ENV 0.700 1 COMPLETE NA 1 G1 -0.0575
2 GEN 0.0280 2 GEN 1.11e- 5 2 G2 0.0570
3 GEN:ENV 0.0567 3 REP(ENV) 9.91e- 8 3 G3 0.279
4 REP:ENV 0.0248 4 ENV 8.26e-17 4 G4 -0.0264
5 Residual 0.0967 5 GEN:ENV 2.15e-11 # ... with 6 more rows
```

GE INTERACTION

Graphical approach genotype's performance across environments shown in line plots and heatmaps

```
ge_plot(data, env, gen, resp, type = 1) - Line plot and heatmap
ge_plot(data_ge2, ENV, GEN, CD) ge_plot(data_ge2, ENV, GEN, CD, type = 2)
```



Numerical approach genotype-environment means and effects, genotype-plus genotype-environment effects, and winner genotypes.

```
Genotype-environment details
ge_details(data, env, gen, resp) - Details for the multi-environment trial data
ge_details(data_ge, ENV, GEN, everything())
Parameters GY HM
1 Mean "2.67" "48.09"
2 SE "0.05" "0.21"
3 SD "0.37" "4.37"
4 CV "34.56" "9.09"
5 Min "0.67 (G10 in E11)" "38 (G2 in E14)"
6 Max "5.09 (G8 in E5)" "58 (G8 in E11)"
7 MinENV "E11 (1.37)" "E14 (41.03)"
8 MaxENV "E3 (4.06)" "E11 (54.2)"
9 MinGEN "G10 (2.47)" "G2 (46.66)"
10 MaxGEN "G8 (3)" "G5 (49.3)"
```

Genotype-environment winners

```
ge_winners(data, env, gen, resp, type = "winners", better = NULL) - Find the winner genotype in each environment
ge_winners(data_ge2, ENV, GEN, c(PH, EH, EP, EL, ED), type = "ranks")
ENV PH EH EP EL ED ENV PH EH EP EL ED
1 A1 H3 H1 H1 H6 H6 1 A1 H3 H1 H1 H6 H6
2 A2 H2 H1 H1 H6 H2 2 A1 H4 H4 H1 H1 H3
3 A3 H13 H13 H6 H4 H13 3 A1 H4 H9 H4 H10 H10
4 A4 H5 H5 H10 H7 H11 # ... with 49 more rows
```

Genotype-environment means

```
ge_means(data, env, gen, resp) - Genotype, Environment and Genotype-environment means
ge <- ge_means(data_ge2, ENV, GEN, PH)$PH
ge$gen_means mat ge$gen_means long ge$gen_means ge$gen_means
GEN_Y HMGV_R RPGV_Y RPGV_Y RPGV_Y HMRPGV_Y HMRPGV_Y
1 H1 2.62 2.58 1 1.05 2.60 1 1.04 2.59 1
2 H10 2.31 2.31 13 0.940 2.34 13 0.937 2.33 13
3 H11 2.39 2.38 10 0.967 2.40 10 0.964 2.40 10
4 H12 2.44 2.43 8 0.988 2.45 8 0.981 2.44 8
# ... with 9 more rows
```

BLUP-based methods

```
Resende_indexes(data) - Harmonic Mean of Genotypic Values (HMGV), Relative Performance of Genotypic Values (RPGV), and Harmonic Mean of Relative Performance of Genotypic Values (HMRPGV)
data_ge2 %>% gamem_met(ENV, GEN, REP, PH) %>% Resende_indexes()
GEN_Y HMGV_R RPGV_Y RPGV_Y RPGV_Y HMRPGV_Y HMRPGV_Y
1 H1 2.62 2.58 1 1.05 2.60 1 1.04 2.59 1
2 H10 2.31 2.31 13 0.940 2.34 13 0.937 2.33 13
3 H11 2.39 2.38 10 0.967 2.40 10 0.964 2.40 10
4 H12 2.44 2.43 8 0.988 2.45 8 0.981 2.44 8
# ... with 9 more rows
```

waasb(data, env, gen, rep, resp, block = NULL, mresp = NULL, wresp = NULL, random = "gen", prob = 0.05, ind_anova = TRUE, verbose = TRUE) - Weighted Average of Absolute Scores from BLUP interaction matrix.

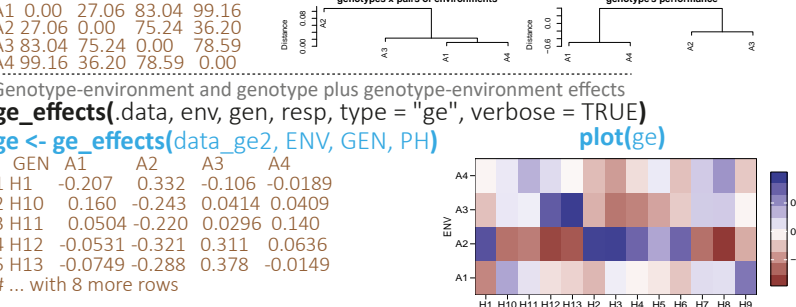
```
mod <- data_ge2 %>% waasb(ENV, GEN, REP, width_of(3))
p <- plot_scores(mod, type = 3) p + theme_metan_minimal()
```

gmd(mod, "WAASB")

```
gen NKR TKW NKE
1 H1 0.55 3.5 2.5
2 H10 0.29 2.5 0.70
3 H11 0.53 0.80 0.78
4 H12 0.43 1.6 1.9
5 H13 0.55 0.42 4.5
6 H2 0.27 3.9 2.2
# ... with 9 more rows
```

fai_blup(data, DI, UI, SI = NULL, mineval = 1, verbose = TRUE) - FAI-BLUP index for multi-trait selection

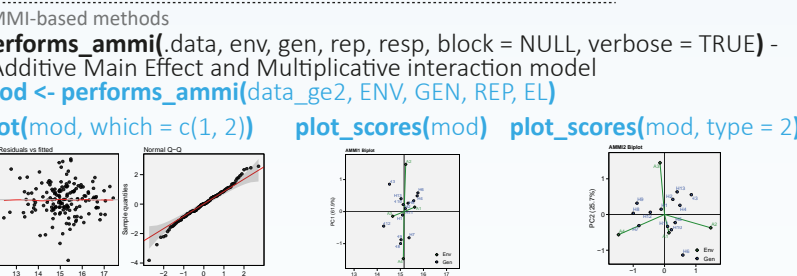
```
fai_mod <- fai_blup(mod, DI = rep("max", 3), UI = rep("min", 3))
plot(fai_mod) plot(fai_mod, radar = FALSE)
```



STABILITY ANALYSIS

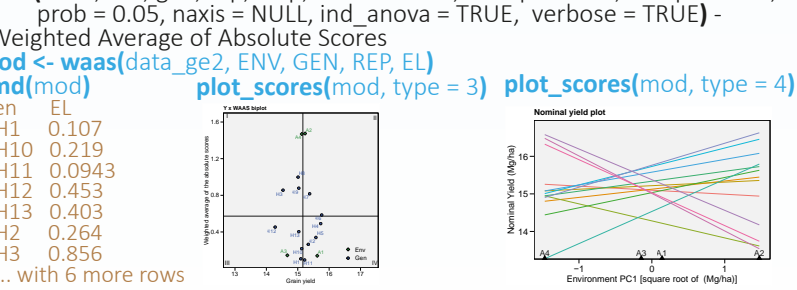
Parametric methods Parametric stability statistics and ranks for the stability indexes

```
ANOVA-based methods
Annicchiario() - Annicchiarico's genotypic confidence index
ecovale() - Wricker's ecovale
Schmidtt() - Genotypic confidence index modified by Schmidtt
Shuklia() - Shukla's stability variance parameter
Regression-based methods
ge_reg() - Eberhart and Russell's regression model
AMMI-based methods
performs_ammfi(data, env, gen, rep, resp, block = NULL, verbose = TRUE) - Additive Main Effect and Multiplicative interaction model
mod <- performs_ammfi(data_ge2, ENV, GEN, REP, EL)
plot(mod, which = c(1, 2)) plot_scores(mod) plot_scores(mod, type = 2)
```



AMMI indexes(data, ordery = NULL, level = 0.95) - AMMI-based stability indexes

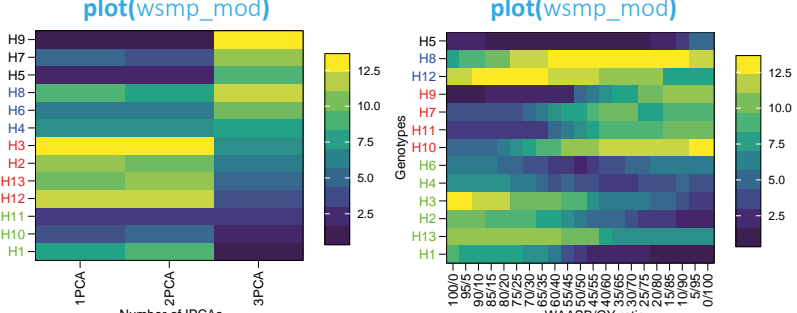
```
print(AMMI_indexes(mod))
waasb(data, env, gen, rep, resp, block = NULL, mresp = NULL, wresp = NULL,
  prob = 0.05, naxis = NULL, ind_anova = TRUE, verbose = TRUE) - Weighted Average of Absolute Scores
mod <- waasb(data_ge2, ENV, GEN, REP, EL)
gmd(mod) plot_scores(mod, type = 3) plot_scores(mod, type = 4)
```



mtsi(data, index = "waasby", SI = 15, mineval = 1, verbose = TRUE) - Olivoto's multi-trait stability index

```
mtsi_mod <- data_ge2 %>% waasb(ENV, GEN, REP, everything()) %>% mtsi()
print(mtsi_mod$sel_difvar) plot(mtsi_mod)
VAR Factor xo Xs SD SDperc
1 EL FA 1 15.2 15.7 0.491 3.24
2 CD FA 1 16.0 16.6 0.587 3.67
3 KW FA 1 17.3 18.4 11.0 6.37
4 NR FA 2 16.1 15.8 -0.306 -1.90
5 NKR FA 2 32.2 34.5 2.22 6.87
6 PERK FA 2 67.4 87.3 -0.151 -0.173
# ... with 9 more rows
```

wmsp(model, mresp = 100, increment = 5, saveWAAS = 50, prob = 0.05, progbar = TRUE) - Weighting between stability and mean performance



BLUP-based methods

```
Resende_indexes(data) - Harmonic Mean of Genotypic Values (HMGV), Relative Performance of Genotypic Values (RPGV), and Harmonic Mean of Relative Performance of Genotypic Values (HMRPGV)
data_ge2 %>% gamem_met(ENV, GEN, REP, PH) %>% Resende_indexes()
GEN_Y HMGV_R RPGV_Y RPGV_Y RPGV_Y HMRPGV_Y HMRPGV_Y
1 H1 2.62 2.58 1 1.05 2.60 1 1.04 2.59 1
2 H10 2.31 2.31 13 0.940 2.34 13 0.937 2.33 13
3 H11 2.39 2.38 10 0.967 2.40 10 0.964 2.40 10
4 H12 2.44 2.43 8 0.988 2.45 8 0.981 2.44 8
# ... with 9 more rows
```

waasb(data, env, gen, rep, resp, block = NULL, mresp = NULL, wresp = NULL, random = "gen", prob = 0.05, ind_anova = TRUE, verbose = TRUE) - Weighted Average of Absolute Scores from BLUP interaction matrix.

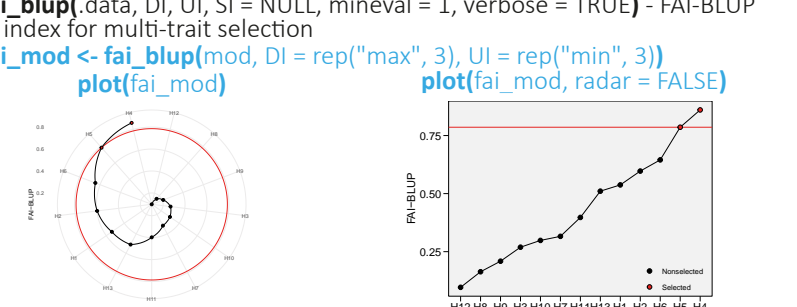
```
mod <- data_ge2 %>% waasb(ENV, GEN, REP, width_of(3))
p <- plot_scores(mod, type = 3) p + theme_metan_minimal()
```

gmd(mod, "WAASB")

```
gen NKR TKW NKE
1 H1 0.55 3.5 2.5
2 H10 0.29 2.5 0.70
3 H11 0.53 0.80 0.78
4 H12 0.43 1.6 1.9
5 H13 0.55 0.42 4.5
6 H2 0.27 3.9 2.2
# ... with 9 more rows
```

fai_blup(data, DI, UI, SI = NULL, mineval = 1, verbose = TRUE) - FAI-BLUP index for multi-trait selection

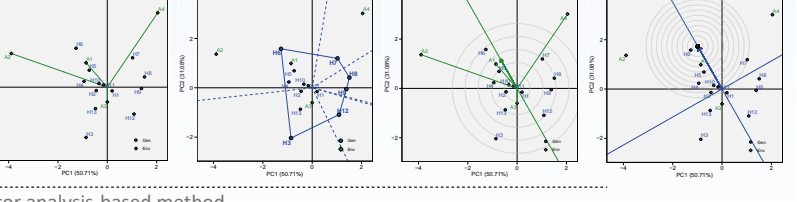
```
fai_mod <- fai_blup(mod, DI = rep("max", 3), UI = rep("min", 3))
plot(fai_mod) plot(fai_mod, radar = FALSE)
```



STABILITY ANALYSIS

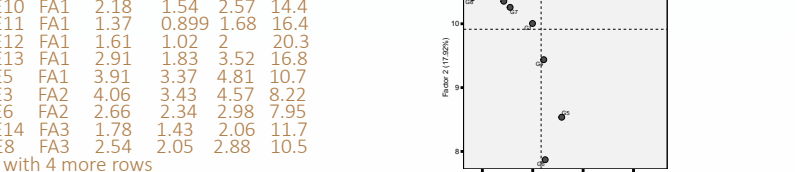
GGE-based method

```
gge(data, env, gen, resp, centering = "environment", scaling = "none",
  svp = "environment", ...) - Genotype plus Genotype-Environment interaction model
m <- gge(data_ge2, ENV, GEN, EL)
plot(m) plot(m, type = 3) plot(m, type = 3) plot(m, type = 8)
```



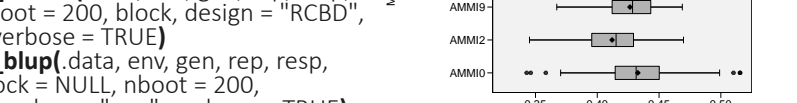
ge_factanal(data, env, gen, rep, resp, mineval = 1, verbose = TRUE) - Stability analysis and environmental stratification

```
mod <- ge_factanal(data_ge, ENV, GEN, REP, GY)
print(mod) plot(mod)
Factor analysis-based method
Env Factor Mean Min Max CV
1 E1 FA 1 2.52 1.97 2.90 13.3
2 E10 FA 1 2.18 1.54 2.57 14.4
3 E11 FA 1 1.37 0.899 1.68 16.4
4 E12 FA 1 1.61 1.02 2 20.3
5 E13 FA 1 2.91 1.83 3.52 16.8
6 E5 FA 1 3.91 3.37 4.81 10.7
7 E3 FA 2 4.06 3.43 4.57 8.22
8 E6 FA 2 2.66 2.34 2.98 7.95
9 E14 FA 3 1.78 1.43 2.06 11.7
10 E8 FA 3 2.54 2.05 2.88 10.5
# ... with 4 more rows >
```



cv_ammfi(data, env, gen, rep, resp, block = NULL, naxis = 2, nboot = 200, design = "RCBD", verbose = TRUE)

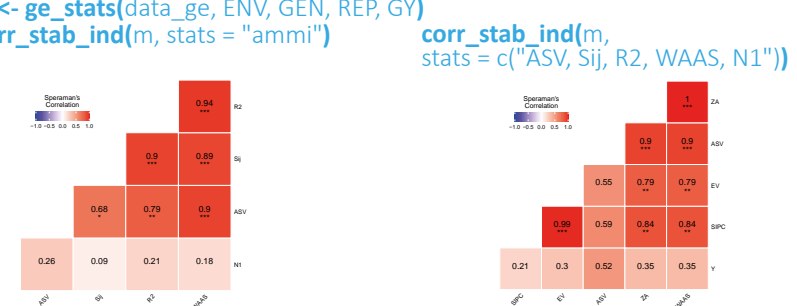
```
cv_ammfi(data, env, gen, rep, resp, nboot = 200, block, design = "RCBD",
  verbose = TRUE)
cv_blup(data, env, gen, rep, resp, block = NULL, nboot = 200,
  random = "gen", verbose = TRUE)
```



Non-parametric methods Rank-based stability measures

```
Foxl(data, env, gen, resp) - The Fox's 'top third' method
Huehnl(data, env, gen, resp) - Huehnl's stability statistics
Superiority(data, env, gen, resp) - Lin and Binns' superiority measure
Thennarasu(data, env, gen, resp) - Thennarasu's stability statistics
```

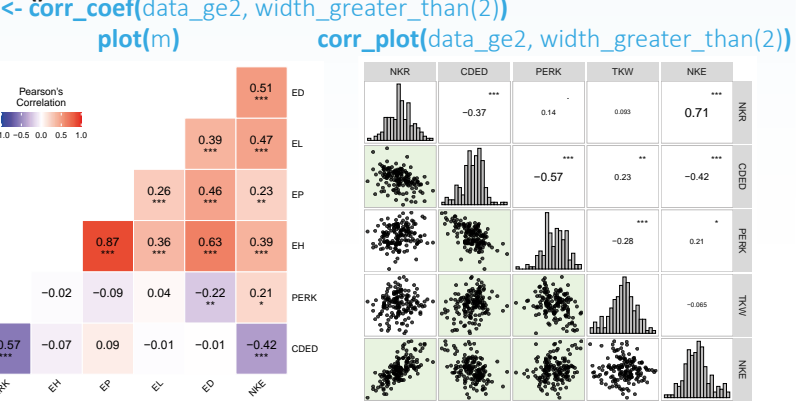
Correlation between stability indexes



BIOMETRICAL MODELS

Correlation coefficient Linear and partial correlation coefficient

```
corr_coeff() - Pearson Correlation Coefficient
corr_plot() - Numerical and graphics correlation coefficient
corr_ci() - Non-parametric confidence interval for correlation coefficient
corr_ss() - Sample size planning for correlation coefficient
lpcorr() - Partial correlation coefficient
m <- corr_coef(data_ge2, width_greater_than(2))
plot(m) corr_plot(data_ge2, width_greater_than(2))
```



Genetic, phenotypic and residual correlations

```
covcor_design() - Genotypic, phenotypic, and residual (co)variance/correlation matrices for designed experiments
Path analysis Path analysis with minimal multicollinearity
colindia() - Collinearity diagnostic
non_collinear_vars() - Select non-collinear predictor variables
path_coeff() - Path analysis with direct and indirect coefficients
```

Canonical correlation coefficient

```
can_corr() - Canonical correlation analysis with collinearity diagnostic, estimation of canonical loads, canonical scores, and hypothesis testing for correlation pairs.
```