

Multi-Environment Trial (MET) Analysis

Guilherme Oliveira, Mandeep Singh, Subash Tapha

2024-11-05

Multi-Environment Trial Analysis - PS 756 project

The following document was developed by Guilherme Oliveira, Mandeep Singh, Subash Tapha, students at South Dakota State University, as a requisite for the course PS 746 - Quantitative Genetics.

The main objective of this project was to conduct Multi Trial Analysis, covering i) GxE analysis (+ GGE analysis) ii) AMMI analysis, iii) GGI analysis, iv) FW analysis and v) genomic selection approach which explores in some way Genotype x Environment interaction (G x E).

For the parts i, ii, iii,iv, we are exploring the open dataset made available by Dias et al. (2018) which contains phenotypic data of five drought tolerance traits, measured in 308 hybrids along eight environments contrasting for water availability. For practical purposes, we are using a subset of 202 hybrids. The traits analyzed were grain yield (GY), ears per plot (EPP), female and male flowering times (FFT and MFT), and anthesis-silking interval (ASI). The source code used is from the metan (Olivoto and Lucio, 2020) and statgenGxE reference manuals.

To complete this task, we are exploring the open dataset made available by Crossa et al. 2013 and Montesinos-Lopez et al. 2016, 2017 which contains the data of 309 double-haploid maize lines conducted in 3 environments and each environments having 3 reps of each line. The traits analyzed were grain yield (yield), anthesis-silking interval (ASI) and plant height (PH). The source code used is from Montesinos-Lopez et al. 2019 using the BMTME R package, where we did just small modifications, however, bringing more interpretations and extra analyses.

Directory

```
setwd("C:/Users/Guilherme.Oliveira/Documents/PS 756 - Final Project")
getwd()
```

```
## [1] "C:/Users/Guilherme.Oliveira/Documents/PS 756 - Final Project"
```

Libraries

The main libraries used for this project to run the Multi-Environment Trial Analysis.

```
library(statgenGxE)
library(metan)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2

## |=====|

## | Multi-Environment Trial Analysis (metan) v1.18.0      |

## | Author: Tiago Olivoto                                |

## | Type 'citation('metan')' to know how to cite metan    |

## | Type 'vignette('metan_start')' for a short tutorial  |

## | Visit 'https://bit.ly/pkgmetan' for a complete tutorial |

## |=====|
```

```
library(tibble)
```

```
##
## Attaching package: 'tibble'

## The following objects are masked from 'package:metan':
##
##   column_to_rownames, remove_rownames, rownames_to_column
```

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.4
## v forcats    1.0.0      v stringr    1.5.1
## v lubridate  1.9.2      v tidyr      1.3.1
## v purrr      1.0.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x forcats::as_factor()      masks metan::as_factor()
## x tibble::column_to_rownames() masks metan::column_to_rownames()
## x dplyr::filter()           masks stats::filter()
## x dplyr::lag()              masks stats::lag()
## x dplyr::recode_factor()     masks metan::recode_factor()
## x tibble::remove_rownames() masks metan::remove_rownames()
## x tidyr::replace_na()        masks metan::replace_na()
## x tibble::rownames_to_column() masks metan::rownames_to_column()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(BMTME)
library(lme4)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

Data

The “pheno” is the dataset which will be used for the parts i, ii, iii, while the “G.Maize” and “Data.Maize” will be used for the parts iv.

```
pheno <- read.table("pheno_corn_MET.txt", header = TRUE)
load("C:/Users/Guilherme.Oliveira/Documents/PS 756 - Final Project/G.Maize.RData")
load("C:/Users/Guilherme.Oliveira/Documents/PS 756 - Final Project/Data.Maize.RData")
```

Data preparation and exploration

Before running the parts i, ii, iii, we need to check some very important points. First, we need to check our “pheno” structure, normality, distribution, descriptive statistic etc.

To complete these tasks, we are using some R basic functions + some functions inside the metan package.

Data strcture

In this analysis the ENV, GEN, REP should be factors, while the traits should be numeric. Also we need, to change our phenotypic data frame to a tibble, being able to run the metan functions.

```
str(pheno)

## 'data.frame':   3434 obs. of  8 variables:
##  $ ENV: chr  "E1" "E2" "E3" "E4" ...
##  $ REP: int   1 1 1 1 1 1 1 1 2 2 ...
##  $ GEN: int   1 1 1 1 1 1 1 1 1 1 ...
##  $ FFT: int  66 62 69 70 63 55 54 56 65 63 ...
##  $ MFT: int  62 61 66 69 50 52 51 53 63 62 ...
##  $ ASI: int   4 1 3 1 13 3 3 3 2 1 ...
##  $ EPP: int  12 18 13 15 7 15 13 7 12 16 ...
##  $ GY : num  1.91 6.63 3.63 5.6 0.97 4.64 4.5 2.55 3.6 5.53 ...
```

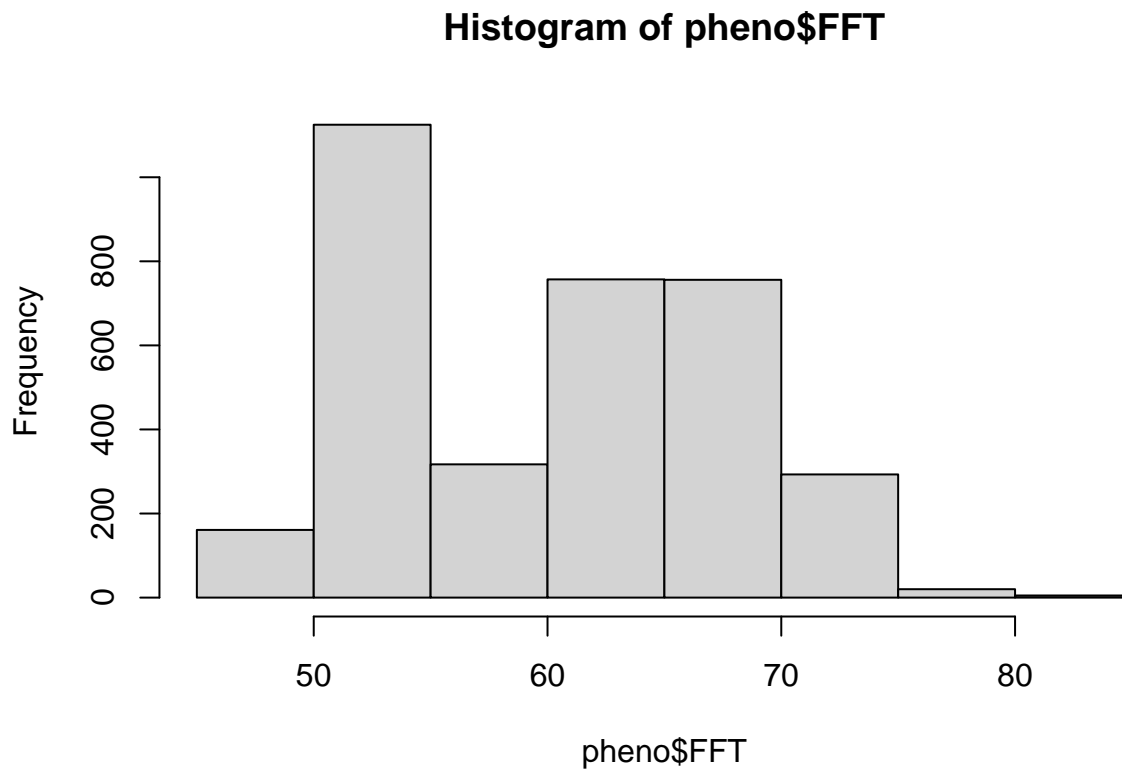
```
pheno$ENV <- as.factor(pheno$ENV)
pheno$GEN <- as.factor(pheno$GEN)
pheno$REP <- as.factor(pheno$REP)
pheno$FFT <- as.numeric(pheno$FFT)
pheno$MFT <- as.numeric(pheno$MFT)
pheno$ASI <- as.numeric(pheno$ASI)
pheno$EPP <- as.numeric(pheno$EPP)
pheno <- as_tibble(pheno)
```

```
str(pheno)
```

```
## tibble [3,434 x 8] (S3: tbl_df/tbl/data.frame)
## $ ENV: Factor w/ 8 levels "E1","E2","E3",...: 1 2 3 4 5 6 7 8 1 2 ...
## $ REP: Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 2 2 ...
## $ GEN: Factor w/ 202 levels "1","3","4","8",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ FFT: num [1:3434] 66 62 69 70 63 55 54 56 65 63 ...
## $ MFT: num [1:3434] 62 61 66 69 50 52 51 53 63 62 ...
## $ ASI: num [1:3434] 4 1 3 1 13 3 3 3 2 1 ...
## $ EPP: num [1:3434] 12 18 13 15 7 15 13 7 12 16 ...
## $ GY : num [1:3434] 1.91 6.63 3.63 5.6 0.97 4.64 4.5 2.55 3.6 5.53 ...
```

Data normality

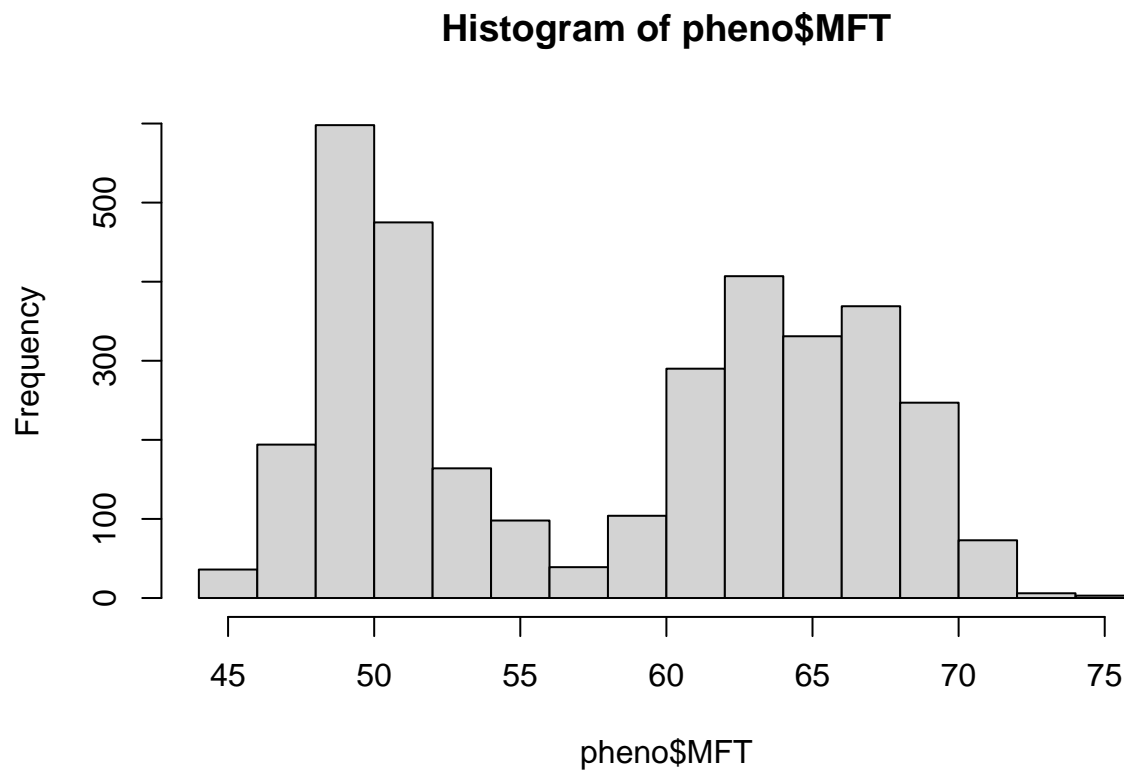
```
hist(pheno$FFT)
```



```
shapiro.test(pheno$FFT)
```

```
##
## Shapiro-Wilk normality test
##
## data:  pheno$FFT
## W = 0.93568, p-value < 2.2e-16
```

```
hist(pheno$MFT)
```

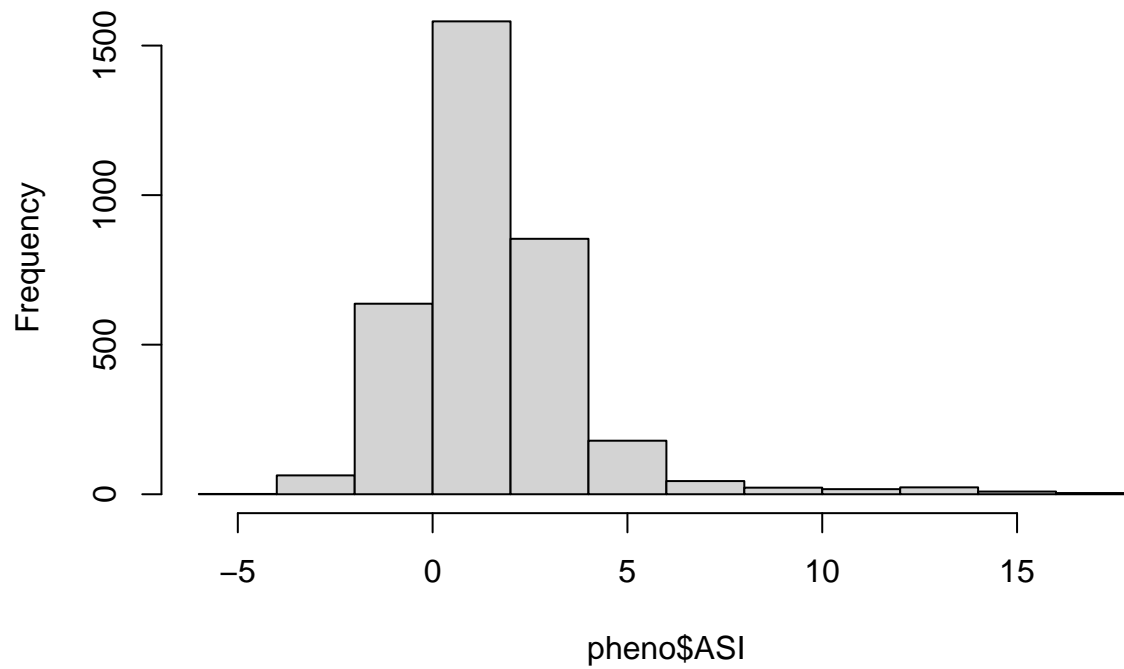


```
shapiro.test(pheno$MFT)
```

```
##  
##  Shapiro-Wilk normality test  
##  
## data:  pheno$MFT  
## W = 0.89781, p-value < 2.2e-16
```

```
hist(pheno$ASI)
```

Histogram of pheno\$ASI

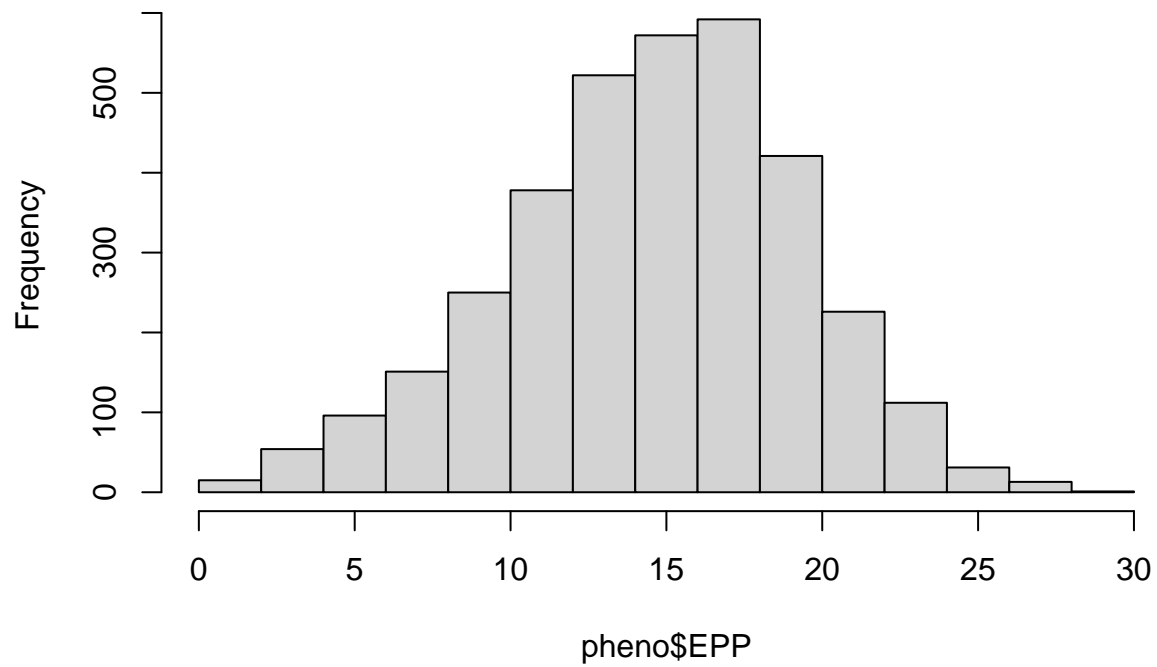


```
shapiro.test(pheno$ASI)
```

```
##  
##  Shapiro-Wilk normality test  
##  
## data:  pheno$ASI  
## W = 0.82689, p-value < 2.2e-16
```

```
hist(pheno$EPP)
```

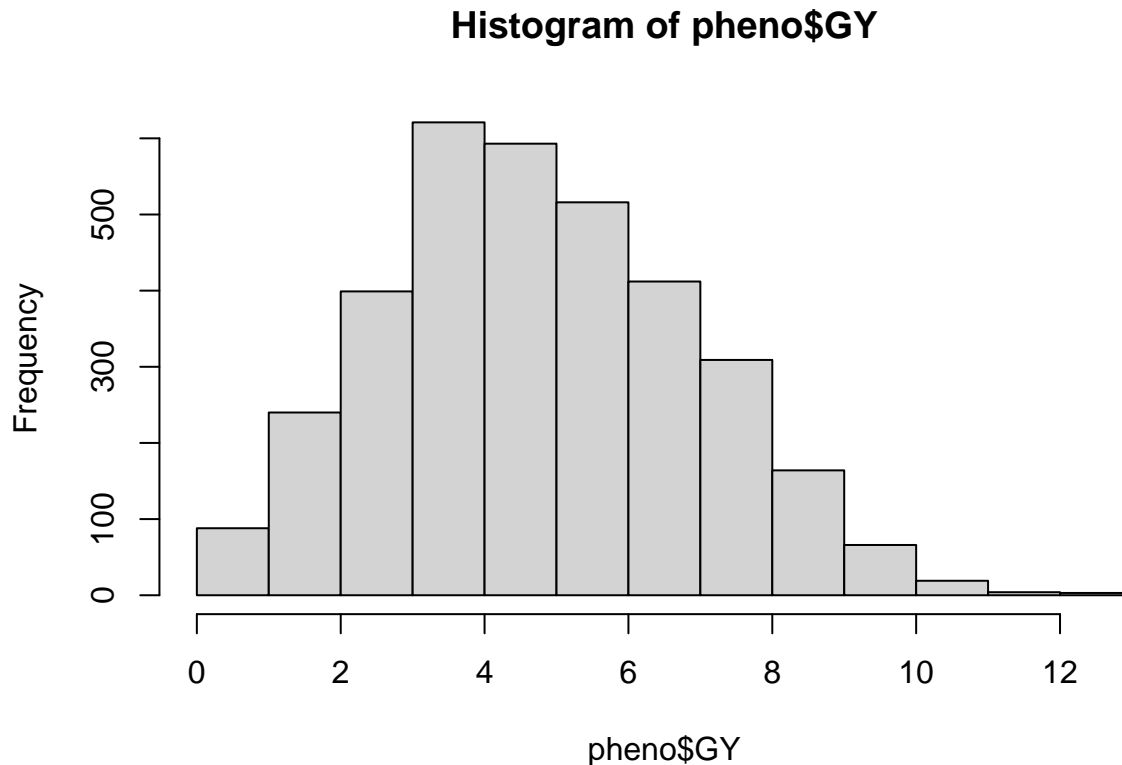
Histogram of pheno\$EPP



```
shapiro.test(pheno$EPP)
```

```
##  
##  Shapiro-Wilk normality test  
##  
## data:  pheno$EPP  
## W = 0.9893, p-value = 2.005e-15
```

```
hist(pheno$GY)
```



```
shapiro.test(pheno$GY)
```

```
##  
##  Shapiro-Wilk normality test  
##  
## data:  pheno$GY  
## W = 0.99075, p-value = 3.805e-14
```

Based on the histograms and Shapiro-Wilks, all the traits does not have a normal distribution. There are different ways to get distributions more close to normality, as example, removing outliers or transforming the data. In our case we tested both approaches, however, we are still unable to get higher values than 0.05 in our Shapiro-Wilk analyses.

The mains reasons to that, it is because we have in our dataset environments with high drought stress affecting the plants, as well as, good environments, and in the moment that we pull together the data, we obtain some extremes values (caused by the drought stress), resulting in some normality deviations.

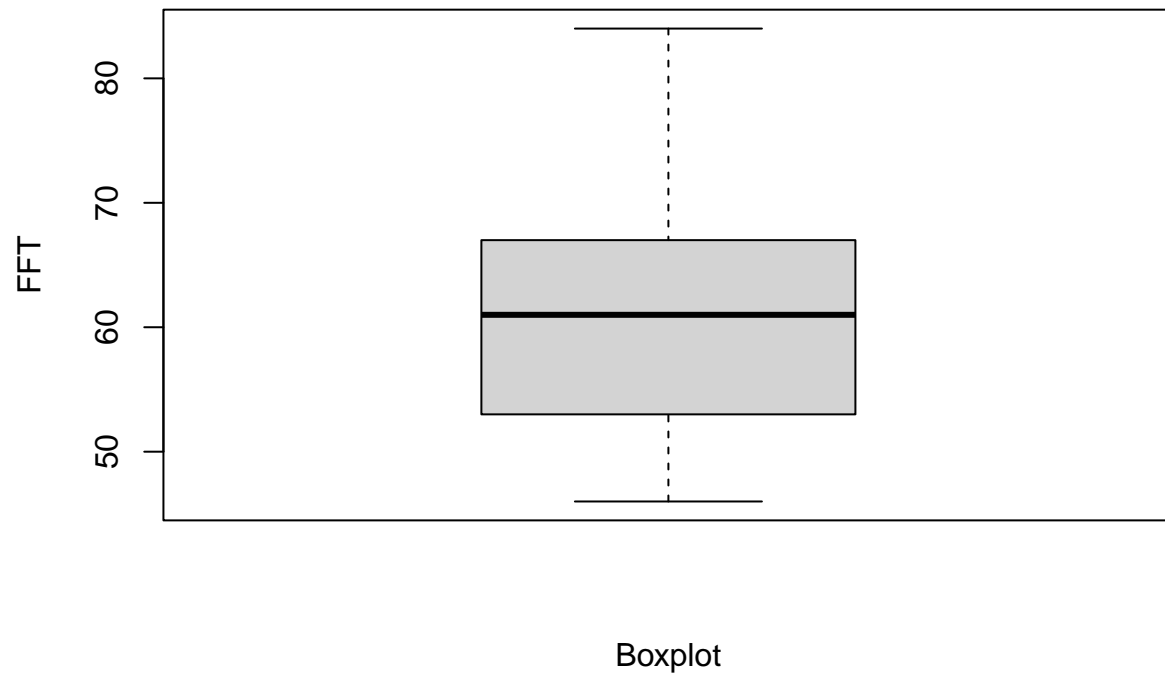
In this sense, we decide to proceed the analysis with our raw values, despite then being completely normal, once, that the main objective of this project is to understand and interpret the MET analysis and develop and/or analyze some previous developed code. Nevertheless, we strongly suggest to pay attention in the data normality and understand how (or not) it can affect the analysis results.

However, in the next codes chunks, we are still providing our code to remove outliers based on the interquartile range (IQR) equations to remove outliers ($Q1 - 1.5 \times IQR$ and $Q3 + 1.5 \times IQR$).

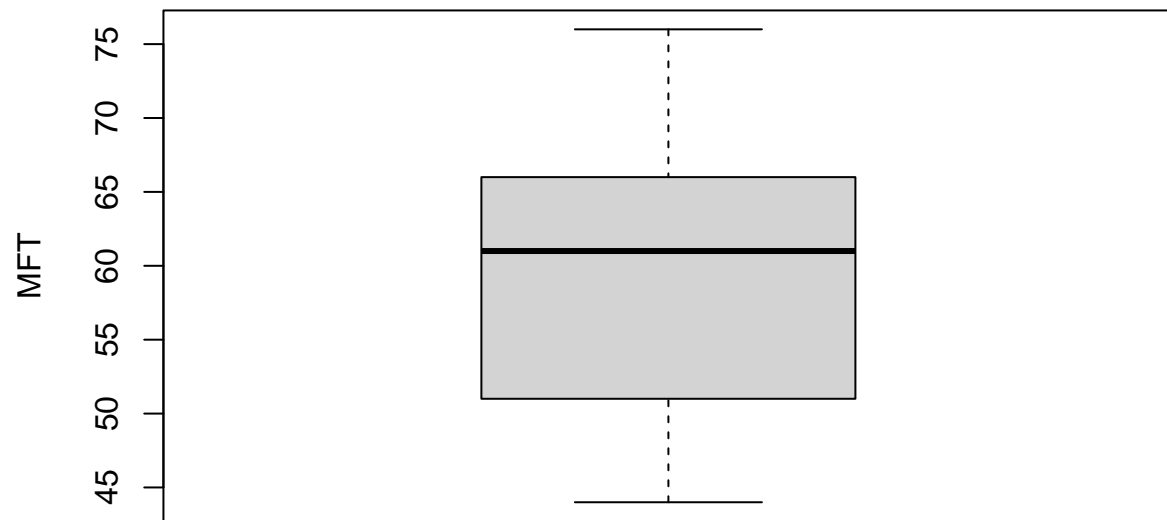
Code for removing outliers

```
# Code to get and see the boxplots
```

```
boxplot.FFT <- boxplot(pheno$FFT, xlab="Boxplot",ylab="FFT")
```

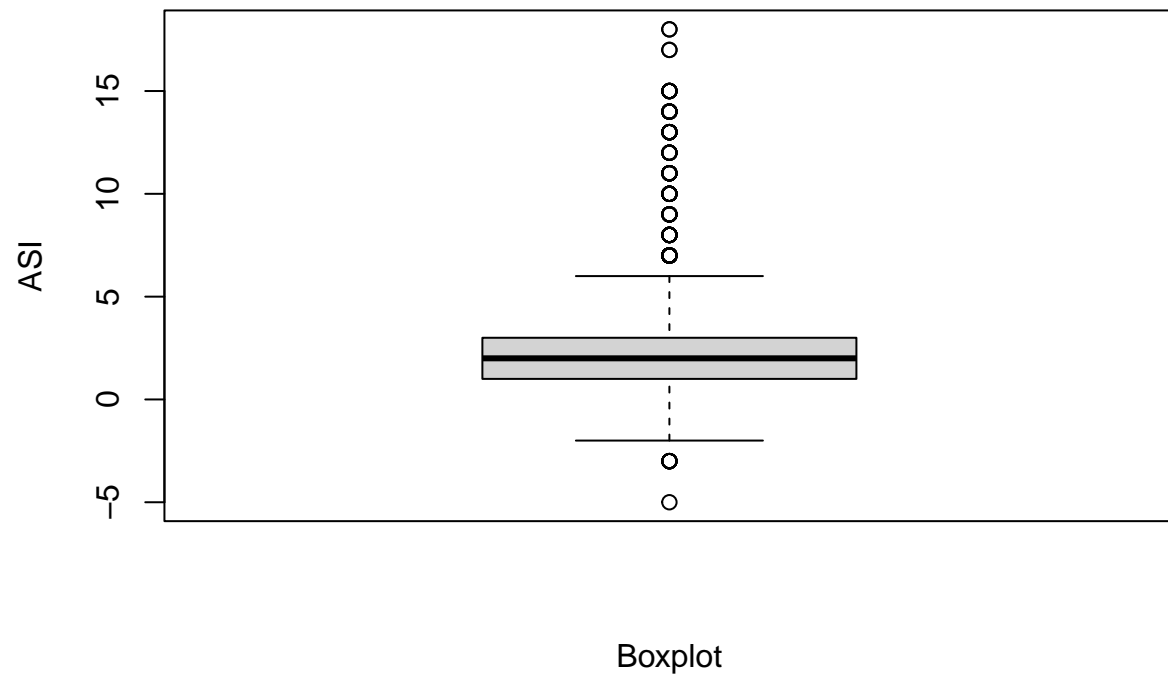


```
boxplot.MFT <- boxplot(pheno$MFT, xlab="Boxplot",ylab="MFT")
```

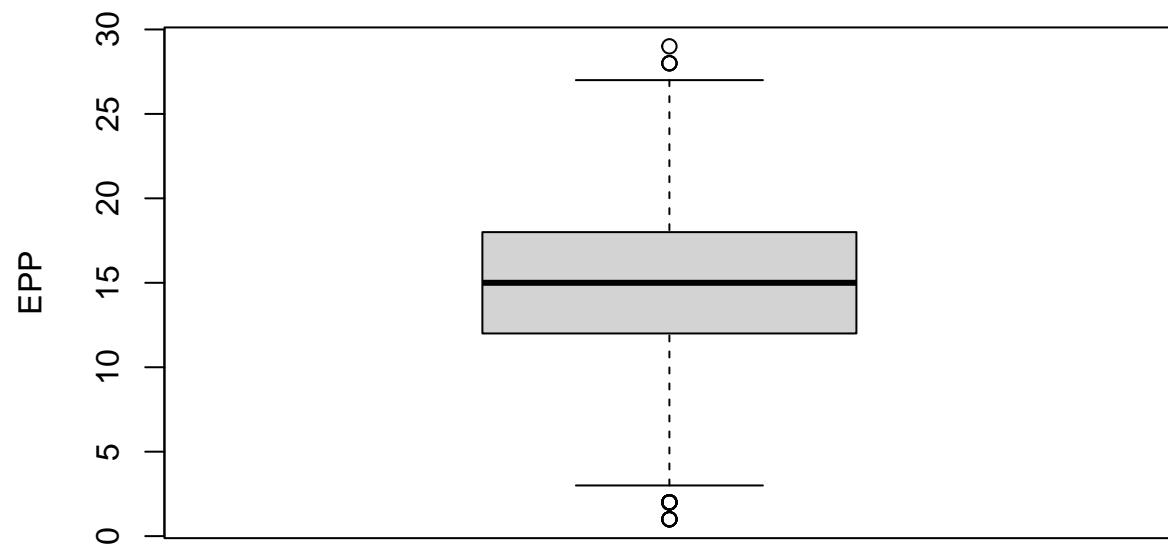


Boxplot

```
boxplot.ASI <- boxplot(pheno$ASI, xlab="Boxplot", ylab="ASI")
```

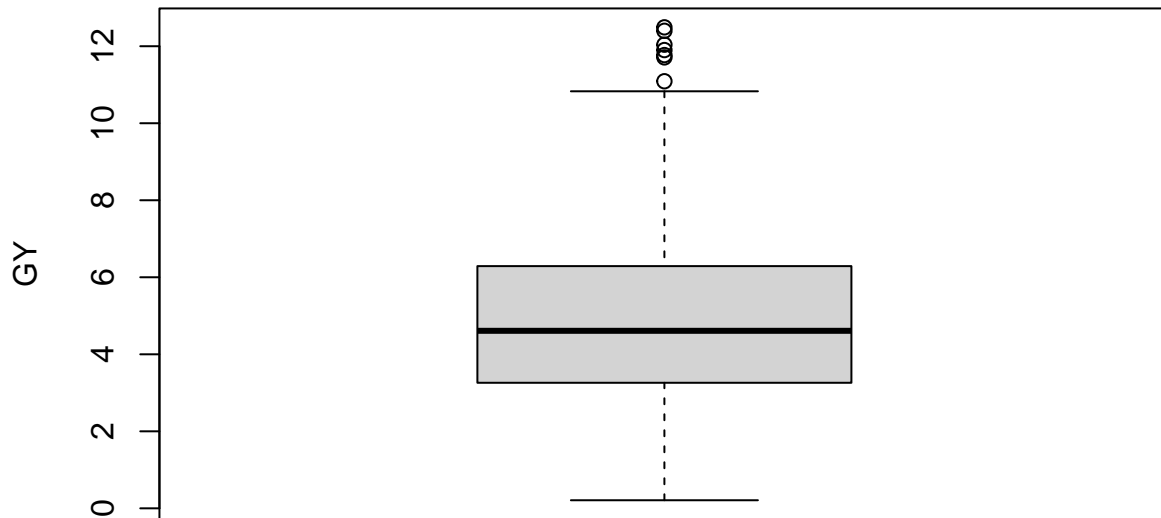


```
boxplot.EPP <- boxplot(pheno$EPP, xlab="Boxplot", ylab="EPP")
```



Boxplot

```
boxplot.GY <- boxplot(pheno$GY, xlab="Boxplot",ylab="GY")
```



Boxplot

```
# Example to remove outliers for a specific trait
```

```
outliers <- boxplot.GY$out; outliers
```

```
## [1] 12.40 12.04 11.09 11.90 11.77 12.49 11.71
```

```
#pheno <- pheno[-which(pheno$GY%in%outliers),]  
#shapiro.test(pheno$GY)
```

Information of the genotypes

The metan package has a bunch of different functions which provide valuable information to check, manipulate, and summarize data. In the next code chunks we are going to show just a small portion of them, before starting the main analyzes required by this project. However, if someone is interested in learning more about these preliminary steps, here is the link for the paper: <https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/2041-210x.13384> which give the access to supplementary information and R codes.

First, the metan package has functions that permits to check means and coefficient of variation (CV) for each genotype across the environments. It can be useful, as a first way to understand about data quality, finding the best and worst ones, as well, based on the CV, having some idea about genotype by environment interaction.

Also, using the metan functions, we can collect the descriptive analysis by trait.

In this analysis, we can observe some very extreme values for CV, as example the ASI CV. In this case, the CV is higher than 100%, meaning that our standard deviation is larger than our mean, in other words, there is a high variability for this trait. Also, we got a high CV for GY, close to 40%.

Again, the main reason to obtain these results is the nature of the environments where the data was collected, having some of them, extreme drought effects. Traits as ASI and GY are extremely affected by drought, so high CVs to these traits, in our analysis conditions are expected.

```
desc_stat(pheno, stats = "all")
```

```
## # A tibble: 5 x 34
##   variable av.dev  ci.t  ci.z  cv gmean hmean  iqr  kurt  mad  max
##   <chr>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ASI        1.48 0.0769 0.0769 107.  1.97  0    2    9.06  1.48  18
## 2 EPP        3.72 0.156  0.156  31.0 14.1 12.7  6   -0.0772 4.45  29
## 3 FFT        6.70 0.251  0.251  12.4 60.2 59.7 14   -1.24  10.4  84
## 4 GY         1.74 0.0713 0.0713  44.5  4.21  3.39 3.03 -0.336  2.21 12.5
## 5 MFT        7.26 0.261  0.261  13.3 58.0 57.4 14.8 -1.54  11.9  76
## # i 23 more variables: mean <dbl>, median <dbl>, min <dbl>, n <dbl>,
## #   n.valid <dbl>, n.missing <dbl>, n.unique <dbl>, ps <dbl>, q2.5 <dbl>,
## #   q25 <dbl>, q75 <dbl>, q97.5 <dbl>, range <dbl>, sd.amo <dbl>, sd.pop <dbl>,
## #   se <dbl>, skew <dbl>, sum <dbl>, sum.dev <dbl>, ave.dev <dbl>,
## #   sum.sq.dev <dbl>, var.amo <dbl>, var.pop <dbl>
```

If necessary, the same function provides information about single locations, where we can confirm our hypothesis that some environments (drought ones) are pushing the CVs to high values.

```
desc_stat(pheno,
  stats = ("mean, se, cv, max, min"),
  by = ENV)
```

```
## # A tibble: 40 x 7
##   ENV  variable  mean    se    cv  max  min
##   <fct> <chr>    <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 E1    ASI      0.932 0.0701 185.    8    -5
## 2 E1    EPP     12.3  0.152  30.4  23    2
## 3 E1    FFT     64.0  0.105   4.03  73   59
## 4 E1    GY       3.53  0.053  37.0  8.36  0.24
## 5 E1    MFT     63.1  0.0785  3.06  70   58
## 6 E2    ASI      0.404 0.0581 289.    4   -3
## 7 E2    EPP     18.1  0.172  19.1  28    5
## 8 E2    FFT     64.6  0.155   4.81  73   58
## 9 E2    GY       6.70  0.078  23.4 10.7  1.31
## 10 E2   MFT     64.2  0.154   4.81  72   57
## # i 30 more rows
```

Something very interesting that also the metan package provides is a function that shows the best lines (higher phenotypic values) in each of these environments. We can see that there is no replicate for GY, meaning that for each environment, there was a different best hybrid.

```
ge_winners(pheno, ENV, GEN, resp = everything())
```

```
## # A tibble: 8 x 6
##   ENV  FFT  MFT  ASI  EPP  GY
##   <fct> <chr> <chr> <chr> <chr> <chr>
## 1 E1    146   74   146   44   287
```

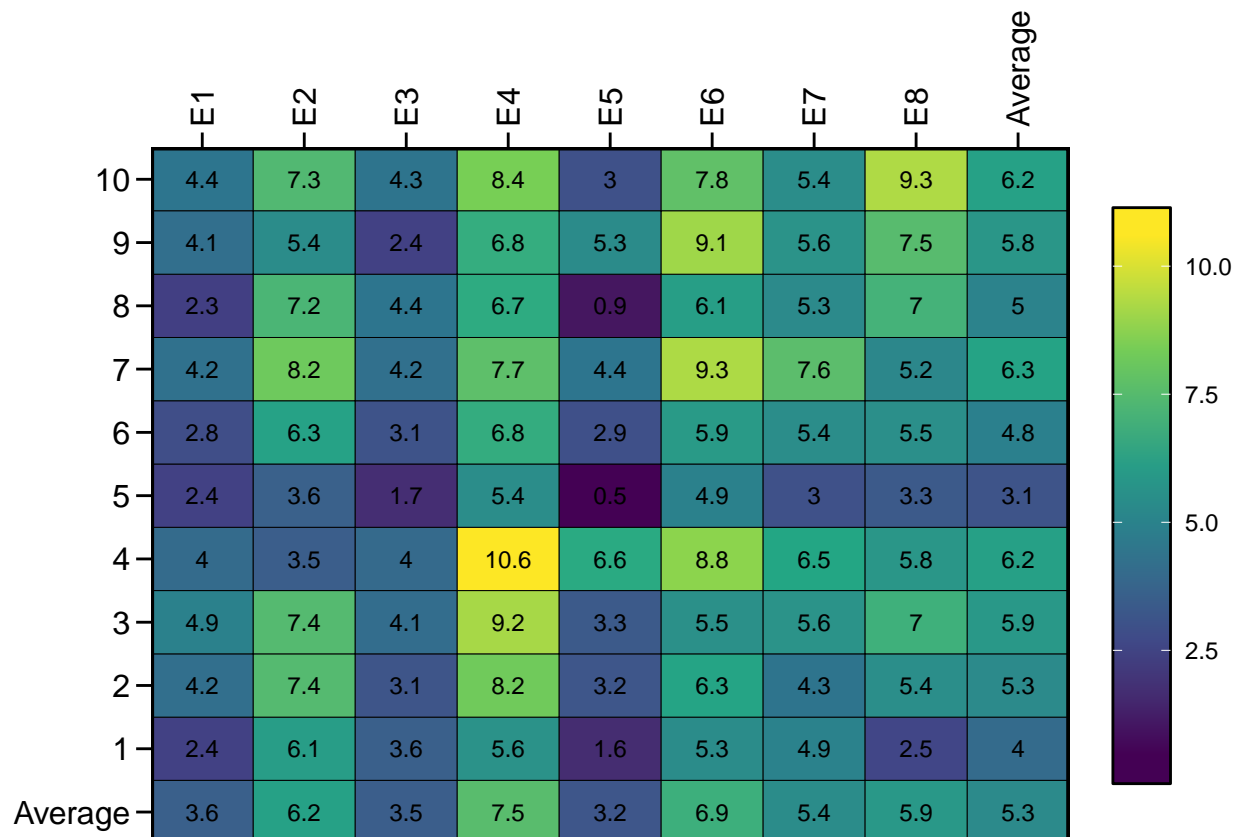
```
## 2 E2    146   236   268   53   101
## 3 E3     22   253    9   191   190
## 4 E4    253   253   224   216   156
## 5 E5    270   16   270   122   234
## 6 E6     73   253   307   289   222
## 7 E7     22   176   276   83    83
## 8 E8    158   148   158   261   69
```

In addition, sometimes numbers in tables are hard to visualize and/or to take other conclusions, which are more easy by plots.

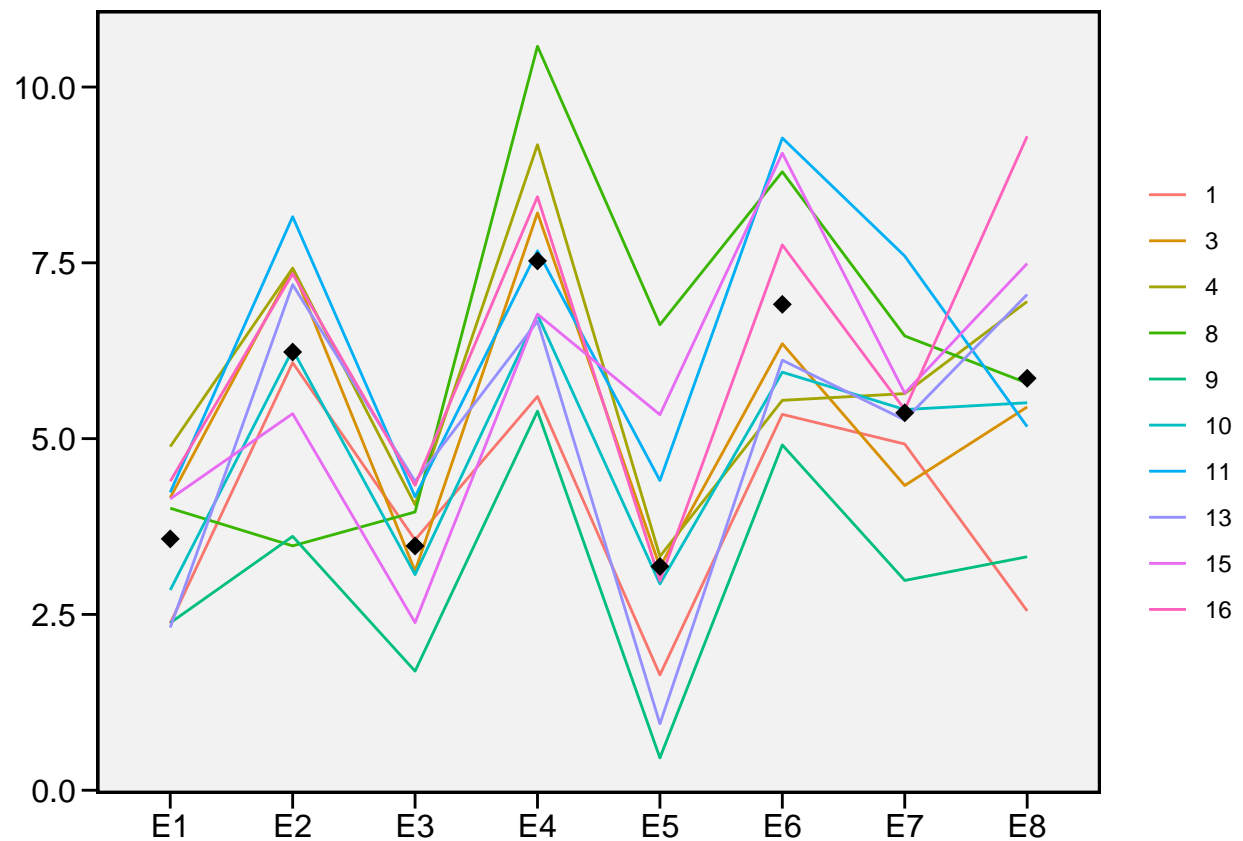
So in the next code chunk, we are showing how to do some heatmaps and line plots to collect more applied information for specific lines across the environments.

So, the first subset, shows how to collect a range of genotypes and the second subset shows how to select very specific lines.

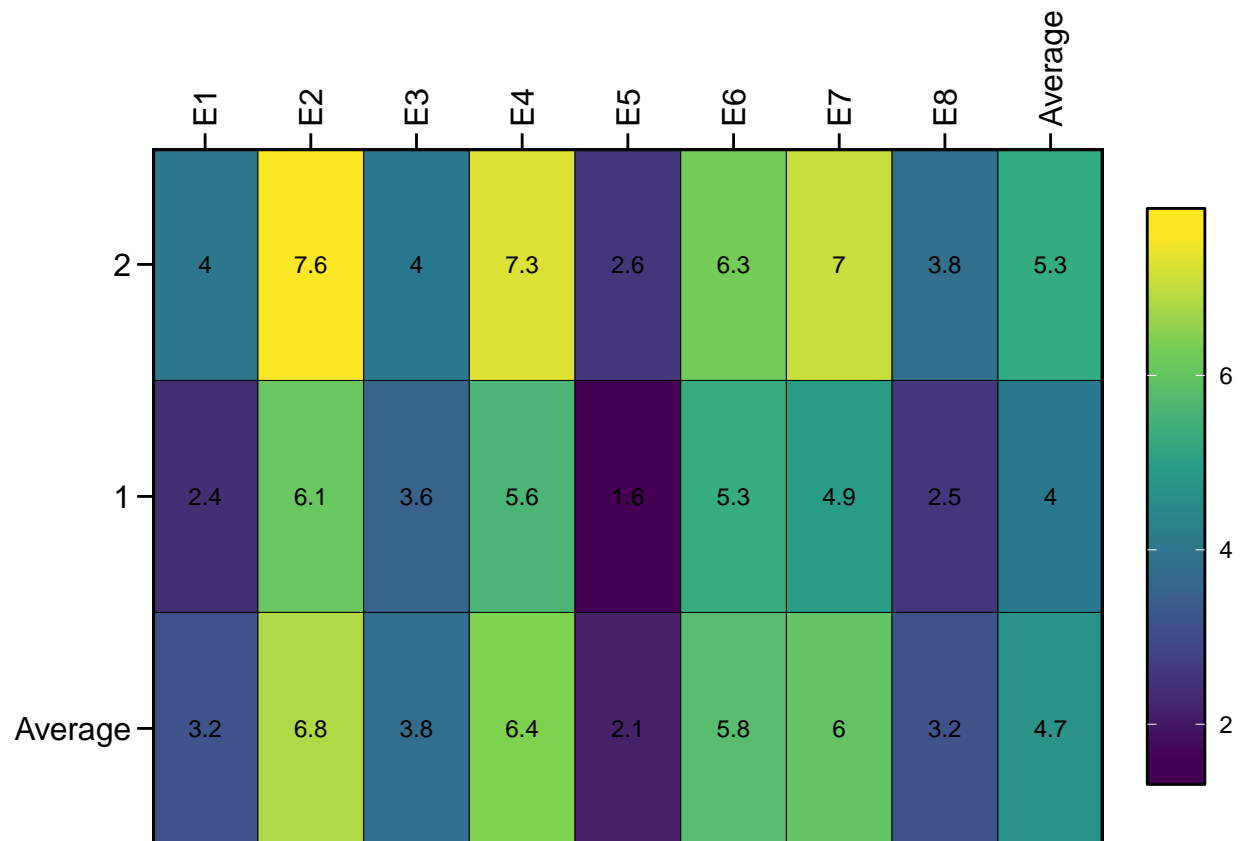
```
subset_pheno <- pheno %>%
  filter(GEN %in% levels(GEN)[1:10]) %>%
  droplevels()
a <- ge_plot(subset_pheno, ENV, GEN, GY); a
```



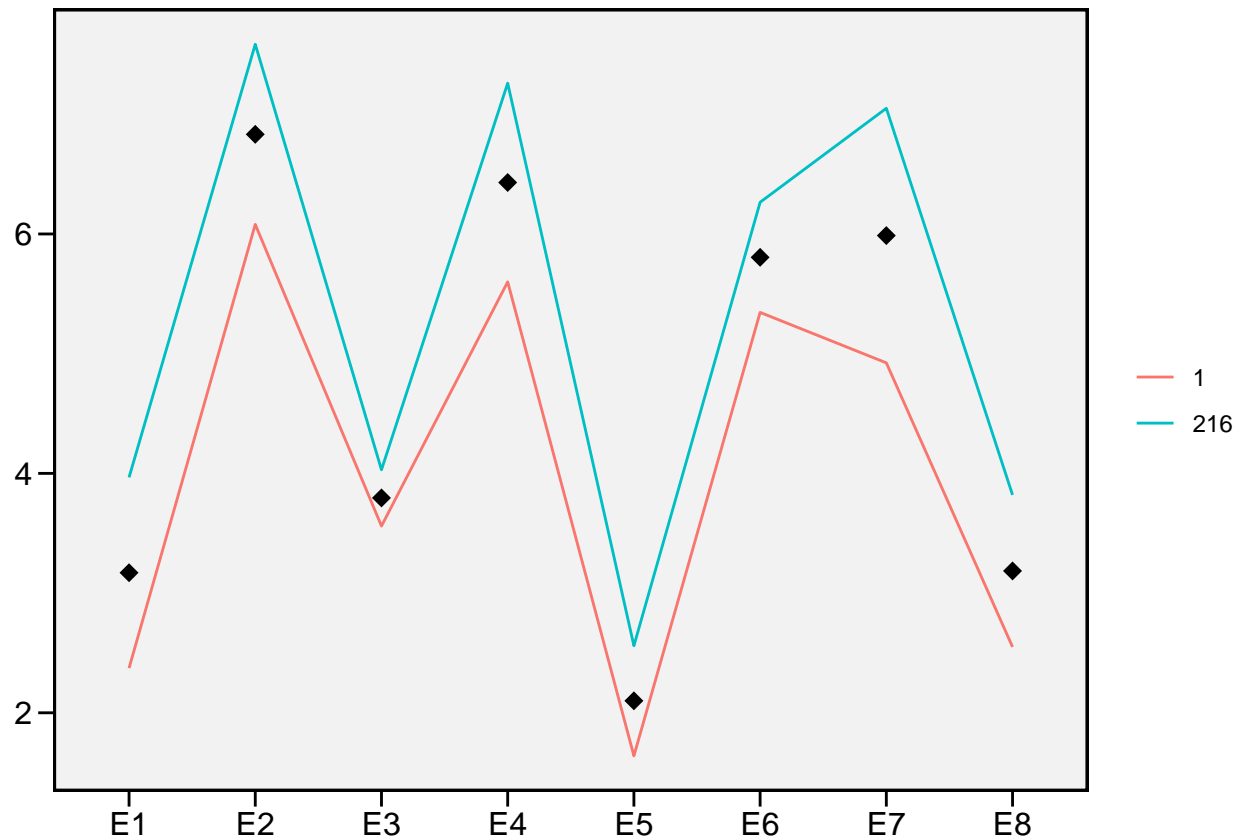
```
b <- ge_plot(subset_pheno, ENV, GEN, GY, type = 2); b
```



```
subset_pheno2 <- pheno %>% filter(GEN %in% c(levels(GEN)[1], levels(GEN)[137]))
c <- ge_plot(subset_pheno2, ENV, GEN, GY); c
```

```
d <- ge_plot(subset_pheno2, ENV, GEN, GY, type = 2); d
```



So, now, we have a good idea about our data, we are going to the main analyses required for this project.

Part i) GxE analysis (+GGE analysis)

In this first step, we are doing some analysis to understand more the genotype x environment interaction. In addition, we are also running some GGE analysis, aiming to understand better the group of environments (mega-environments).

For the first analysis, we are using the `statgenGxE` package. Thus, we need to convert the data according to the package requirements.

```
phenoTD <- statgenSTA::createTD(data = pheno, genotype = "GEN",
                                trial = "ENV")

phenoTD2 <- statgenSTA::createTD(data = subset_pheno, genotype = "GEN",
                                  trial = "ENV")
```

Using a function from the `statgenGxE`, we are running a mixed model, where we are able to extract the variance components.

```
phenoVarComp <- gxeVarComp(TD = phenoTD, trait = "GY")
summary(phenoVarComp)
```

```
## Fitted model formula final mixed model
```

```
##
## GY ~ trial + (1 | genotype) + (1 | genotype:trial)
##
## Sources of variation for fully random model:
## GY ~ (1 | trial) + (1 | genotype) + (1 | genotype:trial)
##
##           Component % Variance expl.
## trial           2.66           52.75 %
## genotype         0.39           7.75 %
## genotype:trial    0.57          11.32 %
## residuals         1.42          28.19 %
##
## Analysis of Variance Table for fully fixed model:
## GY ~ trial + genotype + genotype:trial
##
##           Df Sum Sq Mean Sq F value    Pr(>F)
## trial           7 7579.9 1082.84 824.9253 < 2.2e-16 ***
## genotype       201 1835.0    9.13   6.9550 < 2.2e-16 ***
## genotype:trial 1407 3778.9    2.69   2.0461 < 2.2e-16 ***
## residuals     1818 2386.4    1.31
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vc(phenoVarComp)
```

```
##           Component
## genotype      0.3903468
## genotype:trial 0.5703116
## residuals      1.4205326
```

Once we have the variance components, we also can estimate the heritability for the trait.

```
herit(phenoVarComp)
```

```
## [1] 0.7091811
```

It is a little surprising to get a high heritability (0.71) for grain yield, specifically for the trials conditions from this dataset.

Something very interesting which this package permits us is to predict the genotype main effect based on the mixed model analysis.

```
predGeno <- predict(phenoVarComp)
head(predGeno)
```

```
##   genotype predictedValue
## 1         1         4.075380
## 2         3         4.829526
## 3         4         5.336766
## 4         8         5.579351
## 5         9         3.116764
## 6        10         4.544380
```

```
predGenoTrial <- predict(phenoVarComp, predictLevel = "trial")
head(predGenoTrial)
```

```
##   genotype trial predictedValue
## 1         1    E1         2.594241
## 2         3    E1         3.906923
## 3         4    E1         4.489299
## 4         8    E1         4.113286
## 5         9    E1         2.259634
## 6        10    E1         3.066884
```

In this example, we can see that for example based on the mixed model and evaluation across all the locations, the genotype 8 is the best, and the genotype 9 is the worst, based on the predictions.

Extracting the variance components, getting the heritability, being able to do predictions are main analysis in the GxE spectrum. However, we would like to go more deep, and also finding something is very useful for some research areas, as example plant breeding, mega environments. To complete this task, we are going back to the metan package.

Initially, we need to fit a gge model.

```
gge_model <- gge(pheno, ENV, GEN, GY)
```

As the statgenGxE, we also can do predictions, however, because, the models work a little different, the results will be similar, but not the same results.

```
predict(gge_model)
```

```
## $GY
## # A tibble: 202 x 8
##       E1    E2    E3    E4    E5    E6    E7    E8
## * <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  2.94  5.79  2.56  5.82  2.23  5.51  4.52  2.62
## 2  3.70  7.04  3.57  7.53  3.07  7.03  5.42  5.30
## 3  3.99  7.48  3.96  8.04  3.40  7.57  5.74  6.74
## 4  4.26  8.05  4.22  9.20  3.65  8.21  6.15  5.73
## 5  2.43  4.83  2.01  4.12  1.74  4.43  3.83  3.00
## 6  3.37  6.42  3.20  6.44  2.75  6.32  4.98  5.43
## 7  4.26  8.06  4.21  9.25  3.64  8.21  6.16  5.51
## 8  3.46  6.50  3.38  6.33  2.89  6.45  5.04  6.91
## 9  3.82  7.13  3.83  7.29  3.27  7.19  5.49  7.64
## 10 4.15  7.64  4.28  7.91  3.65  7.83  5.87  9.18
## # i 192 more rows
```

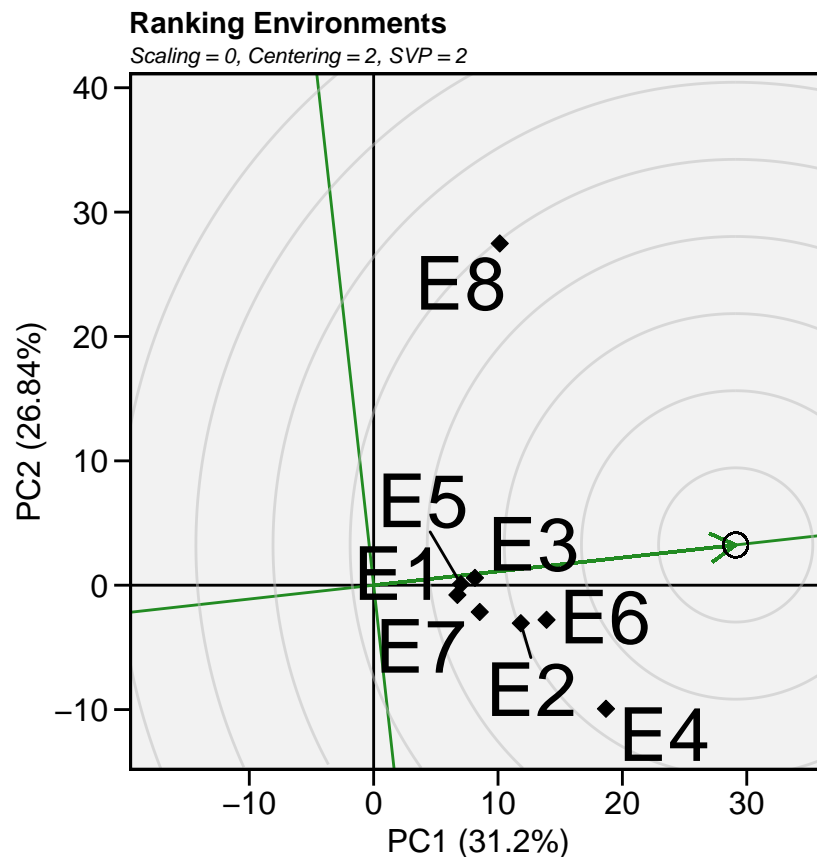
The metan package has a variety of biplots that explore the GGE (Genotype plus GxE), but, for practical purposes, we are only selecting the biplot that ranks the environments and the biplot that brings the relationship between environments.

```
biplot_ranking <- plot(gge_model,
  type = 6,
  col.gen = "black",
  col.env = "black",
```

```
size.text.env = 10,
axis_expand = 1.5)
biplot_ranking
```

```
## Warning in geom_segment(x = 0, y = 0, xend = xcoord, yend = ycoord, arrow = arrow(length = unit(0.15
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

## Warning in geom_point(aes(xcoord, ycoord), shape = 1, size = 4, color = col.stroke): All aesthetics l
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.
```



Basically, in the first biplots, what is close to the black line are considered good environments, while the environments far from the line, differ from the best environments. For example the E5 looks a very good environment, according to this line, while the E8 is not so good (drought stress). It also give to us some idea, about which environments can be grouped as on unique environment, and which ones, does not belong to the “good” pool.

Part iii) AMMI analysis

The AMMI stands for the additive main effect and multiplicative interaction. In this model, the additive main effects are the genotypes and the trials, coming from the ANOVA results, where the multiplicative interaction factors are coming from a PCA analysis on the interactions residuals (genotype by environment means after adjustment for additive genotype and environment effects).

Therefore, in the code chunk below we are running this model for GY trait, aiming to evaluate the genotypes performance across the environments, and finding the stable ones.

Also, from here, we decide, to run the analysis with a subset of genotypes, to show how the graphs look when we have a small or big number of genotypes and how it can affect our conclusions.

```
AMMI_model <- performs_amm_i(pheno, ENV, GEN, REP, GY)
```

```
## variable GY
## -----
## AMMI analysis table
## -----
##      Source   Df Sum Sq Mean Sq F value    Pr(>F) Proportion Accumulated
##      ENV      7   7580 1082.84   90.57 1.28e-07          NA          NA
##    REP(ENV)    9    108   11.96    9.49 3.08e-14          NA          NA
##      GEN    201   1835    9.13    7.25 4.44e-128          NA          NA
##    GEN:ENV 1407   3779    2.69    2.13 5.99e-52          NA          NA
##      PC1    207   2635   12.73   10.11 0.00e+00      37.9      37.9
##      PC2    205   1303    6.36    5.04 0.00e+00      18.7      56.6
##      PC3    203   1047    5.16    4.09 0.00e+00      15.0      71.6
##      PC4    201    829    4.13    3.28 0.00e+00      11.9      83.6
##      PC5    199    561    2.82    2.24 0.00e+00       8.1      91.6
##      PC6    197    360    1.83    1.45 1.00e-04       5.2      96.8
##      PC7    195    224    1.15    0.91 8.01e-01       3.2     100.0
## Residuals 1809   2279    1.26      NA      NA      NA      NA
##      Total 4840  22540    4.66      NA      NA      NA      NA
## -----
##
## All variables with significant (p < 0.05) genotype-vs-environment interaction
## Done!
```

```
AMMI_model_2 <- performs_amm_i(subset_pheno, ENV, GEN, REP, GY)
```

```
## variable GY
## -----
## AMMI analysis table
## -----
##      Source   Df Sum Sq Mean Sq F value    Pr(>F) Proportion Accumulated
##      ENV      7   373.69  53.385  69.460 4.10e-07          NA          NA
##    REP(ENV)    9    6.92   0.769   0.599 7.94e-01          NA          NA
##      GEN      9   150.26  16.696  13.009 1.38e-12          NA          NA
##    GEN:ENV    63   159.39   2.530   1.971 2.05e-03          NA          NA
##      PC1     15   125.53   8.369   6.520 0.00e+00      47.5      47.5
##      PC2     13    64.78   4.983   3.880 1.00e-04      24.5      72.0
##      PC3     11    42.23   3.839   2.990 2.20e-03      16.0      87.9
##      PC4      9    18.15   2.016   1.570 1.38e-01       6.9      94.8
##      PC5      7     7.33   1.047   0.820 5.74e-01       2.8      97.6
##      PC6      5     4.66   0.931   0.730 6.03e-01       1.8      99.3
##      PC7      3     1.81   0.603   0.470 7.04e-01       0.7     100.0
## Residuals    81   103.95   1.283      NA      NA      NA      NA
##      Total   232  1058.70   4.563      NA      NA      NA      NA
## -----
##
```

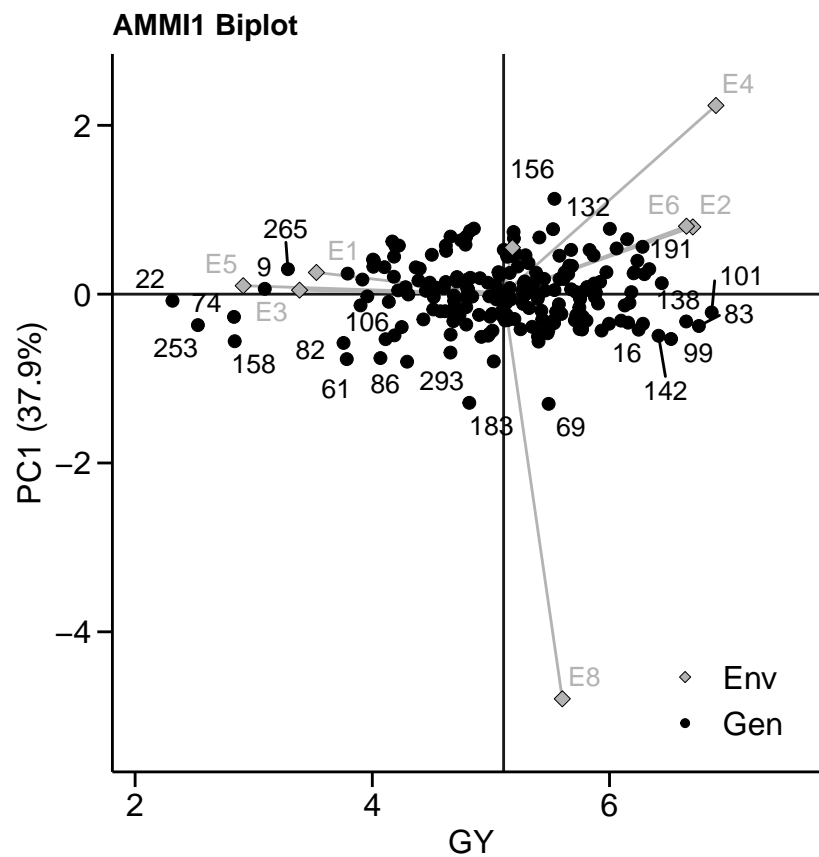
```
## All variables with significant (p < 0.05) genotype-vs-environment interaction
## Done!
```

The values clearly showed a strong genotype x environment interaction in both scenarios.

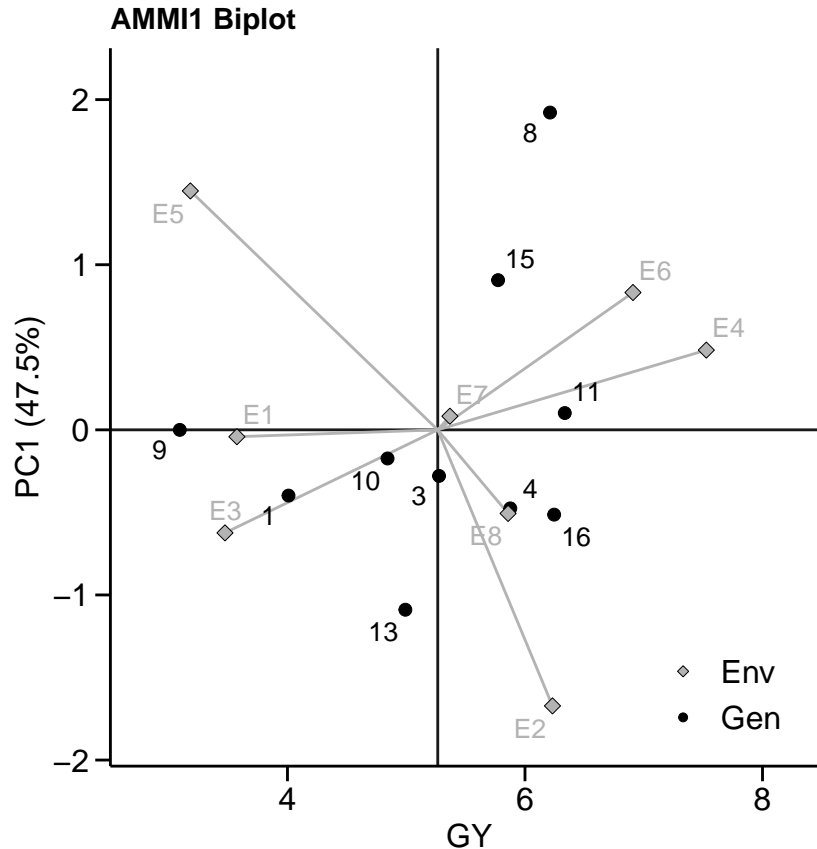
There are different ways to explore the results obtained by the AMMI model. One good way is developing a biplot, where we are using as axis our trait and the first PC.

```
AMMI_biplot_GY <- plot_scores(AMMI_model,
col.gen = "black",
col.env = "gray70",
col.segm.env = "gray70",
plot_theme = theme_metan_minimal())
AMMI_biplot_GY
```

```
## Warning: ggrepel: 181 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```
AMMI_biplot_GY2 <- plot_scores(AMMI_model_2,
col.gen = "black",
col.env = "gray70",
col.segm.env = "gray70",
plot_theme = theme_metan_minimal())
AMMI_biplot_GY2
```



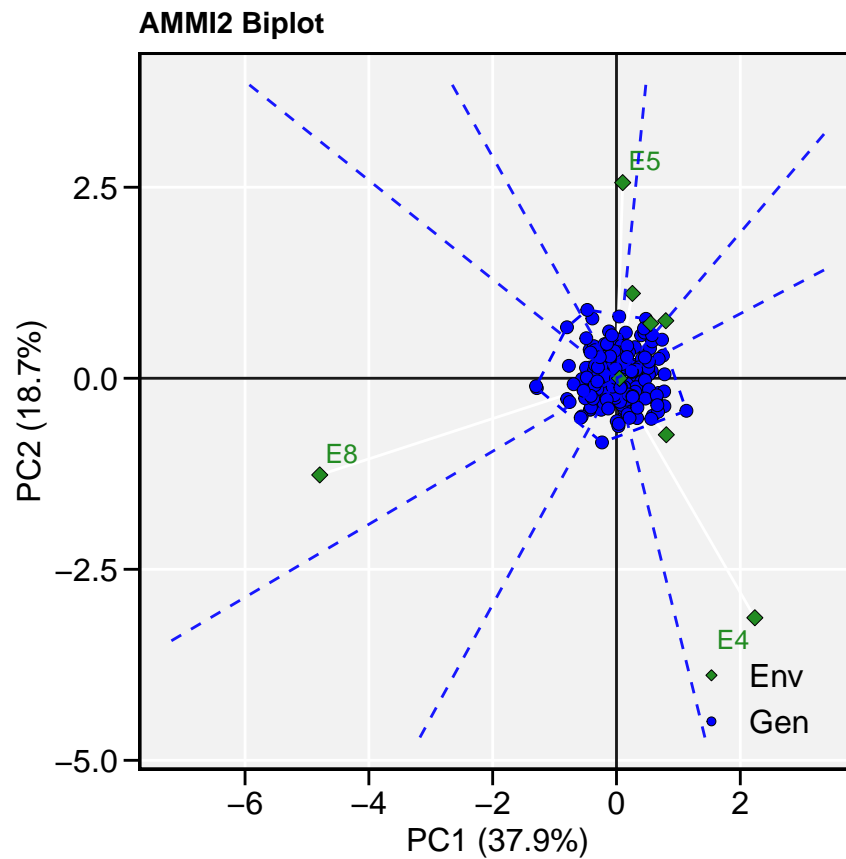
In this biplot we can take a look, from each environments are more similar, or thinking in the other side, finding the environment that have the same best lines. Also we need to pay attention in the quadrants, as well as, in the directions of our environments, because it represents the relationship between environments. For example, E6 and E2 are very similar to each other, however, they are completely different from the E5, E1, E3. The hybrid 101 and 138 apparently have high yield on the environment 6 and 2 and it is low yield in environments 5,3, and 1.

The conclusions are very similar for our small dataset, however we have some deviation about the environments correlation.

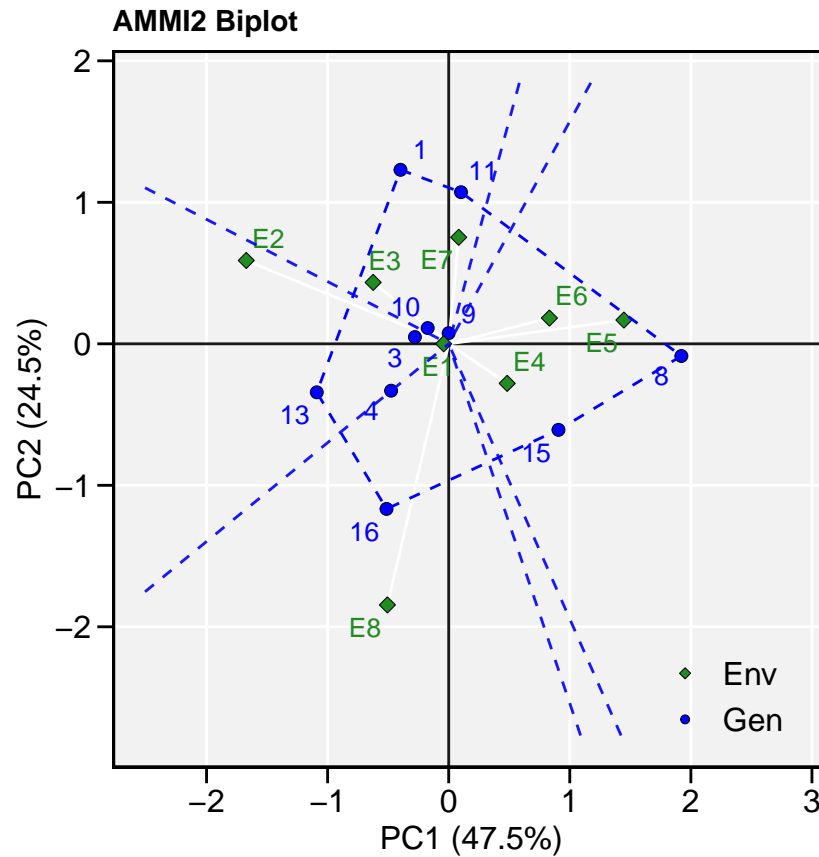
We also can check if we obtain similar results comparing PC1 and PC2.

```
AMMI_biplot_PCs <- plot_scores(AMMI_model,
  type = 2,
  polygon = T,
  col.segm.env = "#FFFFFF00", # Transparent
  axis.expand = 1.5,
  plot_theme = theme_metan(grid = "both"))
AMMI_biplot_PCs
```

```
## Warning: ggrepel: 207 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

```
AMMI_biplot_PCs_2 <- plot_scores(AMMI_model_2,
  type = 2,
  polygon = T,
  col.segm.env = "#FFFFFF00", # Transparent
  axis.expand = 1.5,
  plot_theme = theme_metan(grid = "both"))
AMMI_biplot_PCs_2
```

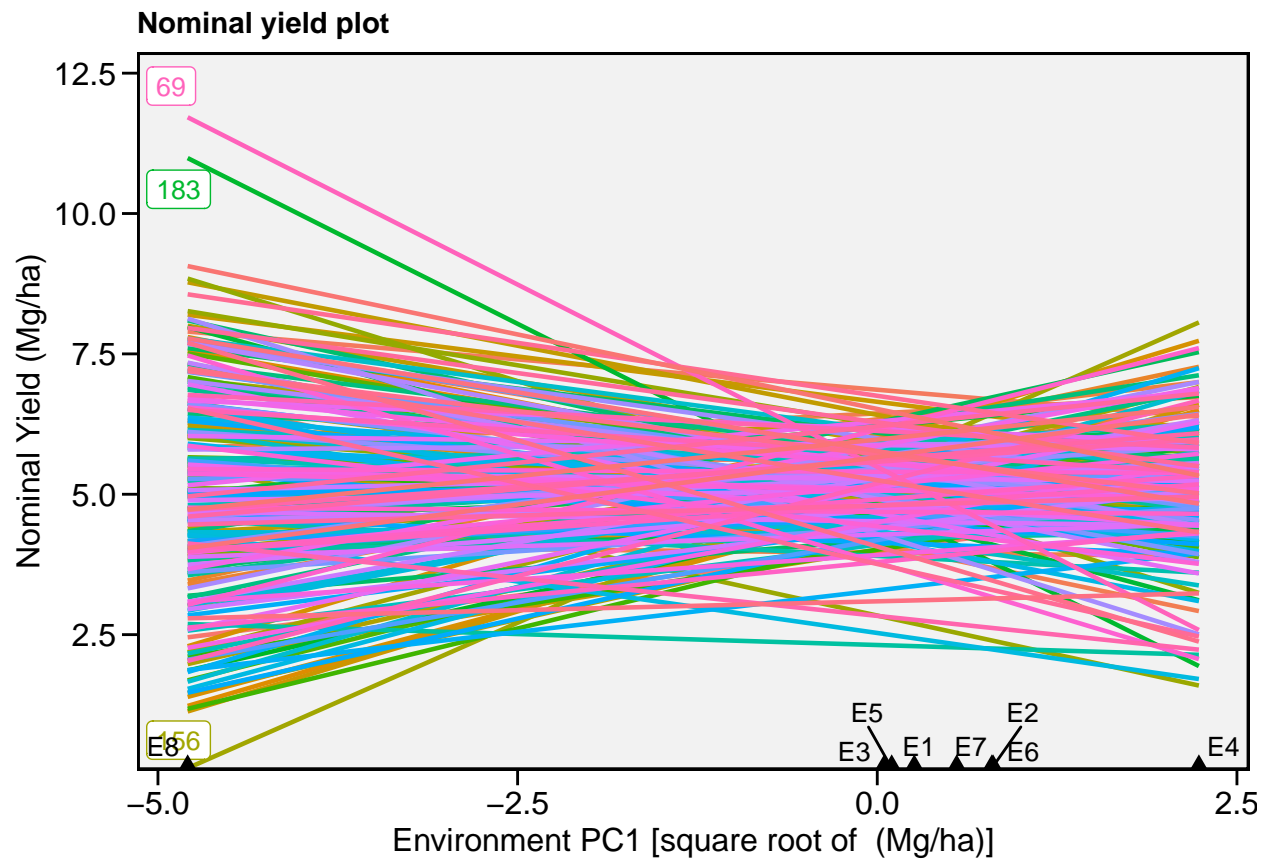


The results are pretty the same.

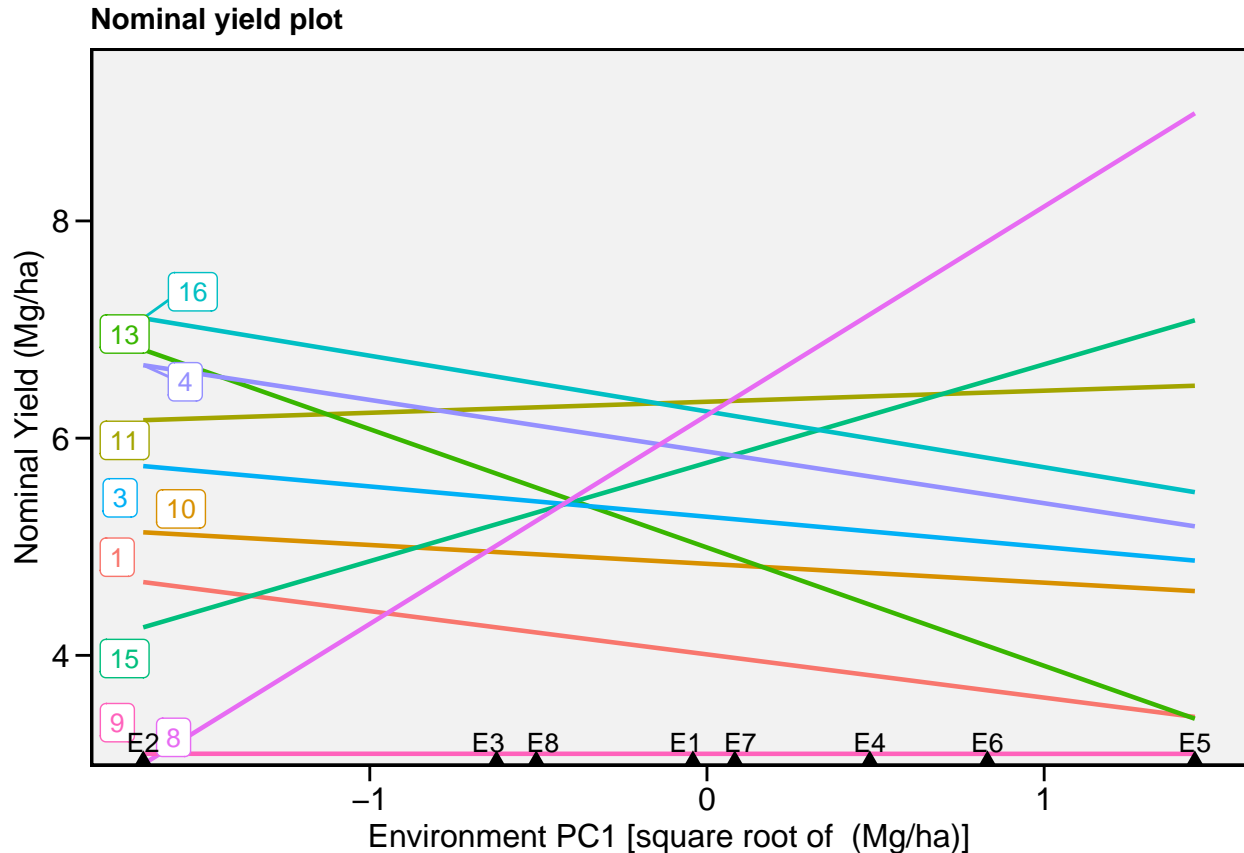
Also we can check “which-won-where” based on our AMMI model.

```
AMMI_biplot_nominal <- plot_scores(AMMI_model, type = 4)
AMMI_biplot_nominal
```

```
## Warning: ggrepel: 199 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```
AMMI_biplot_nominal_2 <- plot_scores(AMMI_model_2, type = 4)
AMMI_biplot_nominal_2
```



In these plots, we can observe the change in rankings, as example, in the first plot where the line 69 is a good one in the E8, however, its rankings drops drastically across better environments.

Part iii) GGI analysis

The GGI stands for genotype by group interaction. The main differences from the previous analysis is that now era grouping some hybrids according to their similarities, considering all the traits, and estimating group-by-environment interactions.

So, the package uses a function to cluster genotypes based on their means for all the traits. Again, here, we are using the pheno and subset_pheno.

```
mean_gen <-
pheno %>%
means_by(GEN) %>%
column_to_rownames("GEN")
```

```
## Warning: 'mean_by()' is deprecated as of metan 1.17.0
## Please use 'mean_by()' instead
```

```
d2 <- clustering(mean_gen)
```

```
mean_gen_2 <-
subset_pheno %>%
```

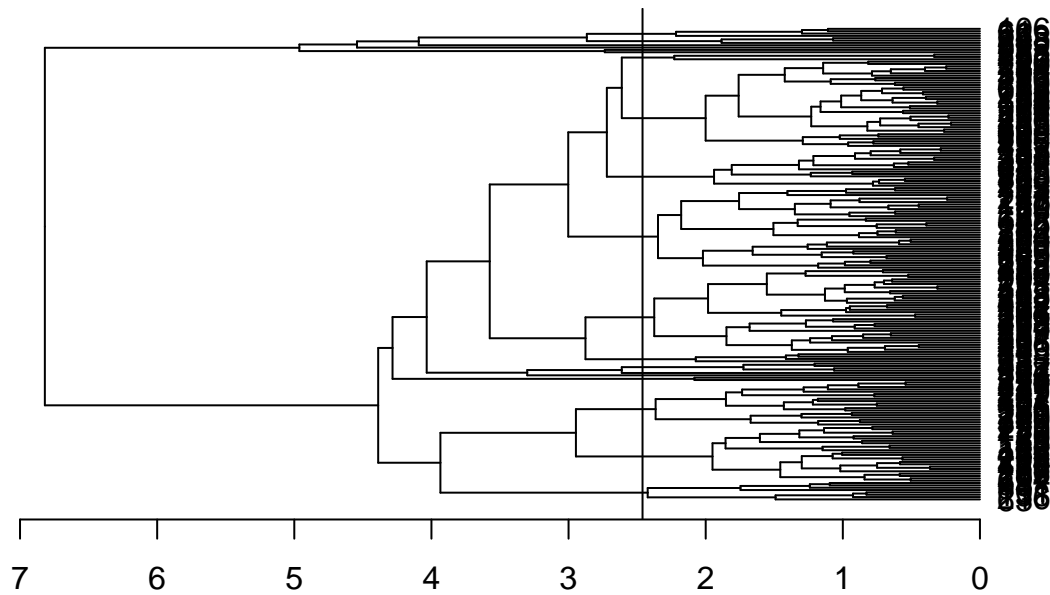
```
means_by(GEN) %>%
column_to_rownames("GEN")
```

```
## Warning: 'mean_by()' is deprecated as of metan 1.17.0
## Please use 'mean_by()' instead
```

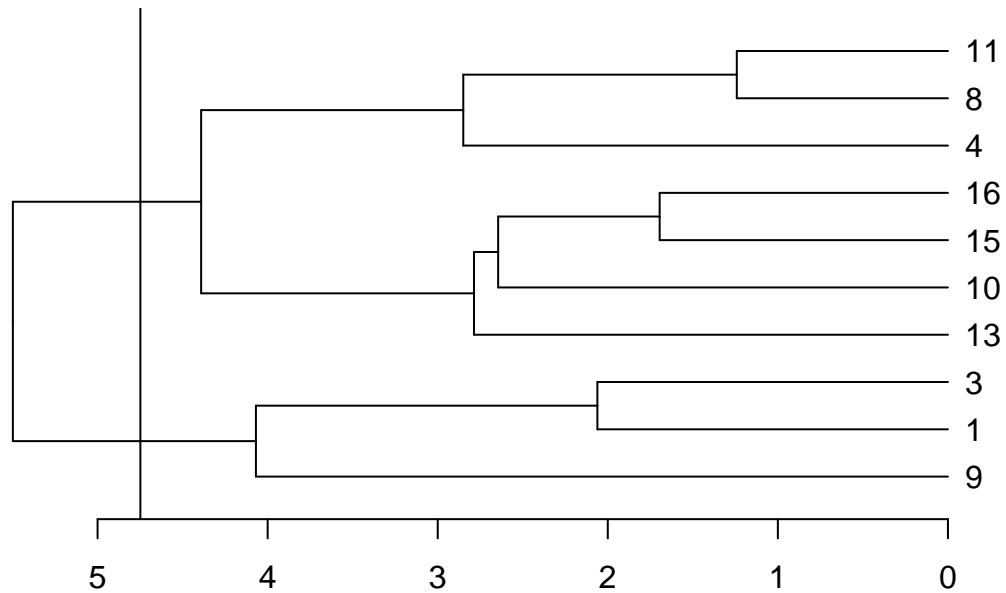
```
d2_2 <- clustering(mean_gen_2)
```

After clustering the genotypes, we can plot this cluster in a dendrogram shape, also, to identify how many cluster we got in this analysis.

```
plot(d2)
```



```
plot(d2_2)
```



For the biggest dataset, we have a lot of cluster, meaning that small number of genotypes in each cluster, while in the subset pheno, based on the threshold, we only have two cluster.

One big problem of this clustering analysis is that we do not know if all the traits are in fact contributing for the cluster building, so, we need to find each ones are important and each ones can be removed.

```
sel_var <- clustering(mean_gen, selvar = TRUE)
```

```
## ASI excluded in this step |===== | 25% GY excluded in this s
## -----
##
## Summary of the adjusted models
## -----
##   Model excluded cophenetic remaining cormantel   pvmantel
## Model 1      -  0.6965867           5 1.0000000 0.000999001
## Model 2     ASI  0.7051845           4 0.9894093 0.000999001
## Model 3      GY  0.7142594           3 0.9869084 0.000999001
## Model 4     MFT  0.7329440           2 0.9616288 0.000999001
## -----
## Suggested variables to be used in the analysis
## -----
## The clustering was calculated with the  Model 4
## The variables included in this model were...
##   FFT EPP
## -----
```

```
sel_var2 <- clustering(mean_gen_2, selvar = TRUE)
```

```
## ASI excluded in this step |===== | 25% GY excluded in this s
## -----
##
## Summary of the adjusted models
## -----
##      Model excluded cophenetic remaining cormantel   pvmantel
## Model 1      -   0.6928961           5 1.0000000 0.000999001
## Model 2      ASI  0.7164971           4 0.9864986 0.000999001
## Model 3      GY   0.7426920           3 0.9756303 0.000999001
## Model 4      MFT  0.7307509           2 0.9710941 0.000999001
## -----
## Suggested variables to be used in the analysis
## -----
## The clustering was calculated with the   Model 3
## The variables included in this model were...
##   FFT MFT EPP
## -----
```

For the biggest dataset, only FFT and EPP were selected, while for the small dataset: FFT, MFT and EPP were selected. Based on these clusters, we can extract implement this information in other analysis and see how this group information can be used in favor of the researcher.

Part iv) FW analysis

The FW stands for Finlay-Wilkinson analysis, where the GxE interaction is estimated heterogeneity of the slopes of a regression of individual genotypic performance on an environmental index.

```
phenoFW <- gxeFw(TD = phenoTD, trait = "GY")
summary(phenoFW)
```

```
## Environmental effects
## =====
##      Trial      EnvEff SE_EnvEff EnvMean SE_EnvMean Rank
## 1      E1 -1.5626337 0.05086093 3.539316 0.4311904    6
## 2      E2  1.5685308 0.05996941 6.667806 0.5341492    3
## 3      E3 -1.6566421 0.05086093 3.445388 0.4449041    7
## 4      E4  1.8280238 0.08139299 6.927077 0.5805836    1
## 5      E5 -2.2546431 0.05996941 2.847898 0.5429888    8
## 6      E6  1.5963523 0.05996941 6.695603 0.5390234    2
## 7      E7  0.0621412 0.05086093 5.162703 0.3412076    5
## 8      E8  0.4188708 0.08139299 5.519128 0.3689585    4
##
## Anova
## =====
##              Df  Sum Sq Mean Sq  F value    Pr(>F)
## Trial           7  7579.9  1082.84  581.7270 < 2.2e-16 ***
## Genotype      201   1835.0     9.13   4.9046 < 2.2e-16 ***
## Sensitivities 201    536.4     2.67   1.4336 0.000101 ***
## Residual     3024   5628.9     1.86
```

```
## Total          3433 15580.2    4.54
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Most sensitive genotypes
## =====
##   Genotype  GenMean SE_GenMean Rank      Sens    SE_Sens MSdeviation
##      66 5.681267  0.3380492   1 1.710121 0.2224153   1.881192
##     196 5.074181  0.3380492   2 1.680427 0.2224153   1.569990
##      40 6.147283  0.3380492   3 1.618015 0.2224153   3.181711
##     222 5.626621  0.3380492   4 1.530812 0.2224153   1.975627
##      63 5.269591  0.3380492   5 1.477241 0.2224153   1.299583
```

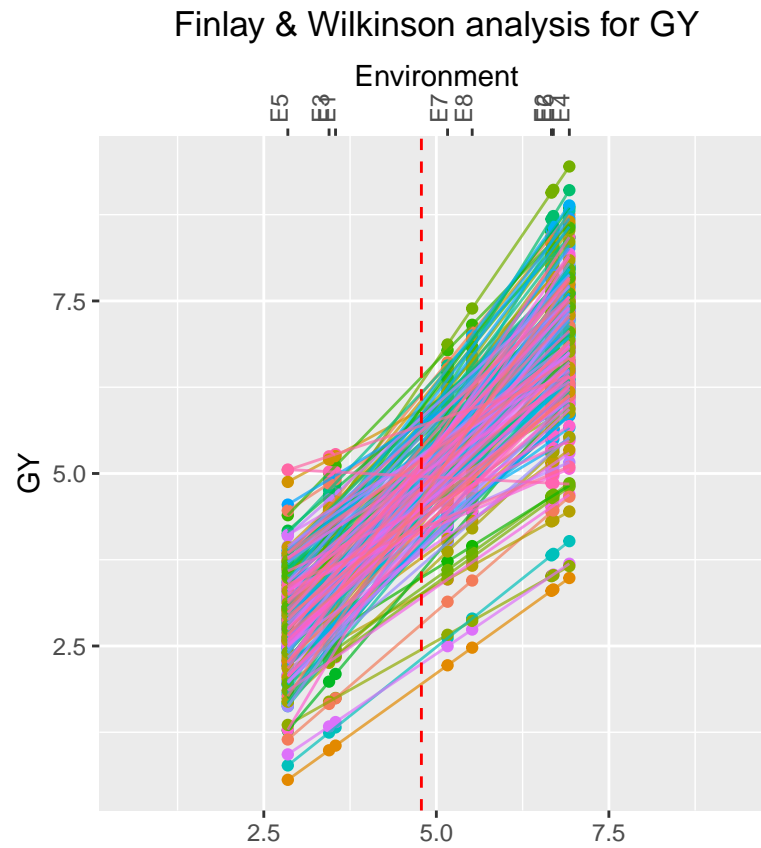
```
phenoFW2 <- gxeFw(TD = phenoTD2, trait = "GY")
summary(phenoFW2)
```

```
## Environmental effects
## =====
##   Trial      EnvEff SE_EnvEff  EnvMean SE_EnvMean Rank
##  1    E1 -1.7015057 0.2325573 3.561427  0.4326692   6
##  2    E2  1.0360834 0.2742051 6.296761  0.4415681   3
##  3    E3 -1.7645439 0.2325573 3.498441  0.4419319   7
##  4    E4  2.1428439 0.3721626 7.402610  0.6306224   1
##  5    E5 -2.1220884 0.2742051 3.141190  0.4983030   8
##  6    E6  1.6418817 0.2742051 6.902061  0.5400925   2
##  7    E7  0.1025548 0.2325573 5.364002  0.3366600   5
##  8    E8  0.6647743 0.3721626 5.925758  0.3907917   4
##
## Anova
## =====
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Trial              7 373.69  53.385 29.2191 < 2.2e-16 ***
## Genotype           9 150.26  16.696  9.1383 7.13e-11 ***
## Sensitivities      9   7.17   0.796  0.4358  0.9139
## Residual          144 263.09   1.827
## Total            169 794.21   4.699
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Most sensitive genotypes
## =====
##   Genotype  GenMean SE_GenMean Rank      Sens    SE_Sens MSdeviation
##      4 6.586575  0.3306879   1 1.205905 0.2165685   1.5558270
##      8 5.032008  0.3306879   2 1.199806 0.2165685   1.8503382
##     10 6.089283  0.3306879   3 1.164750 0.2165685   1.3596097
##      9 5.684087  0.3306879   4 1.027283 0.2165685   2.0816706
##     11 5.208903  0.3306879   5 1.001658 0.2165685   0.9558142
```

It is interesting that we in the FW output, we can obtain also environment rankings based on this model, as well as, getting some environment effects per se and in environments effects on the hybrids.

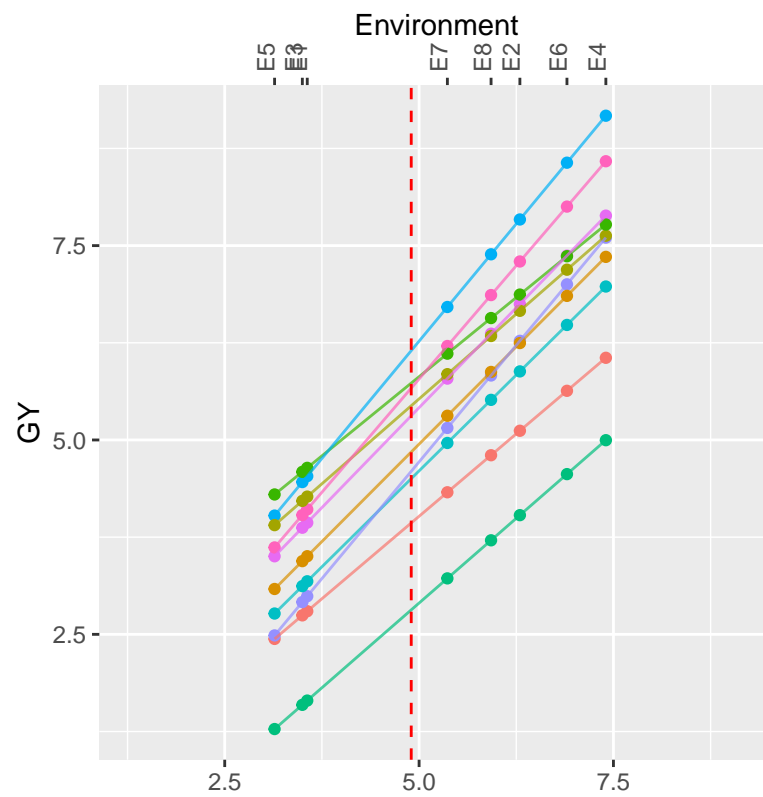
Additionally, we can also plot, to take a look how the hybrids performed across the environments.


```
plot(phenoFW, plotType = "line")
```



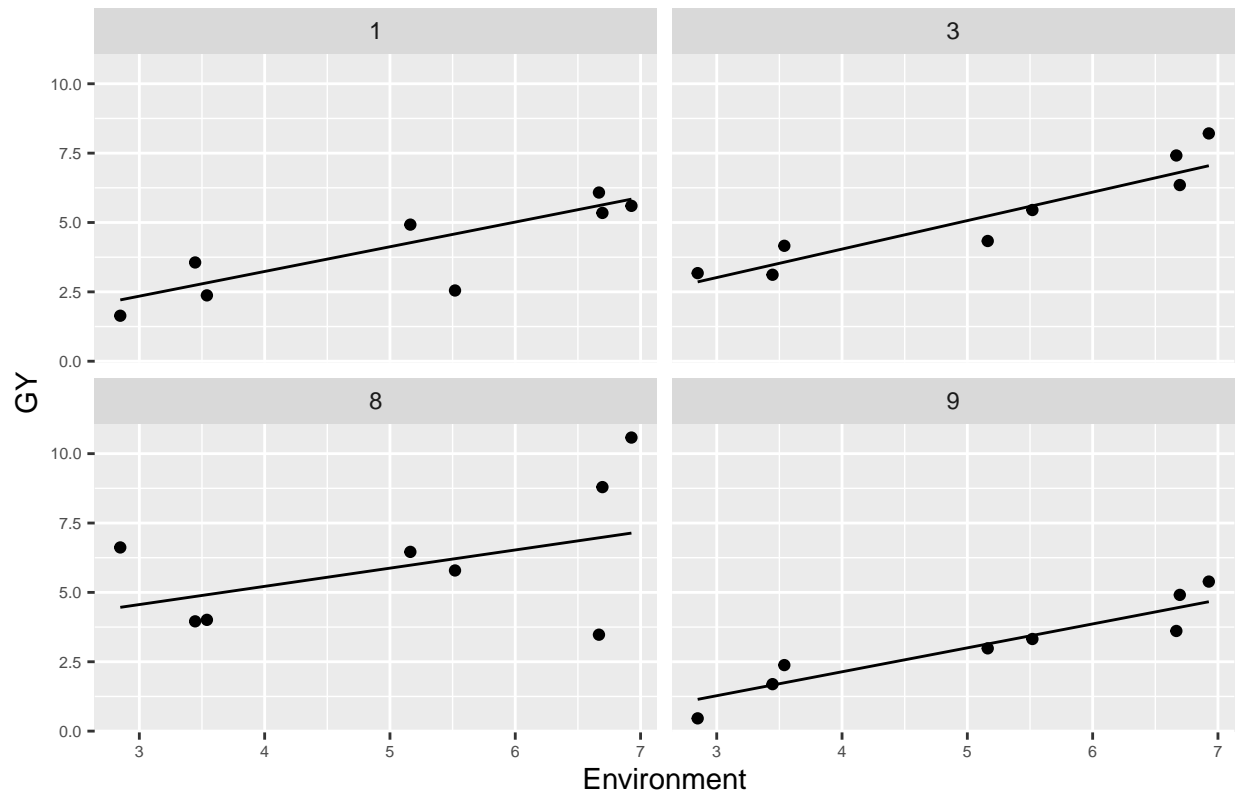
```
plot(phenoFW2, plotType = "line")
```

Finlay & Wilkinson analysis for GY



```
plot(phenoFW, plotType = "trellis", genotypes = c("1", "3", "8", "9"))
```

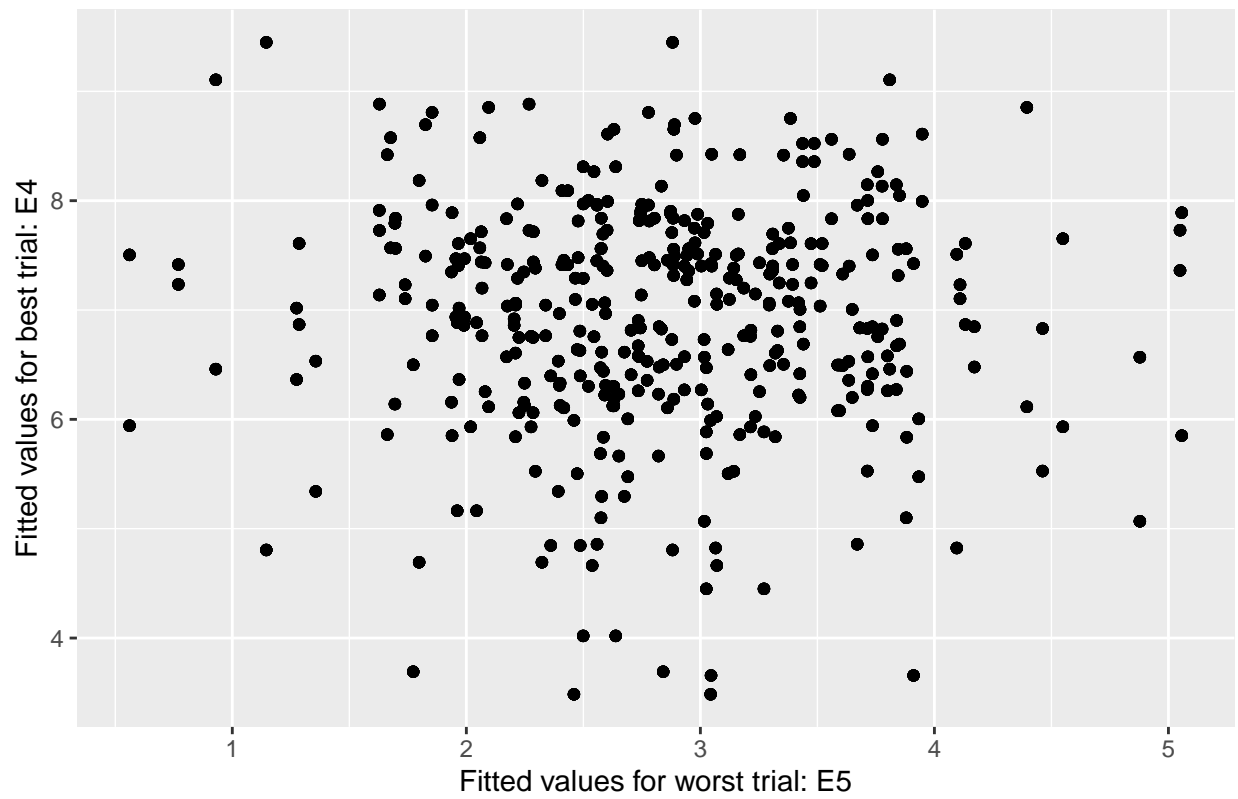
Finlay & Wilkinson analysis for GY



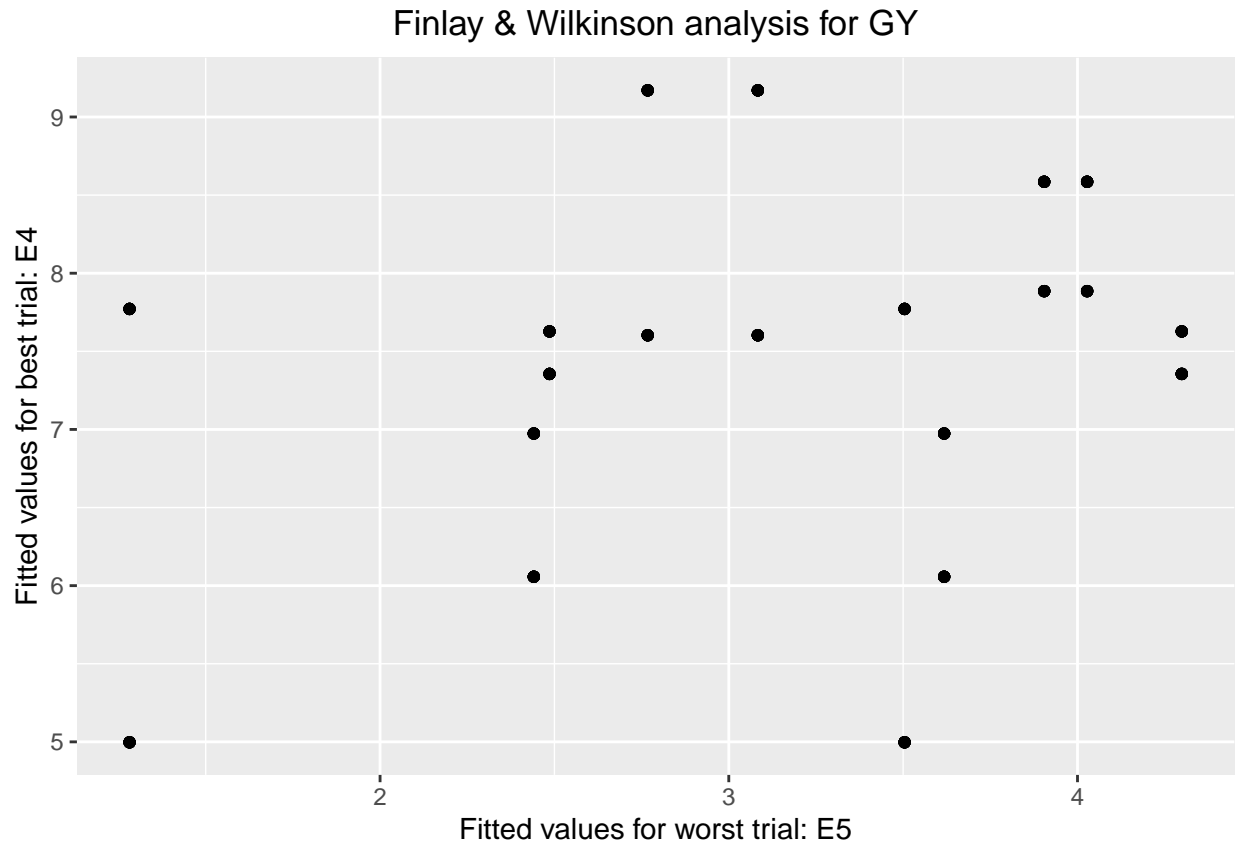
Also, it is possible check the hybrids performance through a scatter plot, comparing their distribution in the best and worst environment.

```
plot(phenoFW, plotType = "scatterFit")
```

Finlay & Wilkinson analysis for GY



```
plot(phenoFW2, plotType = "scatterFit")
```



Part v) Genomic selection

For this part, it is important to reminder that we are using a new dataset that fits better for the genomic selection analysis.

Checking data stuctures

Before running any analysis, it is important to check the data structures, to avoid mistakes and warnings in the R code.

```
str(Data.Maize)
```

```
## 'data.frame':  927 obs. of  6 variables:
## $ Gid  : Factor w/ 309 levels "CKDHL0002","CKDHL0003",...: 1 2 3 4 5 6 7 8 9 10 ...
## $ Env  : Factor w/ 3 levels "EBU","KAK","KTI": 1 1 1 1 1 1 1 1 1 1 ...
## $ Rep  : int   1 1 1 1 1 1 1 1 1 1 ...
## $ Yield: num   6.65 6.1 5.07 6.55 6.82 6.88 6.34 7.09 5.28 6.07 ...
## $ ASI  : num   1.4 1.7 1.6 2.1 2 2.7 2 1.7 2 1.9 ...
## $ PH   : num   2.48 2.45 2.39 2.31 2.28 2.26 2.46 2.39 2.42 2.29 ...
```

```
str(Gg)
```

```
## num [1:309, 1:309] 0.2205 0.1318 0.1392 0.0858 0.1243 ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:309] "V1" "V2" "V3" "V4" ...
## ..$ : chr [1:309] "V1" "V2" "V3" "V4" ...
```

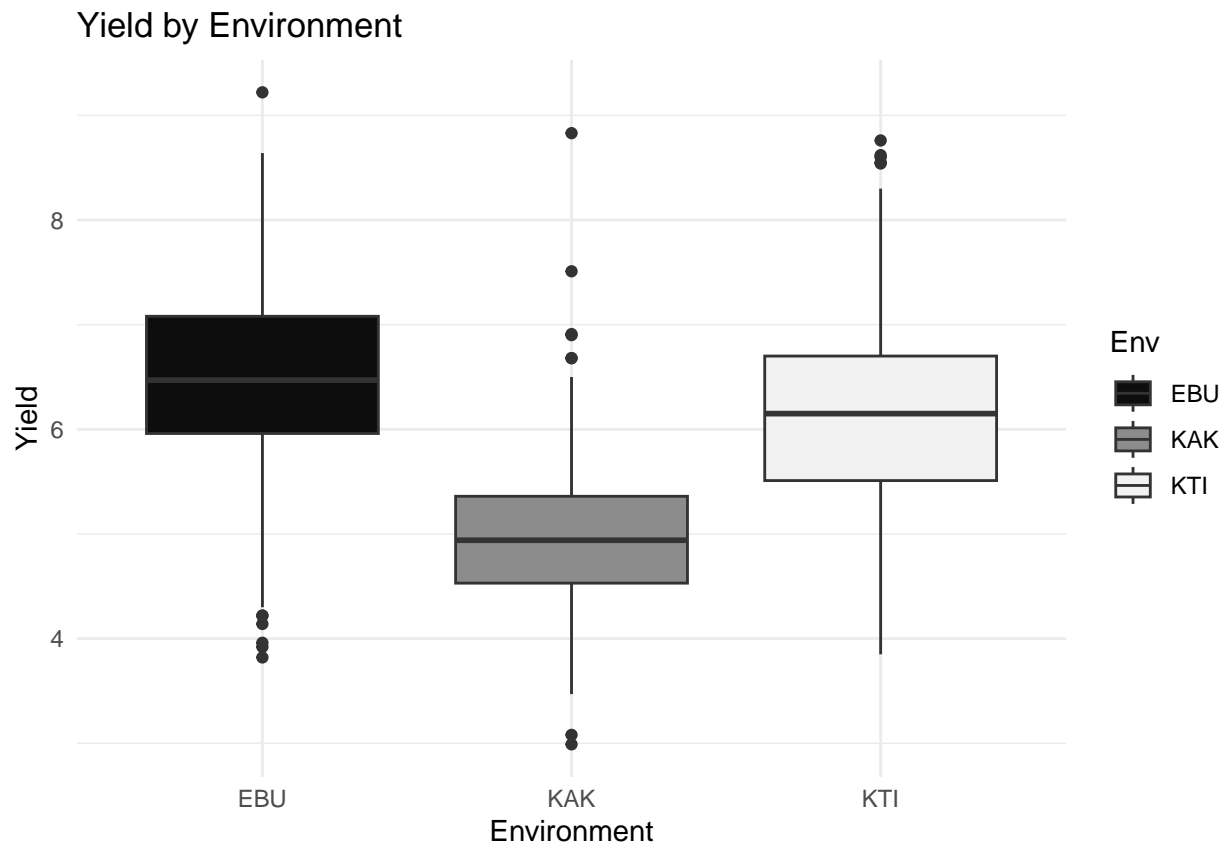
In this case, the REP column is not a factor, so, we need to fix it.

```
Data.Maize$Rep <- as.factor(Data.Maize$Rep)
str(Data.Maize)
```

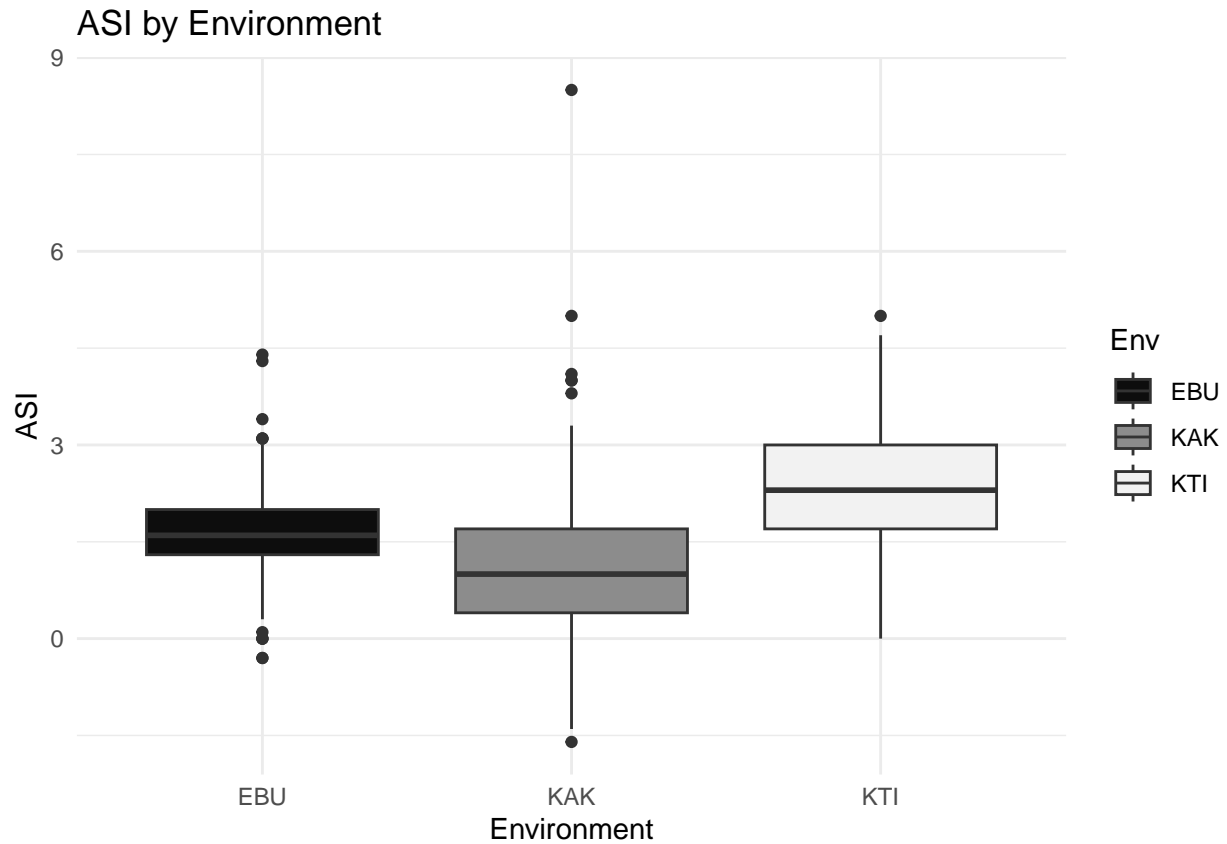
```
## 'data.frame': 927 obs. of 6 variables:
## $ Gid : Factor w/ 309 levels "CKDHL0002","CKDHL0003",...: 1 2 3 4 5 6 7 8 9 10 ...
## $ Env : Factor w/ 3 levels "EBU","KAK","KTI": 1 1 1 1 1 1 1 1 1 1 ...
## $ Rep : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
## $ Yield: num 6.65 6.1 5.07 6.55 6.82 6.88 6.34 7.09 5.28 6.07 ...
## $ ASI : num 1.4 1.7 1.6 2.1 2 2.7 2 1.7 2 1.9 ...
## $ PH : num 2.48 2.45 2.39 2.31 2.28 2.26 2.46 2.39 2.42 2.29 ...
```

Checking data distribution

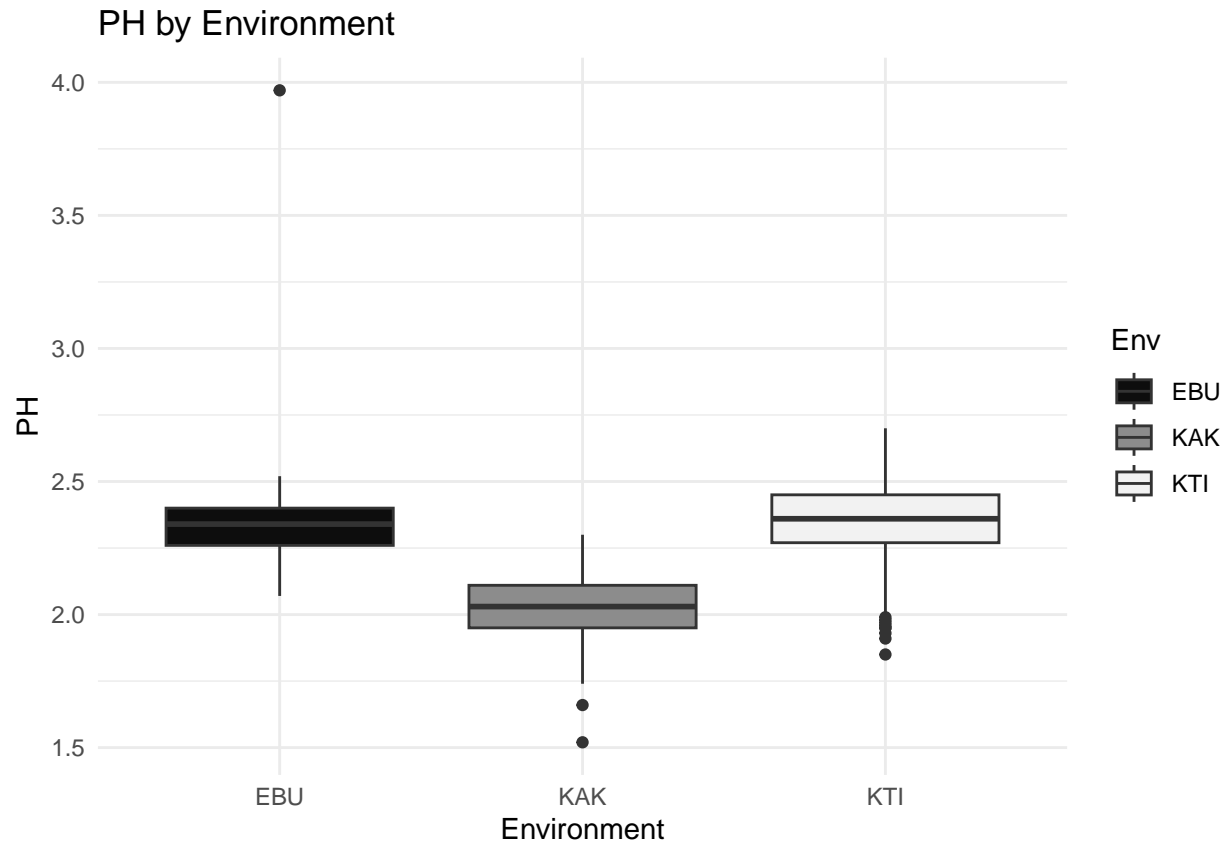
```
ggplot(Data.Maize, aes(x = Env, y = Yield, fill = Env)) +
  geom_boxplot() +
  labs(title = "Yield by Environment", x = "Environment", y = "Yield") +
  scale_fill_manual(values = c("grey5", "grey55", "grey95")) +
  theme_minimal()
```



```
ggplot(Data.Maize, aes(x = Env, y = ASI, fill = Env)) +
  geom_boxplot() +
  labs(title = "ASI by Environment", x = "Environment", y = "ASI") +
  scale_fill_manual(values = c("grey5", "grey55", "grey95")) +
  theme_minimal()
```



```
ggplot(Data.Maize, aes(x = Env, y = PH, fill = Env)) +
  geom_boxplot() +
  labs(title = "PH by Environment", x = "Environment", y = "PH") +
  scale_fill_manual(values = c("grey5", "grey55", "grey95")) +
  theme_minimal()
```



Correlation between environments

```
yield_wide <- Data.Maize %>%
  select(Gid, Env, Yield) %>%
  spread(key = Env, value = Yield)

correlation_matrix <- yield_wide %>%
  select(-Gid) %>%
  cor(use = "complete.obs")

print(correlation_matrix)
```

```
##           EBU          KAK          KTI
## EBU 1.0000000 0.1747700 0.1809038
## KAK 0.1747700 1.0000000 0.2113454
## KTI 0.1809038 0.2113454 1.0000000
```

Correlation between traits

```
compute_correlations <- function(env_data) {
  cor(env_data %>% select(Yield, ASI, PH), use = "complete.obs")
}
```



```

}

correlations_by_env <- Data.Maize %>%
  group_by(Env) %>%
  summarise(correlation_matrix = list(compute_correlations(cur_data()))))

## Warning: There was 1 warning in 'summarise()'.
## i In argument: 'correlation_matrix = list(compute_correlations(cur_data()))'.
## i In group 1: 'Env = EBU'.
## Caused by warning:
## ! 'cur_data()' was deprecated in dplyr 1.1.0.
## i Please use 'pick()' instead.

```

```

correlations_by_env %>%
  filter(Env == "EBU") %>%
  pull(correlation_matrix) %>%
  .[[1]]

```

```

##           Yield      ASI      PH
## Yield  1.0000000 -0.1461431  0.1637575
## ASI   -0.1461431  1.0000000 -0.1200107
## PH     0.1637575 -0.1200107  1.0000000

```

```

correlations_by_env %>%
  filter(Env == "KAK") %>%
  pull(correlation_matrix) %>%
  .[[1]]

```

```

##           Yield      ASI      PH
## Yield  1.0000000 -0.09744938  0.3669753
## ASI   -0.09744938  1.00000000 -0.1638430
## PH     0.36697528 -0.16384300  1.0000000

```

```

correlations_by_env %>%
  filter(Env == "KTI") %>%
  pull(correlation_matrix) %>%
  .[[1]]

```

```

##           Yield      ASI      PH
## Yield  1.0000000 -0.2200647  0.4091269
## ASI   -0.2200647  1.00000000 -0.2222470
## PH     0.4091269 -0.2222470  1.0000000

```

Fixing data strcture (again)

```

Data.Maize <- (Data.Maize[order(Data.Maize$Env,Data.Maize$Gid),])
rownames(Data.Maize)=1:nrow(Data.Maize)
head(Data.Maize)

```

```
##          Gid Env Rep Yield ASI   PH
## 1 CKDHL0002 EBU   1  6.65 1.4 2.48
## 2 CKDHL0003 EBU   1  6.10 1.7 2.45
## 3 CKDHL0004 EBU   1  5.07 1.6 2.39
## 4 CKDHL0005 EBU   1  6.55 2.1 2.31
## 5 CKDHL0007 EBU   1  6.82 2.0 2.28
## 6 CKDHL0008 EBU   1  6.88 2.7 2.26
```

Design of matrices

```
LG <- cholesky(Gg)
ZG <- model.matrix(~0 + as.factor(Data.Maize$Gid))
Z.G <- ZG %*% LG
Z.E <- model.matrix(~0 + as.factor(Data.Maize$Env))
ZEG <- model.matrix(~0 + as.factor(Data.Maize$Gid):as.factor(Data.Maize$Env))
G2 <- kronecker(diag(length(unique(Data.Maize$Env))), data.matrix(Gg))
LG2 <- cholesky(G2)
Z.EG <- ZEG %*% LG2
Y <- as.matrix(Data.Maize[, -c(1, 2, 3)])
```

Fitting the model

```
fm <- BMTME(Y = Y, X = Z.E, Z1 = Z.G, Z2 = Z.EG,
#nIter =15000, burnIn =10000, thin = 2, bs = 50)
nIter =150, burnIn =100, thin = 2, bs = 50)
```

Extracting covariances

```
COV_TraitGenetic <- fm$varTrait
COV_TraitGenetic
```

```
##          Yield    ASI    PH
## [1,] 0.1976 0.0715 0.0281
## [2,] 0.0715 0.3492 0.0121
## [3,] 0.0281 0.0121 0.0192
```

Correlation matrix

```
COR_TraitGenetic <- cov2cor(COV_TraitGenetic)
COR_TraitGenetic
```

```
##          Yield    ASI    PH
## [1,] 1.0000000 0.2721921 0.4562073
## [2,] 0.2721921 1.0000000 0.1477739
## [3,] 0.4562073 0.1477739 1.0000000
```

Correlation matrix between environments

```
COV_EnvGenetic <- fm$varEnv  
COV_EnvGenetic
```

```
##           EBU      KAK      KTI  
## [1,] 0.6045 0.4962 0.3809  
## [2,] 0.4962 0.5320 0.2929  
## [3,] 0.3809 0.2929 0.4278
```

Correlation matrix between traits

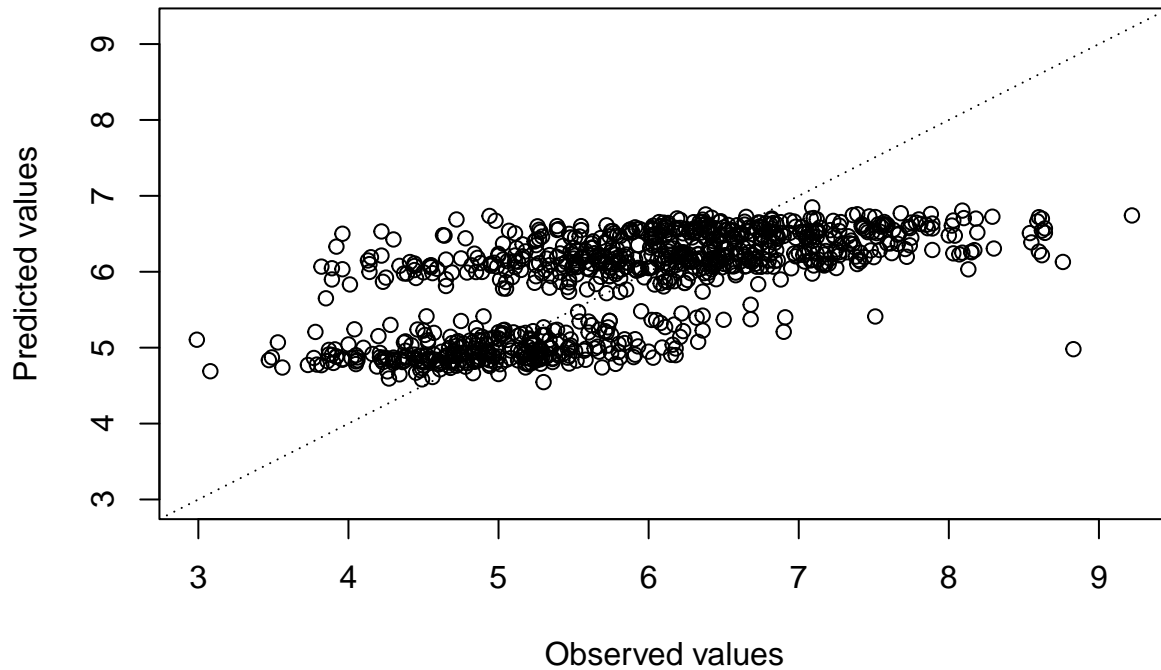
```
COV_ResGenetic <- fm$vare  
COV_ResGenetic
```

```
##           Yield      ASI      PH  
## [1,] 0.6701 -0.1196 0.0274  
## [2,] -0.1196 0.6601 -0.0187  
## [3,] 0.0274 -0.0187 0.0143
```

Predictions

```
plot(fm, trait="Yield")
```

BMTME fitted model in the trait Yield



Cross validation

```
pheno <- data.frame(GID = Data.Maize[, 1], Env =
Data.Maize[, 2], Response = Data.Maize[, 4])
CrossV <- CV.KFold(pheno, DataSetID = "GID", K = 5,
set_seed = 123)
pm <- BMTME(Y = Y, X = Z.E, Z1 = Z.G, Z2 = Z.EG,
nIter = 250, burnIn = 50, thin = 2, bs = 50, testingSet = CrossV)
```

Final results

```
summary(pm)
```

##	Environment	Trait	Pearson	SE_Pearson	MAAPE	SE_MAAPE
## 1	EBU	ASI	0.4679	0.0323	0.3276	0.0172
## 2	EBU	PH	0.2678	0.0458	0.0337	0.0020
## 3	EBU	Yield	0.2995	0.0438	0.1105	0.0059
## 4	KAK	ASI	0.3410	0.0681	0.6133	0.0201
## 5	KAK	PH	0.3053	0.0552	0.0441	0.0023
## 6	KAK	Yield	0.3162	0.0519	0.1033	0.0054
## 7	KTI	ASI	0.1808	0.0592	0.2866	0.0122

## 8	KTI	PH	0.4273	0.0382	0.0470	0.0035
## 9	KTI	Yield	0.2438	0.0437	0.1249	0.0041

References

- Crossa, J., Y. Beyene, S. Kassa, P. Pérez-Rodríguez, J. M. Hickey, et al., 2013 Genomic prediction in maize breeding populations with genotyping-by-sequencing. *G3: Genes/Genomes/Genetics (Bethesda)* 3, 1903–1926. <https://doi.org/10.1534/g3.113.008227>
- Dias, K.O.D.G., Gezan, S.A., Guimarães, C.T. et al. Improving accuracies of genomic predictions for drought tolerance in maize by joint modeling of additive and dominance effects in multi-environment trials. *Heredity* 121, 24–37 (2018). <https://doi.org/10.1038/s41437-018-0053-6>
- Montesinos-López, O. A., A. Montesinos-López, J. Crossa, F. Toledo, O. Pérez-Hernández et al., 2016 A Genomic Bayesian Multi-trait and Multi-environment model. *G3: Genes/Genomes/Genetics (Bethesda)*, 6:2725–2744. <https://doi.org/10.1534/g3.116.032359>
- Montesinos-López, O. A., A. Montesinos-López, J. Crossa, J. C. Montesinos-López, F. J. Luna-Vázquez et al., 2017 A Variational Bayes Genomic-Enabled Prediction Method with Genotype · Environment Interaction. *G3: Genes, Genomes. Genetics* 7: 1833–1853
- Oval A Montesinos-López, Abelardo Montesinos-López, Francisco Javier Luna-Vázquez, Fernando H Toledo, Paulino Pérez-Rodríguez, Morten Lillemo, José Crossa, An R Package for Bayesian Analysis of Multi-environment and Multi-trait Multi-environment Data for Genome-Based Prediction, *G3 Genes/Genomes/Genetics*, Volume 9, Issue 5, 1 May 2019, Pages 1355–1369, <https://doi.org/10.1534/g3.119.400126>