

Douglas Avenir - Test de provenances Californiennes de l'Homol - Analyses des données de résistographe

oughajjibrahim@gmail.com

31 aout 2021

Contents

1	Préliminaires	1
2	Objectifs du script	2
3	Import des données hors résistographe	2
3.1	Données de résistographe	4
4	Préparation des données du résistographe	6
4.1	Suppression de l'écorce.....	6
4.2	Calculs de moyennes des profils	7
4.3	Premiers graphiques descriptifs des données de résisto	7
4.4	Gestion des paramètres de vitesses	10
5	Analyse des valeurs moyennes des sondages au résistographe	15
5.1	Description des données moyennes.....	15
5.2	Comparaisons entre sélectionnés et non sélectionnés	18
6	MP YR 09/0/2017	18
7	Pour la plénière	18
8	Export des données	21

1 Préliminaires

```
# cleaning the workspace
```

```
rm(list = ls())
```

```
# couleurs perso
```

```
couleurs <-rep(c("black", "#E41A1C", "#4C00FFFF", "#4DAF4A", "#A65628",  
                "#FF7F00", "gold", "#984EA3", "#F781BF", "#999999", "#00E5FFFF",  
                "#A0D600FF", "#EBB25EFF", "#377EB8", "darkgreen"),  
              10)
```

```
library(reshape2) # commentaire
```

```
# library(YvesR)
```

```
library(lattice)
```

```
# library(lsmeans)
```

```

# library(multcompView)

# library(lme4)

# function to compute the sample size :
taille.ech <- function(x){
  sum(!is.na(x))
}

# Fonction qui estime une moyenne mais avec `na.rm = T` par défaut
moy_narm <- function(x){
  mean(x, na.rm = T)
}

# Fonction qui permet de centrer-réduire un vecteur :
centre_reduit <- function(x){
  return( (x - mean(x, na.rm = T)) / sd(x, na.rm = T) )
}

```

2 Objectifs du script

Ce script a pour objectif d'analyser les données issues des sondages au résistographe réalisés en mars 2017 lors de la récolte de greffons pour le projet Douglas Avenir.

Tous les arbres du dispositif n'ont pas été sondés à l'aide du résistographe. Les sondages ont été réalisés dans le cadre du projet Douglas Avenir et donc les questions sont orientées dans la cadre de la sélection réalisée.

Les objectifs sont les suivants :

1. Voir si les arbres sélectionnés ont des densités différentes des arbres non sélectionnés. La comparaison avec un "témoin" sera aussi réalisée.
2. Voir si les densités sont différentes au sein des arbres sélectionnés et si une sélection sur ce caractère peut être réalisée.
3. Regarder les liens possibles entre la densité et les autres caractères.

NB : Le résistographe permet d'avoir un profil continu des 2 variables qu'il enregistre : la force nécessaire pour tourner et celle nécessaire pour avancer. **Je n'exploite pas le profil entier mais juste une valeur moyenne.**

3 Import des données hors résistographe

Je charge le fichier qui contient les observations sur le terrain lors de la sélection finale et la sélection du premier jour :

```

obs_sel <- read.table("homol_selection_observations_terrain_2017-03.csv",
  header = T,
  sep = ";",
  stringsAsFactors = F
)

# Je stocke la liste des 24 arbres récoltés :
lst_24_reco <- obs_sel$codes_recoltes[which(!is.na(obs_sel$codes_recoltes))]

dim(obs_sel)

```

```
## [1] 37 6
```

Nombre d'arbres sur lesquels on a récolté des greffons :

```
sum(obs_sel$arbres_recoltes)
```

```
## [1] 24
```

Je charge le jeu de données avec les colonnes contenant les infos sur la sélection *in silico* :

```
load(file = "homolCali_selections.RData")
dim(homolCali)
```

```
## [1] 1360 77
```

Je joint la table de données complètes avec les données d'observations et de sélection du terrain :

```
# Les codes d'individus de la table complète sont du type : CC_CAL_B33
# alors que les codes d'individus de la table de sélections sur le terrain sont
# du type : CAL_B33
# donc sans les 2 lettre d'identification de région au début.
```

```
# Je vais créer une colonne avec les mêmes identifiants dans la table de données
# complètes :
```

```
homolCali$Ind <- paste(homolCali$Fact, "_", homolCali$Bloc, sep = "")
```

```
# Maintenant, je peux joindre les 2 tables sur cette colonne :
```

```
homolCali <- merge(homolCali,
                   obs_sel,
                   by = "Ind",
                   all.x = T
                   )
```

```
dim(homolCali)
```

```
## [1] 1360 83
```

Je charge le jeu de données avec les WA 403 :

```
load(file = "homol403.RData")
dim(homol403)
```

```
## [1] 68 67
```

Je vais une unique table de données avec les 403 et les provenances Californiennes. Pour cela, je dois d'abord faire que les mêmes colonnes existent dans les 2 tables. Comme je ne vais utiliser que les mesures réalisées lors de la dernière campagne, je fais une sélection des colonnes que je vais conserver. Quand les variables ont été ajustées pour l'effet bloc, je ne prends que les valeurs ajustées.

```
# La liste des colonnes que je souhaite avoir :
```

```
lst_col <- c("Ind", "Fact", "Bloc", "Etat", "Provenance", "idInd",
            "No14", "Fh14", "Ba14", "Af14", "Gb14", "Rq14",
            "Ci14_pred", "Nb14_pred", "Deb96_pred", "Ht01_pred",
            "indexsel", "remarques", "arbres_recoltes", "codes_recoltes")
```

```
# Certaines colonnes n'existent pas dans la table `homol403`, je les
# rajoute :
```

```
homol403$Ind <- paste(homol403$Fact, homol403$Bloc, sep = "_")
homol403$idInd <- paste(homol403$etat_prov, homol403$Bloc, sep = "_")
homol403$indexsel <- NA
```

```

homol403$remarques <- NA
homol403$arbres_recoltes <- NA
homol403$codes_recoltes <- NA

# Je classe les colonnes dans le même ordre :
homolCali <- homolCali[, lst_col]
homol403 <- homol403[, lst_col]

# Je colle les 2 tables :
homol403Cali <- rbind(homol403, homolCali)
dim(homol403Cali)

## [1] 1428 20

```

3.1 Données de résistographe

```

lst_fichiers_resisto <- dir(path = "donnees_resisto_pour_analyses")
lst_fichiers_resisto <- paste("donnees_resisto_pour_analyses",
                             lst_fichiers_resisto,
                             sep = "/"
                             )

```

Boucle sur les fichiers pour créer les objets :

```

profils <- vector("list", length(lst_fichiers_resisto))
for(i in 1:length(lst_fichiers_resisto)){
  # stockage du nom de fichier
  f <- lst_fichiers_resisto[i]
  # création de l'objet liste
  profil <- list(fichier = NA,
               ind = NA,
               date = NA,
               heure = NA,
               feed_speed = NA,
               drill_speed = NA,
               valeurs = NA)
  # On stocke les infos
  # nom de fichier
  profil$fichier <- f
  # individus
  profil$ind <- scan(f, skip = 5, nlines = 1, what = "character")
  profil$ind <- gsub("-", "_", profil$ind)
  # date
  profil$date <- scan(f, skip = 6, nlines = 1, what = "character")
  # heure
  profil$heure <- scan(f, skip = 7, nlines = 1, what = "character")
  # feed_speed
  profil$feed_speed <- scan(f, skip = 14, nlines = 1, what = "numeric")
  # drill_speed
  profil$drill_speed <- scan(f, skip = 15, nlines = 1, what = "numeric")
  # table de données
  profil$valeurs <- read.table(f, sep = ";", skip = 129)
  names(profil$valeurs) <- c("drill", "feed")
}

```

```

#
# assign(x = profil$ind, profil)
# nom_profil <- paste("prof", profil$ind, sep = "_")
# assign(x = nom_profil, profil)
# assign(x = substring(f, 15, 22), profil)

profils[[i]] <- profil
}
# Nombre de profils lus et enregistrés :
length(profils)

```

```
## [1] 120
```

```
rm(f)
```

Les noms des individus :

```

noms_ind_prof <- lapply(profils,
                        FUN = function(p){
                          return(p$ind)
                        }
                      )
noms_ind_prof <- unlist(noms_ind_prof)
names(profils) <- noms_ind_prof
names(profils)

```

```

## [1] "HOMOL_CAL_B60" "HOMOL_CAL_B61" "HOMOL_RE2_B47" "HOMOL_RE2_B43"
## [5] "HOMOL_IOH_B44" "HOMOL_RE1_B46" "HOMOL_BUR_B49" "HOMOL_HAY_B40"
## [9] "HOMOL_IOH_B32" "HOMOL_CAL_B33" "HOMOL_VDR_B41" "HOMOL_HOO_B_24"
## [13] "HOMOL_VDR_B29" "HOMOL_SCR_B20" "HOMOL_PIL_B22" "HOMOL_RE1_B14"
## [17] "HOMOL_RE1_B08" "HOMOL_SCR_B17" "HOMOL_WEA_B10" "HOMOL_RE1_B17"
## [21] "HOMOL_RE1_B19" "HOMOL_RE1_B25" "HOMOL_CO2_B25" "HOMOL_VDR_35"
## [25] "HOMOL_RE1_B38" "HOMOL_SCR_B52" "HOMOL_RE2_B52" "HOMOL_CO1_B51"
## [29] "HOMOL_SHA_B53" "HOMOL_LAS_B53" "HOMOL_HAY_B68" "HOMOL_CH1_B26"
## [33] "HOMOL_VDR_B68" "HOMOL_CH2_B68" "HOMOL_HOO_B55" "HOMOL_CO2_B55"
## [37] "HOMOL_PIL_B56" "HOMOL_403_B68" "HOMOL_403_B67" "HOMOL_403_B38"
## [41] "HOMOL_403_B27" "HOMOL_403_B26" "HOMOL_403_B10" "HOMOL_403_B12"
## [45] "HOMOL_403_B01" "HOMOL_403_B02" "HOMOL_403_B07" "HOMOL_403_B24"
## [49] "HOMOL_403_B21" "HOMOL_403_B30" "HOMOL_403_B29" "HOMOL_403_B43"
## [53] "HOMOL_403_B46" "HOMOL_403_B47" "HOMOL_403_B58" "HOMOL_403_B63"
## [57] "HOMOL_HAY_B45" "HOMOL_STH_B44" "HOMOL_VDR_45" "HOMOL_SV1_B60"
## [61] "HOMOL_HAY_B60" "HOMOL_VDR_60" "HOMOL_RE1_B61" "HOMOL_ORL_B62"
## [65] "HOMOL_HAY_B61" "HOMOL_STH_B62" "HOMOL_SCR_B63" "HOMOL_HAY_B62"
## [69] "HOMOL_SV1_B59" "HOMOL_RE1_B59" "HOMOL_SCR_B59" "HOMOL_RE1_B47"
## [73] "HOMOL_VDR_B43" "HOMOL_SV1_B43" "HOMOL_STH_B43" "HOMOL_RE1_B43"
## [77] "HOMOL_ORL_B43" "HOMOL_BUR_B43" "HOMOL_SV1_B42" "HOMOL_VDR_B42"
## [81] "HOMOL_RE1_B42" "HOMOL_ORL_B48" "HOMOL_VDR_B47" "HOMOL_SCR_B57"
## [85] "HOMOL_ORL_B64" "HOMOL_BUR_B63" "HOMOL_HAY_B63" "HOMOL_HAY_B64"
## [89] "HOMOL_RE1_B64" "HOMOL_BUR_B65" "HOMOL_SV1_B48" "HOMOL_RE1_B48"
## [93] "HOMOL_VDR_B48" "HOMOL_RE1_B41" "HOMOL_BUR_B32" "HOMOL_ORL_B31"
## [97] "HOMOL_SCR_B40" "HOMOL_SV1_B40" "HOMOL_BUR_B40" "HOMOL_SV1_B30"
## [101] "HOMOL_HAY_B30" "HOMOL_ORL_B30" "HOMOL_STH_B23" "HOMOL_VDR_B21"
## [105] "HOMOL_ORL_B21" "HOMOL_SV1_B15" "HOMOL_SCR_B07" "HOMOL_STH_B07"
## [109] "HOMOL_HAY_B07" "HOMOL_STH_B02" "HOMOL_BUR_B01" "HOMOL_BUR_B33"
## [113] "HOMOL_SCR_B39" "HOMOL_STH_B56" "HOMOL_BUR_B56" "HOMOL_STH_B66"

```

```
## [117] "HOMOL_SCR_B65" "HOMOL_ORL_B34" "HOMOL_SCR_B24" "HOMOL_VDR_B20"
```

→ Il y a quelques problèmes avec les noms entrés dans la machine.

```
names(profils)[which(nchar(names(profils)) != 13)]
```

```
## [1] "HOMOL_HOO_B_24" "HOMOL_VDR_35" "HOMOL_VDR_45" "HOMOL_VDR_60"
```

Il y a quelques problèmes de noms rentrés dans la machine :

```
tmp <- which(names(profils) == "HOMOL_HOO_B_24")
profils[[tmp]]$ind <- "HOMOL_HOO_B24"
names(profils)[tmp] <- "HOMOL_HOO_B24"
```

```
#
```

```
tmp <- which(names(profils) == "HOMOL_VDR_45")
profils[[tmp]]$ind <- "HOMOL_VDR_B45"
names(profils)[tmp] <- "HOMOL_VDR_B45"
```

```
#
```

```
tmp <- which(names(profils) == "HOMOL_VDR_35")
profils[[tmp]]$ind <- "HOMOL_VDR_B35"
names(profils)[tmp] <- "HOMOL_VDR_B35"
```

```
tmp <- which(names(profils) == "HOMOL_VDR_60")
profils[[tmp]]$ind <- "HOMOL_VDR_B60"
names(profils)[tmp] <- "HOMOL_VDR_B60"
```

```
# lapply(profils,
#         function(x) {
#           plot(x$valeurs$drill,
#                type = "l",
#                xlab = "",
#                ylab = "Résistance")
#           points(x$valeurs$feed,
#                 col = "red",
#                 type = "l")
#           legend("top",
#                 col = c("black", "red"),
#                 legend = c("Rotation", "Avancement"),
#                 pch = "-")
#         })
```

4 Préparation des données du résistographe

4.1 Suppression de l'écorce

Je fais une boucle pour supprimer la partie correspondant à l'écorce.

```
profils_sauvEcorce <- profils
profils <- lapply(profils,
  FUN = function(p) {
    # longueur du profil
    long_p <- dim(p$valeurs)[1]
    # J'enlève les 200 premières valeurs et
    # les 300 dernières
    p$valeurs <- p$valeurs[200:(long_p - 300),]
```

```

    return(p)
  }
)

```

4.2 Calculs de moyennes des profils

Je vais estimer une moyenne pour les 2 valeurs enregistrées lors d'un sondage. Mon tableau de données va donc devenir un tableau rectangulaire classique avec une ligne par observation/sondage.

*# Je fais un `sapply` unique dans lequel j'extraie les 5 infos dont j'ai besoin
et qui vont constituer une ligne de ma data.frame :*

```

moy_profils <- sapply(profils,
  FUN = function(p){
    tmp_vec <- c(p$ind,
                 p$feed_speed,
                 p$drill_speed,
                 mean(p$valeurs$feed),
                 mean(p$valeurs$drill)
                )
    names(tmp_vec) <- c("ind",
                       "feed_speed",
                       "drill_speed",
                       "moy_feed",
                       "moy_drill"
                      )
    return(tmp_vec)
  }
)
# L'objet produit est une matrice mais les lignes sont en colonnes et vice-versa
moy_profils <- t(moy_profils)
# Je convertis l'objet en data.frame
moy_profils <- as.data.frame(moy_profils,
                             stringsAsFactors = F)
# Je convertis les colonnes de données quanti en numeric :
moy_profils[,2:5] <- apply(moy_profils[2:5], 2, as.numeric)

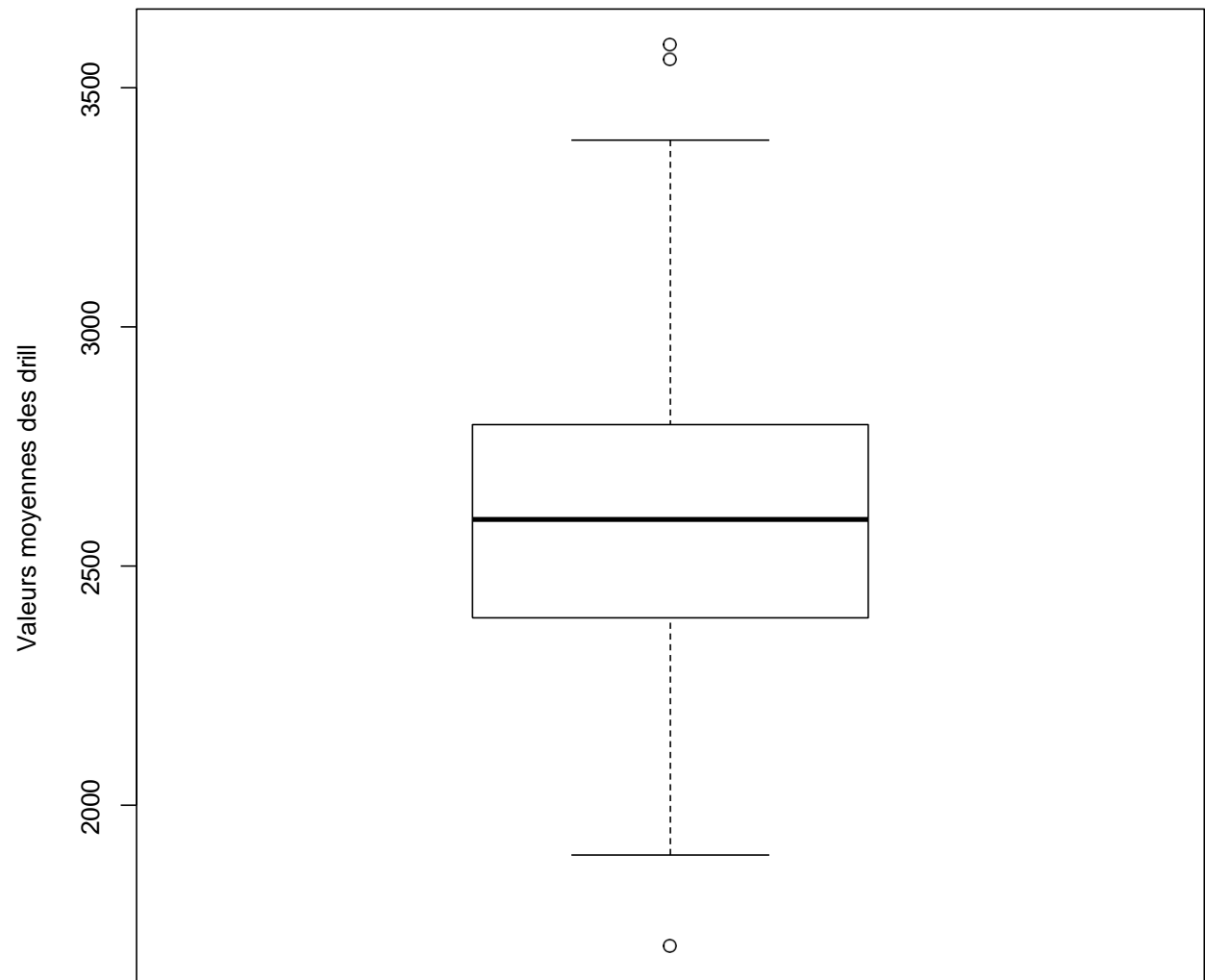
```

4.3 Premiers graphiques descriptifs des données de résisto

```

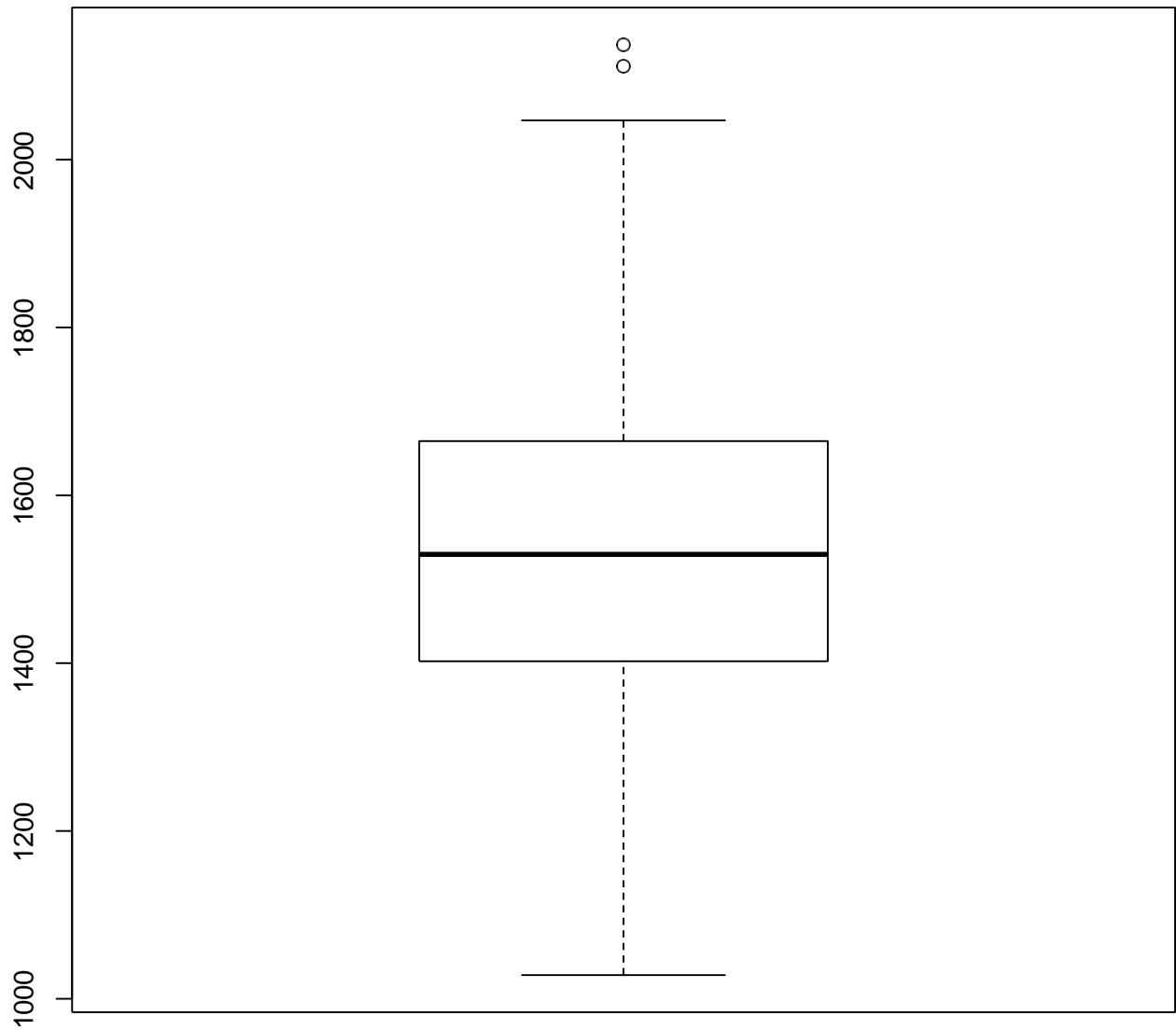
boxplot(moy_profils$moy_drill,
        ylab = "Valeurs moyennes des drill")

```

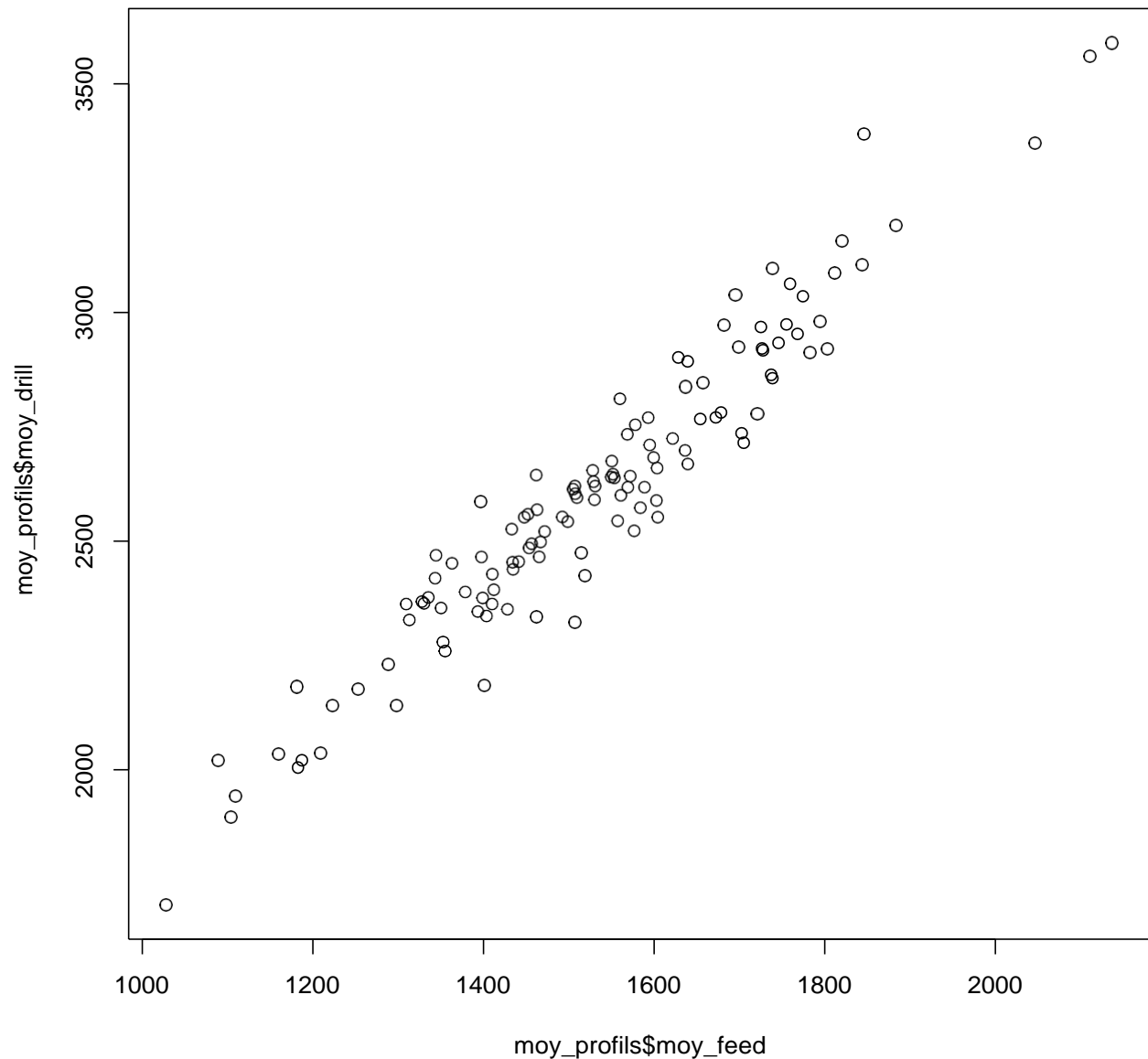


```
boxplot(moy_profils$moy_feed,  
        main = "Valeurs moyennes des feed")
```


Valeurs moyennes des feed



```
plot(moy_profils$moy_feed, moy_profils$moy_drill)
```



Résultats :

- Pas grand chose à dire sur les distributions.
- Les 2 variables sont assez corrélées.

4.4 Gestion des paramètres de vitesses

Je vérifie les vitesses de drill et feed :

```
table(moy_profils$drill_speed, useNA = "ifany")
```

```
##
## 1500 2500
## 116   4
```

```
table(moy_profils$feed_speed, useNA = "ifany")
```

```
##
```

```
## 500 1000
## 3 117
```

Résultats :

→ Pour chaque paramètres, 2 valeurs ont été utilisées selon les sondages.

→ La majorité des sondages a été réalisée avec une même valeur.

Les 4 sondages réalisés avec la vitesse drill_speed de 2500 :

```
moy_profils[which(moy_profils$drill_speed == "2500"),]
```

```
##                ind feed_speed drill_speed moy_feed moy_drill
## HOMOL_CAL_B60 HOMOL_CAL_B60      1000      2500 1462.207  2334.761
## HOMOL_CAL_B61 HOMOL_CAL_B61      1000      2500 1159.930  2034.099
## HOMOL_RE2_B47 HOMOL_RE2_B47      1000      2500 1089.488  2020.843
## HOMOL_RE2_B43 HOMOL_RE2_B43      1000      2500 1410.637  2363.034
```

```
tmp <- moy_profils[which(moy_profils$drill_speed == "2500"),"ind"]
```

Résultats :

→ Ces sondages avec une drill_speed différente ont par contre la valeur de feed_speed utilisée pour la plupart des sondages.

Ont-ils été récoltés pour les greffons ?

```
tmp[which(tmp %in% lst_24_reco)]
```

```
## [1] "HOMOL_CAL_B60" "HOMOL_RE2_B43"
```

```
rm(tmp)
```

Résultats :

→ Il y a 2 arbres récoltés dans les 4 avec la drill_speed différente.

Les 3 sondages réalisés avec la vitesse feed_speed de 500 :

```
moy_profils[which(moy_profils$feed_speed == 500),]
```

```
##                ind feed_speed drill_speed moy_feed moy_drill
## HOMOL_BUR_B49 HOMOL_BUR_B49      500      1500 1028.388  1704.881
## HOMOL_HOO_B24 HOMOL_HOO_B24      500      1500 1223.251  2140.715
## HOMOL_LAS_B53 HOMOL_LAS_B53      500      1500 1288.834  2230.400
```

```
tmp <- moy_profils[which(moy_profils$feed_speed == 500),"ind"]
```

Résultats :

→ Ces sondages avec une feed_speed différente ont par contre la valeur de drill_speed utilisée pour la plupart des sondages.

Ont-ils été récoltés pour les greffons ?

```
tmp[which(tmp %in% lst_24_reco)]
```

```
## [1] "HOMOL_BUR_B49" "HOMOL_HOO_B24"
```

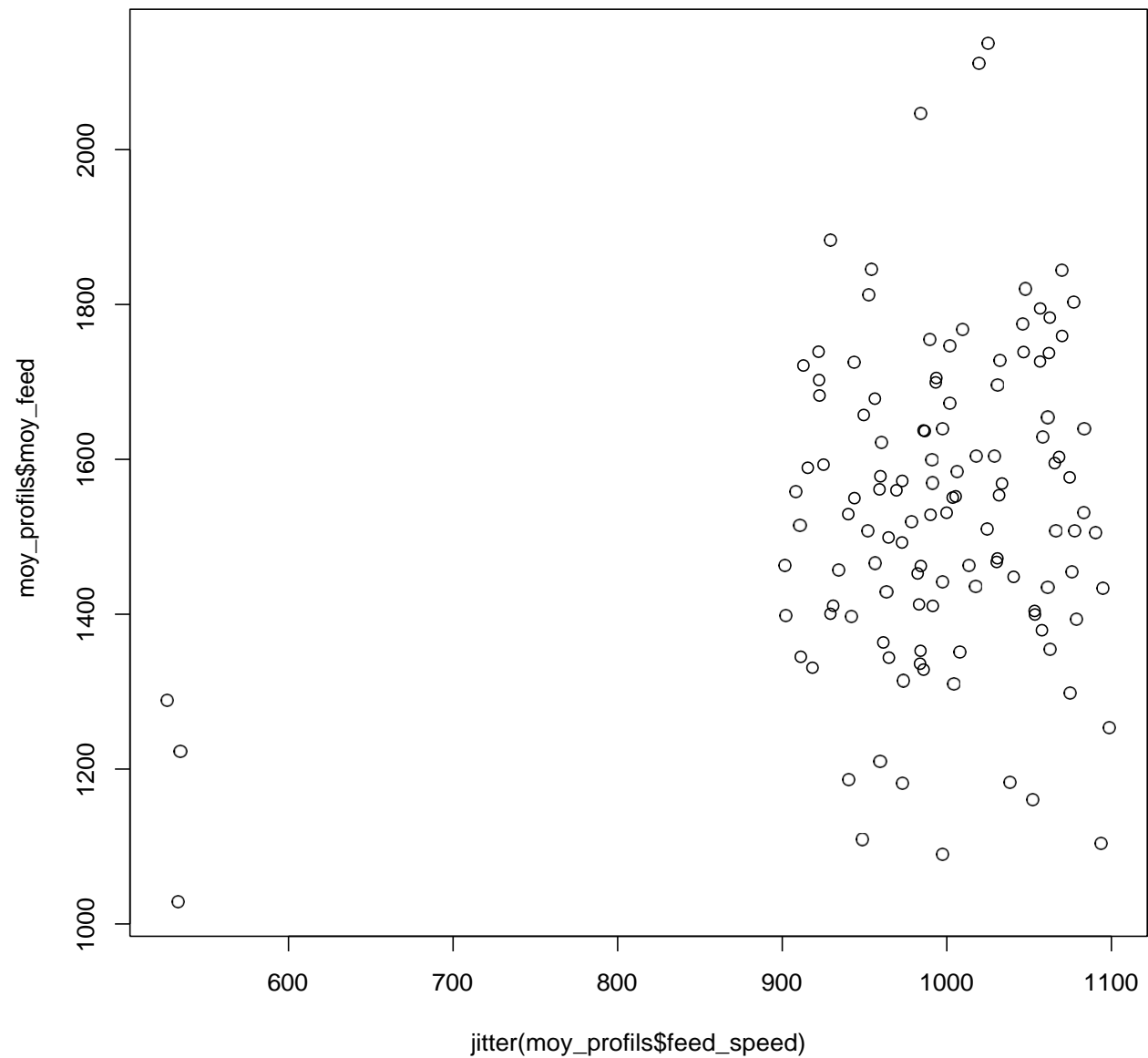
```
rm(tmp)
```

Résultats :

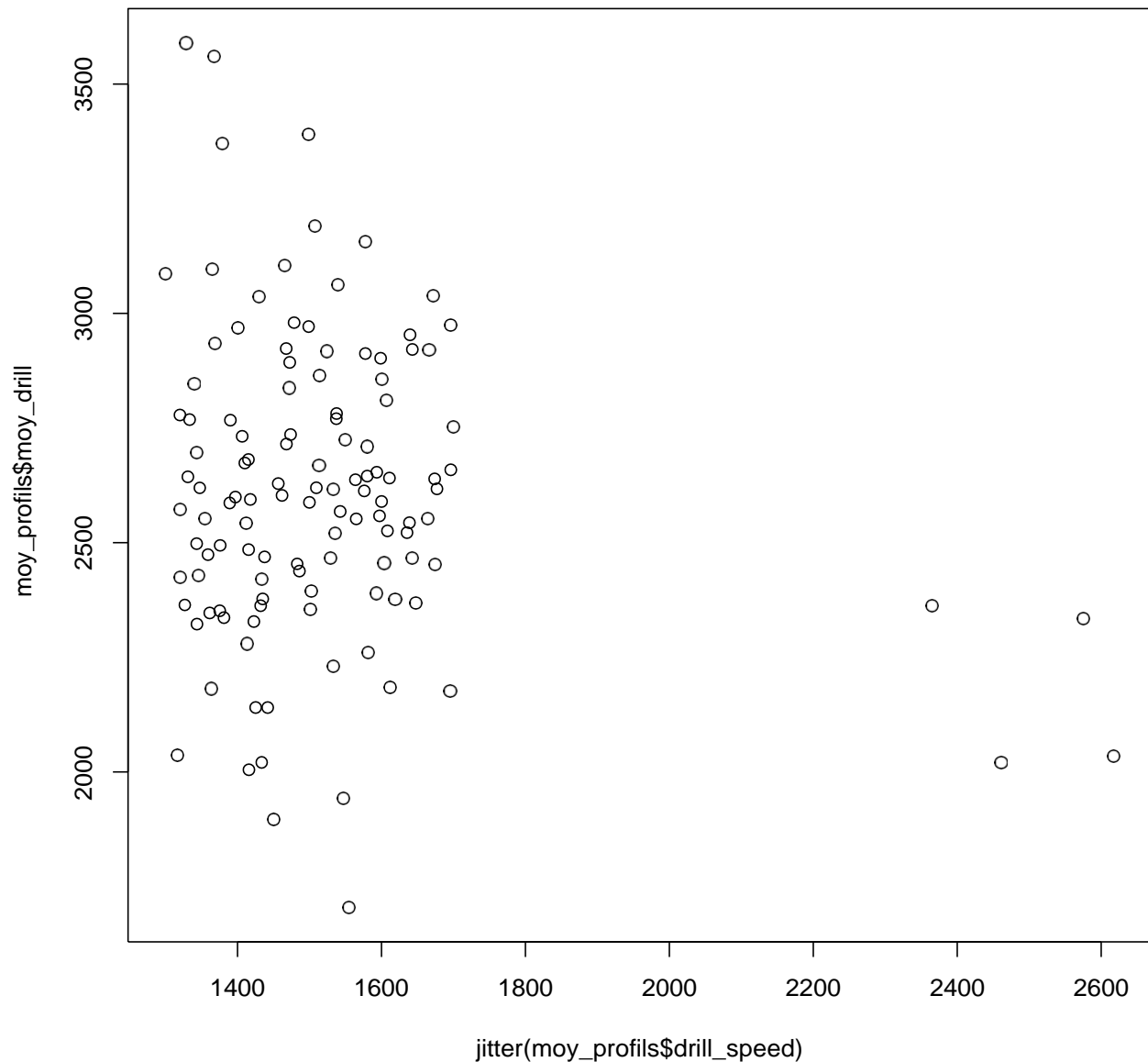
→ Il y a 2 arbres récoltés avec cette vitesse de 500.

Comparaison des valeurs obtenues avec les 2 valeurs de vitesses :

```
plot(jitter(moy_profils$feed_speed), moy_profils$moy_feed)
```



```
plot(jitter(moy_profils$drill_speed), moy_profils$moy_drill)
```

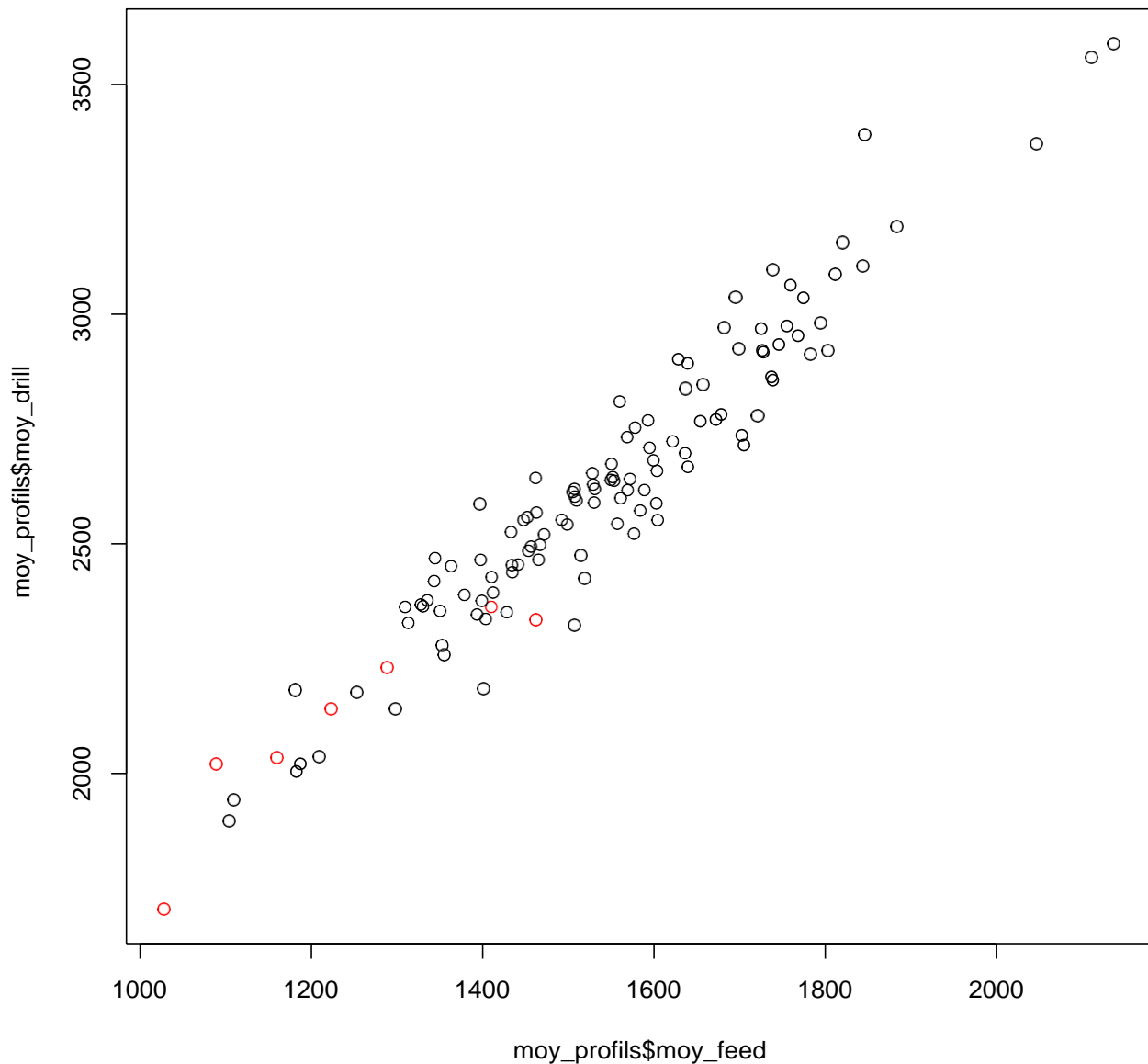


26/04/2017 - Après discussion avec Leo, je décide de ne pas prendre en compte les données obtenues avec des vitesses différentes. En effet, les paramètres de vitesses ont une influence sur les valeurs enregistrées mais il n'est pas possible d'appliquer facilement une correction. Il ne vaut mieux pas changer les paramètres de vitesses si l'on souhaite comparer les profils entre eux.

Estimation d'une des valeurs par corrélation entre les 2 mesures - Comme les deux mesures sont corrélées, j'avais pensé estimé la valeur "manquante" d'une des variables en utilisant la valeur mesurée pour l'autre variable.

Le graphique des deux variables avec les points pour lesquels une des deux n'est pas mesurée avec la vitesse majoritaire :

```
# indice des sondages avec une vitesse différente pour l'une des 2 vitesses
vit_dif <- moy_profils$feed_speed == 500 | moy_profils$drill_speed == 2500
plot(moy_profils$moy_feed, moy_profils$moy_drill,
     col = vit_dif + 1
)
```



Résultats :

- Les points sont parfaitement dans la relation entre les deux paramètres.
- Les graphiques des valeurs drill/feed en fonction des vitesses ne montrent pas de différences flagrantes entre les 2 groupes.

Les points sont dans la relation, ce n'est pas attendu. . . En effet, si on part du constat que les 2 variables d'un sondage sont corrélées, et si on fait l'hypothèse que la valeurs de vitesse modifie la valeur mesurée, pour un individu, si une des variable est mesurée avec la valeur "classique" de vitesse et si l'autre variable est mesurée avec une valeur "rare", cette variable devrait avoir une valeur différente de l'attendu et donc le couple des deux variables devrait sortir de la relation.

03/05/2017 - Ceci n'est pas élucidé.

J'enlève les valeurs obtenues avec les paramètres de vitesses minoritaires.

```
moy_profils$moy_feed[which(moy_profils$feed_speed == 500)] <- NA
moy_profils$moy_drill[which(moy_profils$drill_speed == 2500)] <- NA
```

5 Analyse des valeurs moyennes des sondages au résistographe

5.1 Description des données moyennes

On ajoute une colonne avec la provenance des individus et une autre avec les blocs à partir du code d'individu :

```
# La provenance :  
moy_profils$prov <- substr(moy_profils$ind, 7, 9)  
# Le bloc :  
moy_profils$bloc <- substr(moy_profils$ind, 11, 13)
```

Les provenances représentées :

```
sort(table(moy_profils$prov, useNA = "ifany"),  
      decreasing = T)
```

```
##  
## 403 RE1 VDR SCR HAY BUR ORL STH SV1 CAL RE2 CO2 HOO IOH PIL CH1 CH2 CO1  
## 19 15 12 11 10 9 8 8 8 3 3 2 2 2 2 1 1 1  
## LAS SHA WEA  
## 1 1 1
```

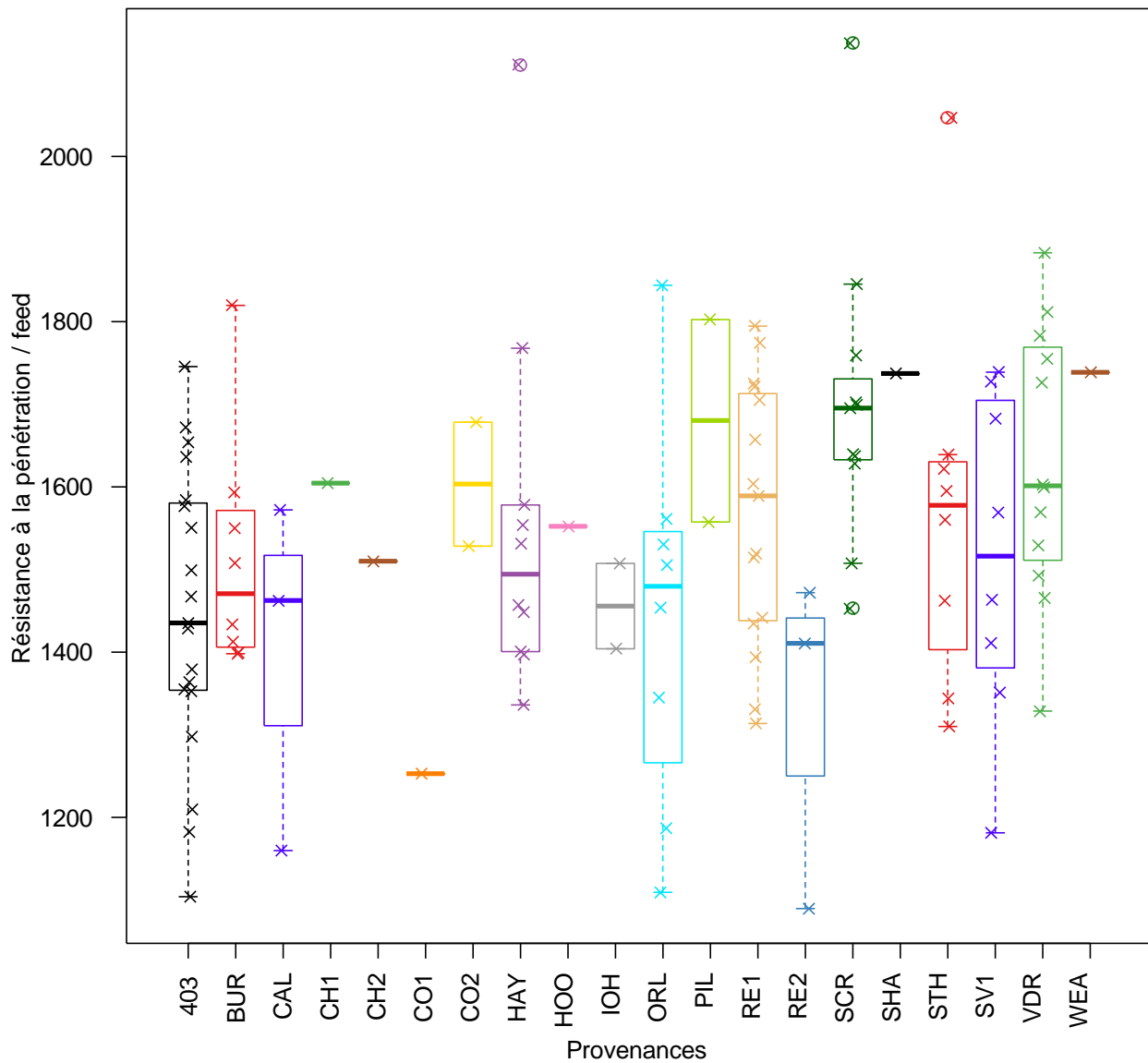
Résultats :

→ A part le témoin 403, 8 provenances sont bien représentées avec au moins 8 sondages mais les 12 autres ne sont représentées que par 1 à 3 sondages.

Les distributions des valeurs moyennes par sondage pour chaque provenance :

```
boxplot(moy_profils$moy_feed ~ moy_profils$prov,  
        las = 2,  
        ylab = "Résistance à la pénétration / feed",  
        xlab = "Provenances",  
        main = "Moyennes des résistances à la pénétration",  
        border = couleurs  
        )  
stripchart(moy_profils$moy_feed ~ moy_profils$prov,  
           vert = T,  
           method = "jitter",  
           pch = 4,  
           las = 2,  
           col = couleurs,  
           add = T  
           )
```

Moyennes des résistances à la pénétration



Résultats :

→ Il y a des différences de distributions entre certaines provenances qui ont plutôt des valeurs hautes comme SCR, VDR ou RE1 par exemple et d'autres qui ont des valeurs plus basses comme 403, BUR, HAY...

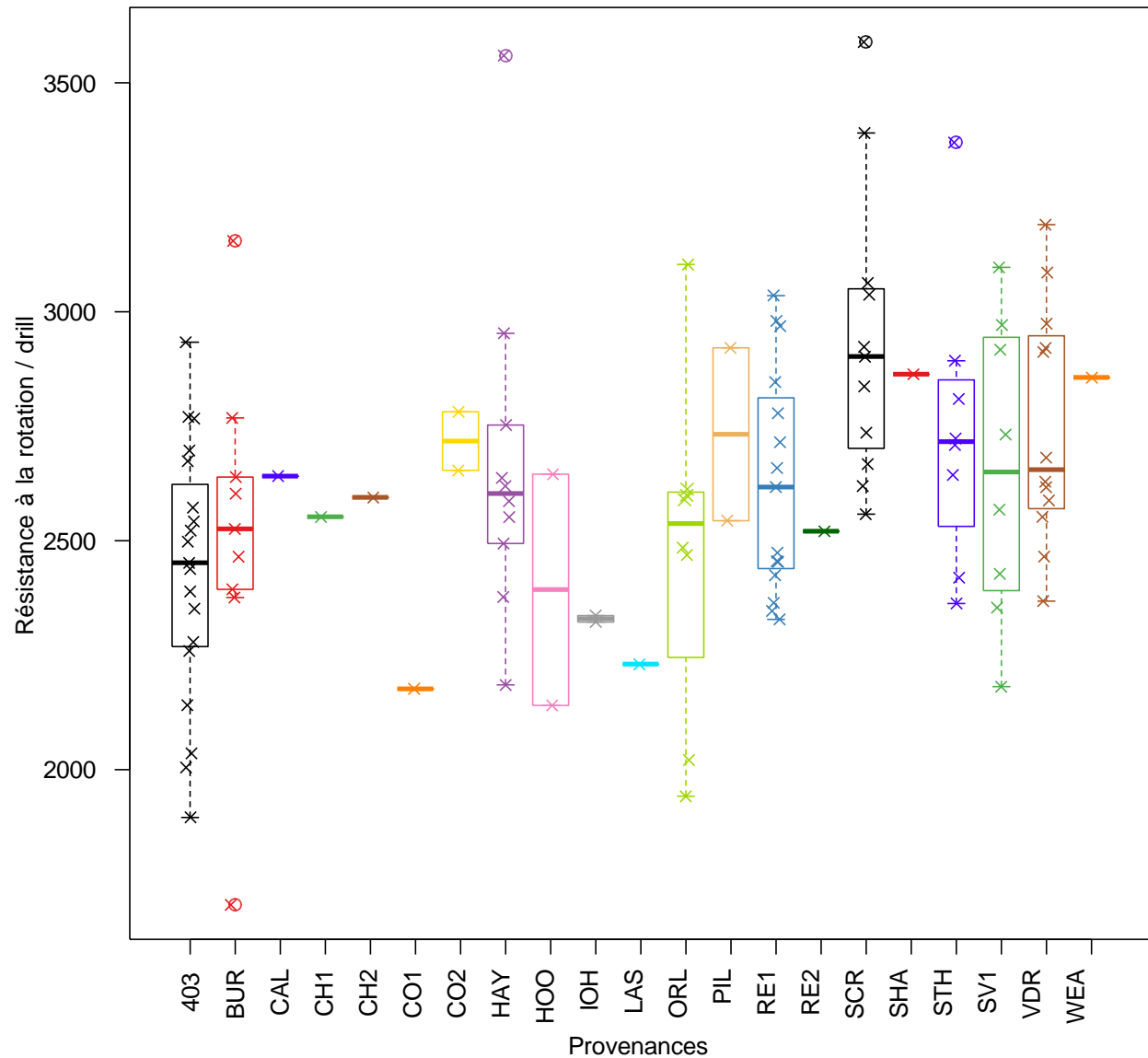
→ Il y a aussi des différences entre les individus d'une même provenance, qui du même ordre que celles entre les provenances.

```
boxplot(moy_profils$moy_drill ~ moy_profils$prov,
  las = 2,
  ylab = "Résistance à la rotation / drill",
  xlab = "Provenances",
  main = "Moyennes des résistances à la rotation",
  border = couleurs
)
```



```
stripchart(moy_profils$moy_drill ~ moy_profils$prov,
  vert = T,
  method = "jitter",
  pch = 4,
  las = 2,
  col = couleurs,
  add = T
)
```

Moyennes des résistances à la rotation



Résultats :

→ Mêmes constats que pour la résistance à la pénétration.

5.2 Comparaisons entre sélectionnés et non sélectionnés

Je joins la table des données dendro et des données résisto :

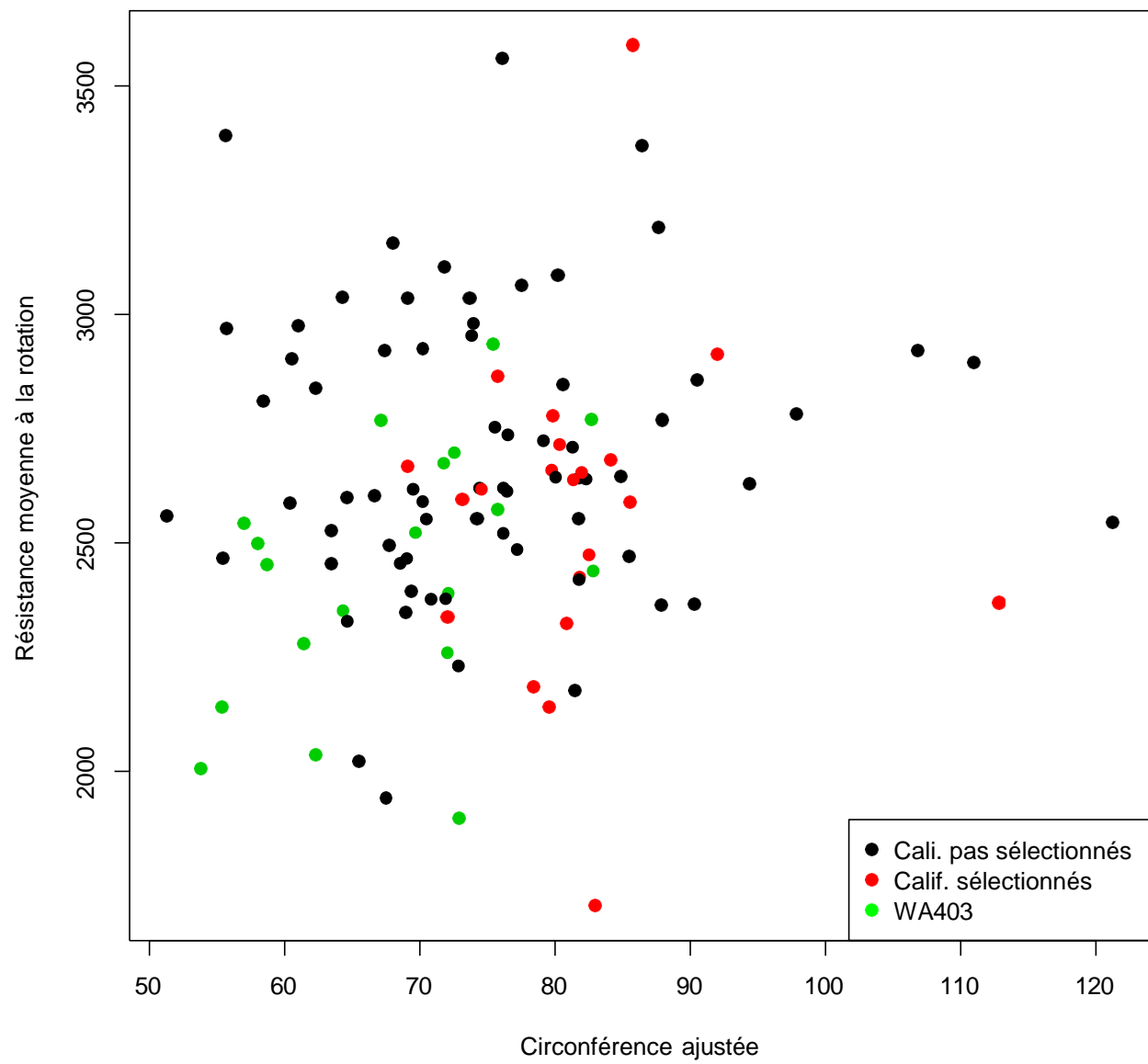
```
# Les codes d'individus dans la table de mesures dendro sont du type  
# `BUR_B12` or, ceux de la table des mesures résisto ont un préfixe `HOMOL_`.  
# Je modifie ça :  
moy_profils$prov_bloc <- paste(moy_profils$prov, moy_profils$bloc, sep = "_")  
  
# Je joint les deux tables :  
toto <- merge(moy_profils, homol403Cali,  
              by.x = "prov_bloc",  
              by.y = "Ind",  
              all.x = TRUE)  
dim(toto)  
  
## [1] 121 27
```

6 MP YR 09/0/2017

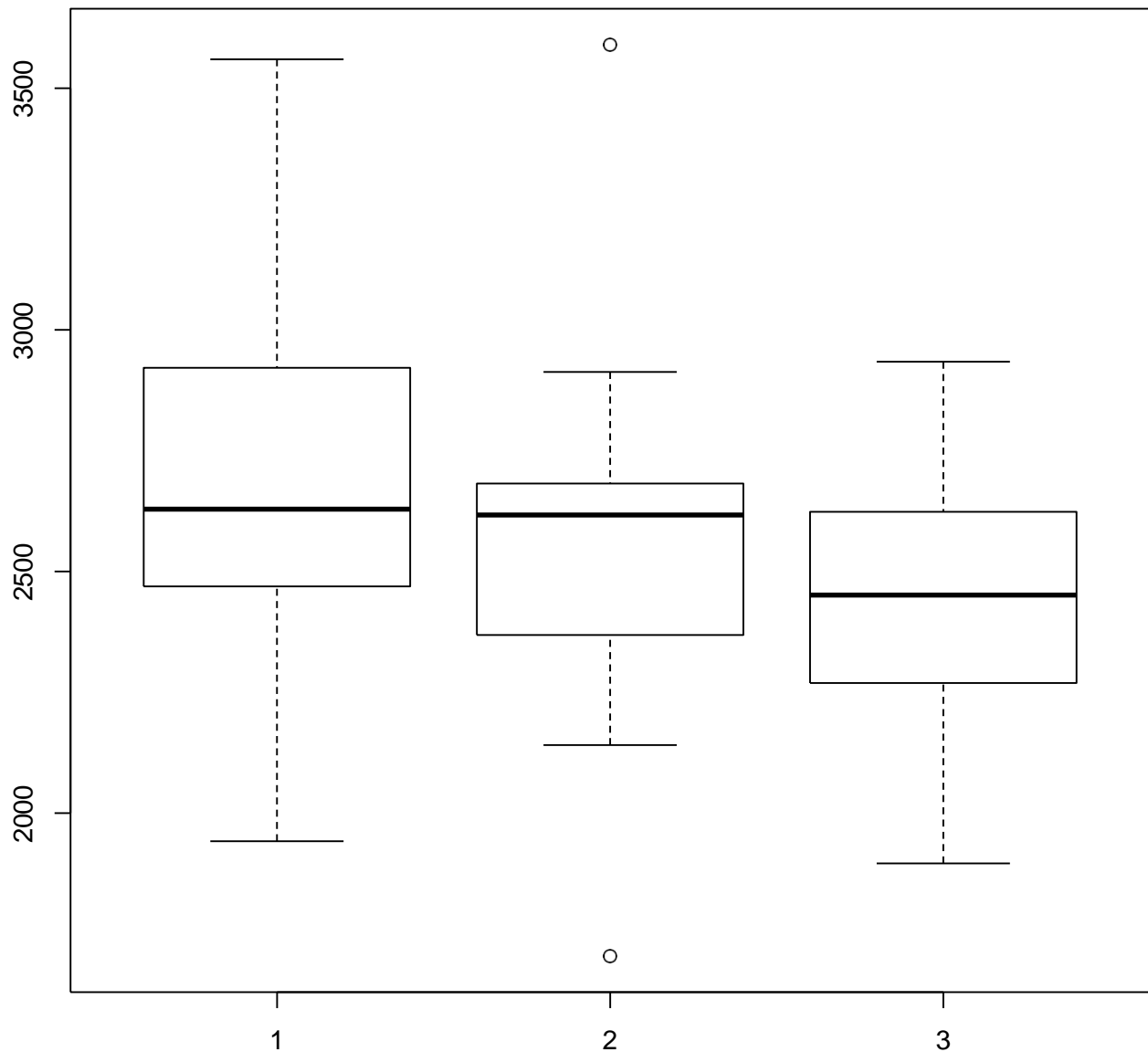
Provenance SV1 de OR dans le lot donc il faut que je récupère les données correspondantes. ça se passe dans le script 2.

7 Pour la plénère

```
# je fais une colonne avec 2 si récotlé, 1 si Cali et 3 si SZ403  
toto$statut <- 1  
toto$statut[which(toto$arbres_recoltes == 1)] <- 2  
toto$statut[which(substr(toto$prov_bloc, 1, 3) == "403")] <- 3  
  
plot(toto$Ci14_pred, toto$moy_drill,  
      xlab = "Circonférence ajustée",  
      ylab = "Résistance moyenne à la rotation",  
      col = toto$statut,  
      pch = 19)  
legend("bottomright",  
       col = c("black", "red", "green"),  
       legend = c("Cali. pas sélectionnés", "Calif. sélectionnés", "WA403"),  
       pch = 19)
```



```
boxplot(toto$moy_drill ~ toto$statut)
```



```
# toto <- merge(moy_resisto, obs_sel, by.x = "ind", by.y = "Ind", all = T)
# # on d gage les cas pour lesquels on n'a pas de resisto :
#
# toto[which(is.na(toto$drill)),]
# toto <- toto[-which(is.na(toto$drill)),]
# # Les arbres qui ont un NA pour la colonne de s lection sont des 403, on met l'info
# toto$seljour1[which(is.na(toto$seljour1))] <- "403"
#
# toto <- toto[-which(toto$ind == "HOMOL_PIL_B56"),]
# table(toto$seljour1)
```

```
# boxplot(toto$drill ~ toto$seljour1)
```

Relation avec la circonf

```
# homolCali$ind_resisto <- paste("HOMOL", homolCali$Fact, homolCali$Bloc, sep = "_")
# toto <- merge(toto, homolCali[,c("ind_resisto", "Ci14_pred", "Ht01_pred")],
```

```

# by.x = "ind",
# by.y = "ind_resisto",
# all.x = T,
# all.y = F)
# plot(toto$Ci14_pred, toto$drill,
# col = as.factor(toto$seljour1))
# plot(toto$Ht01_pred, toto$drill,
# col = as.factor(toto$seljour1))

# plot(toto$Ci14_pred ~ as.factor(toto$seljour1))
# plot(toto$Ht01_pred ~ as.factor(toto$seljour1))

# tapply(toto$Ci14_pred, toto$seljour1, mean)
# tapply(toto$Ht01_pred, toto$seljour1, mean)
# tapply(toto$feed, toto$seljour1, mean)

# ratrap <- toto[order(toto$Ci14_pred, decreasing = T),]
# ratrap
# ratrap[1:8,]

```

8 Export des données

We select the interesting columns:

```

# homol <- homol[, c("Fact", "Bloc", "Htinc10y", "Ci14", "No14", "Vi14", "Fh14pr", "Ba14", "Af14", "Nb14",
# save(homol, file = "homol.RData")

```