

# t.p.é\*

Une collection d'outils pour  
(s')initier à la publication  
en éducation et valoriser la  
transformation pédagogique

\* transformer et publier en sciences de l'éducation



Ces outils sont organisés en trois volets :

- accompagner / structurer pour les accompagnant-es
- rédiger pour les équipes enseignantes débutant en recherche-action
- s'enrichir / ressources une série de références et de guides

Ils ont été développés par la Chaire recherche-action sur l'innovation pédagogique de l'Université Paris Saclay et l'institut Villebon - *Georges Charpak*, en collaboration avec l'UQAM, et sont le fruit du travail de Marine Moyon, Frédéric Bouquet, Jeanne Parmentier, et Martin Riopel, et d'Emmanuel Ahr pour l'outil EVA.

L'exploration, la conception de la charte graphique et la mise en forme des outils ont été réalisées par Dalva Rospape et Marie Jouble.

Retrouvez tous les outils sur :

<https://cep.villebon-charpak.fr/tpe>

## → s'enrichir / ressources

### stat 101

Ce document est un guide pour mener des analyses statistiques sur R (niveau avancé).

## création d'un répertoire de travail sur l'ordinateur

Le plus simple est de créer, sur le bureau de votre ordinateur, un dossier que l'on pourra nommer ChaireIP (pour Chaire Innovation Pédagogique). Ce dossier constituera votre répertoire de travail. Vous pourrez y déposer et y enregistrer tous vos fichiers de travail en lien avec votre projet d'innovation pédagogique.

## définition du répertoire de travail sur R

Une fois la console R ouverte, il vous suffit d'aller dans "Files", sélectionner le dossier "ChaireIP" enregistré sur le bureau d'ordinateur, cliquer sur "More" puis "Set as Working Directory". Sur la console s'affichera alors automatiquement : `setwd(~ChaireIP)`

## chargement du fichier de données enregistré au format xlsx et création de dataframe sous R

Création d'un dataframe nommé DF. Nous souhaitons que DF contiennent les données enregistrées dans le 1er onglet d'un fichier excel nommé "mon\_dataset\_ChaireIP.xlsx" et enregistré à ce chemin `C:/Users/marine.moyon/Desktop/ChaireIP/mon_dataset_ChaireIP.xlsx`

```
library(readxl)
DF <- read_excel(«C:/Users/marine.moyon/Desktop/ChaireIP/mon_dataset_ChaireIP .xlsx»,1)
```

Le visualisateur interne permet de visualiser les données. Pour cela, utiliser la fonction `View` et indiquer le nom du dataframe

```
View(DF)
```

## exploration des variables contenues dans le dataframe

La commande `names` permet d'avoir accès au nom des variables contenues dans le dataframe `DF`

```
names(DF)
## [1] «Pp»
## [3] «Sexe»
## [5] «Groupe_TD»
## [7] «Note_test_decembre»
## [9] «Score_motivation»
## [11] «Score_sentiment_auto_efficacite»
«Age»
«Specialite_choisie»
«Note_test_novembre»
«Temps_revision»
«Score_estime_de_soi»
«Statut»
## [13] «Admission» «condition»
## [15] «score_attention_T0» «score_attention_T1»
```

Il est possible de n'afficher que les `n` premières variables du dataframe `DF`. Par exemple nous pouvons choisir de n'afficher que les 4 premières variables

```
names(DF)[1:4]
## [1] «Pp»           «Age»           «Sexe»
## [4] «Specialite_choisie»
```

La commande `str` permet de décrire le contenu du dataframe `DF` (nombre de participants ; nombre de variables ; intitulé des variables ; mode de représentation de la variable. `int` = variable quanti et premières valeurs ; `factor` = variable quali avec le nombre de niveaux et premières observations

```
str(DF)
## tibble [22 x 16] (S3: tbl_df/tbl/data.frame)
## $Pp
## $Age
19 ...
## $ Sexe
## $ Specialite_choisie «physique» ...
: num[1:22]12345678910... : num[1:22]181920181819182018
: chr [1:22] «F» «F» «F» «F» ...
: chr [1:22] «chimie» «physique» «maths»
## $ Groupe_TD
##$ Note_test_novembre ...
##$ Note_test_decembre 5 ...
##$ Temps_revision
```

```
30 290 ...
## $ Score_motivation
.
## $Score_estime_de_soi
## $ Score_sentiment_auto_efficacite: num [1:22] 2 4 1 5 2 4 4 1 2
3 ...
## $ Statut
nt» «primant» ...
## $ Admission
## $ condition
tin» ...
## $ score_attention_T0
## $ score_attention_T1
..
: chr [1:22] «primant» «doublant» «prima
: chr [1:22] «oui» «oui» «non» «oui» ...
: chr [1:22] «matin» «matin» «matin» «ma
:num[1:22]4365812368... :num[1:22]1491111126541311.
```

Il est possible qu'une variable qualitative ne soit pas reconnue comme telle.  
Dans ce cas, la forcer à devenir qualitative

```
Groupe_TD <- factor (DF$Groupe_TD)
```

Il est possible d'associer des étiquettes à certaines modalités. Par ex, il est possible de spécifier que le groupe de TD 1 correspond au Groupe du lundi matin et que le groupe de TD 2 correspond au Groupe du vendredi midi.

```
Groupe_TD<-factor(DF$Groupe_TD, levels=c(1,2), labels=c(« Groupe du
lundi mat in », « Groupe du vendredi midi»))
#une rapide vérification est alors recommandée
table(Groupe_TD, deparse.level=2, useNA =>always»)
## Groupe_TD
## Groupe du lundi matin Groupe du vendredi midi <NA> ## 11 11 0
```

La commande nrow permet de savoir combien de lignes comporte le tableau de données constituant le dataframe DF

```
nrow(DF)
## [1] 22
```

La commande `summary` est intéressante. Elle permet d'obtenir un résumé numérique univarié pour chacune des variables. Pour les variables numériques seront indiqués les indicateurs de tendance centrale, de dispersion et d'étendue. Pour les variables qualitatives, un tableau d'effectif sera associé à chacune des modalités.

```
summary(DF)
##      Pp          Age          Sexe
## Min.   : 1.00   Min.   :17.00   Length:22
## 1st Qu.: 6.25   1st Qu.:18.00
## Median :11.50   Median :18.50
## Mean   :11.50   Mean   :18.73
## 3rd Qu.:16.75   3rd Qu.:19.00
## Max.   :22.00   Max.   :21.00
##
Specialite_choisie
Length:22
Class :character
## Groupe_TD
## Min.   :1.0
## 1st Qu.:1.0
## Median :1.5
## Mean   :1.5   Mean   :12.00   Mean   :13.73   Mean
:119.7
## 3rd Qu.:2.0   3rd Qu.:15.00   3rd Qu.:15.00   3rd Qu.:130.0 ## Max.   :2.0
Max.   :19.00   Max.   :19.00   Max.   :380.0 ##
Note_test_novembre
Min.   : 2.00
1st Qu.:10.25
Median :12.50
Note_test_decembre
Min.   : 9.00
1st Qu.:12.00
Median :14.00
Temps_revision
Min.   : 10.0
1st Qu.: 48.5
Median : 95.0
## Score_motivation Score_estime_de_soi ## Min.   : 3.000 Min.   :1.000
## 1st Qu.: 7.000 1st Qu.:2.000
## Median : 9.000 Median :3.000
## Mean   : 8.952   Mean   :2.864
## 3rd Qu.:12.000   3rd Qu.:4.000
## Max.   :14.000   Max.   :5.000
## NA's   :1
Score_sentiment_auto_efficacite
Min.   :1.000
1st Qu.:2.000
Median :2.500
Mean   :2.727
3rd Qu.:4.000
Max.   :5.000
Class :character
Mode   :character   Mode   :character
```

```
##      Statut      Admission      condition
T0
## Length:22      Length:22      Length:22
## Class :character Class :character Class :character
## Mode :character Mode :character Mode :character ##
##
##
##
## score_attention_T1
## Min.   : 1.000
## 1st Qu.: 4.250
## Median : 7.000
## Mean   : 7.773
## 3rd Qu.:11.000
## Max.   :15.000
##
score_attention_
Min.   : 1.000
1st Qu.: 4.000
Median : 5.500
Mean   : 6.409
3rd Qu.: 8.000
Max.   :14.000
```

Pour une présentation plus élégante, il est possible de faire appel à la variable `describe`. Apparaîtront alors les variables numériques quantitatives (nom, moyenne, écart-type, médiane, valeur minimale, valeur maximale, nombre de valeurs valides) puis les variables catégorielles (modalités, nombre de sujets associés à chacune des modalités et pourcentage de sujets associés à chacune des modalités)

```
library(prettyR)
describe(DF, num.desc=c(«mean»,«sd»,«median»,«min»,«max»,«valid.n»))
## Description of DF
##
## Numeric
## mean
## Pp 11.50
## Age 18.73
## Groupe_TD      1.50
## Note_test_novembre 12.00  4.67  12.5  2  19
## Note_test_decembre 13.73  2.64  14.0  9  19
## Temps_revision 119.68 100.07
## Score_motivation
## Score_estime_de_soi ##Score_sentiment_auto_efficacite ## score_
attention_T0
## score_attention_T1
##
## Factor
##
```

```

## Sexe F H
## Count 11 11
## Percent 50 50
## Mode >1 mode
##
95.0 10 380      22
 9.0  3  14      21
 3.0  1   5      22
 2.5  1   5      22
  sd median
6.49  11.5
0.98  18.5
0.51   1.5
min max valid.n
  1  22      22
 17  21      22
  1   2      22
8.95  3.28
2.86  1.36
2.73  1.28
6.41  3.62   5.5  1  14      22
7.77  3.99   7.0  1  15      22
##
##
##
## Mode physique
##
Specialite_choisie physique chimie maths Count 8.00 7.00 7.00
Percent 36.36 31.82 31.82
## Statut      primant doublant
## Count      12.00   10.00
## Percent    54.55   45.45
## Mode primant
##
## Admission   oui   non
## Count      15.00  7.00
## Percent    68.18 31.82
## Mode oui
##
## condition  matin  soir
## Count       11   11
## Percent     50   50
## Mode >1 mode

```

## Exploration plus fine d'une variable du dataframe

### Variable qualitative

Il est possible de n'afficher que les n premières modalités prises par la variable "Note\_test\_novembre" contenue dans le dataframe DF. Par exemple, nous pouvons choisir de n'afficher que les dix premières variables

```
head(factor(DF$Specialite_choisie), n = 10)
## [1] chimie physique maths physique physique maths chimie physiq
ue
## [9] chimie maths
## Levels: chimie maths physique
```

La fonction table permet d'obtenir un tableau d'effectif, avec le nombre d'observations associées à chacune des modalités

```
table(DF$Specialite_choisie, deparse.level=2, useNA =>always)
## DF$Specialite_choisie
## chimie maths physique <NA> ## 7 7 8 0
```

Il est aussi possible d'obtenir un tableau de contingence en croisant deux variables d'intérêt, par exemple pour les variables Statut et Admission

```
table(DF$Statut,DF$Admission, deparse.level=2, useNA =>always)
head(factor(DF$Specialite_choisie), n = 10)
## [1] chimie physique maths physique physique maths chimie physiq
ue
## [9] chimie maths
## Levels: chimie maths physique
table(DF$Specialite_choisie, deparse.level=2, useNA =>always)
## DF$Specialite_choisie
## chimie maths physique <NA> ## 7 7 8 0
##           DF$Admission
## DF$Statut non oui <NA>
##  doublant  3  7  0
##  primant   4  8  0
##  <NA>      0  0  0
xtabs(~Statut+ Admission, DF)
##           Admission
## Statut    non oui
##  doublant  3  7
##  primant   4  8
```

## Kit stat\_R Niveau débutant

Il est possible d'obtenir des pourcentages

```
tab<-table(DF$Statut,DF$Admission, deparse.level=2) (prop.
table(tab))*100
##           DF$Admission
## DF$Statut   non     oui
##  doublant 13.63636 31.81818
##  primant  18.18182 36.36364
summary(DF$Specialite_choisie)
##   Length      Class      Mode
##      22 character character
(prop.table(tab))*100
##           DF$Admission
## DF$Statut   non     oui
##  doublant 13.63636 31.81818
##  primant  18.18182 36.36364
summary(DF$Specialite_choisie)
##   Length      Class      Mode
##      22 character character
```

pour voir les modalités qui sont utilisées pour cette variable

```
unique(DF$Specialite_choisie)
## [1] «chimie» «physique» «maths»
```

pour que R renvoie les numéros d'observations pour lesquelles la valeur prise par "Spécialité choisie" vaut "maths"

```
which(DF$Specialite_choisie==»maths»)
## [1]  3  6 10 11 16 18 21
```

### Variable quantitative

Il est possible de visualiser l'ensemble des observations d'une variable (par ex, la variable "Note-test\_novembre") du dataframe DF

DF\$Note\_test\_novembre

```
## [1] 12 11 16 14 13  9 11 15 18  7  2 19  2 14 16 18  6 15 11
10 12 13
```

Pour connaître le nombre total d'observations de la variable "Note\_test\_novembre" contenue dans le dataframe DF

```
length(DF$Note_test_novembre)
## [1] 22
```

Il est possible d'afficher la 1ère observation de la variable "Note\_test\_novembre" contenue dans le dataframe DF

```
DF$Note_test_novembre[1]
## [1] 12
```

## Kit stat\_R Niveau débutant

Il est possible d'afficher les 5 premières observations de la variable "Note\_test\_novembre" contenue dans le dataframe DF

```
DF$Note_test_novembre[1:5]
## [1] 12 11 16 14 13
#Autre commande possible
head(DF$Note_test_novembre, n=5)
## [1] 12 11 16 14 13
```

Il est possible de connaître le nombre d'observations répondant à la condition valeur supérieure à 10» pour la variable "Note\_test\_novembre"

```
table(DF$Note_test_novembre>10)
##
## FALSE  TRUE
##      6   16
```

Il est possible de calculer un seul indicateur, de tendance centrale, de dispersion ou d'étendue

```
#Pour calculer la valeur minimale
min(DF$Age, na.rm=TRUE)
## [1] 17
#Pour calculer la valeur médiane
median(DF$Age, na.rm=TRUE)
## [1] 18.5
#Pour calculer la valeur moyenne
mean(DF$Age, na.rm=TRUE)
## [1] 18.72727
#Pour calculer l'écart-type
sd(DF$Age, na.rm=TRUE)
## [1] 0.9847319
```

Résumé numérique de la variable quanti d'intérêt

```
summary(DF$Note_test_novembre)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      2.00  10.25   12.50   12.00   15.00   19.00
```

Pour calculer des moyennes conditionnelles moyenne de la note au test de novembre quand Groupe de TD = 1 et quand groupe TD = 2

```
tapply(DF$Note_test_novembre, DF$Groupe_TD, mean, na.rm=TRUE)
##      1      2
## 11.54545 12.45455
# autre commande possible
aggregate(Note_test_novembre~Groupe_TD, DF, mean)
##      Groupe_TD Note_test_novembre
## 1      1      11.54545
## 2      2      12.45455
# autre commande possible
by(DF$Note_test_novembre, DF$Groupe_TD, mean, na.rm=TRUE)
## DF$Groupe_TD: 1
## [1] 11.54545
```

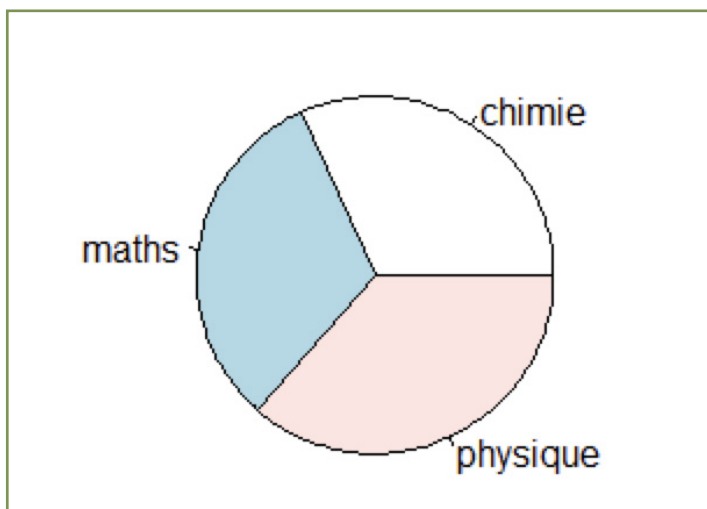
```
## -----  
## DF$Groupe_TD: 2  
## [1] 12.45455
```

## Exploration graphique

Visualisation de données qualitatives

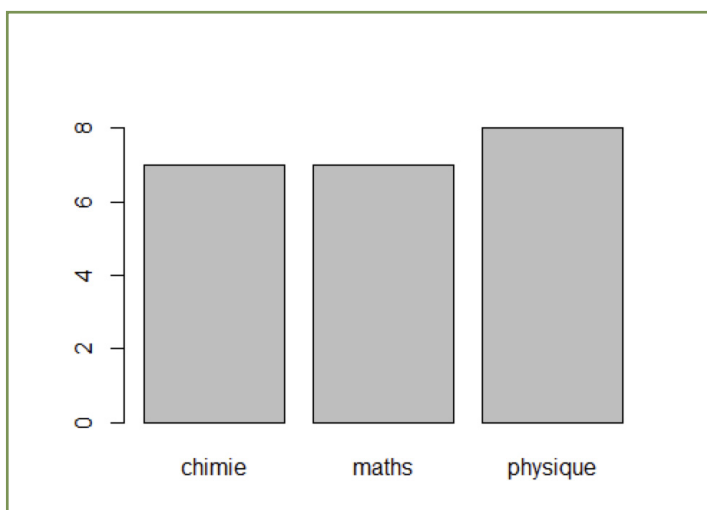
Pour un diagramme circulaire

```
pie(table(DF$Specialite_choisie))
```



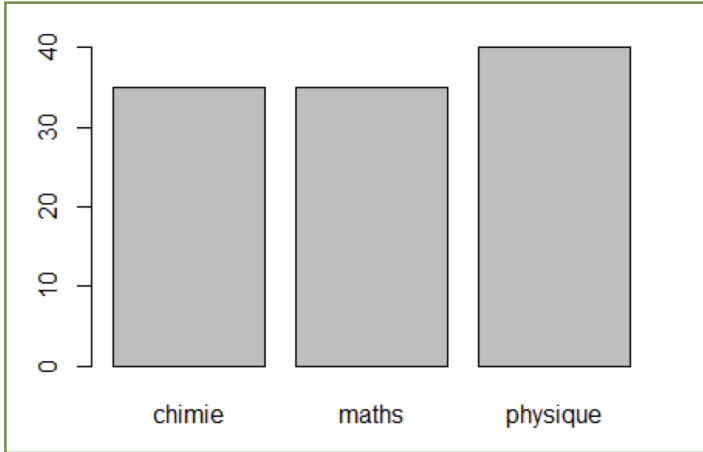
Pour un diagramme en bâton

```
barplot(table(DF$Specialite_choisie))
```



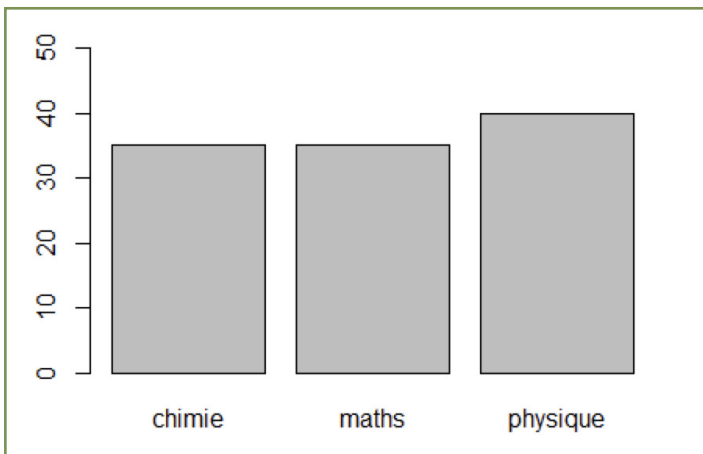
Pour avoir des % (Pour DF n=20)

```
barplot(table(DF$Specialite_choisie)*100/20)
```



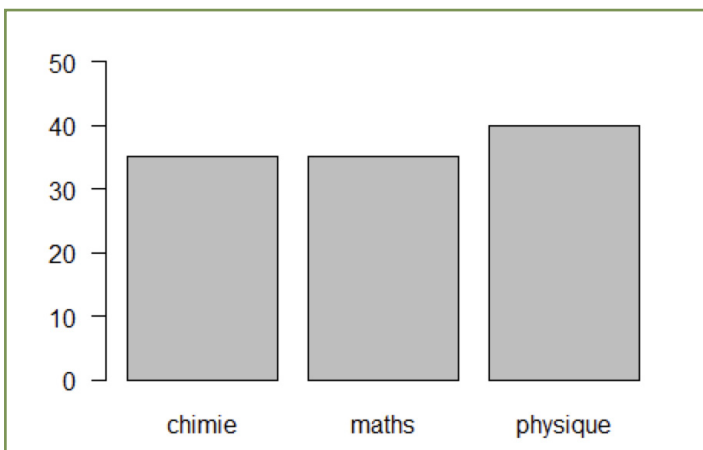
Pour avoir un axe y compris entre 0 et 50

```
barplot(table(DF$Specialite_choisie)*100/20, ylim=c(0,50))
```



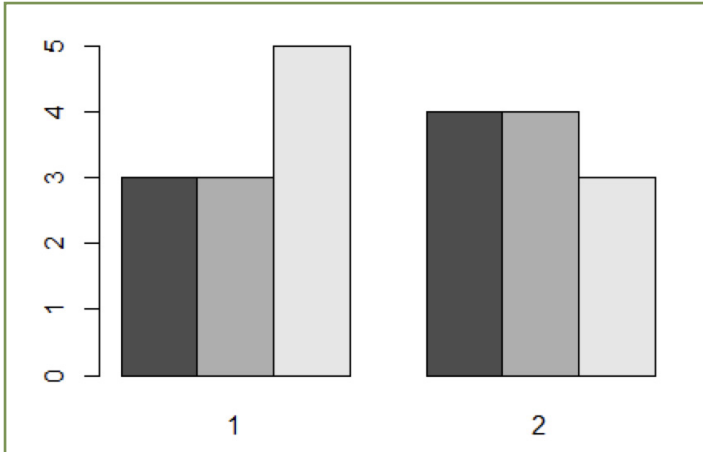
Pour orienter les labels horizontalement sur l'axe des ordonnées

```
barplot(table(DF$Specialite_choisie)*100/20, ylim=c(0,50), las=1)
```

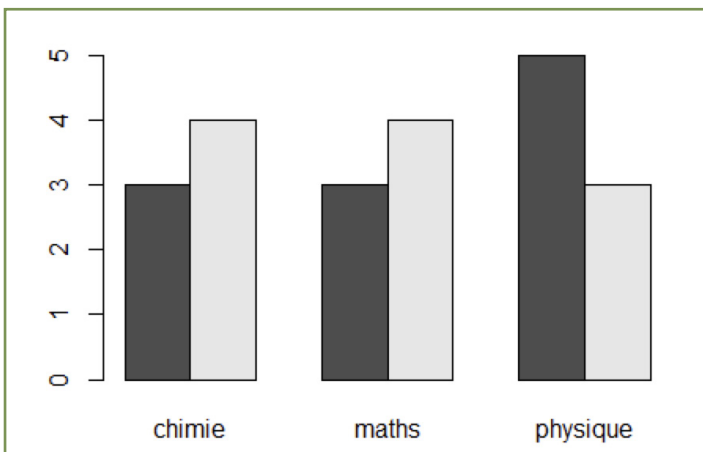


Pour afficher deux variables

```
barplot(xtabs(~Specialite_choisie + Groupe_TD, DF), beside=TRUE)
```



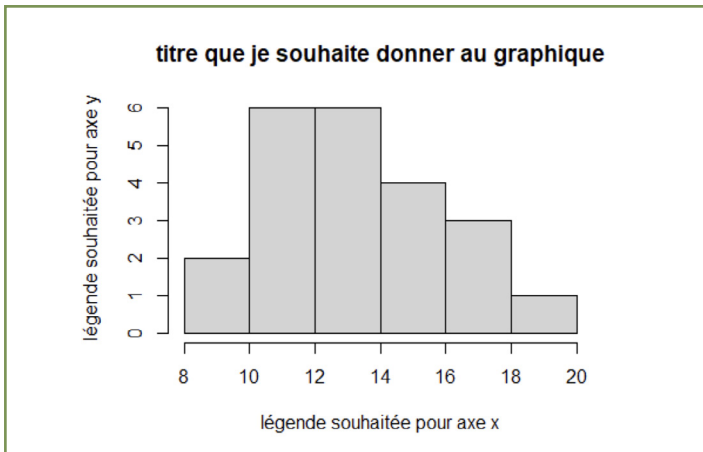
```
barplot(xtabs(~Groupe_TD + Specialite_choisie, DF), beside=TRUE)
```



## Visualisation de données quantitatives

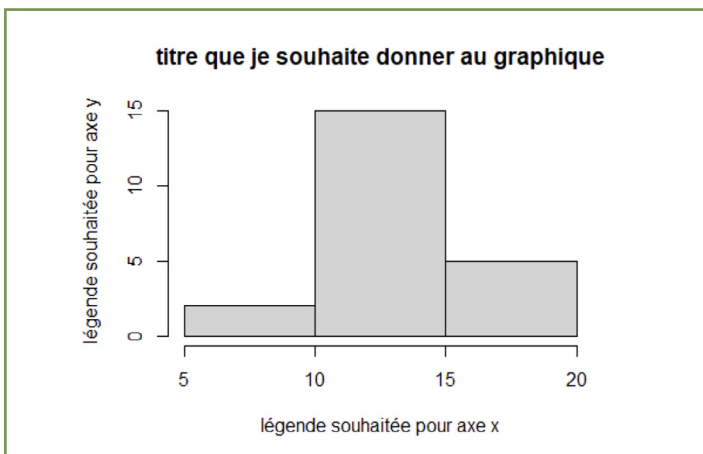
pour un histogramme représentant une variable continue

```
hist(DF$Note_test_decembre, main=»titre que je souhaite donner au graphique«, xlab=»légende souhaitée pour axe x«, ylab=» légende souhaitée pour axe y«)
```



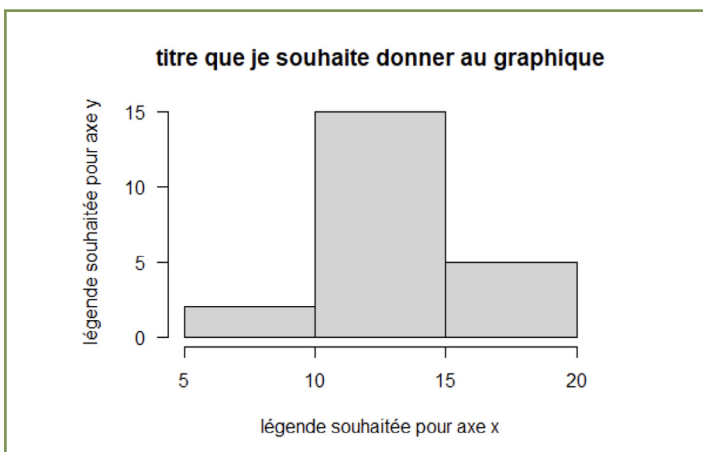
Si je ne veux que 3 batons

```
hist(DF$Note_test_decembre, nclass=3, main=»titre que je souhaite donner au graphique», xlab=»légende souhaitée pour axe x», ylab=»légende souhaitée pour axe y»)
```



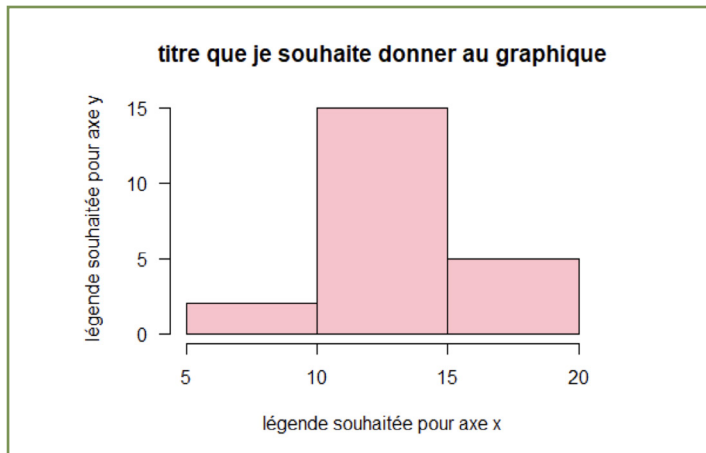
Pour réorienter les labels

```
hist(DF$Note_test_decembre, nclass=3, las=1, main=»titre que je souhaite donner au graphique», xlab=»légende souhaitée pour axe x», ylab=»légende souhaitée pour axe y»)
```



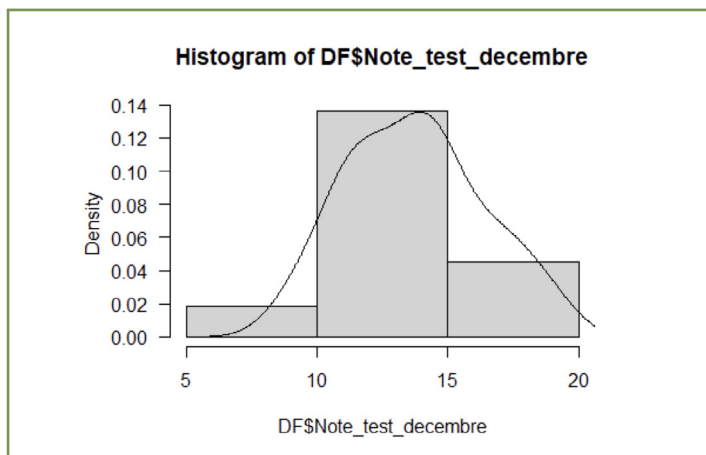
il est possible de choisir la couleur des bâtons

```
hist(DF$Note_test_decembre, nclass=3, las=1, main=»titre que je  
souhaite donner au graphique«, xlab=»légende souhaitée pour axe x«,  
ylab=» légende souhaitée pour axe y«, col=»pink«)
```



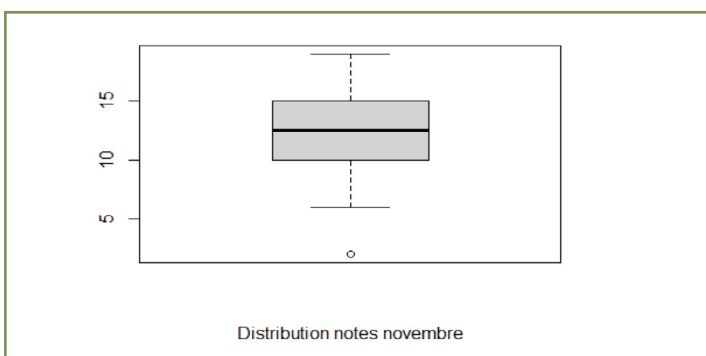
Il est aussi possible d'utiliser une densité non paramétrique

```
hist(DF$Note_test_decembre, nclass=3, prob=TRUE, las=1)  
lines(density(DF$Note_test_decembre, na.rm=TRUE))
```

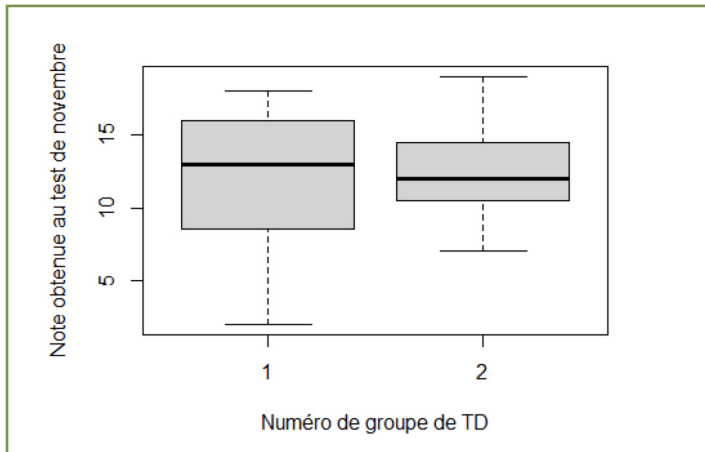


Pour une boîte à moustaches

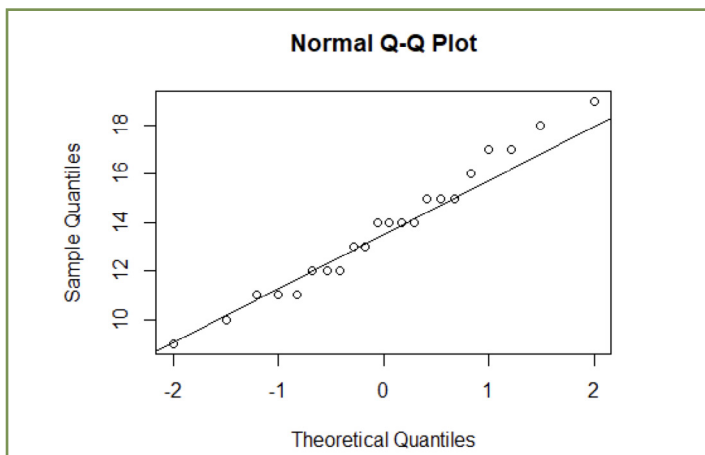
```
boxplot(DF$Note_test_novembre, xlab=»Distribution notes novembre«)
```



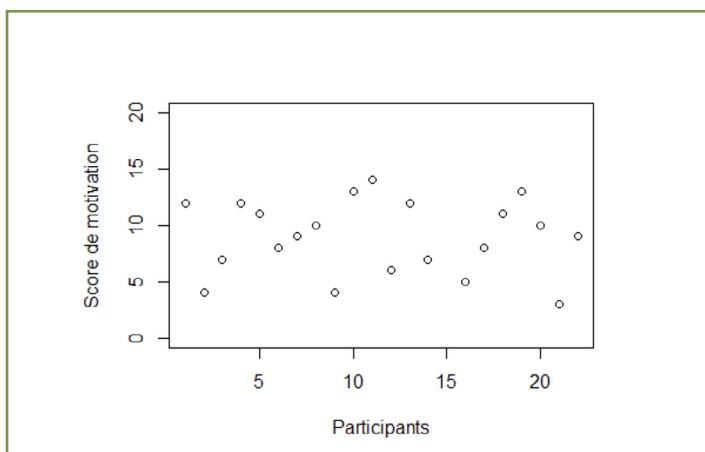
```
#pour deux variables quanti  
boxplot(DF$Note_test_novembre~DF$Groupe_TD,ylab = «Note obtenue au  
test de novembre», xlab=»Numéro de groupe de TD»)
```



```
qqnorm(DF$Note_test_decembre)  
qqline(DF$Note_test_decembre)
```

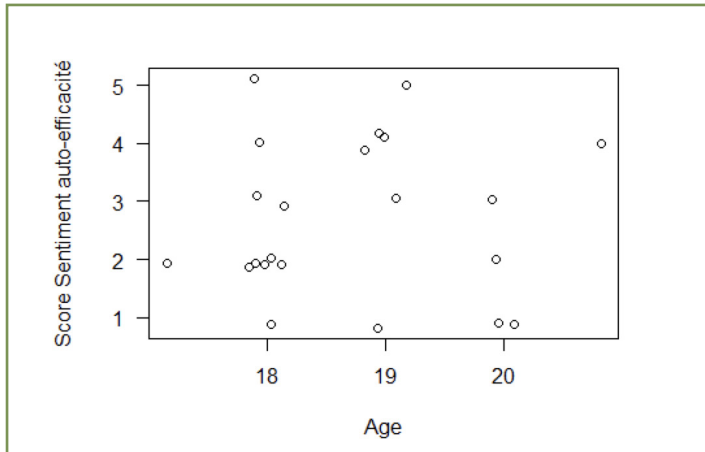


```
plot(DF$Score_motivation, ylab=»Score de motivation»,  
xlab=»Participants», ylim=c(0,20))
```



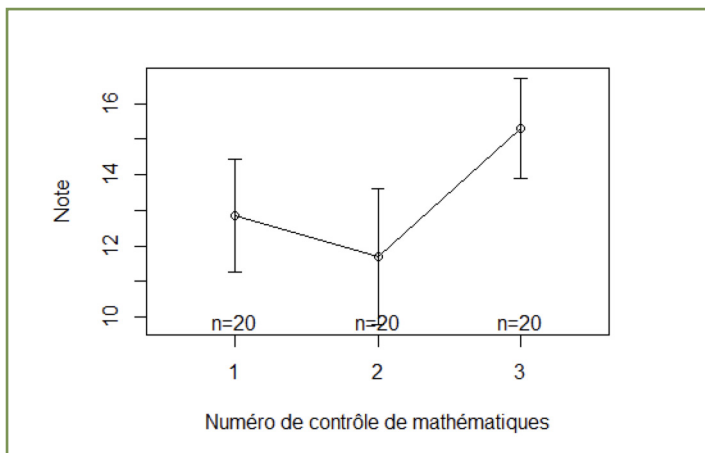
## Distribution conjointe de 2 variables quantitatives

```
plot(jitter(DF$Age),jitter(DF$Score_sentiment_auto_efficacite), las=1,  
ylab=»Score Sentiment auto-efficacité», xlab=»Age«)
```



## pour avoir évolution temporelle de la distribution d'une variable aléatoire quanti

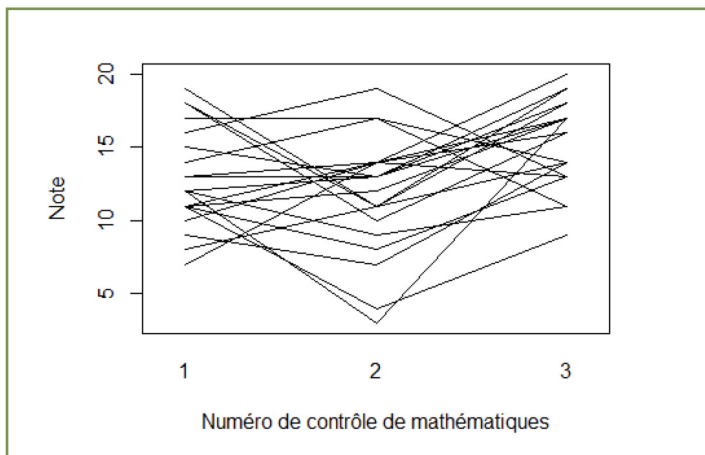
```
DF2 <- read_excel(«C:/Users/marine.moyon/Desktop/ChaireIP/mon_  
dataset_ChairesIP.xlsx»,2)  
library(gplots)  
##  
## Attaching package: 'gplots'  
## The following object is masked from 'package:stats':  
##  
## lowess  
plotmeans(DF2$Note~DF2$Maths_numero_de_controle, gap=0,  
barcol=»black«, ylab=»Note«, xlab=»Numéro de contrôle de  
mathématiques«)  
## Warning in plot.xy(xy.coords(x, y), type = type, ...): «gap» n'est  
pas un  
## paramètre graphique
```



## Kit stat\_R Niveau débutant

Si on ne veut pas évolution moyenne des participants au fil du temps mais évolution de chaque sujet.

```
interaction.plot(DF2$Maths_numero_de_controle, DF2$Pp, DF2$Note,  
lty=1, legend=FALSE, ylab=»Note«, xlab=»Numéro de contrôle de  
mathématiques«)
```



## Application d'un filtre pour réduire le dataset

Afficher le DF mais uniquement pour les participants qui ont choisie la spécialité mathématiques

```
DFshort<-subset(DF, Specialite_choisie ==»maths«)
```

Afficher les valeurs prises pour la variable Age, uniquement pour les participants qui ont choisie la spécialité mathématiques

```
DFshort2<-subset(DF, Specialite_choisie ==»maths«, Age)
```

pour les participants dont la spécialité choisie est maths et pour qui la note au test de décembre est strictement supérieure à 10

```
DFshort3<-subset(DF, Spécialité_choisie==»maths« & Note_test_  
decembre>10)
```

## Tests Statistiques

### Test de Chi-2

```
chisq.test(DF$Statut, DF$Admission, correct=FALSE) # Bien mettre
correct=FALSE sinon R propose un test de Chi2 avec correction de
continuité ; test plus robuste mais nettement moins puissant.
## Warning in chisq.test(DF$Statut, DF$Admission, correct = FALSE):
Chi-squared
## approximation may be incorrect
##
## Pearson's Chi-squared test
##
## data: DF$Statut and DF$Admission
## X-squared = 0.027937, df = 1, p-value = 0.8673
```

## Test exact de Fisher

```
fisher.test(DF$Statut, DF$Admission)
##
## Fisher's Exact Test for Count Data
##
## data: DF$Statut and DF$Admission
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.09212345 7.26648360
## sample estimates:
## odds ratio
## 0.8631582
```

## Test de Corrélation

### Calculer un coefficient de corrélation de Pearson

```
cor(DF$Age, DF$Score_estime_de_soi, use=»complete.obs»)
## [1] 0.04215805
#autre commande possible
cor.test(DF$Age, DF$Score_estime_de_soi)
##
## Pearson's product-moment correlation
##
## data: DF$Age and DF$Score_estime_de_soi
## t = 0.1887, df = 20, p-value = 0.8522
```

## Kit stat\_R Niveau débutant

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.3863169  0.4556674
## sample estimates:
##      cor
## 0.04215805
```

### Calculer un coefficient de corrélation de Spearman

```
cor.test(DF$Age, DF$Score_estime_de_soi, method=»spearman»)
## Warning in cor.test.default(DF$Age, DF$Score_estime_de_soi, method
=
## «spearman»): Cannot compute exact p-value with ties
##
## Spearman's rank correlation rho
##
## data:  DF$Age and DF$Score_estime_de_soi
## S = 1605.4, p-value = 0.6789
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.09352807
```

## Test T de Student pour échantillons indépendants

variable à étudier tilde la variable qui détermine les sous groupes. var.equal=TRUE  
sinon par défaut, R va utiliser l'approximation de Welch

```
t.test(DF$Note_test_decembre~DF$Groupe_TD, var.equal=TRUE)
##
## Two Sample t-test
##
## data:  DF$Note_test_decembre by DF$Groupe_TD
## t = -1.1386, df = 20, p-value = 0.2683
## alternative hypothesis: true difference in means is not equal to
0
## 95 percent confidence interval:
## -3.604520  1.059065
## sample estimates:
## mean in group 1 mean in group 2
##      13.09091      14.36364
# autre commande qui revient au même
t.test(DF$Note_test_decembre[DF$Groupe_TD==1],DF$Note_test_
decembre[DF$Groupe_TD==2], var.equal=TRUE )
##
## Two Sample t-test
##
## data:  DF$Note_test_decembre[DF$Groupe_TD == 1] and DF$Note_
test_decembre[DF$Groupe_TD == 2]
```

```
## t = -1.1386, df = 20, p-value = 0.2683
## alternative hypothesis: true difference in means is not equal to
## 0
## 95 percent confidence interval:
## -3.604520  1.059065
## sample estimates:
## mean of x mean of y
## 13.09091  14.36364
```

### Test de Wilcoxon

```
wilcox.test(DF$Note_test_novembre~DF$Groupe_TD)
## Warning in wilcox.test.default(x = c(12, 16, 13, 11, 18, 2, 2, 16,
## 18, 6, :
## cannot compute exact p-value with ties
##
## Wilcoxon rank sum test with continuity correction
##
## data:  DF$Note_test_novembre by DF$Groupe_TD
## W = 61.5, p-value = 0.9737
## alternative hypothesis: true location shift is not equal to 0
```

### Test T de Student pour échantillons appariés

```
t.test(DF$Note_test_novembre,DF$Note_test_decembre, paired=TRUE)
##
## Paired t-test
##
## data:  DF$Note_test_novembre and DF$Note_test_decembre
## t = -1.3719, df = 21, p-value = 0.1846
## alternative hypothesis: true difference in means is not equal to
## 0
## 95 percent confidence interval:
## -4.3456070  0.8910616
## sample estimates:
## mean of the differences
## -1.727273
```

### Anova

```
Anova(lm(note ~ niveau*conditionc ,data=DF),type='III')
```

### Modèles de régression

```
Chargement de la fonction PRE
PRE<-function(model){
  library(car)
  A<-Anova(model, type=»III»)
  R<-A[1]
  dim<-dim(R)
  d<-dim[1]
```

```
SSE<-R[d,1]
ETASQP<- R/(SSE+R)
ANOVA<-cbind(A,ETASQP); ANOVA<-round(ANOVA, digits=3)
colnames(ANOVA)<-c(«SumSq», «Df», «F», «p», «PRE»)
ANOVA [d,c(3,4,5)]<-»»
print(ANOVA)
}
```

### Régression simple : un facteur continu

```
DF <- read_excel(«C:/Users/marine.moyon/Desktop/ChaireIP/dataset_
demo_regressions.xlsx»,1)
fit1<-lm(pourcentage_reussite_eval_formative~moyenne_bac, DF)
summary(fit1)
##
## Call:
## lm(formula = pourcentage_reussite_eval_formative ~ moyenne_bac,
##     data = DF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.404  -7.225  -1.417   7.979  19.147
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  38.7891     6.7837   5.718 2.02e-05 ***
## moyenne_bac   1.9445     0.5566   3.494 0.00259 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.605 on 18 degrees of freedom
## Multiple R-squared:  0.4041, Adjusted R-squared:  0.371
## F-statistic: 12.2 on 1 and 18 DF, p-value: 0.002594
confint(fit1)
##              2.5 %      97.5 %
## (Intercept) 24.5371061 53.041121
## moyenne_bac  0.7751308  3.113905
PRE(fit1)
## Loading required package: carData
##              SumSq Df      F      p    PRE
## (Intercept) 3016.272  1 32.695    0 0.645
## moyenne_bac 1125.927  1 12.205 0.003 0.404
## Residuals  1660.564 18
```

Régression simple, plan inter (VD quantitative, VD catégorielle k=2) ; soit l'équivalent du test de t pour échantillons indépendants

```
DF<-within(DF, {
  supportc<--0.5*(support==»Papier») +0.5*(support==»Num»)})

fit2<-lm(pourcentage_reussite_eval_formative~supportc, DF)
summary(fit2)
```

```
##
## Call:
## lm(formula = pourcentage_reussite_eval_formative ~ supportc,
##     data = DF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.812  -8.278   3.722   9.772  13.708
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    67.362     3.667   18.37 4.17e-13 ***
## supportc       22.157     9.890    2.24  0.0379 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11 on 18 degrees of freedom
## Multiple R-squared:  0.218, Adjusted R-squared:  0.1746
## F-statistic: 5.019 on 1 and 18 DF, p-value: 0.03793
confint(fit2)
##              2.5 %    97.5 %
## (Intercept) 59.657159 75.06729
## supportc    1.378161 42.93618
PRE(fit2)
##              SumSq Df      F      p    PRE
## (Intercept) 40839.021  1 337.365    0 0.949
## supportc    607.539  1   5.019 0.038 0.218
## Residuals   2178.952 18
```

Régression simple, plan intra (VD quantitative, VD catégorielle k=2); soit l'équivalent du test de t pour échantillons appariés

```
DF2 <- read_excel(«C:/Users/marine.moyon/Desktop/ChaireIP/dataset_
demo_regressions.xlsx»,2)

Wdiff<-DF2$difficile-DF2$facile

fit3<-lm(Wdiff~1, DF2)

summary(fit3)
##
## Call:
## lm(formula = Wdiff ~ 1, data = DF2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.8636 -1.6136  0.1364  1.8864  4.1364
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -9.1364     0.5556  -16.45  1.8e-13 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.606 on 21 degrees of freedom
confint(fit3)
##           2.5 %      97.5 %
## (Intercept) -10.2917 -7.981029
PRE(fit3)
##           SumSq Df      F p    PRE
## (Intercept) 1836.409  1 270.456 0 0.928
## Residuals   142.591 21
```

Régression multiple : VD quantitative, deux VI quantitatives continues

```
fit4<-lm(niveau_anglais~pied, DF2)
summary(fit4)
##
## Call:
## lm(formula = niveau_anglais ~ pied, data = DF2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.9419 -3.4387 -0.4438  3.7839 10.7043
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -73.0915     7.1159  -10.27 2.01e-09 ***
## pied          3.5009      0.2014   17.39 1.54e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.246 on 20 degrees of freedom
## Multiple R-squared:  0.9379, Adjusted R-squared:  0.9348
## F-statistic: 302.3 on 1 and 20 DF, p-value: 1.537e-13
confint(fit4)
##           2.5 %      97.5 %
## (Intercept) -87.935094 -58.247887
## pied         3.080896   3.920961
PRE(fit4)
##           SumSq Df      F p    PRE
## (Intercept) 2903.005  1 105.504 0 0.841
## pied        8317.506  1 302.283 0 0.938
## Residuals   550.313 20
fit5<-lm(niveau_anglais~pied+age, DF2)
summary(fit5)
##
## Call:
## lm(formula = niveau_anglais ~ pied + age, data = DF2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.709 -3.393  0.755  3.140  6.351
##
```

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12.6278    18.6334  -0.678  0.50613
## pied         0.8294     0.7998   1.037  0.31277
## age          2.5658     0.7521   3.411  0.00293 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.238 on 19 degrees of freedom
## Multiple R-squared:  0.9615, Adjusted R-squared:  0.9575
## F-statistic: 237.3 on 2 and 19 DF,  p-value: 3.633e-14
confint(ft5)
##           2.5 %    97.5 %
## (Intercept) -51.6278496 26.372340
## pied        -0.8446857  2.503500
## age          0.9915711  4.140042
PRE(ft5)
##           SumSq Df      F      p  PRE
## (Intercept)  8.249  1  0.459 0.506 0.024
## pied        19.314  1  1.075 0.313 0.054
## age         209.033  1 11.637 0.003  0.38
## Residuals   341.280 19
```

Régression multiple : VD quantitative, une VI catégorielle et une VI quantitative

```
DF2<-within(DF2, {
  sexec<--0.5*(sexe==»homme«)+0.5*(sexe==»femme«)})

fit6<-lm(note_TD~sexec, DF2)
summary(fit6)
##
## Call:
## lm(formula = note_TD ~ sexec, data = DF2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.5636 -0.7659 -0.0273  1.2091  3.8364
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.9273     0.4345  29.751  <2e-16 ***
## sexec        -2.2727     0.8690  -2.615  0.0166 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.038 on 20 degrees of freedom
## Multiple R-squared:  0.2548, Adjusted R-squared:  0.2176
## F-statistic: 6.839 on 1 and 20 DF,  p-value: 0.01657
confint(fit6)
##           2.5 %    97.5 %
## (Intercept) 12.020884 13.8336611
## sexec      -4.085504 -0.4599505
```

```

PRE(ft6)
##              SumSq Df          F      p    PRE
## (Intercept) 3676.516  1 885.113    0 0.978
## sexec       28.409  1   6.839 0.017 0.255
## Residuals   83.075 20
ft7<-lm(note_TD~sexec+temps_apprentissage+sexec*temps_apprentissage,
DF2)
summary(ft7)
##
## Call:
## lm(formula = note_TD ~ sexec + temps_apprentissage + sexec *
##      temps_apprentissage, data = DF2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.20491 -0.42826  0.05831  0.38674  1.13773
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.17547    0.33957  27.021 5.07e-16 ***
## sexec          3.52507    0.67914   5.191 6.16e-05
## ***
## temps_apprentissage    0.58469    0.04754  12.299 3.39e-10
## ***
## sexec:temps_apprentissage -0.92085    0.09508  -9.685 1.46e-08
## ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6537 on 18 degrees of freedom
## Multiple R-squared:  0.931, Adjusted R-squared:  0.9195
## F-statistic: 80.97 on 3 and 18 DF, p-value: 1.208e-10
confint(ft7)
##              2.5 %      97.5 %
## (Intercept)    8.462066  9.8888776
## sexec          2.098262  4.9518852
## temps_apprentissage    0.484808  0.6845659
## sexec:temps_apprentissage -1.120604 -0.7210887
PRE(ft7)
##              SumSq Df          F      p    PRE
## (Intercept)    311.974  1 730.135  0 0.976
## sexec          11.512  1  26.942  0 0.599
## temps_apprentissage    64.630  1 151.258  0 0.894
## sexec:temps_apprentissage    40.078  1  93.797  0 0.839
## Residuals         7.691 18
Effets simples
DF2<-within(DF2, {
  sexec<--0.5*(sexe==»homme») + 0.5*(sexe==»femme»)
  sexedH<-0*(DF2$sexe==»homme») + 1*(DF2$sexe==»femme»)
  sexedF<--1*(DF2$sexe==»homme») + 0*(DF2$sexe==»femme»)
})

ft8<-lm(note_TD~sexedH+temps_apprentissage+sexedH*temps_

```

```

apprentissage, DF2)
summary(fit8)
##
## Call:
## lm(formula = note_TD ~ sexedH + temps_apprentissage + sexedH *
##     temps_apprentissage, data = DF2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.20491 -0.42826  0.05831  0.38674  1.13773
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      7.41293     0.54604   13.576 6.76e-11
## ***
## sexedH           3.52507     0.67914    5.191 6.16e-05
## ***
## temps_apprentissage  1.04511     0.08002   13.060 1.28e-10
## ***
## sexedH:temps_apprentissage -0.92085     0.09508   -9.685 1.46e-08
## ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6537 on 18 degrees of freedom
## Multiple R-squared:  0.931, Adjusted R-squared:  0.9195
## F-statistic: 80.97 on 3 and 18 DF, p-value: 1.208e-10
confint(fit8)
##              2.5 %      97.5 %
## (Intercept)  6.2657410  8.5601289
## sexedH       2.0982617  4.9518852
## temps_apprentissage  0.8769893  1.2132312
## sexedH:temps_apprentissage -1.1206044 -0.7210887
PRE(fit8)
##              SumSq Df      F p    PRE
## (Intercept)  78.748  1 184.301 0 0.911
## sexedH       11.512  1  26.942 0 0.599
## temps_apprentissage  72.881  1 170.569 0 0.905
## sexedH:temps_apprentissage 40.078  1  93.797 0 0.839
## Residuals      7.691 18
fit9<-lm(note_TD~sexedF+temps_apprentissage+sexedF*temps_
apprentissage, DF2)
summary(fit9)
##
## Call:
## lm(formula = note_TD ~ sexedF + temps_apprentissage + sexedF *
##     temps_apprentissage, data = DF2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.20491 -0.42826  0.05831  0.38674  1.13773
##
## Coefficients:

```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    10.93801    0.40381  27.087 4.85e-16
## ***
## sexedF          3.52507    0.67914   5.191 6.16e-05
## ***
## temps_apprentissage  0.12426    0.05135   2.420  0.0263 *
## sexedF:temps_apprentissage -0.92085    0.09508  -9.685 1.46e-08
## ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6537 on 18 degrees of freedom
## Multiple R-squared:  0.931, Adjusted R-squared:  0.9195
## F-statistic: 80.97 on 3 and 18 DF, p-value: 1.208e-10
confint(fit9)
##              2.5 %      97.5 %
## (Intercept)    10.08963480 11.7863820
## sexedF          2.09826172  4.9518852
## temps_apprentissage  0.01638162  0.2321457
## sexedF:temps_apprentissage -1.12060439 -0.7210887
PRE(fit9)
##              SumSq Df      F      p      PRE
## (Intercept)    313.500  1 733.706    0 0.976
## sexedF          11.512  1  26.942    0 0.599
## temps_apprentissage  2.502  1   5.856 0.026 0.245
## sexedF:temps_apprentissage 40.078  1  93.797    0 0.839
## Residuals          7.691 18
```

### Régression multiple : Plan Mixte

```
DF3 <- read_excel(«C:/Users/marine.moyon/Desktop/ChaireIP/dataset_
demo_regressions.xlsx»,3)
DF3<-within(DF3, {
Wdiff<-difficile-facile
Wmoy<-rowMeans(cbind(facile, difficile))
conditionc<- -0.5*(condition==»seul») + 0.5*(condition==»pair»)
})

fit10<-lm(Wdiff~conditionc, DF3)
summary(fit10)
##
## Call:
## lm(formula = Wdiff ~ conditionc, data = DF3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.4545 -2.1818  0.1818  1.3636  5.5455
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -10.1818    0.6914 -14.725 3.38e-12 ***
## conditionc   -3.2727    1.3829  -2.367  0.0282 *
## ---
```

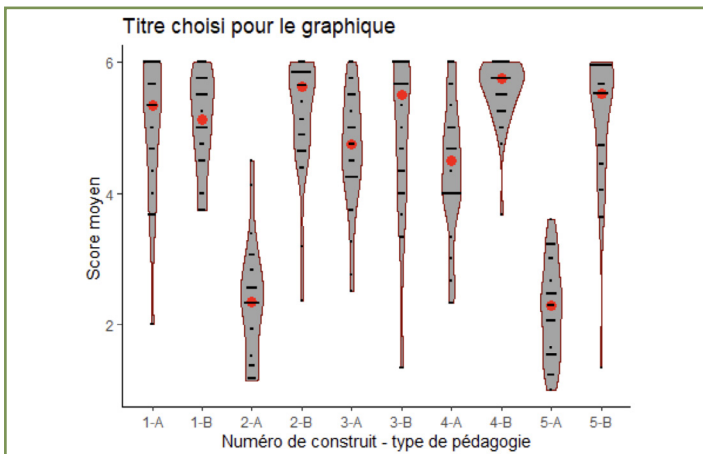
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.243 on 20 degrees of freedom
## Multiple R-squared:  0.2188, Adjusted R-squared:  0.1797
## F-statistic: 5.601 on 1 and 20 DF,  p-value: 0.02816
#ligne 1 = effet principal de la variable intra
#ligne 2 = effet d'interaction
PRE(fit10)
##              SumSq Df          F      p    PRE
## (Intercept) 2280.727  1 216.837    0 0.916
## conditionc   58.909  1   5.601 0.028 0.219
## Residuals   210.364 20
fit11<-lm(Wmoy~conditionc, DF3)
summary(fit11)
##
## Call:
## lm(formula = Wmoy ~ conditionc, data = DF3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.59091 -0.59091 -0.09091  0.40909  2.90909
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.0909      0.3174  44.399 < 2e-16 ***
## conditionc   -2.0000      0.6347  -3.151  0.00503 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.489 on 20 degrees of freedom
## Multiple R-squared:  0.3317, Adjusted R-squared:  0.2983
## F-statistic: 9.928 on 1 and 20 DF,  p-value: 0.005028
# ligne 2 = effet principal de la variable inter
PRE(fit11)
##              SumSq Df          F      p    PRE
## (Intercept) 4368.182  1 1971.282    0 0.99
## conditionc   22.000  1   9.928 0.005 0.332
## Residuals   44.318 20
```

### Représentation graphique plus aboutie

```
library(ggplot2)
DF <- read_excel(«C:/Users/marine.moyon/Desktop/ChaireIP/graphes.
xlsx»,1)

data_summary <- function(x) {
  m <- mean(x)
  ymin <- m-sd(x)
  ymax <- m+sd(x)
  return(c(y=m,ymin=ymin,ymax=ymax))
}
```

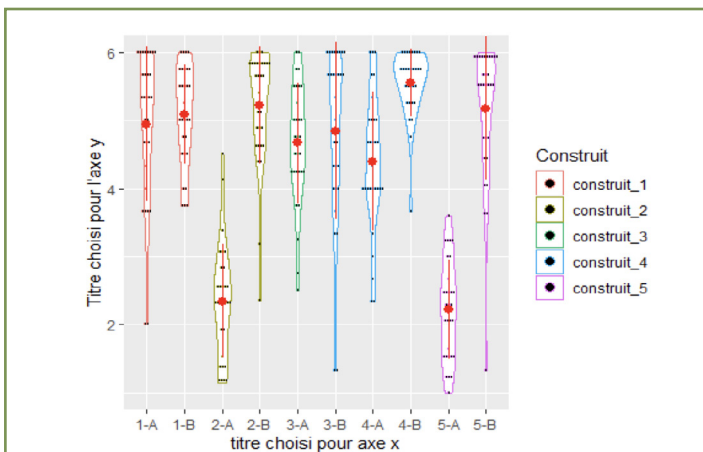
```
ggplot(DF, aes(x=Type_moyenne, y=Score,)) + geom_
violin(position=position_dodge(1),fill='#A4A4A4', color=»darkred») +
stat_summary(fun.y=median, geom=»point», size=3, color=»red») +
coord_cartesian(ylim=c(1,6))+ xlab(«Numéro de construit - type de
pédagogie») + ggtitle(«Titre choisi pour le graphique»)+ ylab(«Score
moyen «)+ geom_dotplot(binaxis='y', stackdir='center', dotsize=0.2,
position=»dodge», color=»black») + theme_classic()
## Warning: `fun.y` is deprecated. Use `fun` instead.
## `stat_bindot()` using `bins = 30`. Pick better value with
`binwidth`.
```



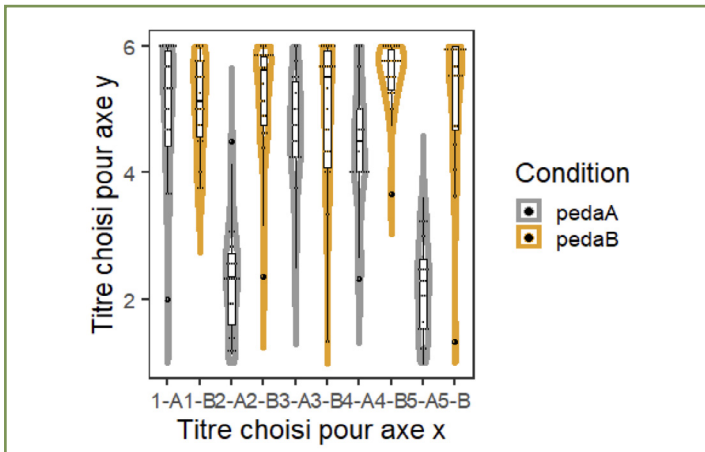
```
ggplot(DF, aes (x=Type_moyenne, y=Score, color=Construit))+ geom_
violin()+ geom_dotplot(binaxis='y', stackdir='center', dotsize=0.2)
+coord_cartesian(ylim=c(1,6)) + stat_summary(fun.data=data_summary,
geom=»pointrange», color=»red») +
xlab(«titre choisi pour axe x») + ylab(«Titre choisi pour l'axe
y»)+ theme(
```

```
panel.grid.major = element_line(size = 0.5, linetype = 'solid',
colour = «white»),
panel.grid.minor = element_line(size = 0.25, linetype = 'solid',
colour = «white»)
```

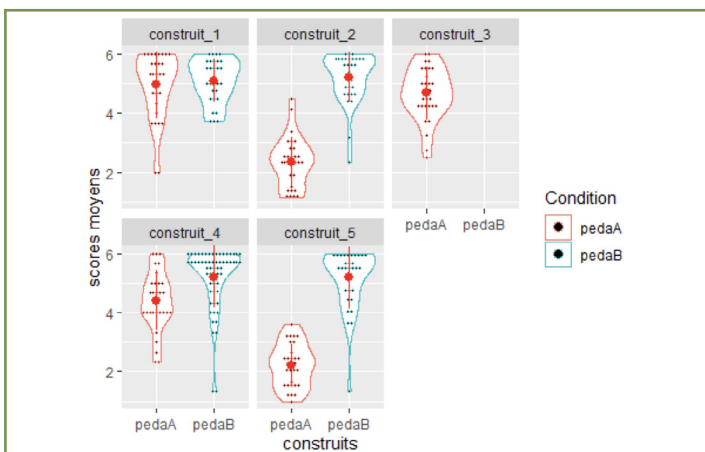
```
)
## `stat_bindot()` using `bins = 30`. Pick better value with
`binwidth`.
```



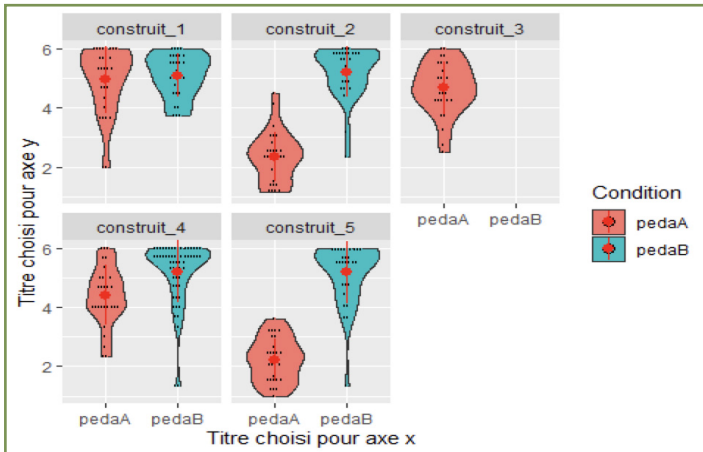
```
ggplot(DF, aes(x=Type_moyenne, y=Score, color=Condition)) + scale_
color_manual(values=c('#999999','#E69F00')) + geom_violin(trim=FALSE,
width=0.8, size=2)+ geom_boxplot(width=0.2, color=»black») + geom_
dotplot(binaxis='y', stackdir='center', dotsize=0.2)+ theme_bw(base_
size = 17)+ theme(
  panel.grid.major = element_line(size = 0.5, linetype = 'solid',
  colour = «white»),
  panel.grid.minor = element_line(size = 0.25, linetype = 'solid',
  colour = «white»)
)+ xlab(«Titre choisi pour axe x») + ylab(«Titre choisi pour axe
y»)+ylim(1,6)
## `stat_bindot()` using `bins = 30`. Pick better value with
`binwidth`.
## Warning: Removed 1225 rows containing missing values (geom_
violin).
```



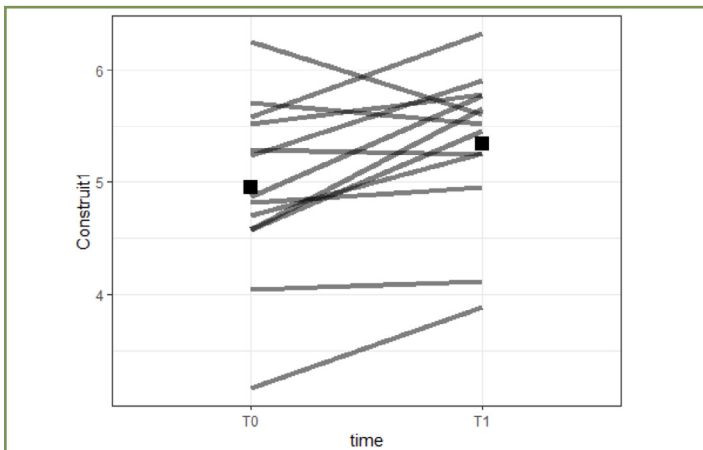
```
ggplot(DF, aes (x=Condition, y=Score, color=Condition))+geom_
violin()+ geom_dotplot(binaxis='y', stackdir='center', dotsize=0.7)
+ facet_wrap(~Construit)+ coord_cartesian(ylim=c(1,6)) + stat_
summary(fun.data=data_summary, geom=»pointrange«, color=»red») +
  xlab(«construits») + ylab(«scores moyens»)
## `stat_bindot()` using `bins = 30`. Pick better value with
`binwidth`.
```



```
ggplot(DF, aes (x=Condition, y=Score, fill=Condition))+geom_violin()+
geom_dotplot(binaxis='y', stackdir='center', dotsize=0.5) + facet_
wrap(~Construit)+ coord_cartesian(ylim=c(1,6)) + stat_summary(fun.
data=data_summary, geom=»pointrange«, color=»red«) +
xlab(«Titre choisi pour axe x») + ylab(«Titre choisi pour axe y»)
## `stat_bindot()` using `bins = 30`. Pick better value with
`binwidth`.
```

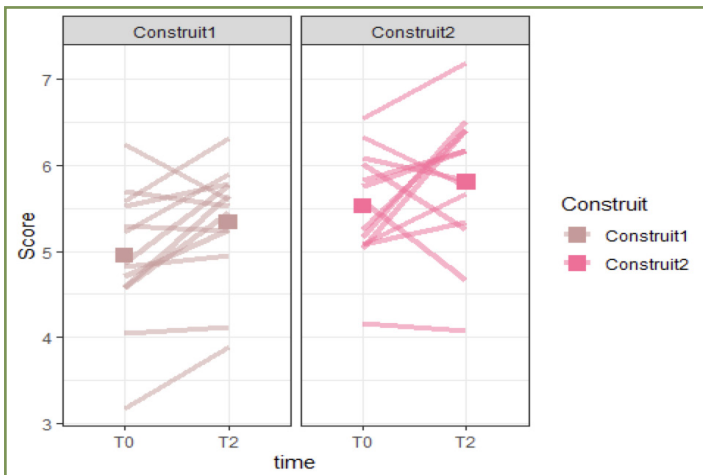


```
DF <- read_excel(«C:/Users/marine.moyon/Desktop/ChaireIP/graphes.
xlsx»,2)
DF<- as.data.frame(DF)
TIME=rbind('Pre', 'Pre', 'Post', 'Post')
CONDITION=rbind('Construit1', 'Construit2', 'Construit1', 'Construit2')
idata=idata<-data.frame (CONDITION, TIME)
D1 <- reshape (DF, varying=c(«Construit1.T0», «Construit1.T1»),
idvar=»pp«, direction=»long«)
ggplot(D1, aes(x=time, y=Construit1, group=pp))+geom_line(size=1.5,
alpha=.5)+theme_bw()+stat_summary(aes(group=1), geom=»point«, fun.
y=mean, shape=15, size=4)
## Warning: `fun.y` is deprecated. Use `fun` instead.
```



```
DF <- read_excel(«C:/Users/marine.moyon/Desktop/ChaireIP/graphes.
xlsx»,3)
DF<- as.data.frame(DF)
TIME=rbind('Pre', 'Pre', 'Post', 'Post')
idata=idata<-data.frame (TIME)
D1<- reshape (DF,varying=c(«Score.T0», «Score.T2»), idvar=»pp»,
direction=»long«)
```

```
ggplot(D1, aes(x=time, y=Score, color=Construit,
group=subject))+geom_line(size=1.5, alpha=.5)+facet_
grid(~Construit)+theme_bw()+stat_summary(aes(group=1),
geom=»point«, fun.y=mean, shape=15, size=4)+scale_color_
manual(values=c(«#CC9999»,»#FF6699«))
## Warning: `fun.y` is deprecated. Use `fun` instead.
```



```
DF <- read_excel(«C:/Users/marine.moyon/Desktop/ChaireIP/graphes.
xlsx»,3)
DF<- as.data.frame(DF)
TIME=rbind('Pre', 'Pre', 'Post', 'Post')
idata=idata<-data.frame (TIME)
D1<- reshape (DF,varying=c(«Score.T0», «Score.T2»), idvar=»pp»,
direction=»long«)
```

```
ggplot(D1, aes(x=time, y=Score, color=Construit,
group=subject))+geom_line(size=1.5, alpha=.5)+facet_
grid(~Construit+Genre)+theme_bw()+stat_summary(aes(group=1),
geom=»point«, fun.y=mean, shape=15, size=4)+scale_color_
manual(values=c(«#d175b8»,»#617a89«))
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

# Kit stat\_R Niveau débutant

