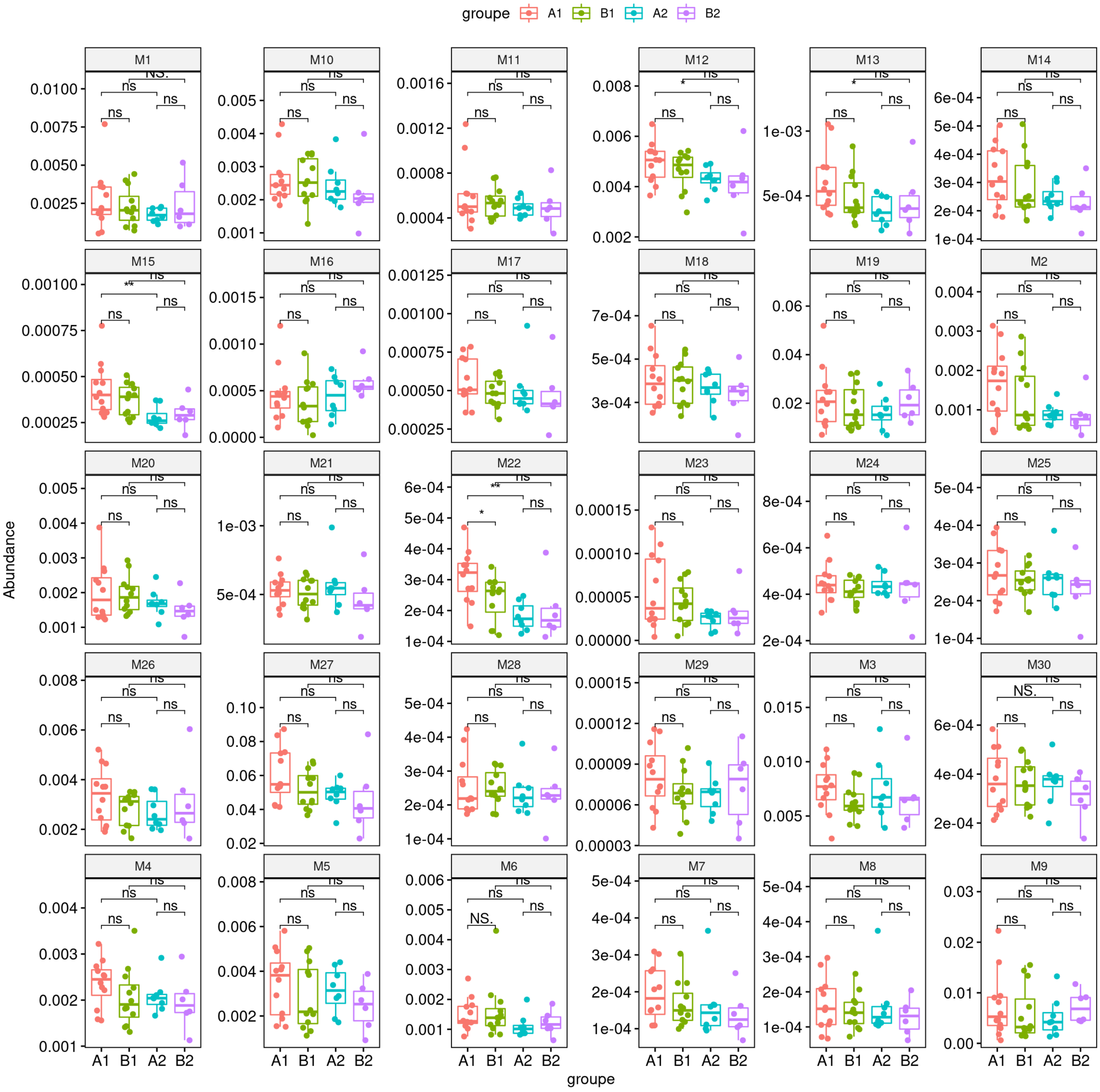
Une planche de graphiques

Spoiler alert



```r
library(openxlsx)
library(ggplot2)
library(tidyr)
library(dplyr)
library(tibble)
```

Importation et présentation des données

Abondance de 30 métabolites pour 38 observations réparties en 4 groupes

```
metabolite <- read.xlsx("metab.xlsx")
metabolite$groupe <- factor(metabolite$groupe,
                                 levels=c("A1","B1","A2","B2"))
summary(metabolite)
```

```
##        M1                  M2                  M3                 M4
##  Min.   :0.0005254   Min.   :0.0003557   Min.   :0.002925   Min.   :0.001129
##  1st Qu.:0.0014447   1st Qu.:0.0006366   1st Qu.:0.005598   1st Qu.:0.001758
##  Median :0.0019777   Median :0.0008990   Median :0.006657   Median :0.002077
##  Mean   :0.0023243   Mean   :0.0012556   Mean   :0.007003   Mean   :0.002140
##  3rd Qu.:0.0029574   3rd Qu.:0.0018142   3rd Qu.:0.008238   3rd Qu.:0.002522
##  Max.   :0.0076899   Max.   :0.0031358   Max.   :0.013001   Max.   :0.003504
##        M5                  M6                  M7
##  Min.   :0.0009152   Min.   :0.0006342   Min.   :0.0000688
##  1st Qu.:0.0018662   1st Qu.:0.0010311   1st Qu.:0.0001098
##  Median :0.0029078   Median :0.0012418   Median :0.0001535
##  Mean   :0.0030473   Mean   :0.0014035   Mean   :0.0001694
##  3rd Qu.:0.0040656   3rd Qu.:0.0016013   3rd Qu.:0.0002111
##  Max.   :0.0058096   Max.   :0.0042977   Max.   :0.0003652
##        M8                  M9                 M10
##  Min.   :6.202e-05   Min.   :0.0006151   Min.   :0.0009801
##  1st Qu.:1.071e-04   1st Qu.:0.0029987   1st Qu.:0.0020337
##  Median :1.406e-04   Median :0.0045993   Median :0.0023198
##  Mean   :1.507e-04   Mean   :0.0064434   Mean   :0.0025062
##  3rd Qu.:1.730e-04   3rd Qu.:0.0090120   3rd Qu.:0.0028410
##  Max.   :3.739e-04   Max.   :0.0222578   Max.   :0.0042863
##       M11                 M12                M13                  M14
##  Min.   :0.0002614   Min.   :0.002135   Min.   :0.0002098   Min.   :0.0001195
##  1st Qu.:0.0004246   1st Qu.:0.004089   1st Qu.:0.0003610   1st Qu.:0.0002138
##  Median :0.0004975   Median :0.004573   Median :0.0004302   Median :0.0002469
##  Mean   :0.0005394   Mean   :0.004587   Mean   :0.0004959   Mean   :0.0002764
##  3rd Qu.:0.0005895   3rd Qu.:0.005141   3rd Qu.:0.0005756   3rd Qu.:0.0003415
##  Max.   :0.0012367   Max.   :0.006486   Max.   :0.0010539   Max.   :0.0005065
##       M15                 M16                 M17
##  Min.   :0.0001815   Min.   :2.291e-05   Min.   :0.0002101
##  1st Qu.:0.0002695   1st Qu.:2.537e-04   1st Qu.:0.0004140
##  Median :0.0003272   Median :4.641e-04   Median :0.0004816
##  Mean   :0.0003599   Mean   :4.526e-04   Mean   :0.0005101
##  3rd Qu.:0.0004250   3rd Qu.:5.702e-04   3rd Qu.:0.0005751
##  Max.   :0.0007758   Max.   :1.197e-03   Max.   :0.0009213
##       M18                 M19                M20                  M21
##  Min.   :0.0001488   Min.   :0.006853   Min.   :0.0007242   Min.   :0.0001917
##  1st Qu.:0.0002971   1st Qu.:0.011803   1st Qu.:0.0013885   1st Qu.:0.0004253
##  Median :0.0003715   Median :0.016610   Median :0.0016649   Median :0.0005286
##  Mean   :0.0003806   Mean   :0.019409   Mean   :0.0018355   Mean   :0.0005226
##  3rd Qu.:0.0004486   3rd Qu.:0.024679   3rd Qu.:0.0022176   3rd Qu.:0.0006001
##  Max.   :0.0006537   Max.   :0.051902   Max.   :0.0038765   Max.   :0.0009875
##       M22                 M23                 M24
##  Min.   :0.0001147   Min.   :4.095e-06   Min.   :0.0002164
##  1st Qu.:0.0001568   1st Qu.:2.040e-05   1st Qu.:0.0003963
##  Median :0.0002418   Median :3.146e-05   Median :0.0004330
##  Mean   :0.0002434   Mean   :4.158e-05   Mean   :0.0004347
##  3rd Qu.:0.0002925   3rd Qu.:5.554e-05   3rd Qu.:0.0004681
##  Max.   :0.0004692   Max.   :1.300e-04   Max.   :0.0006875
##       M25                 M26                M27                M28
##  Min.   :0.0001040   Min.   :0.001629   Min.   :0.02285   Min.   :0.0001008
##  1st Qu.:0.0002203   1st Qu.:0.002172   1st Qu.:0.04220   1st Qu.:0.0002034
##  Median :0.0002558   Median :0.002947   Median :0.05243   Median :0.0002291
##  Mean   :0.0002574   Mean   :0.003003   Mean   :0.05299   Mean   :0.0002460
##  3rd Qu.:0.0002883   3rd Qu.:0.003554   3rd Qu.:0.05962   3rd Qu.:0.0002698
```

```
##  Max.   :0.0003942   Max.   :0.006034   Max.   :0.08730   Max.   :0.0004238  
##       M29                 M30            groupe 
##  Min.   :3.515e-05   Min.   :0.0001366   A1:12  
##  1st Qu.:5.891e-05   1st Qu.:0.0002730   B1:12  
##  Median :7.035e-05   Median :0.0003661   A2: 8  
##  Mean   :7.322e-05   Mean   :0.0003574   B2: 6  
##  3rd Qu.:8.831e-05   3rd Qu.:0.0004212          
##  Max.   :1.158e-04   Max.   :0.0005836          
```

```
head(metabolite)
```

```
##             M1           M2          M3          M4          M5          M6
## 1 0.0019497303 0.0009976818 0.005072411 0.001583864 0.002111279 0.001022931
## 2 0.0021994583 0.0004305712 0.007059411 0.001773148 0.002922657 0.001118970
## 3 0.0019755328 0.0015505552 0.006012389 0.002276433 0.003616820 0.001564363
## 4 0.0006109359 0.0004927399 0.010356908 0.002214662 0.001517629 0.001285894
## 5 0.0015460363 0.0019539049 0.011131809 0.002525373 0.001881233 0.001228354
## 6 0.0018525065 0.0011651985 0.006663383 0.002733360 0.005809555 0.001249846
##             M7           M8          M9         M10          M11         M12
## 1 0.0001088416 6.670640e-05 0.001797589 0.002438347 0.0003045516 0.003647887
## 2 0.0001090255 7.117895e-05 0.004543185 0.002179489 0.0004631070 0.004411536
## 3 0.0003023043 1.056315e-04 0.002996306 0.002126223 0.0003786612 0.005402707
## 4 0.0001507176 2.070830e-04 0.016050478 0.002033345 0.0006182604 0.004830741
## 5 0.0002611004 1.551742e-04 0.022257759 0.002332064 0.0006151416 0.006485944
## 6 0.0002125418 1.479881e-04 0.003708689 0.002743453 0.0004567703 0.005051771
##            M13          M14          M15          M16          M17          M18
## 1 0.0003550040 0.0002140638 0.0003044825 0.0003427589 0.0003572941 0.0002825989
## 2 0.0005174194 0.0001826632 0.0002804458 0.0002286762 0.0004739388 0.0002522240
## 3 0.0004331546 0.0002477032 0.0003259931 0.0001055723 0.0004826770 0.0003256138
## 4 0.0003652183 0.0002564543 0.0007757909 0.0004552280 0.0005090956 0.0002841940
## 5 0.0006020554 0.0002922910 0.0005687291 0.0011965096 0.0007846011 0.0005147707
## 6 0.0004601564 0.0003145901 0.0003627486 0.0005267340 0.0007029728 0.0006537269
##          M19         M20          M21          M22          M23          M24
## 1 0.01069387 0.001262816 0.0003511469 0.0001487752 2.530329e-05 0.0003748787
## 2 0.01270249 0.001370909 0.0004284019 0.0002725926 2.384110e-05 0.0004172943
## 3 0.01177159 0.001452808 0.0005482662 0.0002346421 1.775549e-05 0.0004433637
## 4 0.03503002 0.001225808 0.0005115621 0.0002712070 3.131458e-05 0.0004172762
## 5 0.05190167 0.002617515 0.0005729502 0.0003475241 4.095478e-06 0.0006517901
## 6 0.01661872 0.002282455 0.0007620550 0.0003296081 2.477516e-05 0.0005478172
##            M25         M26        M27          M28          M29          M30
## 1 0.0001721437 0.002133350 0.04148790 0.0001733488 4.308620e-05 0.0002326818
## 2 0.0001924611 0.002460116 0.05349343 0.0001876042 7.318912e-05 0.0002124977
## 3 0.0002629841 0.002034839 0.05460285 0.0001825373 6.950998e-05 0.0002901140
## 4 0.0002217138 0.003712321 0.04205769 0.0002708714 8.408064e-05 0.0002720904
## 5 0.0003787150 0.004648184 0.06862181 0.0003919751 1.141668e-04 0.0005107015
## 6 0.0003942078 0.002749467 0.05497483 0.0004238065 7.366480e-05 0.0005836174
##   groupe
## 1     A1
## 2     A1
## 3     A1
## 4     A1
## 5     A1
## 6     A1
```

```
as_tibble(metabolite)
```

```
## # A tibble: 38 × 31
##          M1       M2      M3      M4      M5       M6       M7        M8      M9
##       <dbl>    <dbl>   <dbl>   <dbl>   <dbl>    <dbl>    <dbl>     <dbl>   <dbl>
##  1 0.00195  0.000998 0.00507 0.00158 0.00211 0.00102  0.000109 0.0000667 0.00180
##  2 0.00220  0.000431 0.00706 0.00177 0.00292 0.00112  0.000109 0.0000712 0.00454
##  3 0.00198  0.00155  0.00601 0.00228 0.00362 0.00156  0.000302 0.000106  0.00300
##  4 0.000611 0.000493 0.0104  0.00221 0.00152 0.00129  0.000151 0.000207  0.0161
##  5 0.00155  0.00195  0.0111  0.00253 0.00188 0.00123  0.000261 0.000155  0.0223
##  6 0.00185  0.00117  0.00666 0.00273 0.00581 0.00125  0.000213 0.000148  0.00371
##  7 0.000525 0.000881 0.00849 0.00156 0.00155 0.000760 0.000109 0.000116  0.00938
##  8 0.00769  0.00292  0.00831 0.00322 0.00402 0.00176  0.000309 0.000277  0.00905
##  9 0.00361  0.00314  0.00970 0.00263 0.00507 0.00208  0.000207 0.000297  0.00816
## 10 0.00305  0.00232  0.00760 0.00254 0.00490 0.00270  0.000256 0.000215  0.00466
## # … with 28 more rows, and 22 more variables: M10 <dbl>, M11 <dbl>, M12 <dbl>,
## #   M13 <dbl>, M14 <dbl>, M15 <dbl>, M16 <dbl>, M17 <dbl>, M18 <dbl>,
## #   M19 <dbl>, M20 <dbl>, M21 <dbl>, M22 <dbl>, M23 <dbl>, M24 <dbl>,
## #   M25 <dbl>, M26 <dbl>, M27 <dbl>, M28 <dbl>, M29 <dbl>, M30 <dbl>,
## #   groupe <fct>
```

```
table(metabolite$groupe)
```

```
## 
## A1 B1 A2 B2 
## 12 12  8  6
```

Représentation souhaitée (pour un seul métabolite)

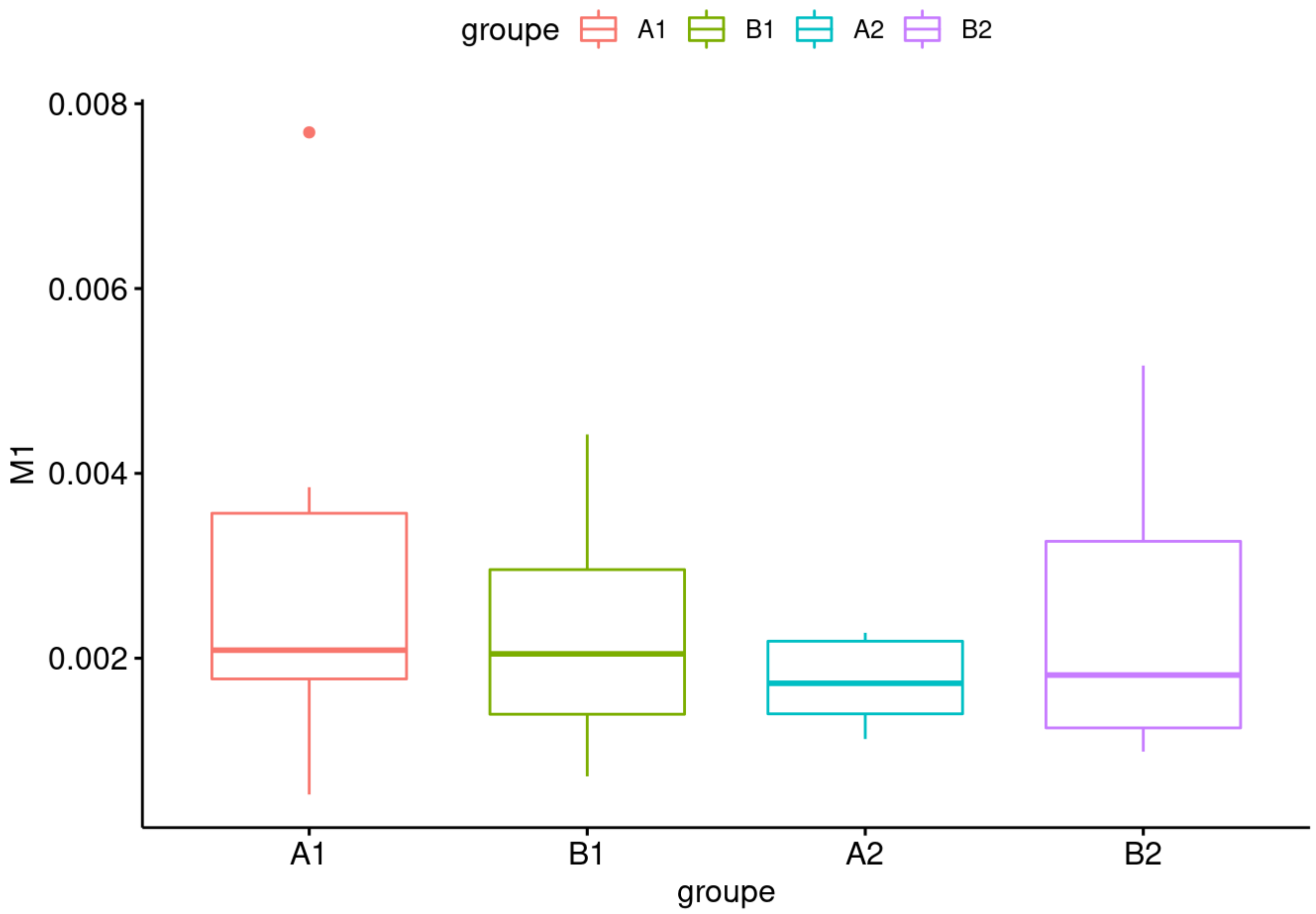
Représentation de boxplots (avec les points) par groupe avec p-value (ou symbole) d’un test de comparaison entre groupes.

Utilisons le package `ggpubr` dont le “slogan” est **‘ggplot2’ Based Publication Ready Plots**.

```
library(ggpubr)
```

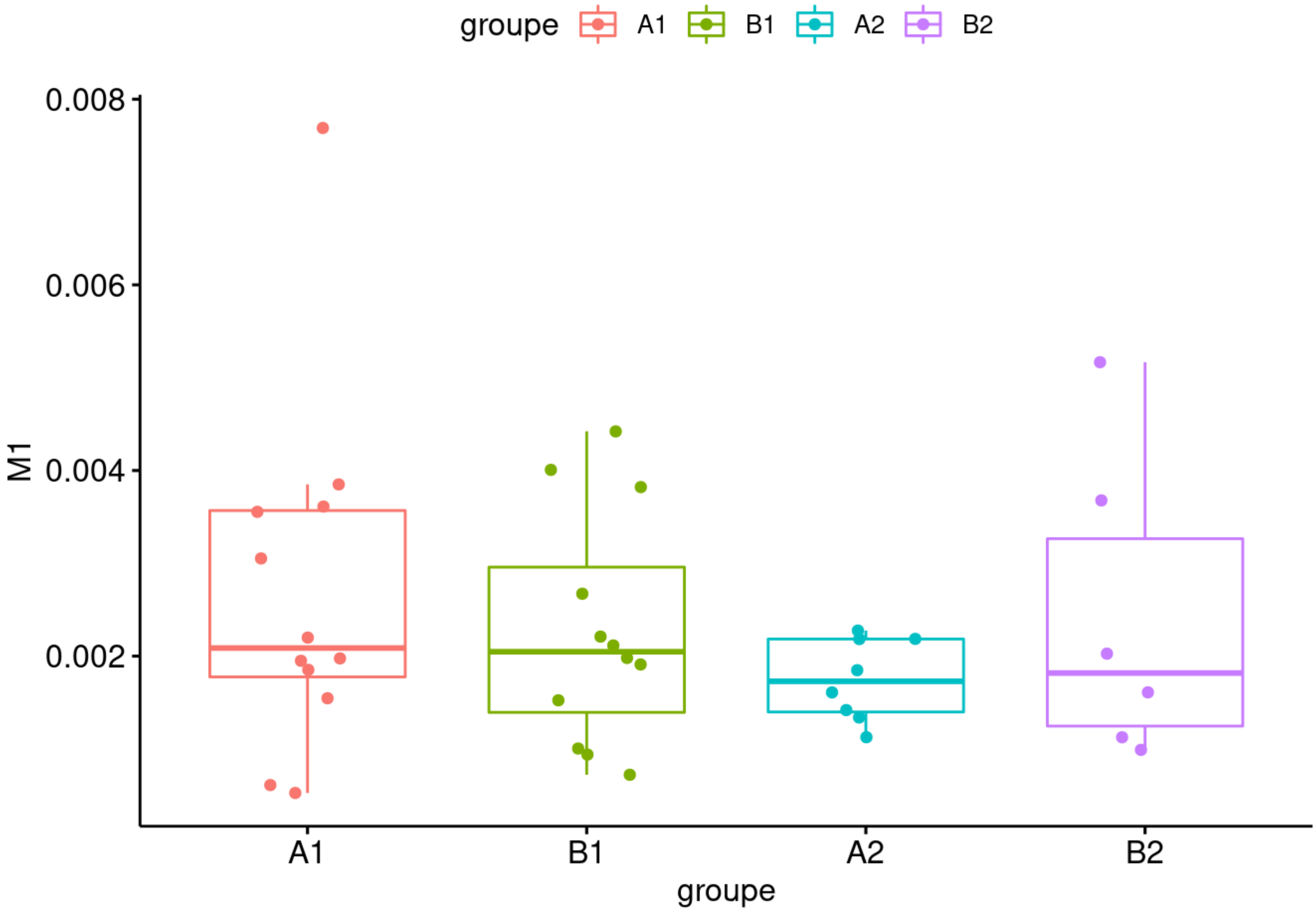
Juste les boxplots

```
Boxplot_ggpubr <- ggboxplot(data = metabolite,
                            x = "groupe", y = "M1",
                            color = "groupe")
Boxplot_ggpubr
```



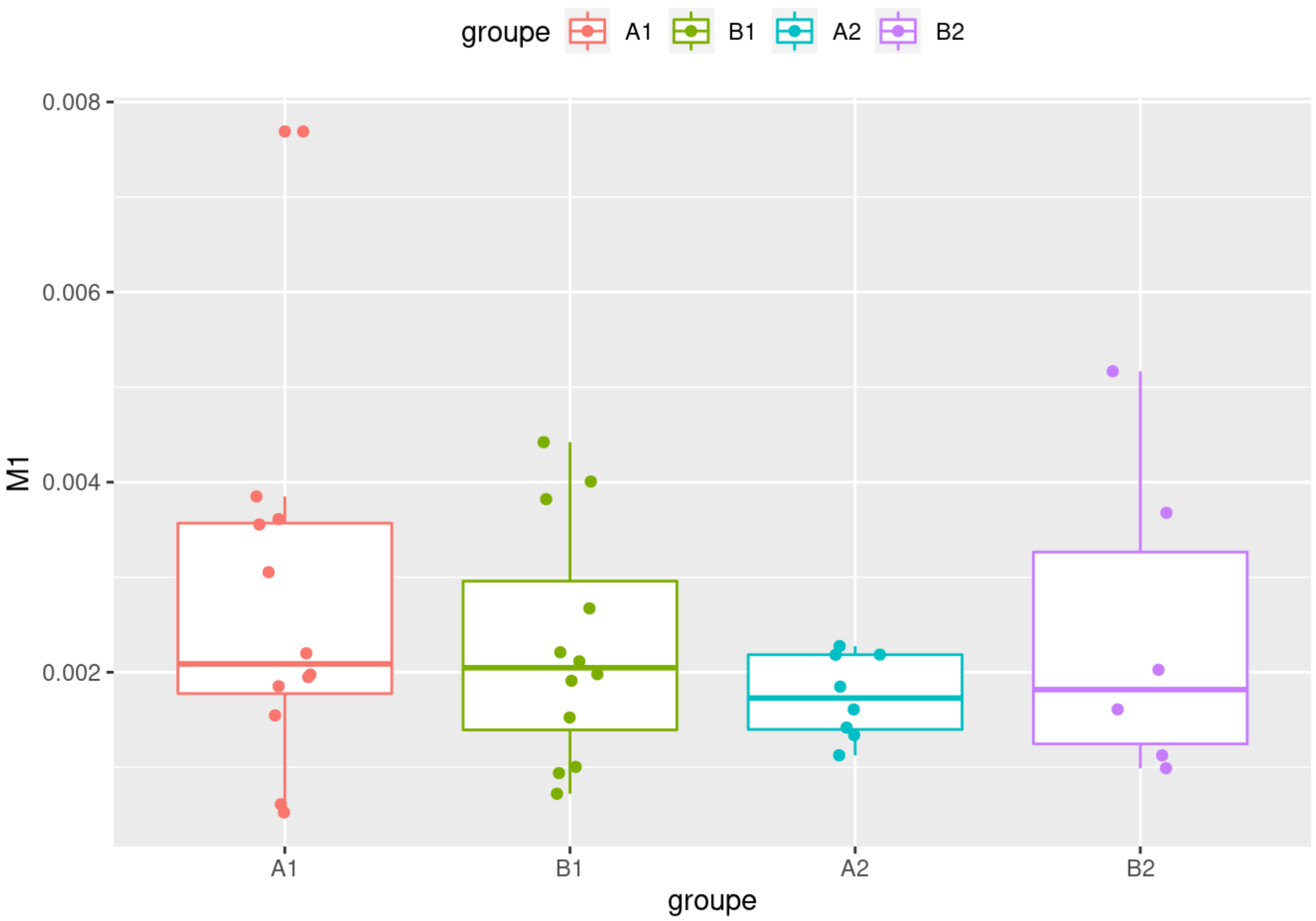
Avec les points

```
Boxplot_ggpubr <- ggboxplot(data = metabolite,
                            x = "groupe", y = "M1",
                            color = "groupe",
                            add="jitter")
Boxplot_ggpubr
```



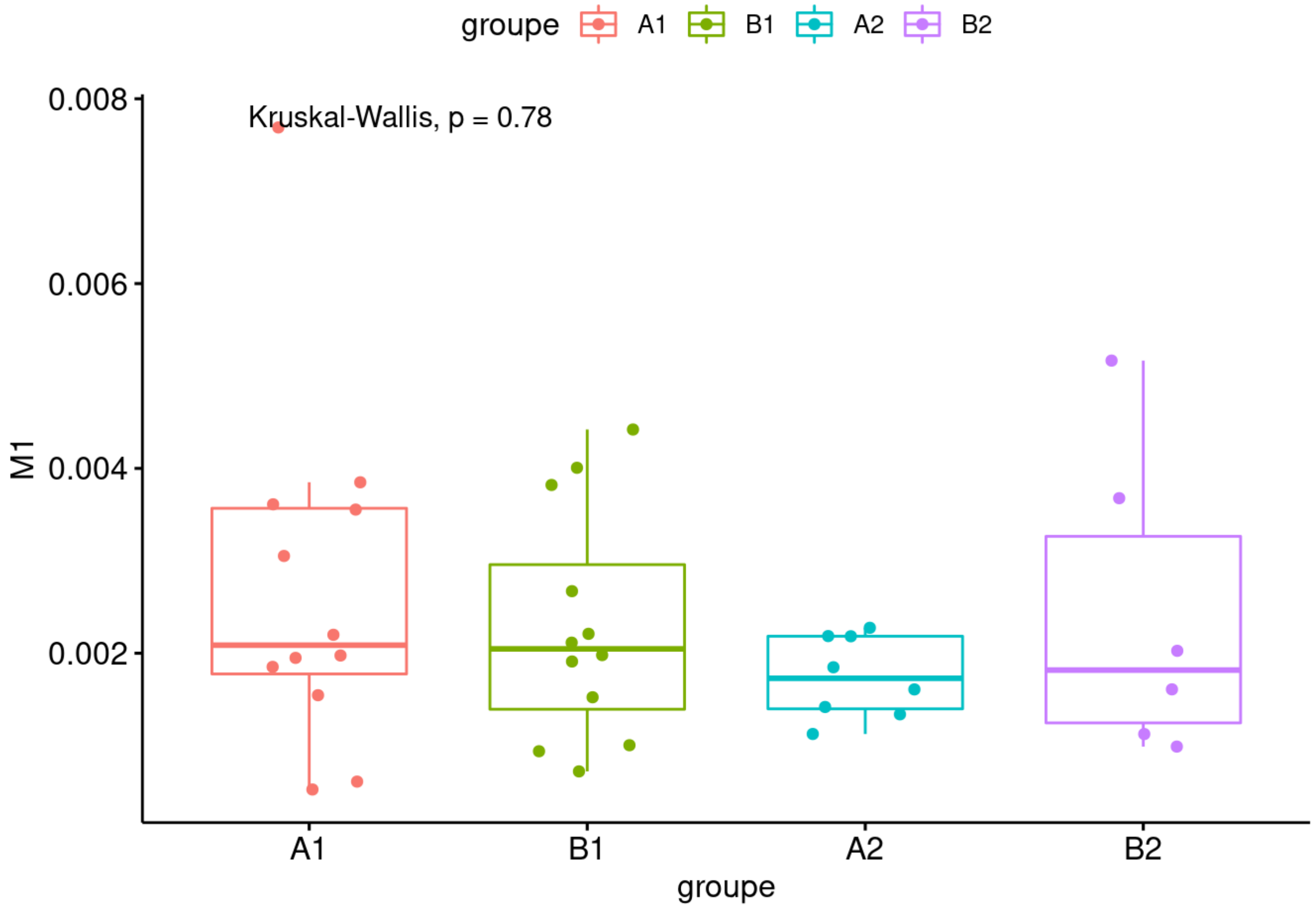
Jusqu'ici on pouvait faire la même chose directement avec `ggplot2` à un thème prêt.

```
ggplot(metabolite, aes(x = groupe, y = M1, color = groupe)) + geom_boxplot() + geom_j
itter(width=0.1) +
  theme(legend.position = "top")
```



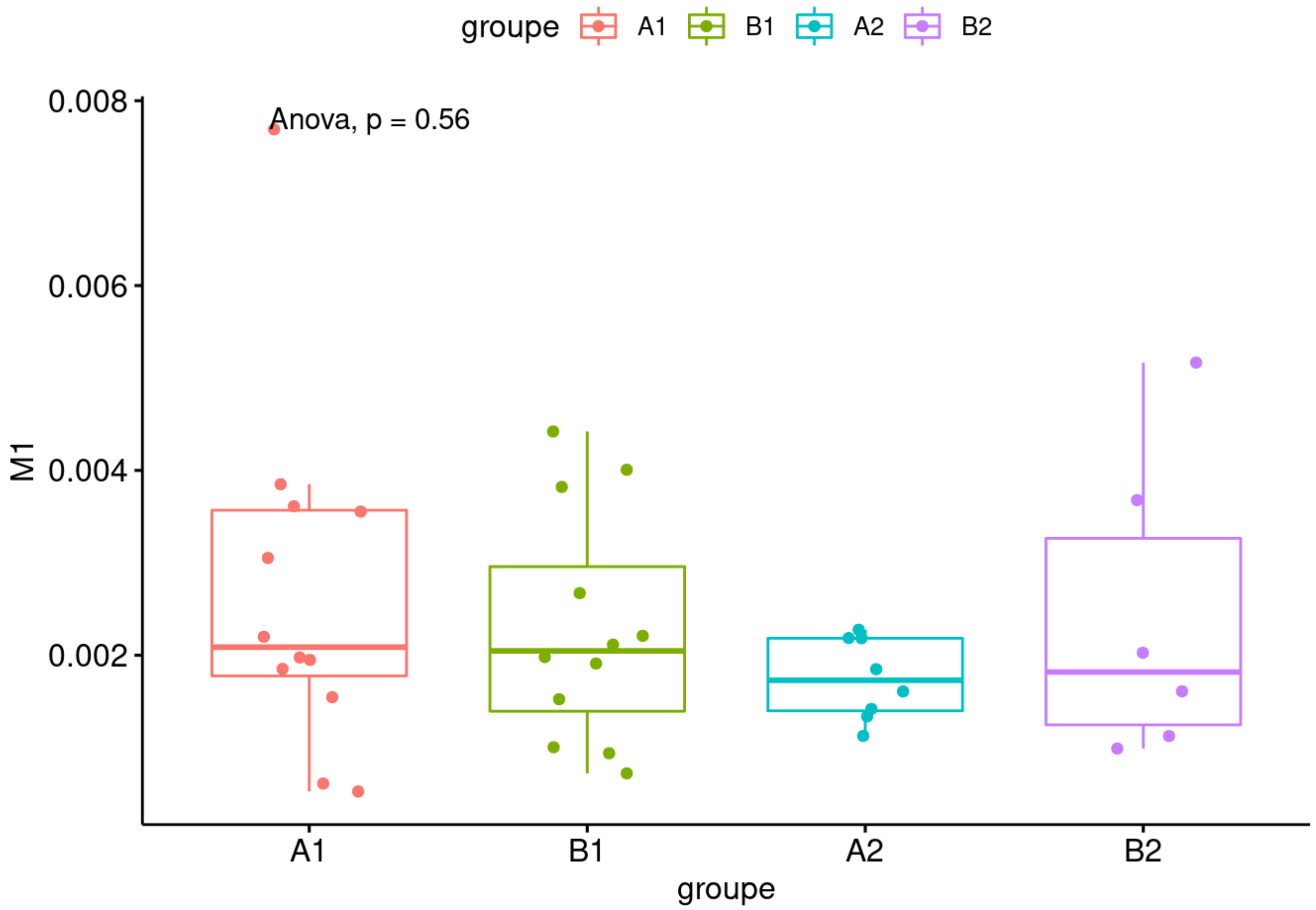
Mais ce qui est souhaité est (presque) ceci avec un test de Kruskal-Wallis de comparaison des 4 groupes :

```
Boxplot_ggpubr + stat_compare_means()
```



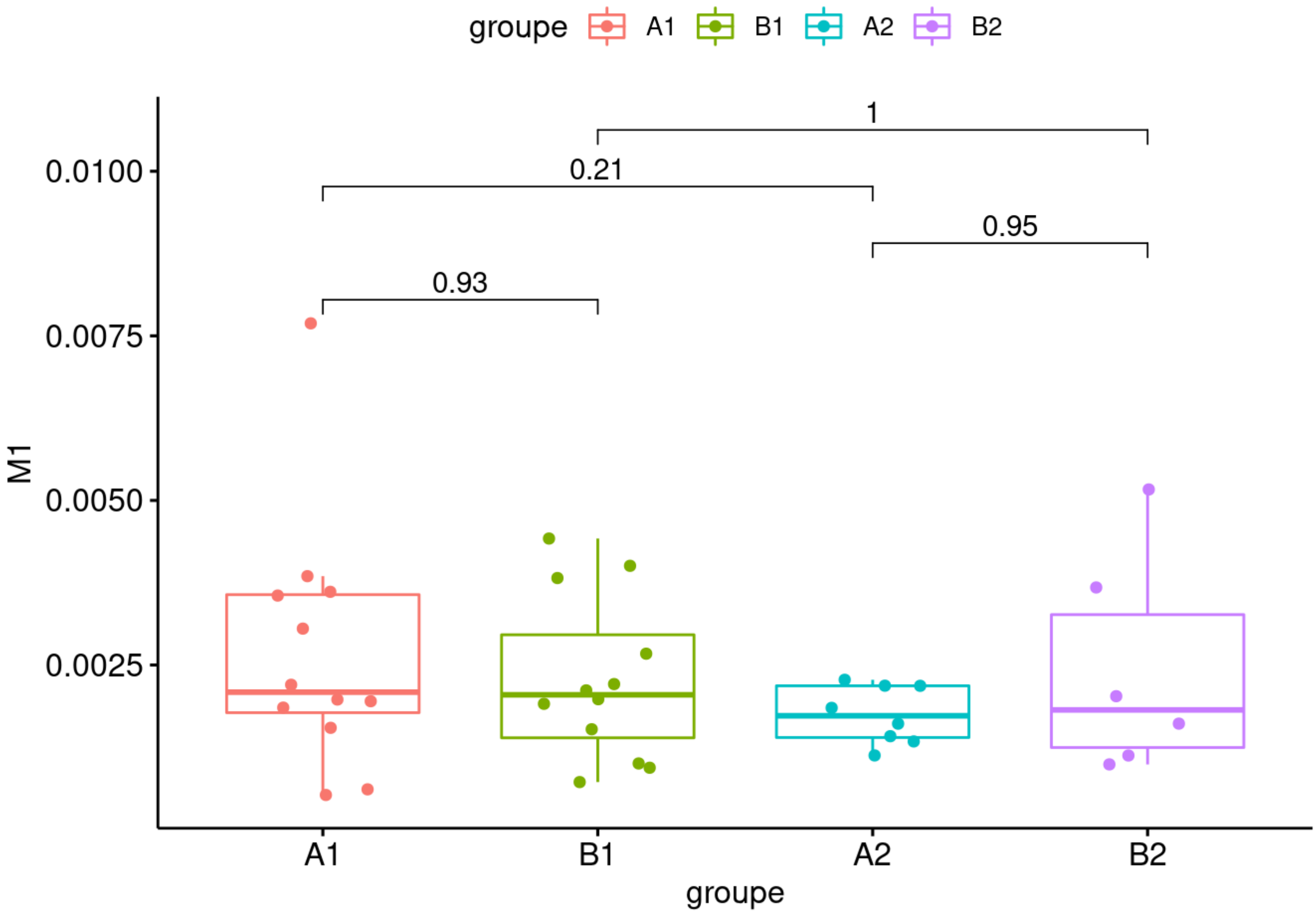
Pour une ANOVA :

```
Boxplot_ggpubr + stat_compare_means(method = "anova")
```



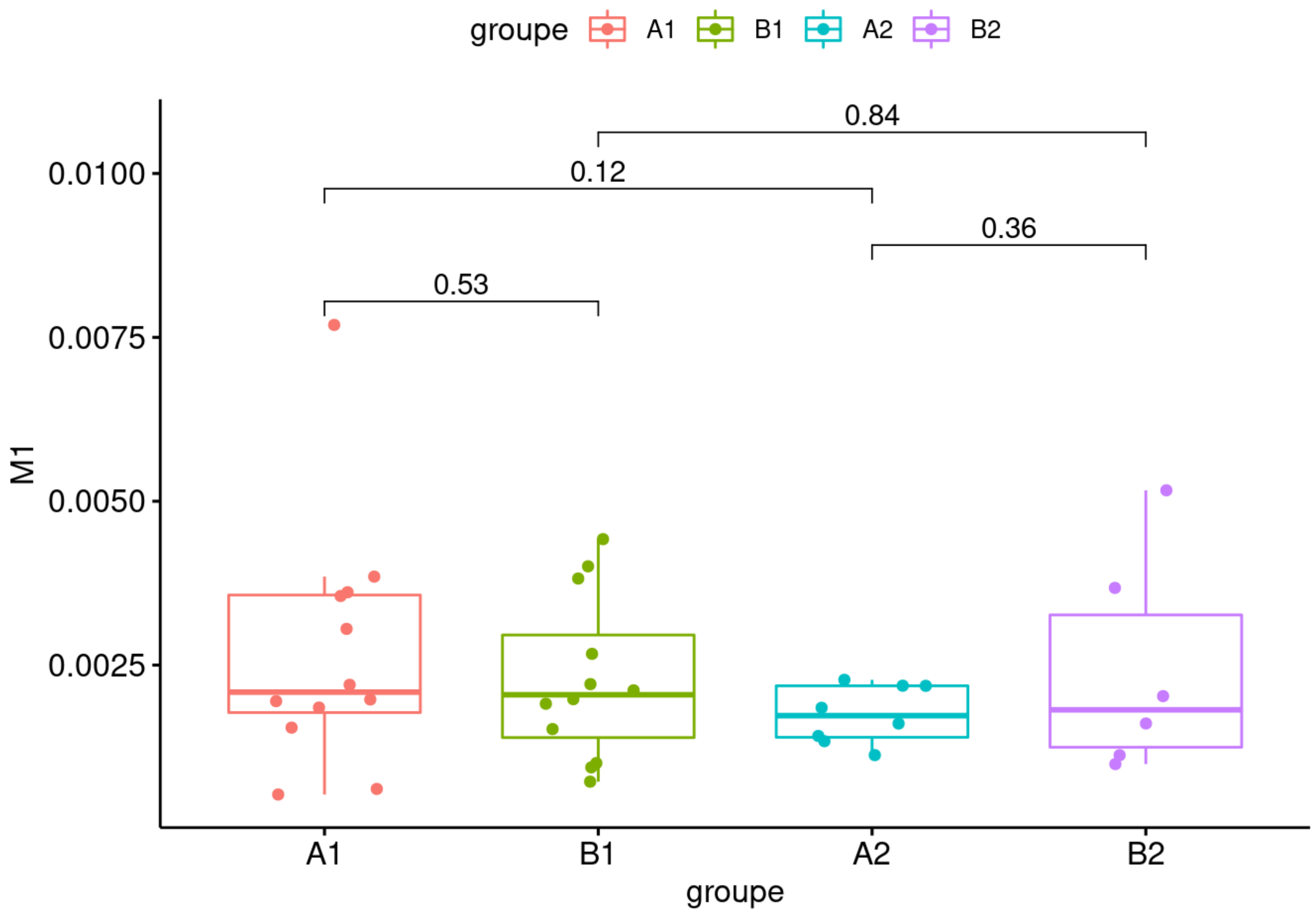
et finalement ceci avec les comparaisons deux-à-deux effectuée par un test de Wilcoxon :

```
MesComparaisons <- list( c("A1", "B1"),
                         c("A2", "B2"),
                         c("A1", "A2"),
                         c("B1", "B2"))
Boxplot_ggpubr + stat_compare_means(comparisons = MesComparaisons)
```



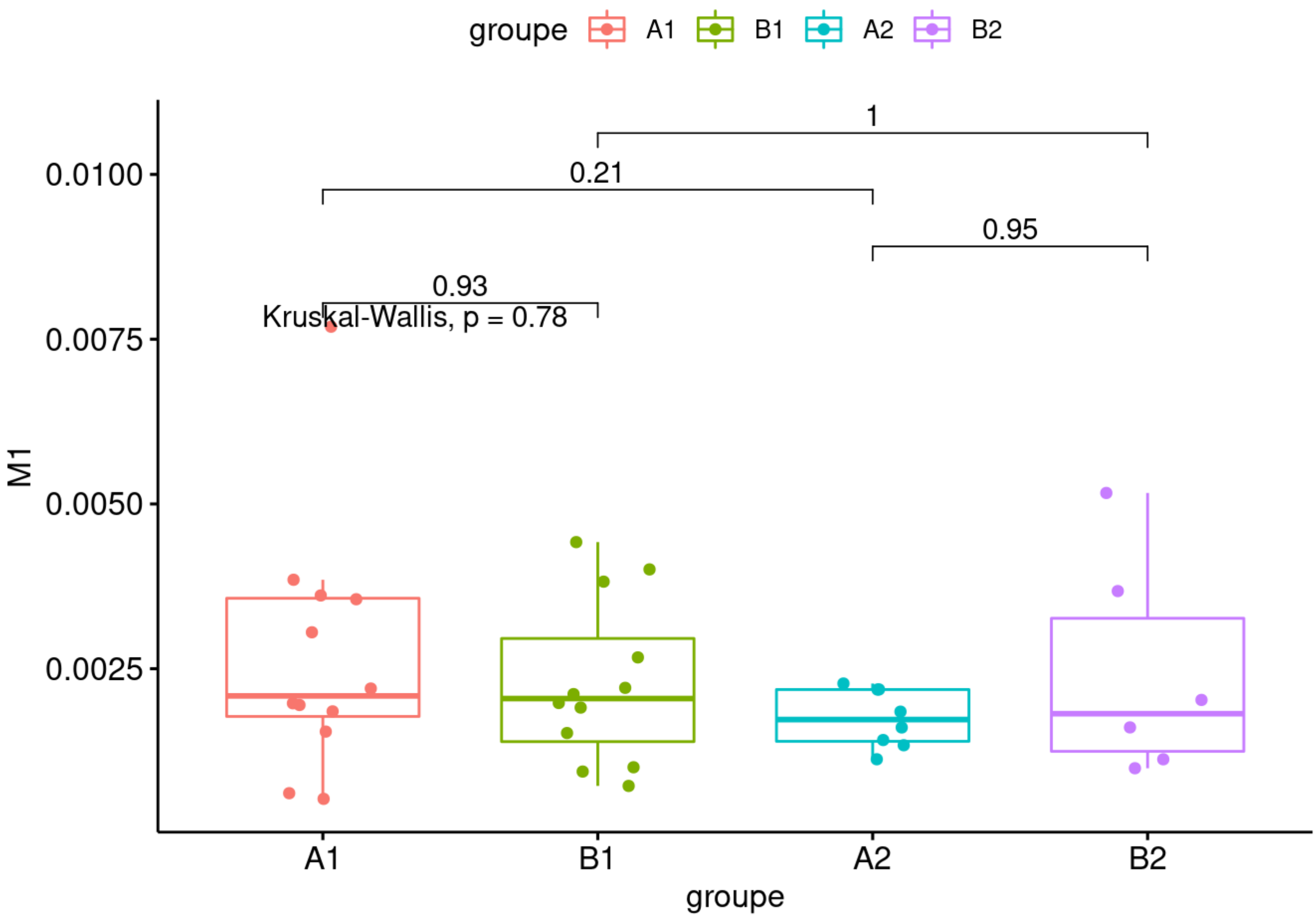
Pour réaliser des tests de Student :

```
Boxplot_ggpubr + stat_compare_means(comparisons = MesComparaisons,
                                    method = "t.test")
```



Pour afficher à la fois test global et comparaison deux-à-deux (avec un petit défaut dans l'affichage) :

```
Boxplot_ggpubr +
  stat_compare_means(comparisons = MesComparaisons) +
  stat_compare_means()
```



Représenter tous les métabolites

Nécessite de modifier la structure du jeu de données qui était pourtant *tidy*.

> **La *tidyness* ultime reviendrait-elle à rassembler toutes les données numériques en une seule colonne ?**

`pivot_longer`

```
metabolite_longer <- tidyr::pivot_longer(metabolite, !groupe,
                                names_to="Metab", values_to = "Abundance")
dim(metabolite_longer)
```

```
## [1] 1140    3
```

```
metabolite_longer
```

```
## # A tibble: 1,140 × 3
##    groupe Metab Abundance
##    <fct>  <chr>     <dbl>
##  1 A1     M1    0.00195
##  2 A1     M2    0.000998
##  3 A1     M3    0.00507
##  4 A1     M4    0.00158
##  5 A1     M5    0.00211
##  6 A1     M6    0.00102
##  7 A1     M7    0.000109
##  8 A1     M8    0.0000667
##  9 A1     M9    0.00180
## 10 A1     M10   0.00244
## # … with 1,130 more rows
```

```
summary(metabolite_longer)
```

```
##  groupe      Metab             Abundance
##  A1:360   Length:1140        Min.   :0.0000041
##  B1:360   Class :character   1st Qu.:0.0002711
##  A2:240   Mode  :character   Median :0.0005297
##  B2:180                      Mean   :0.0037821
##                              3rd Qu.:0.0023096
##                              Max.   :0.0872987
```

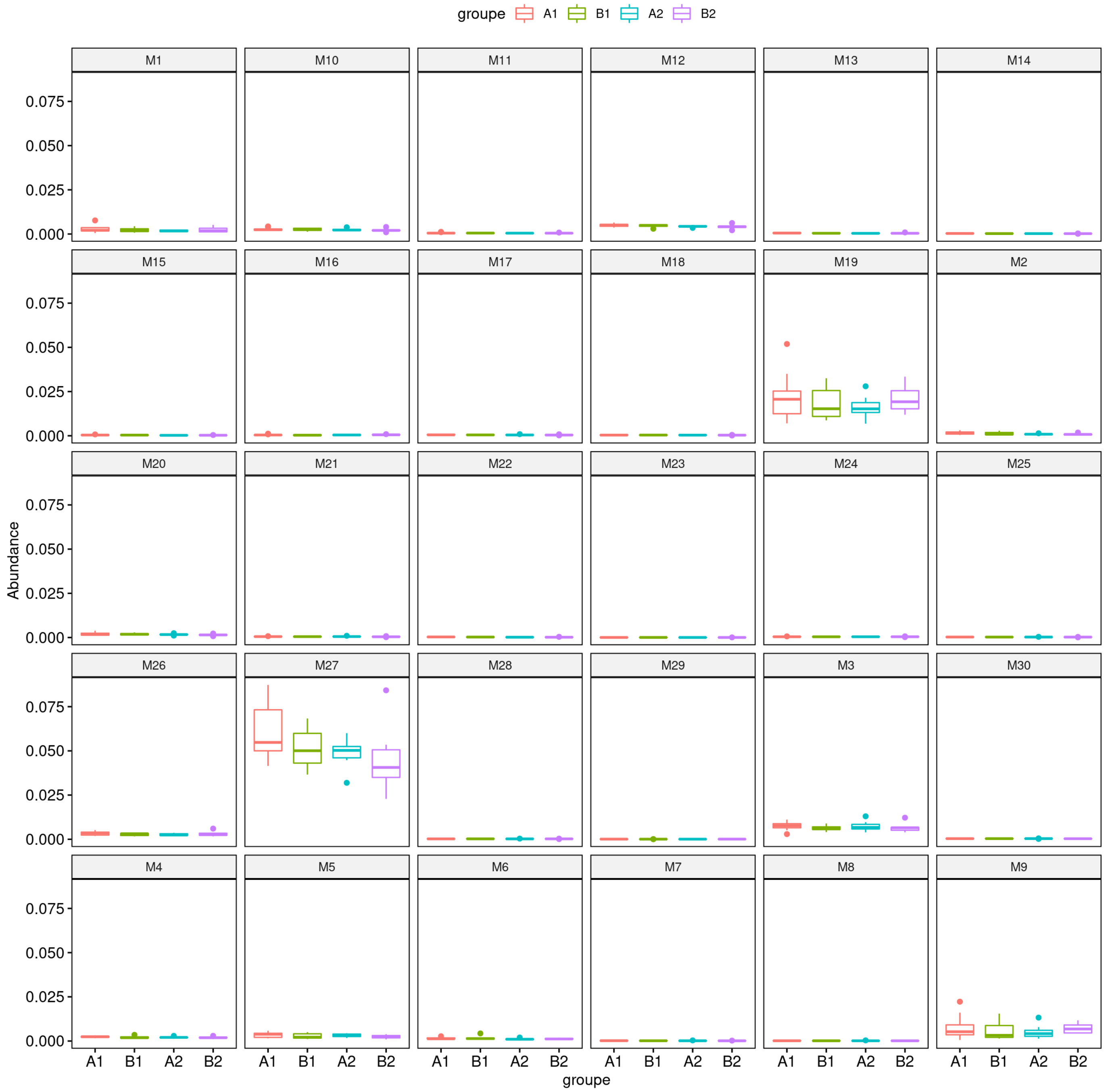
```
table(metabolite_longer$Metab)
```

```
## 
##  M1 M10 M11 M12 M13 M14 M15 M16 M17 M18 M19  M2 M20 M21 M22 M23 M24 M25 M26 M27 
##  38  38  38  38  38  38  38  38  38  38  38  38  38  38  38  38  38  38  38  38 
## M28 M29  M3 M30  M4  M5  M6  M7  M8  M9 
##  38  38  38  38  38  38  38  38  38  38 
```

Graphique de base

Avec `ggpubr`

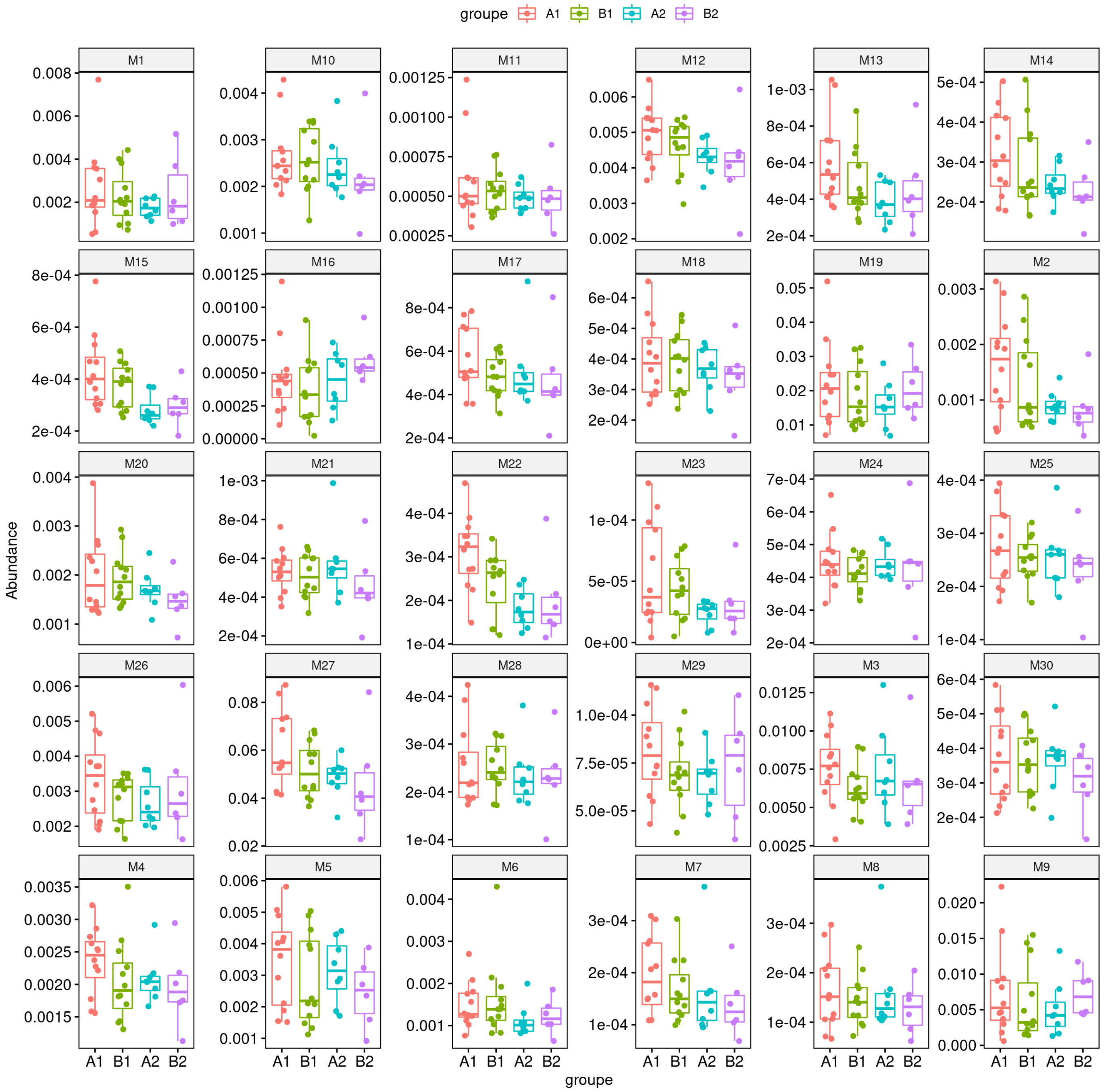
```
ggboxplot(data = metabolite_longer,
          x = "groupe", y = "Abundance", color = "groupe",
          facet.by = "Metab")
```



Pour libérer les échelles verticales et ajouter les points en plus des boxplots :

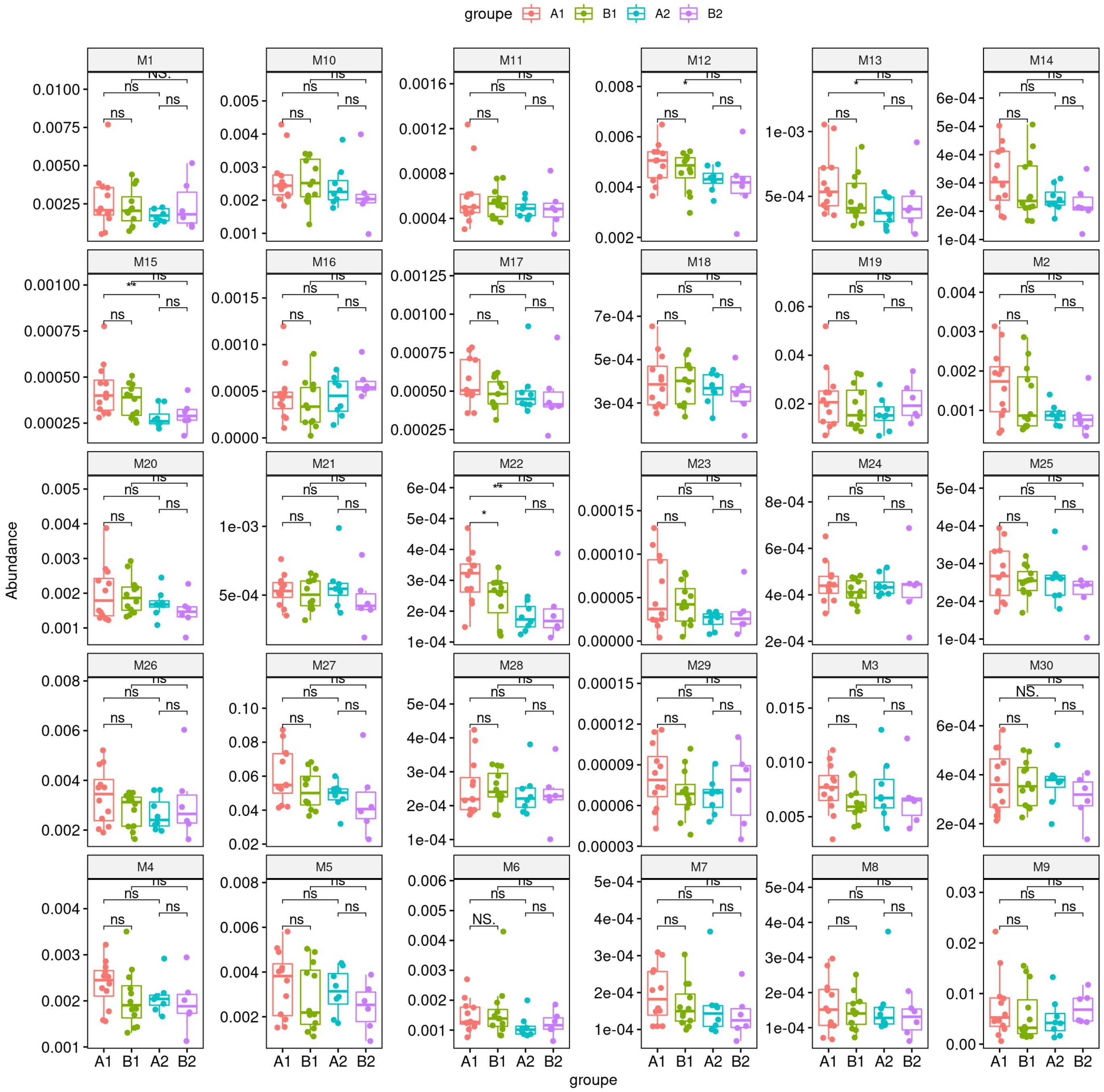
```
Boxplot_ggpubr_tous <- ggboxplot(data = metabolite_longer,
                                 x = "groupe", y = "Abundance",
                                 color = "groupe",
                                 facet.by = "Metab",
                                 scales = "free_y",
                                 add = "jitter")

Boxplot_ggpubr_tous
```



Pour ajouter un symbole de significativité :

```
Boxplot_ggpubr_tous + stat_compare_means(comparisons = MesComparaisons,
                                         label = "p.signif")
```



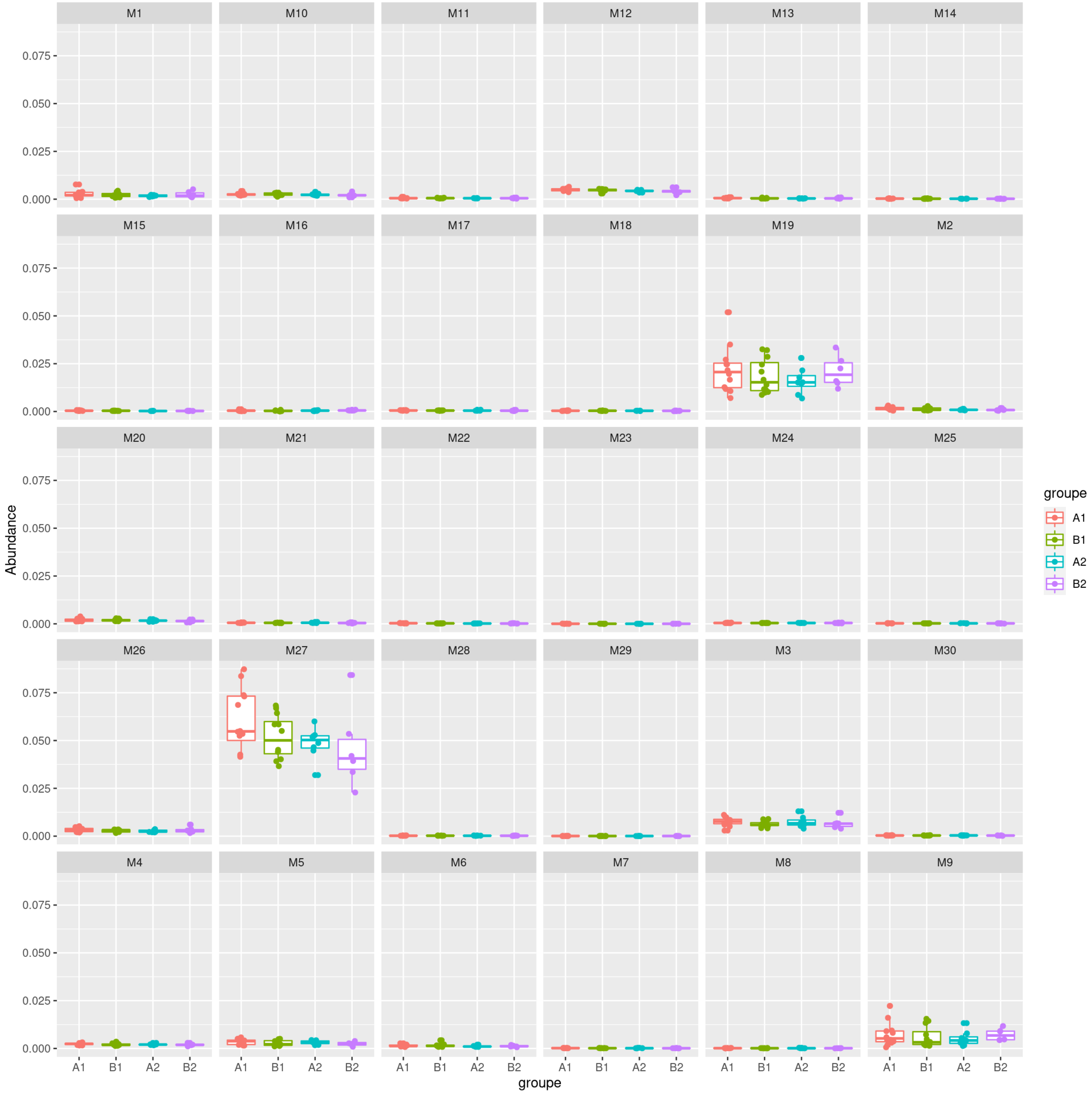
Avec `ggplot2`

Boxplot + points avec `groupe` en abscisse et en couleur et `Abundance` en ordonnée.

```
Boxplot_gg2 <- ggplot(data = metabolite_longer,
                  aes(x = groupe, y = Abundance, color = groupe)) +
  geom_boxplot() + geom_jitter(width=0.1)
```

`facet_wrap`

```
Boxplot_gg2 + facet_wrap(~Metab)
```



Comme précedemment, on va là aussi libérer les échelles verticales.

`facet_wrap avec scales = "free_y"`

Pour relacher cette contrainte :

```
Boxplot_gg2 + facet_wrap(~Metab, scales = "free_y")
```

On va voir maintenant comment ajouter les p-values ou les symboles de significativité des tests et des segments ou autres pour indiquer les paires testées.

Importation des p-values

Les p-values sont dans un fichier à part.

```
Pvalues <-  read.xlsx("Pvalues.xlsx", sheet=1)
Pvalues
```

```
##   group1 group2      M1       M2       M3      M4      M5       M6      M7
## 1     B1     A1 0.93099 0.370840 0.099877 0.12602 0.26024 1.000000 0.37084
## 2     B2     A2 0.94972 0.490840 0.572760 0.66200 0.22844 0.662000 0.66200
## 3     A1     A2 0.20302 0.058729 0.562830 0.20302 0.61603 0.082579 0.37495
## 4     B1     B2 1.00000 0.437080 0.891620 0.89162 0.75027 0.437080 0.43708
##        M8      M9     M10     M11      M12      M13     M14       M15     M16
## 1 0.58336 0.37084 0.97697 0.97697 0.402500 0.140960 0.31232 0.2854800 0.54437
## 2 0.75458 0.14186 0.49084 0.85181 0.754580 0.754580 0.41359 0.5727600 0.49084
## 3 0.84705 0.37495 0.46359 0.56283 0.049141 0.027891 0.15349 0.0038135 0.90787
## 4 0.68197 0.29078 0.29078 0.43708 0.249620 0.891620 0.29078 0.1505100 0.10246
##       M17     M18     M19     M20     M21       M22     M23     M24     M25
## 1 0.31232 0.93099 0.43573 0.79501 0.83986 0.0463870 0.70745 0.28548 0.58336
## 2 0.57276 0.57276 0.34499 0.18115 0.22844 0.9497200 0.75458 0.85181 0.41359
## 3 0.33485 0.78714 0.29762 0.72845 0.90787 0.0038135 0.20302 0.84705 0.61603
## 4 0.43708 0.43708 0.38451 0.15051 0.38451 0.2907800 0.38451 0.82008 0.43708
##       M26      M27     M28     M29     M30
## 1 0.28548 0.236580 0.70745 0.19393 0.83986
## 2 0.85181 0.490840 0.85181 0.66200 0.34499
## 3 0.15349 0.097164 0.67132 0.17697 1.00000
## 4 0.96359 0.249620 0.49364 0.68197 0.38451
```

On va préparer un nouveau data.frame `df_annotation` sur la base de ces p-values.

Créer les symbôles de significativité

```
symbol <- apply(Pvalues[,-c(1,2)], 1,
                  function(x){
                    symnum(x,
                           cutpoints = c(0,.001,.01,.05, .1, 1),
                           symbols = c("****","***","**","*","."))})
df_annotation <- as.data.frame(symbol)
colnames(df_annotation) <- c("B1_A1","B2_A2","A1_A2","B1_B2")
df_annotation <- rownames_to_column(df_annotation,"Metab")
df_annotation
```

```
##    Metab B1_A1 B2_A2 A1_A2 B1_B2
## 1      M1     .     .     .     .
## 2      M2     .     .     *     .
## 3      M3     *     .     .     .
## 4      M4     .     .     .     .
## 5      M5     .     .     .     .
## 6      M6     .     .     *     .
## 7      M7     .     .     .     .
## 8      M8     .     .     .     .
## 9      M9     .     .     .     .
## 10   M10     .     .     .     .
## 11   M11     .     .     .     .
## 12   M12     .     .    **     .
## 13   M13     .     .    **     .
## 14   M14     .     .     .     .
## 15   M15     .     .   ***     .
## 16   M16     .     .     .     .
## 17   M17     .     .     .     .
## 18   M18     .     .     .     .
## 19   M19     .     .     .     .
## 20   M20     .     .     .     .
## 21   M21     .     .     .     .
## 22   M22    **     .   ***     .
## 23   M23     .     .     .     .
## 24   M24     .     .     .     .
## 25   M25     .     .     .     .
## 26   M26     .     .     .     .
## 27   M27     .     .     *     .
## 28   M28     .     .     .     .
## 29   M29     .     .     .     .
## 30   M30     .     .     .     .
```

Où placer les p-values ou les symbôles ?

> **Attention bricolage à venir !**

Pour placer les p-values au dessus des boxplots (et des points), il faut faire de la place ! Commençons par repérer le mininum et le maximum de chaque métabolite

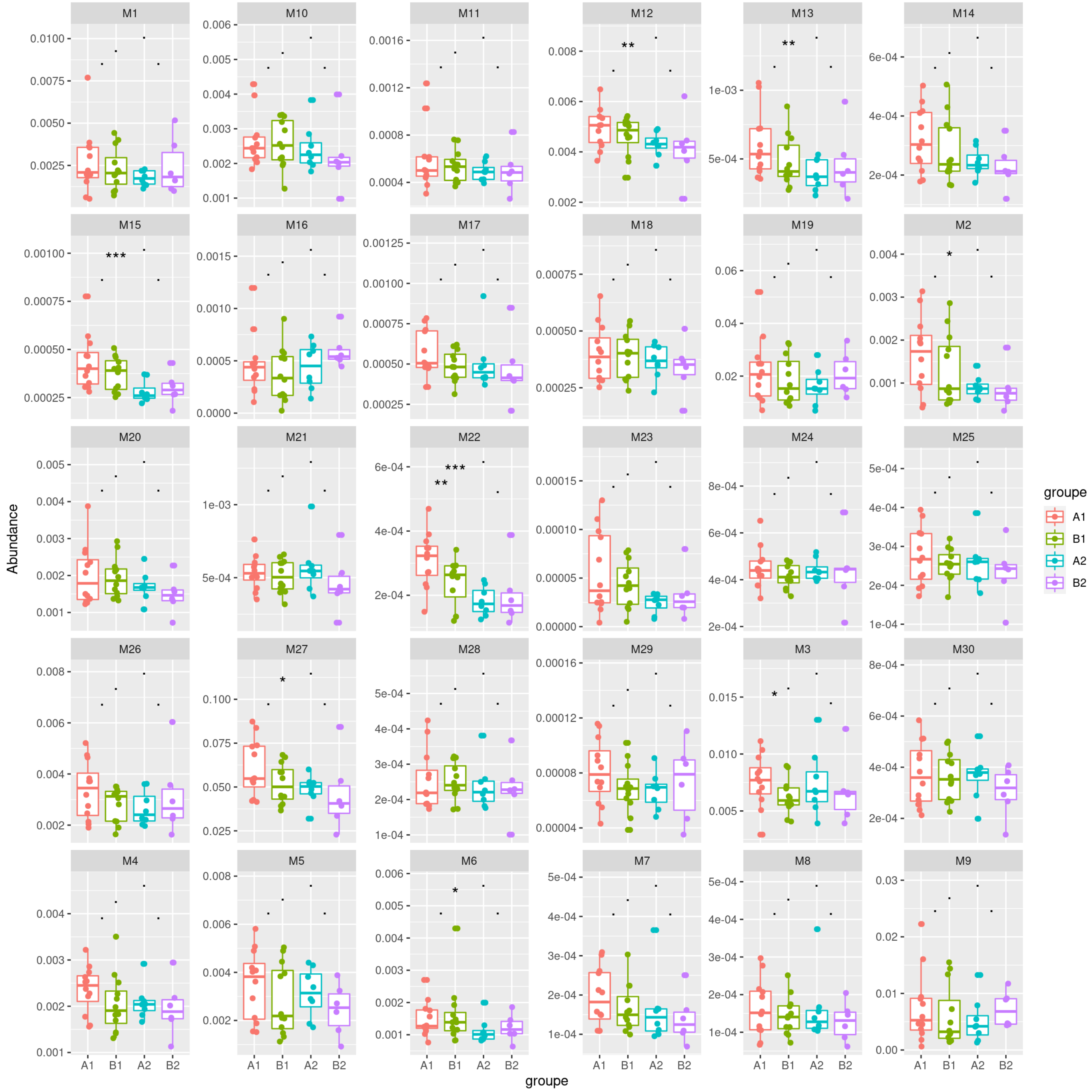
```
df_annotation$min <- apply(metabolite[,1:30], 2, min)
df_annotation$max <- apply(metabolite[,1:30], 2, max)
df_annotation
```

```
##    Metab B1_A1 B2_A2 A1_A2 B1_B2          min          max
## 1     M1     .     .     .     . 5.253883e-04 0.0076898600
## 2     M2     .     .     *     . 3.556876e-04 0.0031358399
## 3     M3     *     .     .     . 2.925137e-03 0.0130006779
## 4     M4     .     .     .     . 1.128671e-03 0.0035038216
## 5     M5     .     .     .     . 9.152375e-04 0.0058095546
## 6     M6     .     .     *     . 6.342488e-04 0.0042976851
## 7     M7     .     .     .     . 6.880382e-05 0.0003651556
## 8     M8     .     .     .     . 6.201652e-05 0.0003739061
## 9     M9     .     .     .     . 6.150863e-04 0.0222577590
## 10   M10     .     .     .     . 9.801484e-04 0.0042863095
## 11   M11     .     .     .     . 2.613533e-04 0.0012366796
## 12   M12     .     .    **     . 2.135441e-03 0.0064859444
## 13   M13     .     .    **     . 2.097845e-04 0.0010539301
## 14   M14     .     .     .     . 1.195333e-04 0.0005065246
## 15   M15     .     .   ***     . 1.814674e-04 0.0007757909
## 16   M16     .     .     .     . 2.290687e-05 0.0011965096
## 17   M17     .     .     .     . 2.101212e-04 0.0009212711
## 18   M18     .     .     .     . 1.488194e-04 0.0006537269
## 19   M19     .     .     .     . 6.853101e-03 0.0519016723
## 20   M20     .     .     .     . 7.241679e-04 0.0038764847
## 21   M21     .     .     .     . 1.916745e-04 0.0009875093
## 22   M22    **     .   ***     . 1.147069e-04 0.0004692441
## 23   M23     .     .     .     . 4.095478e-06 0.0001300232
## 24   M24     .     .     .     . 2.163686e-04 0.0006875205
## 25   M25     .     .     .     . 1.040086e-04 0.0003942078
## 26   M26     .     .     .     . 1.628724e-03 0.0060340080
## 27   M27     .     .     *     . 2.284627e-02 0.0872986452
## 28   M28     .     .     .     . 1.008304e-04 0.0004238065
## 29   M29     .     .     .     . 3.514620e-05 0.0001158070
## 30   M30     .     .     .     . 1.365880e-04 0.0005836174
```

Et calculons les coordonnées des positions des p-values en prenant en ordonnée une valeur légèrement supérieure au maximum de chaque métabolite.

```
df_annotation$x_pval_1 <- 1.5
df_annotation$y_pval_1 <- 1.15*df_annotation$max
df_annotation$x_pval_2 <- 3.5
df_annotation$y_pval_2 <- 1.15*df_annotation$max
df_annotation$x_pval_3 <- 2
df_annotation$y_pval_3 <- 1.25*df_annotation$max
df_annotation$x_pval_4 <- 3
df_annotation$y_pval_4 <- 1.35*df_annotation$max
```

```
Boxplot_gg2_Annotate <- ggplot(data = metabolite_longer,
                              aes(x=groupe, y=Abundance, color=groupe)) +
  geom_boxplot() +
  facet_wrap(~Metab, scales = "free_y") +
  geom_jitter(width=0.1) +
  geom_text(data = df_annotation,
            aes(x = x_pval_1,  y = y_pval_1,
                label = B1_A1), color="black", size=5) +
  geom_text(data = df_annotation,
            aes(x = x_pval_2,  y = y_pval_2,
                label = B2_A2), color="black", size=5) +
  geom_text(data = df_annotation,
            aes(x = x_pval_3,  y = y_pval_3,
                label = A1_A2), color="black", size=5) +
  geom_text(data = df_annotation,
            aes(x = x_pval_4,  y = y_pval_4,
                label = B1_B2), color="black", size=5)
Boxplot_gg2_Annotate
```



Ajouter les segments

- Compléter `df_annotation` avec les coordonnées des extrémités des segments

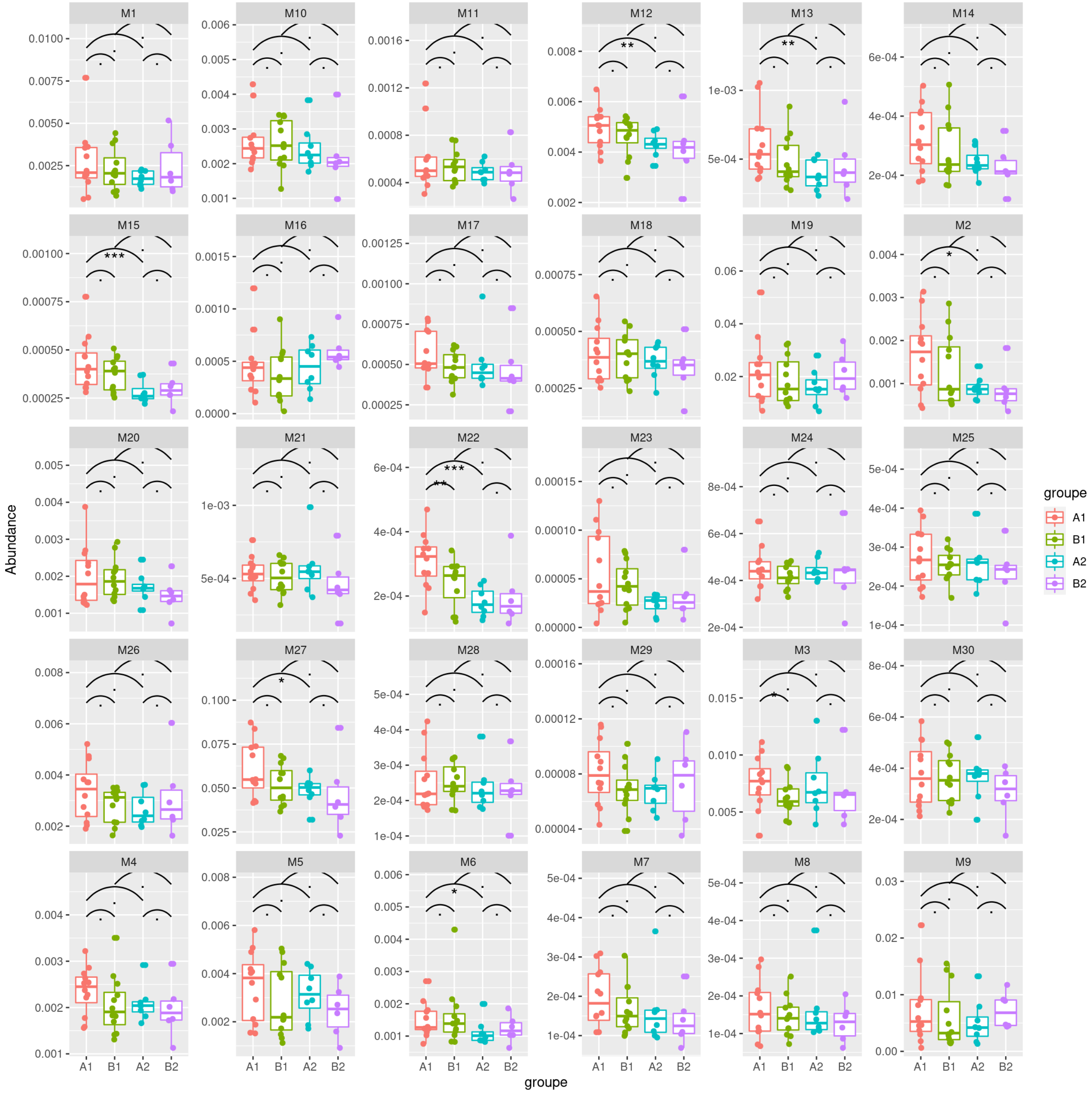
```
df_annotation$x_bar_1 <- 1
df_annotation$y_bar_1 <- 1.13*df_annotation$max
df_annotation$x_end_bar_1 <- 2
df_annotation$y_end_bar_1 <- 1.13*df_annotation$max
df_annotation$x_bar_2 <- 3
df_annotation$y_bar_2 <- 1.13*df_annotation$max
df_annotation$x_end_bar_2 <- 4
df_annotation$y_end_bar_2 <- 1.13*df_annotation$max
df_annotation$x_bar_3 <- 1
df_annotation$y_bar_3 <- 1.23*df_annotation$max
df_annotation$x_end_bar_3 <- 3
df_annotation$y_end_bar_3 <- 1.23*df_annotation$max
df_annotation$x_bar_4 <- 2
df_annotation$y_bar_4 <- 1.33*df_annotation$max
df_annotation$x_end_bar_4 <- 4
df_annotation$y_end_bar_4 <- 1.33*df_annotation$max
```

- Ajouter les segments avec `geom_segment`

```
Boxplot_gg2_Annotate +
    geom_segment(data = df_annotation,
                 aes(x = x_bar_1, y = y_bar_1,
                     xend = x_end_bar_1, yend = y_end_bar_1), color="black") +
    geom_segment(data = df_annotation,
                 aes(x = x_bar_2, y = y_bar_2,
                     xend = x_end_bar_2, yend = y_end_bar_2), color="black") +
    geom_segment(data = df_annotation,
                 aes(x = x_bar_3, y = y_bar_3,
                     xend = x_end_bar_3, yend = y_end_bar_3), color="black") +
    geom_segment(data = df_annotation,
                 aes(x = x_bar_4, y = y_bar_4,
                     xend = x_end_bar_4, yend = y_end_bar_4), color="black")
```

- Cela pourrait aussi être des courbes avec `geom_curve`

```
Boxplot_gg2_Annotate +
    geom_curve(data = df_annotation,
                aes(x = x_bar_1, y = y_bar_1,
                    xend = x_end_bar_1, yend = y_end_bar_1),
              color="black", curvature = -0.5) +
    geom_curve(data = df_annotation,
                aes(x = x_bar_2, y = y_bar_2,
                    xend = x_end_bar_2, yend = y_end_bar_2),
              color="black", curvature = -0.5) +
    geom_curve(data = df_annotation,
                aes(x = x_bar_3, y = y_bar_3,
                    xend = x_end_bar_3, yend = y_end_bar_3),
              color="black", curvature = -0.5) +
    geom_curve(data = df_annotation,
                aes(x = x_bar_4, y = y_bar_4,
                    xend = x_end_bar_4, yend = y_end_bar_4),
              color="black", curvature = -0.5)
```



- Ajuster l’échelle verticale

Pour que la courbe ne sorte pas du graphique, on peut réaliser un dernier petit ajustement avec `geom_blank` et un nouveau data.frame (et/ou modifier la courbure).

```
df_dummy <- select(df_annotation, Metab, min, max)
df_dummy$groupe <- "A1"
df_dummy$min <- 0.9*df_dummy$min
df_dummy$max <- 1.5*df_dummy$max
```

```
Boxplot_gg2_Annotate +
    geom_curve(data = df_annotation,
                 aes(x = x_bar_1, y = y_bar_1,
                     xend = x_end_bar_1, yend = y_end_bar_1),
               color="black", curvature = -0.25) +
    geom_curve(data = df_annotation,
                 aes(x = x_bar_2, y = y_bar_2,
                     xend = x_end_bar_2, yend = y_end_bar_2),
               color="black", curvature = -0.25) +
    geom_curve(data = df_annotation,
                 aes(x = x_bar_3, y = y_bar_3,
                     xend = x_end_bar_3, yend = y_end_bar_3),
               color="black", curvature = -0.25) +
    geom_curve(data = df_annotation,
                 aes(x = x_bar_4, y = y_bar_4,
                     xend = x_end_bar_4, yend = y_end_bar_4),
               color="black", curvature = -0.25) +
  geom_blank(data  = df_dummy, aes(x = groupe, y = max)) +
  geom_blank(data  = df_dummy, aes(x = groupe, y = min))
```

En résumé

- `ggpubr` pourquoi pas.
- quelques découvertes `ggplot2` : `scales = "free_y"`, `geom_curve`, `geom_blank`, manipulation de plusieurs data.frames pour un seul graphique.
- `plus tidy que tidy ?`

Bonus : barplot + barres d’erreur

```
df_bar <- summarise(group_by(metabolite_longer, groupe, Metab),
                    mean=mean(Abundance),
                    stdev=sd(Abundance))


Barplot_ggplot2 <- ggplot(df_bar) +
  geom_bar(aes(x=groupe, y=mean, fill = groupe), stat="identity", alpha=0.5) +
  geom_errorbar(aes(x=groupe, ymin=mean-stdev, ymax=mean+stdev),
                size=.3, width=.3) +
  facet_wrap(~Metab, scales = "free_y") +
  geom_jitter(data=metabolite_longer, aes(x=groupe, y=Abundance, color=groupe),
              width=0.1, size=1, alpha=0.3) +
  scale_colour_manual(values = c("rosybrown","tomato","midnightblue", "purple")) +
  scale_fill_manual(values = c("rosybrown","tomato","midnightblue", "purple")) +
  theme(strip.text = element_text(face="bold")) +
  geom_text(data = df_annotation, aes(x = x_pval_1,  y = y_pval_1,
                                      label = B1_A1),
            color="black", size=3) +
  geom_text(data = df_annotation, aes(x = x_pval_2,  y = y_pval_2,
                                      label = B2_A2),
            color="black", size=3) +
  geom_text(data = df_annotation, aes(x = x_pval_3,  y = y_pval_3,
                                      label = A1_A2),
            color="black", size=3) +
  geom_text(data = df_annotation, aes(x = x_pval_4,  y = y_pval_4,
                                      label = B1_B2),
            color="black", size=3) +
  geom_segment(data = df_annotation,
             aes(x = x_bar_1, y = y_bar_1,
                 xend = x_end_bar_1, yend = y_end_bar_1),
             color="black") +
  geom_segment(data = df_annotation,
               aes(x = x_bar_2, y = y_bar_2,
                   xend = x_end_bar_2, yend = y_end_bar_2),
               color="black") +
  geom_segment(data = df_annotation,
               aes(x = x_bar_3, y = y_bar_3,
                   xend = x_end_bar_3, yend = y_end_bar_3),
               color="black") +
  geom_segment(data = df_annotation,
               aes(x = x_bar_4, y = y_bar_4,
                   xend = x_end_bar_4, yend = y_end_bar_4),
               color="black")

Barplot_ggplot2
```


M1
M10
M11
M12
M13
M14
M15
M16
M17
M18
M19
M2
M20
M21
M22
M23
M24
M25
M26
M27
M28
M29
M3
M30
M4
M5
M6
M7
M8
M9
mean
groupe
A1
B1
A2
B2
groupe
A1
B1
A2
B2