Compte-rendu Statistiques – TP1

Statistiques descriptives et visualisation d'information

Université de Tours
Moreau Clément

30 November 2020

Rendu des questions

Exercice 1

On considère le data frame généré par le code suivant :

```
T <- data.frame(V1=rep(c(1, NA), 3), V2=c(seq(1,5),NA))
T
```

```
##   V1 V2
## 1  1  1
## 2 NA  2
## 3  1  3
## 4 NA  4
## 5  1  5
## 6 NA NA
```

1. Modifier la valeur située ligne 3, colonne 1 de T par la valeur 10.

   ```
   T[3,1] <- 10
   T
   ```

   ```
   ##   V1 V2
   ## 1  1  1
   ## 2 NA  2
   ## 3 10  3
   ## 4 NA  4
   ## 5  1  5
   ## 6 NA NA
   ```

2. Dans la colonne 2, remplacer toutes les valeurs ≥ 4 par la valeur 20. On pourra utiliser la commande `ifelse(phi, valT, valF)` qui rend la valeur `valT` si la condition logique `phi` est vérifiée et `valF` sinon.

   ```
   T$V2 <- ifelse(T$V2 >= 4, 20, T$V2)
   T
   ```

   ```
   ##   V1 V2
   ## 1  1  1
   ## 2 NA  2
   ## 3 10  3
   ```

```
## 4 NA 20
## 5  1 20
## 6 NA NA
```

3. On peut détecter si une valeur possède la valeur `NA` grâce à la commande `is.na()`.

   Remplacer toutes les valeurs `NA` de T par la valeur 0.

```
for(j in 1:ncol(T)) {
  T[, j] <- ifelse(is.na(T[, j]), 0, T[, j])
}
T
```

```
##   V1 V2
## 1  1  1
## 2  0  2
## 3 10  3
## 4  0 20
## 5  1 20
## 6  0  0
```

4. Ajouter une nouvelle colonne à T qui est la moyenne des deux colonnes V_1 et V_2.

```
T$V3 <- (T$V1 + T$V2)/2
T
```

```
##   V1 V2   V3
## 1  1  1  1.0
## 2  0  2  1.0
## 3 10  3  6.5
## 4  0 20 10.0
## 5  1 20 10.5
## 6  0  0  0.0
```

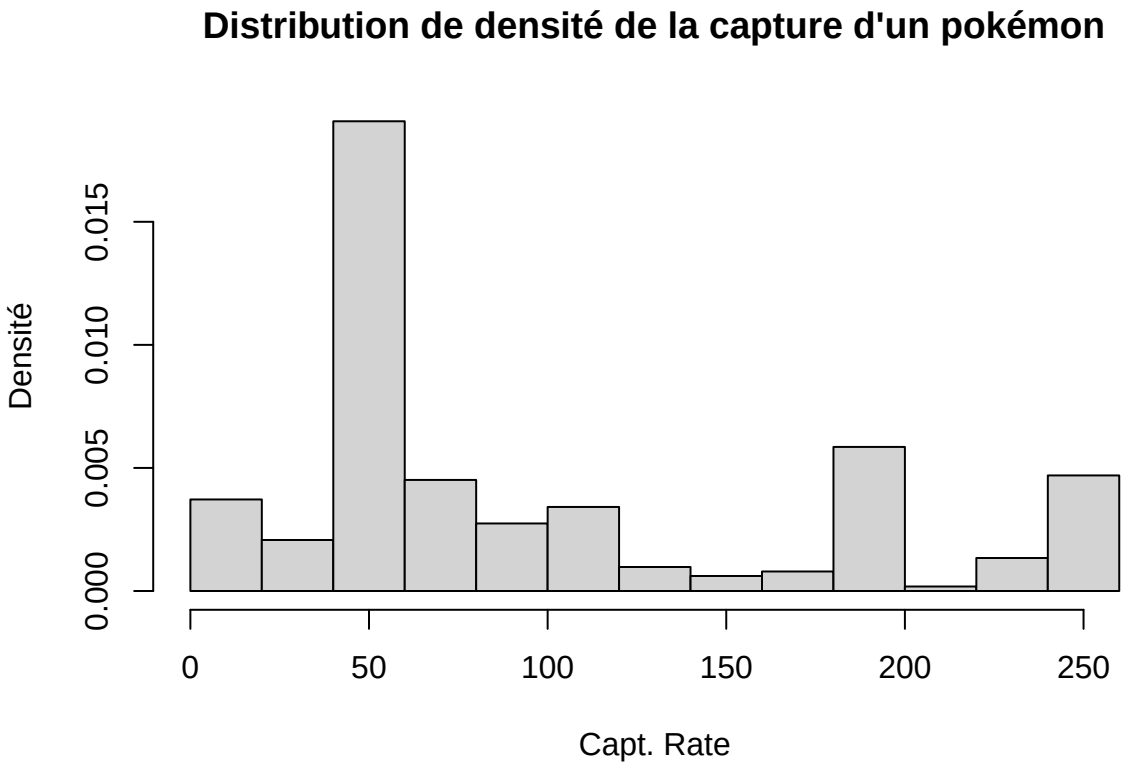
5. Ajouter au dataframe `poke` la nouvelle colonne `base_stats` qui est correspond à la somme des différentes statistiques du pokémon, c.a.d des variables `hp`, `atk`, ..., `spd`.

```
poke$base_stats <- (poke$hp +
                    poke$atk +
                    poke$def +
                    poke$sp_atk +
                    poke$sp_def + poke$spd)
```

Exercice 2

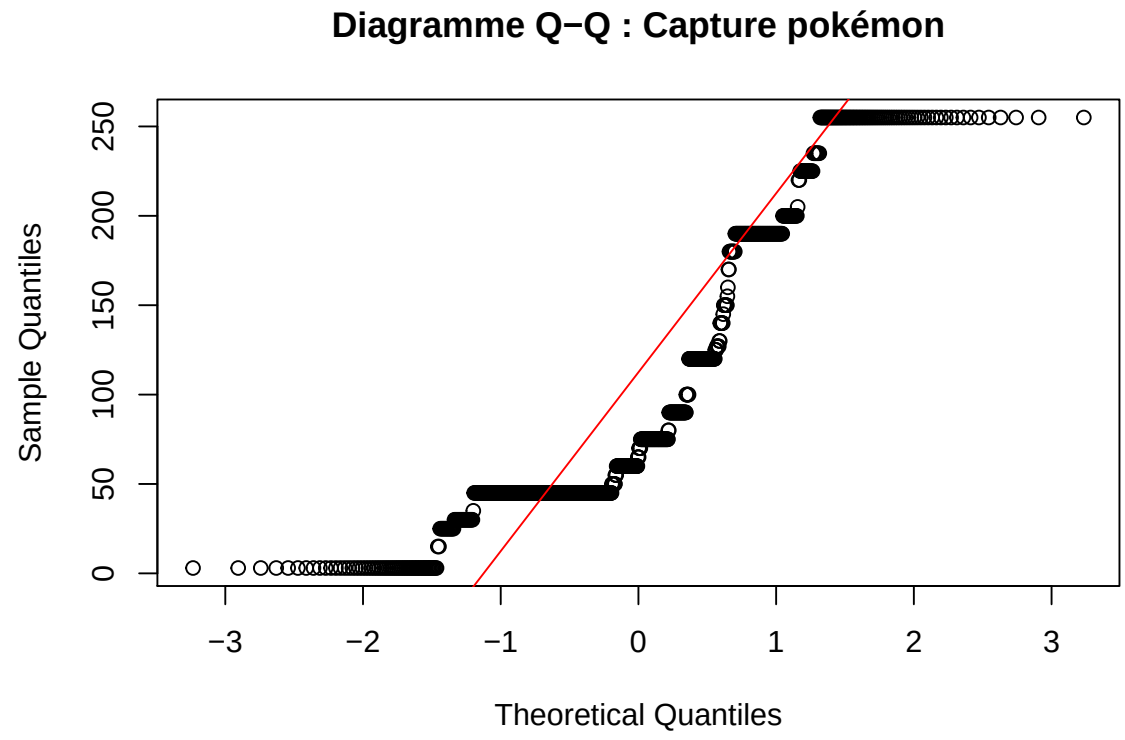
1. Dresser l'histogramme de la variable `capt_rate`. Cette variable suit-elle une loi Normale ? Expliquer votre réponse.

```
hist(poke$capt_rate,
     main = "Distribution de densité de la capture d'un pokémon",
     freq = FALSE, xlab = "Capt. Rate", ylab = "Densité")
```



L'histogramme semble indiquer que la variable ne suit pas une loi normale. On peut corroborer cette hypothèse à l'aide un Q-Q plot.

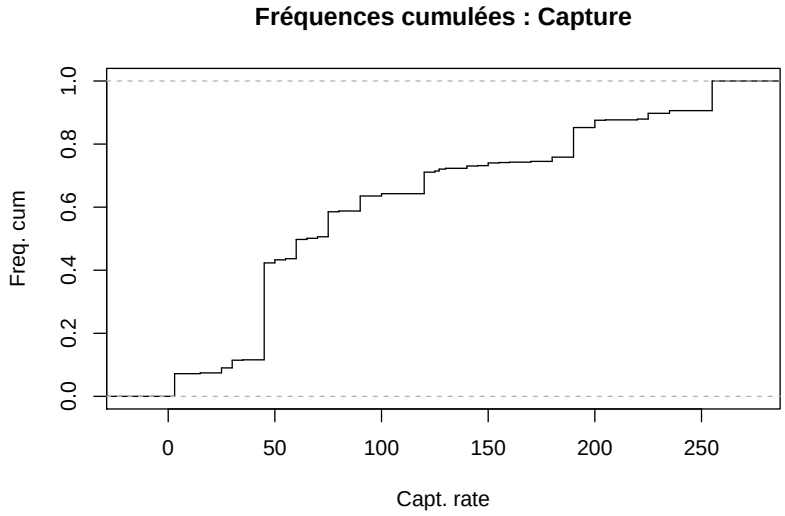
```
qqnorm(poke$capt_rate, main = "Diagramme Q-Q : Capture pokémon")
qqline(poke$capt_rate, col = "red")
```



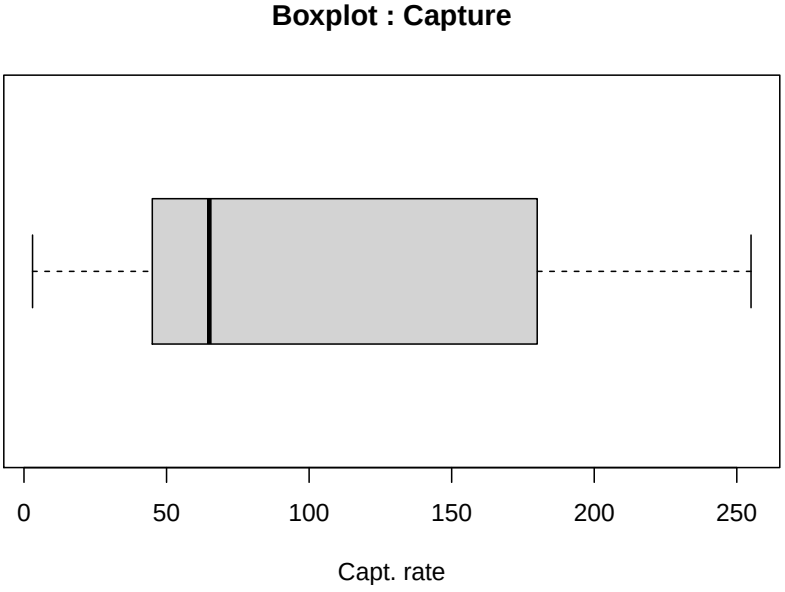
L'ajustement n'est pas respecté, on peut rejeter l'hypothèse d'une loi Normale. La variable semble distribuée par pallier.

2. Dresser les graphiques des fréquences cumulées croissantes et la boite à moustaches de `capt_rate`. Que constatez-vous ? Quelles sont les similitudes entre ces deux graphiques ?

```
plot(ecdf(poke$capt_rate),
     verticals = TRUE,
     do.points = FALSE,
     xlab = "Capt. rate",
     ylab = "Freq. cum",
     main = "Fréquences cumulées : Capture")
```




```
boxplot(poke$capt_rate,
        horizontal = TRUE,
        xlab = "Capt. rate",
        main = "Boxplot : Capture")
```



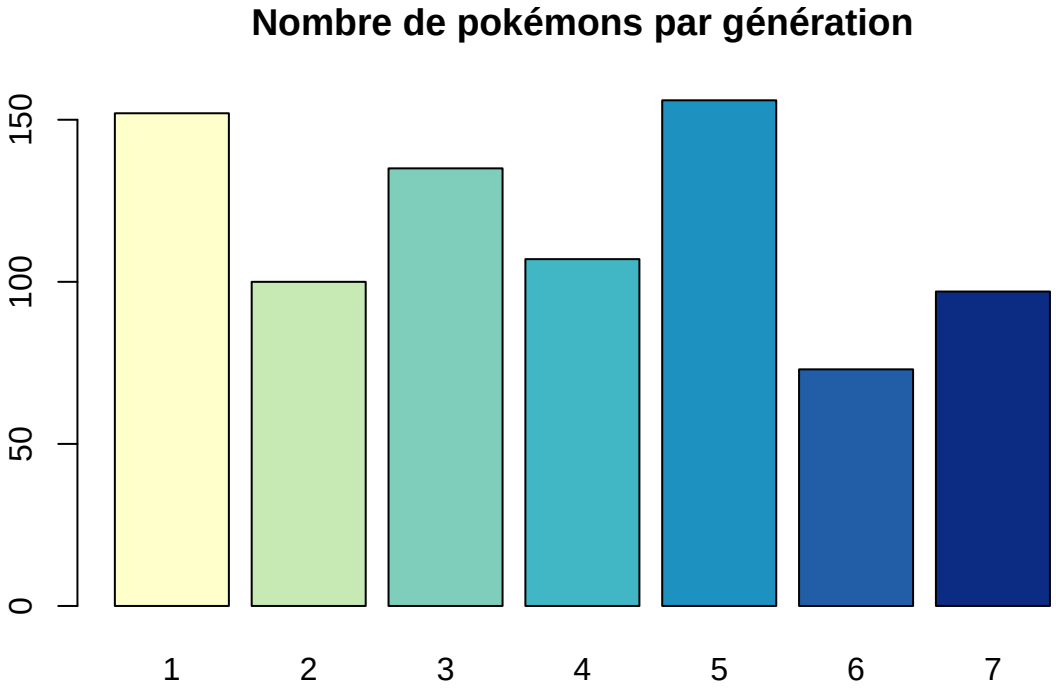
On peut voir la boxplot comme une vue simplifiée des fréquences cumulées croissantes (fcc). La

boxplot indique principalement les informations des quantiles Q_1 et Q_3 ainsi que la médiane. Ces informations peuvent aussi être visualisées sur le graphique des fcc mais pour tout percentile p.

Exercice 3

Code pour le graphique:

```
library(RColorBrewer)
barplot(table(poke$generation),
        col = brewer.pal(n = 7, name = "YlGnBu"),
        main = "Nombre de pokémons par génération")
```



Exercice 4

1. Quel est le coefficient de corrélation entre les variables `sp_atk` et `base_stats`. Pouvez-vous expliquez pourquoi ce score est élevé ?

   ```
   cor(poke$base_stats, poke$sp_atk)
   ```

   ```
   ## [1] 0.7432076
   ```

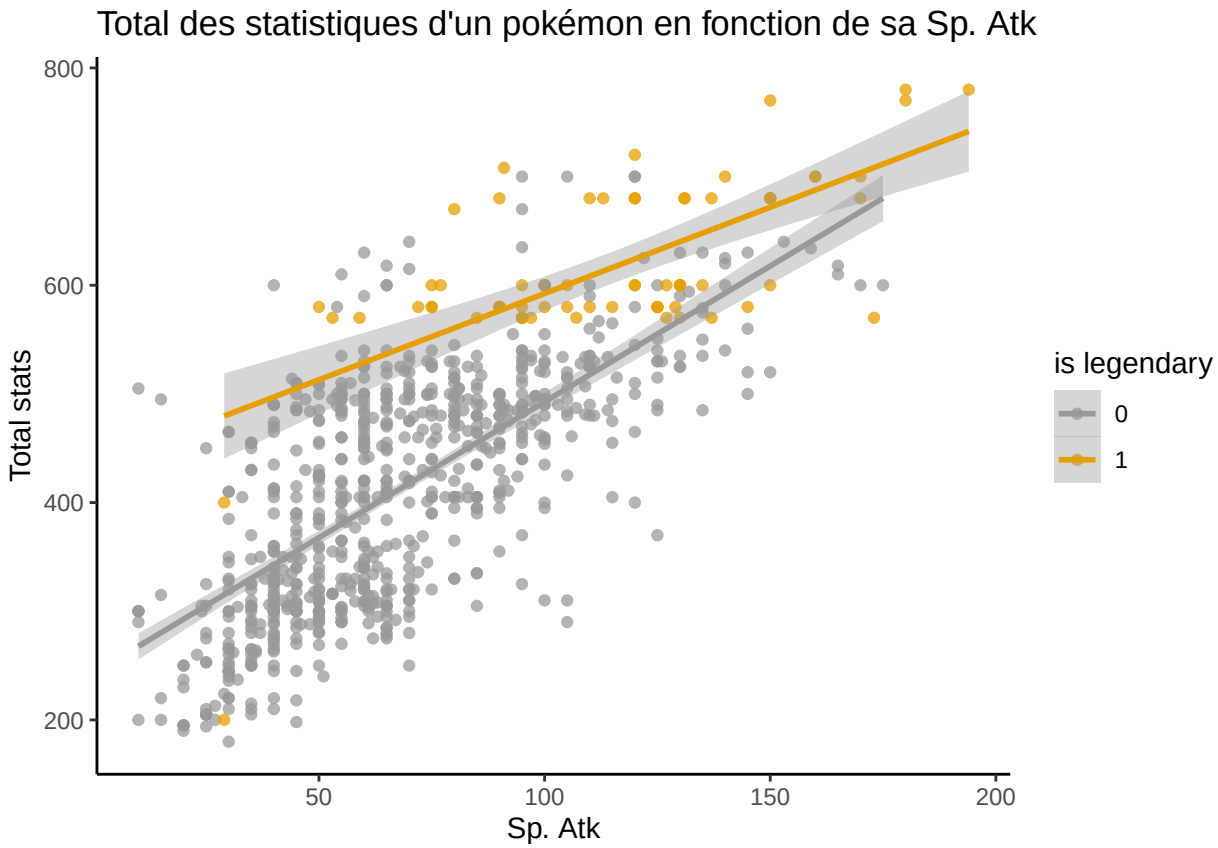
   Le score est élevé car la variable ***`sp_atk`*** *fait partie de la combinaison linéaire qui a servi à créer* ***`base_stats`****.*

2. Donner le code R permettant de générer le graphique ci-dessous.

```
library(ggplot2)
poke$is_legendary <- as.factor(poke$is_legendary)
p <- ggplot(poke, aes(x = sp_atk, y = base_stats)) +
     geom_point(aes(color = is_legendary), alpha = 0.75) +
     geom_smooth(aes(color = is_legendary), method = "lm") +
     scale_color_manual(values=c('#999999','#E69F00')) +
```

```
     theme_classic() +
     labs(color = "is legendary",
          x = "Sp. Atk",
          y = "Total stats",
          title = "Total des statistiques d'un pokémon en fonction de sa Sp. Atk")
p
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Exercice 5

Supposons que l'on veuille visualiser, pour chaque type, la proportion de chaque autre type de Pokémon. Par exemple, pour le type `grass`, on souhaite connaitre la proportion de pokémons de types `grass / t` où `t` est un type quelconque.

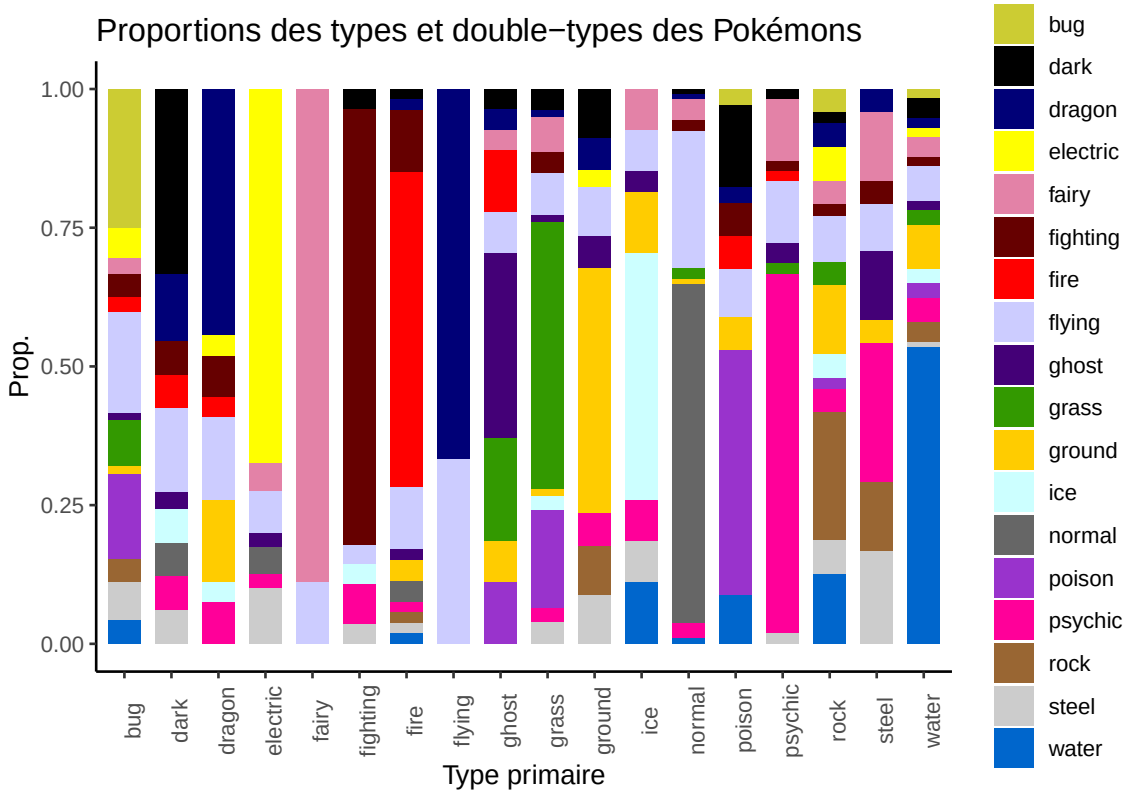
Plus formellement, on souhaite visualiser la proportion de chaque combinaison de types (t_1, t_2).

1. Créer un dataframe doté de deux colonnes `t1` et `t2` où chaque ligne correspond à un pokémon et où `t1` correspond au type primaire du pokémon et `t2` son type secondaire. Dans le cas où `type2 = NA`, `t2` prendra la valeur de `type1`. Par exemple, si on a `type1 = grass` et `type2 = NA`, on affectera à `t2` également la valeur `grass` ce qui correspond à dire que le pokémon est de type "plante" pur.

```
T <- data.frame(type1 = poke$type1,
                type2 = ifelse(is.na(poke$type2), poke$type1, poke$type2))
```

2. Donner le code R permettant de générer le graphique des proportions des types secondaires pour chaque type de pokémon.

```r
p <- ggplot(T) +
  geom_bar(width=0.7,
           mapping = aes(x = type1, fill = type2),
           position = "fill") +
  scale_fill_manual(values= unlist(lapply(sort(unique(T$type2)), col_type))) +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(fill = "Type",
       x = "Type primaire",
       y = "Prop.",
       title = "Proportions des types et double-types des Pokémons")
p
```

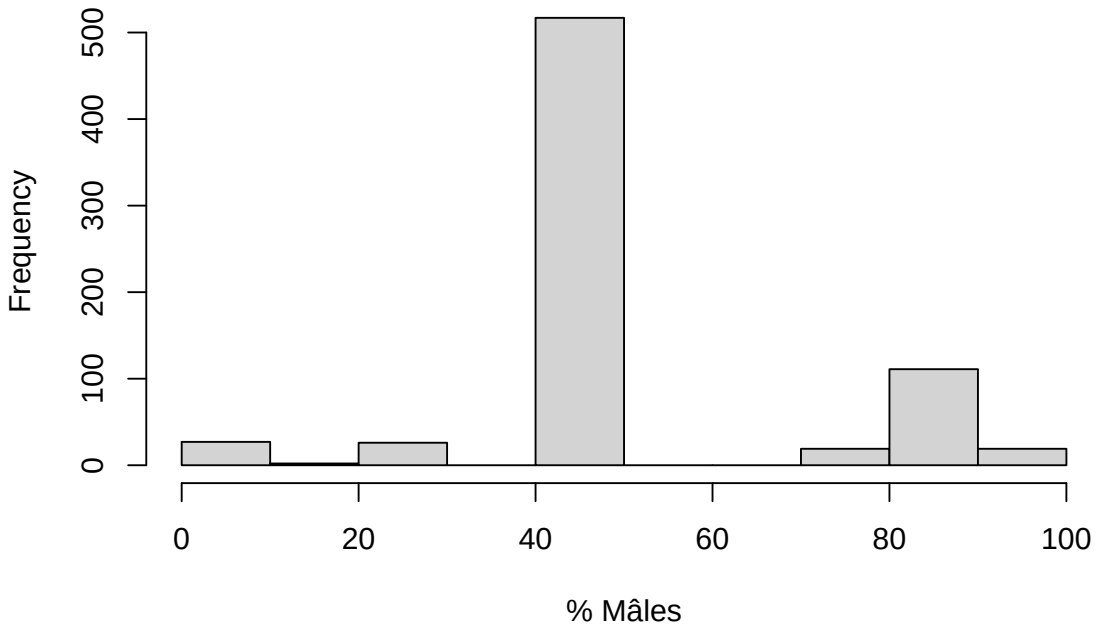


Exercice 6

On souhaite vérifier l'hypothèse que les pokémons roses sont davantage genrés comme femelle (`percentage_male` < 50) que les autres pokémons.

1. Dresser l'histogramme de la variable `percentage_male` et proposer une méthode de discrétisation pour cette variable.

```r
hist(poke$percentage_male,
     xlab = "% Mâles",
     main = "Histogramme de la proportion de mâles chez les pokémons")
```



On remarque 3 étendues principales de la variables ***percentage_male****:*

- $\leq 30\%$
- 50%
- $\geq 80\%$

On peut garder l'esprit de la méthode de Jenks et attribuer les labels selon les conditions suivantes:

- *Si* `percentage_male` $< 50 \Rightarrow$ *F qui indique une prévalence des femelles.*
- *Si* `percentage_male` $= 50 \Rightarrow$ *N, pour aucune prévalence (Neutre).*
- *Si* `percentage_male` $> 50 \Rightarrow$ *M, qui indique une prévalence des mâles.*

```
poke$percentage_male_disc <- ifelse(is.na(poke$percentage_male), NA,
                                    ifelse(poke$percentage_male < 50, "F",
                                    ifelse(poke$percentage_male == 50, "N", "M")))
```

2. Créer une nouvelle variable `is_pink` qui vaut 1 si le pokémon est rose et 0 sinon.

```
is_pink <- ifelse(poke$color == "Pink", 1, 0)
```

3. Proposer une méthode permettant vérifier l'hypothèse puis conclure.

   Deux méthodologies sont valables pour corroborer notre hypothèse que les pokémons roses sont davantage genrés comme femelle:

   - *À l'aide de notre discrétisation de* ***percentage_male*** *et de la variable* ***is_pink****, on peut effectuer un test du chi2 et regarder la p-valeur associée:*

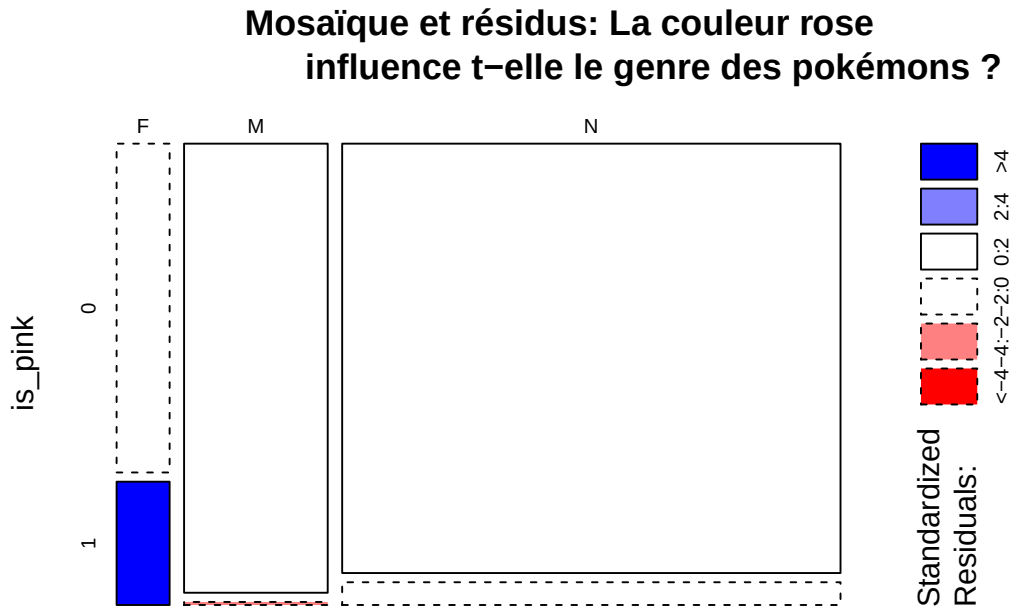
     ```
     chisq.test(table(poke$percentage_male_disc, is_pink))
     ```

     ```
     ## Warning in chisq.test(table(poke$percentage_male_disc, is_pink)): Chi-squared
     ## approximation may be incorrect
     ```

```
##
##  Pearson's Chi-squared test
##
## data:  table(poke$percentage_male_disc, is_pink)
## X-squared = 53.93, df = 2, p-value = 1.946e-12
```

- *Établir le diagramme mosaïque avec les résidus de Pearson et constater s'il on observe un écart à la norme ($r > 2$) dans la case correspondant à l'événement* (`percentage_male_disc` $= F) \cap$ (`is_pink` $= 1$)*:*

```
mosaicplot(table(poke$percentage_male_disc, is_pink),
           shade = TRUE,
           main = "Mosaïque et résidus: La couleur rose
                   influence t-elle le genre des pokémons ?")
```



Dans les deux cas l'hypothèse est vérifiée. La p-valeur est clairement en dessous de 0.05 et l'on constate effectivement une cellule bleue foncée à l'intersection voulue.

Exercice 7

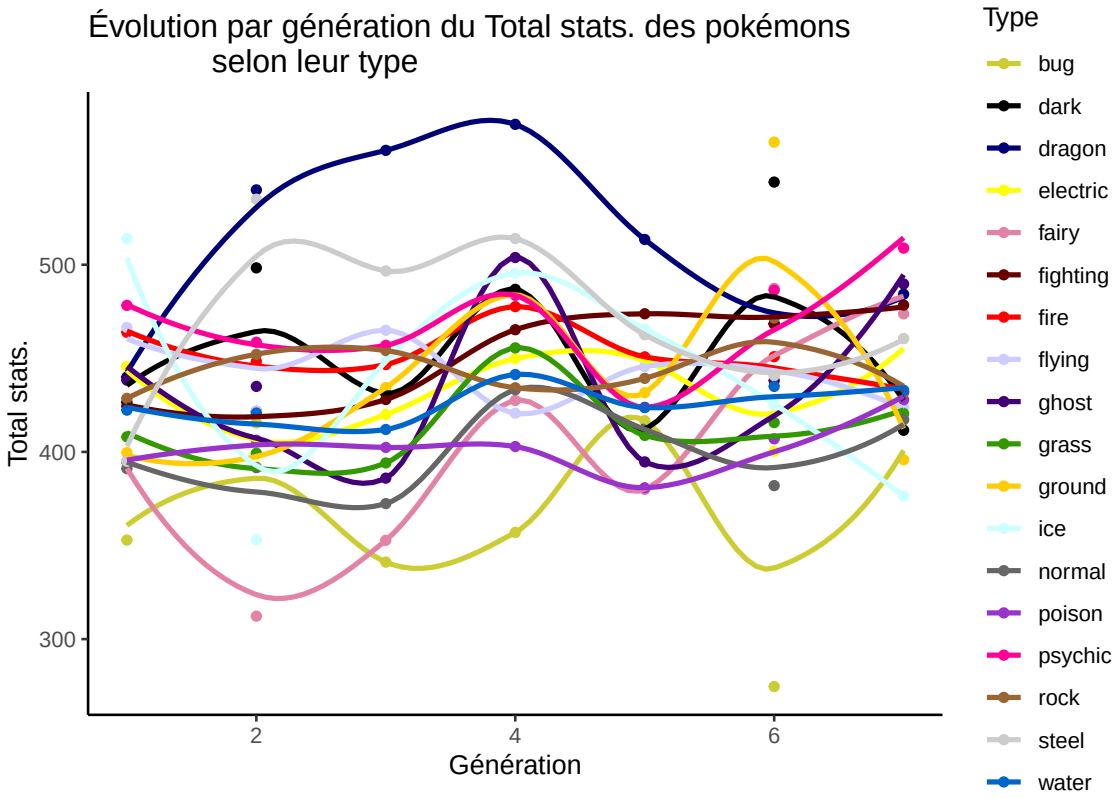
1. Code pour le graphique:

```
T1 <- data.frame(poke[, c("generation", "type1", "base_stats")]) ;
names(T1) <- c("generation", "type", "base_stats")
T2 <- subset(poke, !is.na(type2))[, c("generation", "type2", "base_stats")] ;
names(T2) <- c("generation", "type", "base_stats")
#
T <- data.frame(rbind(T1, T2))
summary <- data_summary(T,
                        varname = "base_stats",
                        groupnames = c("type","generation"))
```

```
## Loading required package: plyr
```

```
#
p <- ggplot(summary, aes(x=generation, y=base_stats, color=type)) +
  geom_point()+
  scale_color_manual(values= unlist(lapply(sort(unique(summary$type)), col_type))) +
  geom_smooth(se = FALSE) +
  theme_classic() +
  labs(title="Évolution par génération du Total stats. des pokémons
              selon leur type",
       x = "Génération",
       y = "Total stats.",
       color = "Type")
p
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



2. Quelles analyses peut-on en tirer ? Quel est son défaut majeur ?

   On remarque sur ce graphique que le type de pokémons, au fil des générations, le plus favorisé quant à leur base de statistiques moyenne est le type Dragon. Le plus défavorisé est le type Insecte. Le reste des types fluctue selon les générations entre une `base_stats` $\in [380, 500]$.

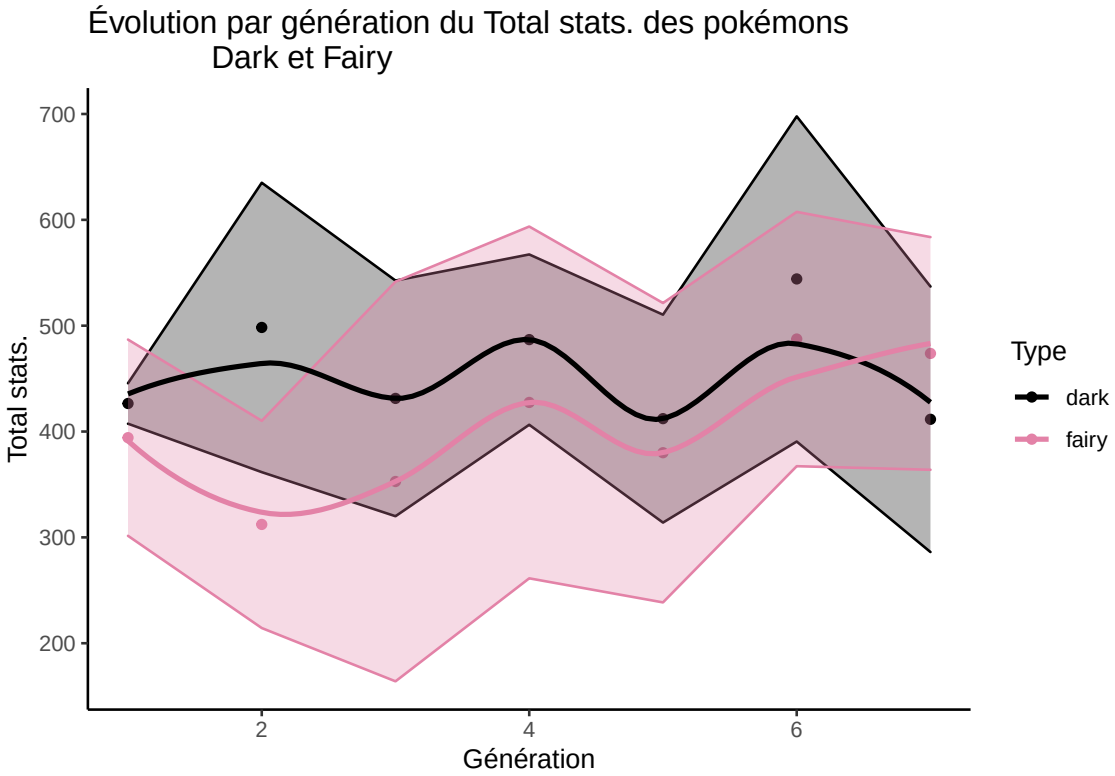
   Le défaut majeur de ce graphique est la lisibilité, il est très difficile de tirer des observations individuelles par type. Une possibilité pour plus de lisibilité du graphique serait de faire apparaitre uniquement certains types ou valeurs en fonction de l'objectif communicationnelle.

2. Reprendre le graphique précédent et faire figurer uniquement les courbes des types `dark` et `fairy`. On ajoutera également un ruban pour générer les écarts-types.

```
summary <- subset(summary, type == "fairy" | type == "dark")
```

```
p <- ggplot(summary, aes(x=generation, y=base_stats, color=type)) +
  geom_point()+
  scale_color_manual(values = unlist(lapply(sort(unique(summary$type)), col_type))) +
  geom_ribbon(aes(ymin = base_stats-sd, ymax=base_stats+sd, fill = type),
              alpha=0.3, show.legend = FALSE)  +
  scale_fill_manual(values = unlist(lapply(sort(unique(summary$type)), col_type))) +
  geom_smooth(se = FALSE) +
  theme_classic() +
  labs(title="Évolution par génération du Total stats. des pokémons
              Dark et Fairy",
       x="Génération",
       y = "Total stats.",
       color = "Type")
p
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



On voit sur ce graphique que durant presque toutes les générations, les pokémons Dark étaient globalement plus puissants que les pokémons Fairy. La courbe noire est presque toujours au dessus de la courbe rose de même que le ruban noire des écarts-types. Néanmoins, on observe un revirement pour la dernière génération où le point et la courbe roses sont au dessus des deux indicateurs noirs. De plus, le ruban noire est en dessous du rose ce qui indique que les pokémons Dark sont globalement plus faibles que les Fairy pour la génération 7.

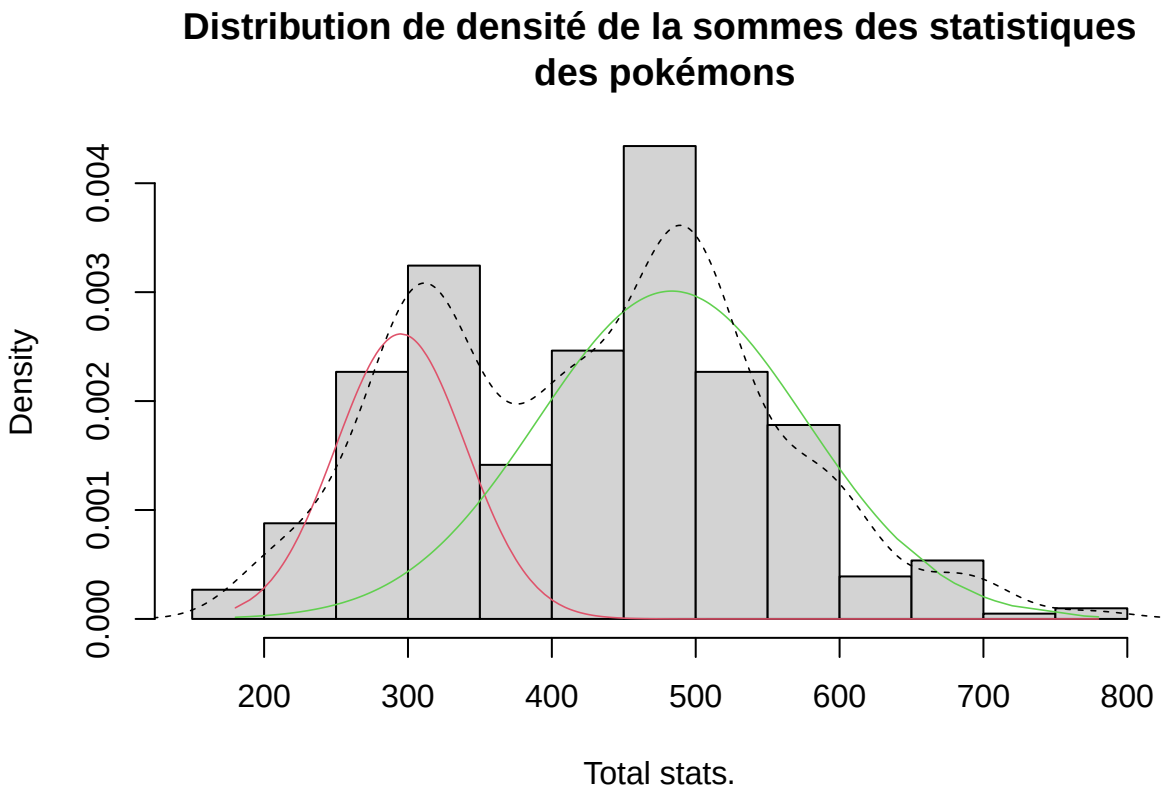
Exercice 8

```
library("mixtools")
```

```
## mixtools package, version 1.2.0, Released 2020-02-05
```

```
## This package is based upon work supported by the National Science Foundation under Grant No. SES-0518
EM_poke <- normalmixEM(poke$base_stats, mu = c(320, 480), sigma=c(50,100), k=2)

plot(EM_poke, which=2,
     xlab2 = "Total stats.",
     main2 = "Distribution de densité de la sommes des statistiques\n des pokémons",
     lwd2=0.8)
lines(density(EM_poke$x), lty=2, lwd=0.8)
```



1. Quel phénomène selon vous peut expliquer l'apparition de mélanges gaussiens ?

   Un mélange gaussien au sein d'une variable est caractérisé par des sous groupes d'individus $G_1, G_2, ...G_p$ *de proportion* $\pi_1, ..., \pi_p$ *qui suivent chacun une loi normale de moyenne* μ_k *et de matrice de variance-covariance* Σ_k*. On note* $\theta = \{\mu_{i=1,...,p}, \Sigma_{i=1,...,p}\}$ *la mixture à g composantes. Ainsi, on a:*

$$P(\theta) = \sum_{i=1}^{p} \pi_i \mathcal{N}(\mu_i, \Sigma_i)$$

2. Déterminer une méthodologie d'analyse, que vous détaillerez, permettant d'expliquer la présence des deux gaussiennes sur le graphique précédent.

   Compte tenu de la mixture observée, on peut chercher le groupe d'individus caractérisé par la gaussienne rouge et le groupe d'individus caractérisé par la gaussienne verte.

   On note $\mu_1 = 320, \sigma_1 = 50$ *respectivement la moyenne et l'écart-type de la gaussienne rouge (resp.* $\mu_2 = 480, \sigma_2 = 100$*). Ainsi, pour chaque individu i, si* $\texttt{base\_stats}_i \in [\mu_1 - 2\sigma_1, \mu_1 + 2\sigma_1]$*, on admettra que* $i \in G_1$*, (resp. pour la gaussienne verte). On notera que i peut appartenir à la fois à* G_1 *et* G_2*.*

Enfin, on analysera chacun des groupes G_1 et G_2 afin de déterminer la prévalence d'une caractérisque au sein de chacun des groupes.

```
mu_1 <- 320
sigma_1 <- 50
G_1 <- subset(poke, base_stats >= (mu_1 - 2 * sigma_1)
              & base_stats <= (mu_1 + 2 * sigma_1))

mu_2 <- 480
sigma_2 <- 100
G_2 <- subset(poke, base_stats >= (mu_2 - 2 * sigma_2)
              & base_stats <= (mu_2 + 2 * sigma_2))
```

3. Conclure votre analyse et expliquer à quoi correspond, en terme d'individus, chacune de ces gaussiennes.

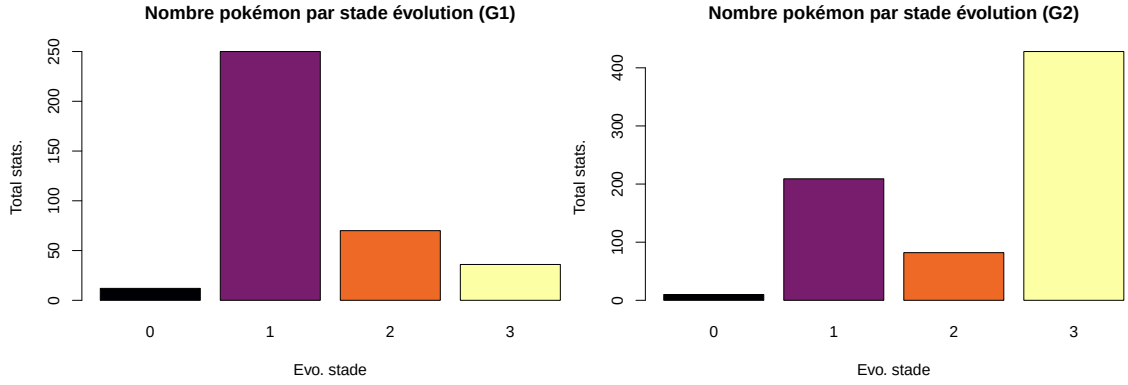
On peut s'intéresser à la variable ***`evolving_stade`*** *de chacun de groupes.*

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
barplot(table(G_1$evolving_stade),
        col = inferno(4),
        ylab = "Total stats.",
        xlab = "Evo. stade",
        main = "Nombre pokémon par stade évolution (G1)")

barplot(table(G_2$evolving_stade),
        col = inferno(4),
        ylab = "Total stats.",
        xlab = "Evo. stade",
        main = "Nombre pokémon par stade évolution (G2)")
```



On voit ici que G_2 comporte beaucoup plus de pokémons d'évolution terminale (stade 3), qui sont les plus forts et donc avec une ***`base_stats`*** *élevée, tandis que G_1 concentre lui les pokémons des stades inférieurs (stade 1 principalement).*

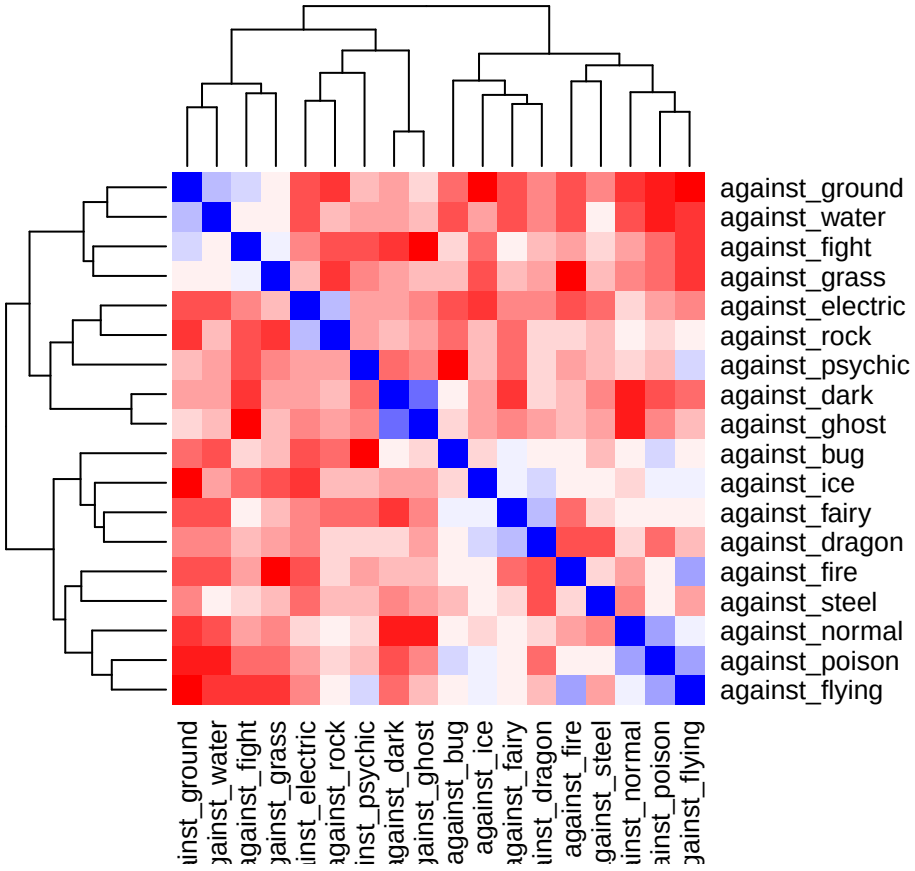
Exerice 9

```
library("corrplot")
```

```
## corrplot 0.84 loaded
```

```r
source("http://www.sthda.com/upload/rquery_cormat.r")

num_var <- poke[,25:ncol(poke)-1]
col<- colorRampPalette(c("red", "white", "blue"))(20)
rquery.cormat(num_var, graphType="heatmap", col = col)
```



Donner une analyse du dernier diagramme (carte de chaleur et dendrogramme).

Une case bleue signifie que l'efficacité du type T_1 et du type T_2 sont positivement corrélée. Autrement dit, on a de bonne chance qui si la variable `against_T1` *soit haute pour un pokémon, la variable* `against_T2` *le soit aussi (resp. basse). À l'inverse, une case rouge indique que si* `against_T1` *est importante, alors* `against_T2` *sera probablement faible. Le dendrogramme en haut et sur le côté groupe les variables en fonction de leur corrélation. Plus celle-ci est forte, plus les variables sont agrégées bas dans l'arbre du dendrogramme (i.e. considérées comme similaires).*