PROJET R - GARY DJIGO

REPORTING

Récupérons le dataset "worldHapinessReport"

```
data=read.csv("worldHappinessReport.csv")
head(data)
```

```
##   X     Country Happiness       GDP SocialSupport Health   Freedom  Generosity
## 1 1 Afghanistan  2.661718  7.497755     0.4908801   52.8 0.4270109 -0.11219829
## 2 2     Albania  4.639548  9.376145     0.6376983   68.4 0.7496110 -0.03264282
## 3 3     Algeria  5.248912  9.540639     0.8067539   65.7 0.4366705 -0.19152211
## 4 4   Argentina  6.039330  9.848709     0.9066991   68.6 0.8319662 -0.18259992
## 5 5     Armenia  4.287736  9.081095     0.6979249   66.6 0.6136971 -0.13395788
## 6 6   Australia  7.257038 10.706581     0.9499578   73.3 0.9105502  0.30877295
##   Corruption PositiveAffect NegativeAffect ConfidenceInGovernment
## 1  0.9543926      0.4963486      0.3713262              0.2611785
## 2  0.8761346      0.6692409      0.3338841              0.4577375
## 3  0.6997742      0.6419796      0.2887100                     NA
## 4  0.8410525      0.8094226      0.2917173              0.3054303
## 5  0.8646833      0.6250138      0.4371487              0.2469010
## 6  0.4113465      0.7800789      0.2253610              0.4534070
##                          Region
## 1                 Southern Asia
## 2    Central and Eastern Europe
## 3 Middle East and Northern Africa
## 4     Latin America and Caribbean
## 5    Central and Eastern Europe
## 6     Australia and New Zealand
```

STATISTIQUE UNIVARIÉ

La variable que nous allon étudier est le : GDP (Gross Domestic Product). C'est l'équivalent du PIB (produit interieur brut).

Etudions la moyenne de cette variable sans tenir compte des valeurs de manquantes :

```
GDP = data$GDP
mean(GDP, na.rm = TRUE)
```

```
## [1] 9.298247
```

Faisons un résumé de celle-ci :

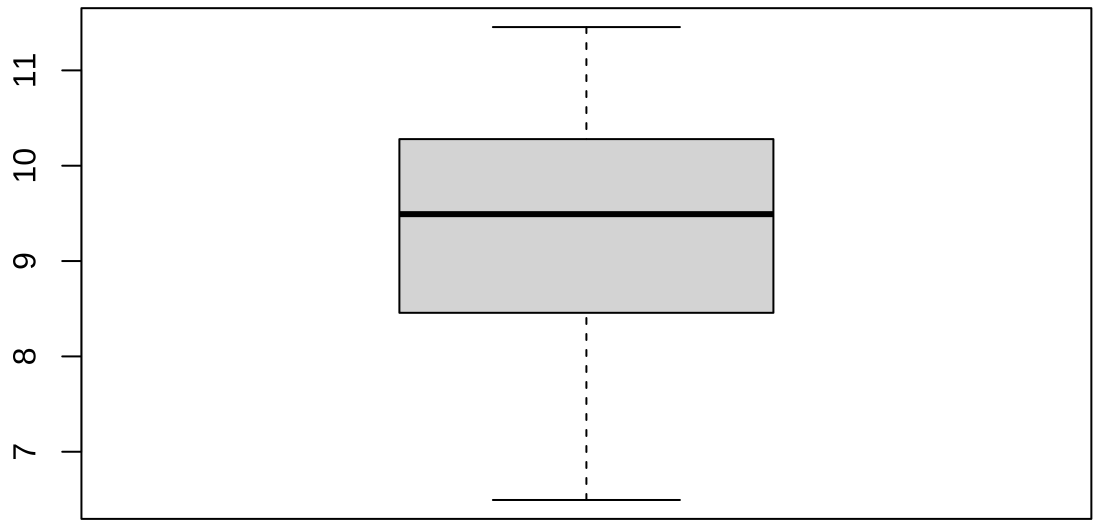
```r
summary(GDP, na.rm = TRUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   6.494   8.457   9.492   9.298  10.279  11.454       4
```

Nous remarquons que nous avons 4 valeurs manquantes pour cette variable.

Réalisons un boxplot et vérifions également l'écart type de la variable :

```r
boxplot(GDP, na.rm = TRUE)
```



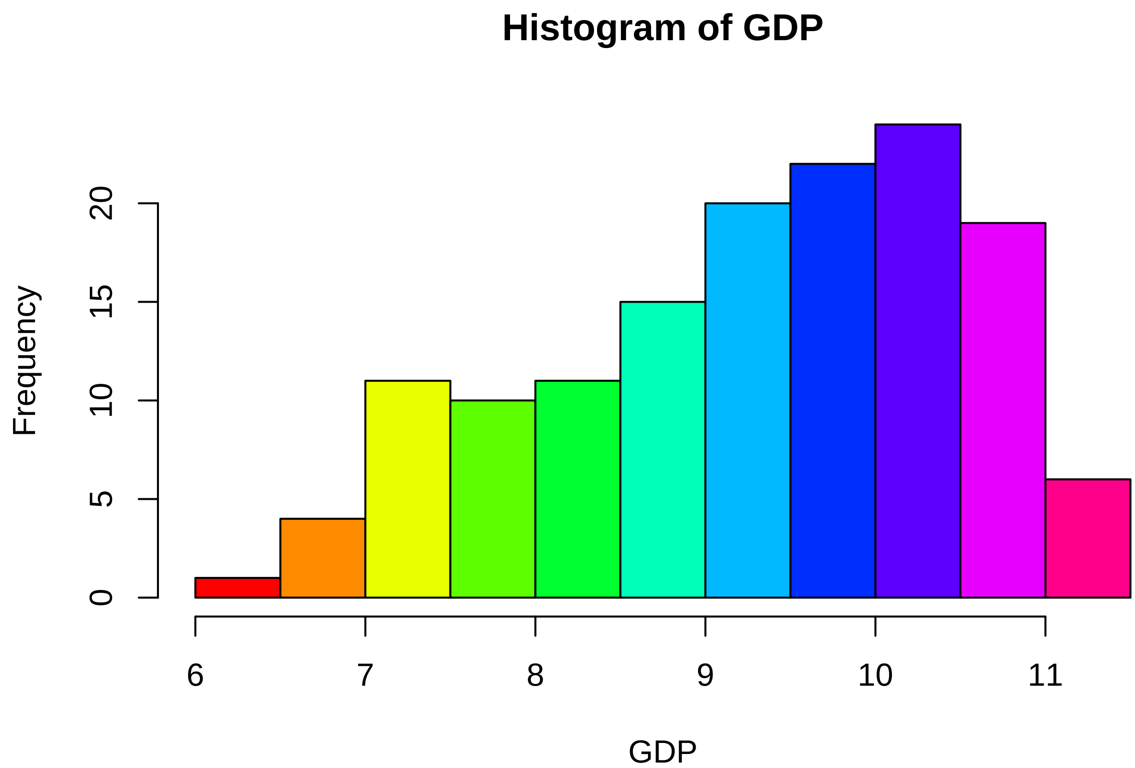

```r
sd(GDP, na.rm = TRUE)
```

```
## [1] 1.199979
```

Nous remarquons que la majorité de nos données GDP se situent entre 8.4 et 10.2. Toutes données se situant en dessous de 6.94 et au dessus de 11.45 seront considérées comme abbérantes. L'écart type est de 1.199979.

Vérifions si cette variable suit la loi normale :

```r
hist(GDP, col =rainbow(11))
```



Nous remarquons que nous sommes beaucoup plus proche d'une loi poisson que d'une loi normale.

```
shapiro.test(GDP)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  GDP
## W = 0.96199, p-value = 0.000539
```

pvalue plus petit que 5% donc nous pouvons rejeter l hypothese de normalite donc les prix ne sont pas issues du hasard!!!

Nous recherchons l'intervalle de confiance à 90% de la moyenne (9.208247) :

```
t.test(GDP)
```

```
##
##  One Sample t-test
##
## data:  GDP
## t = 92.661, df = 142, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  9.099879 9.496615
```

```
## sample estimates:
## mean of x
##  9.298247
```

L'intervalle de confiance à 90% de la moyenne est inférieur à 9.09 donc elle n'est pas fiable.

Nous allons observer le Top 10 des pays ayant le plus grand GDP.

```r
country=data$Country
df=data.frame(country,GDP)

attach(df)
```

```
## Les objets suivants sont masqués _par_ .GlobalEnv:
##
##     country, GDP
```

```r
ndf <- df[order(-GDP,country),]

head(ndf,10)
```

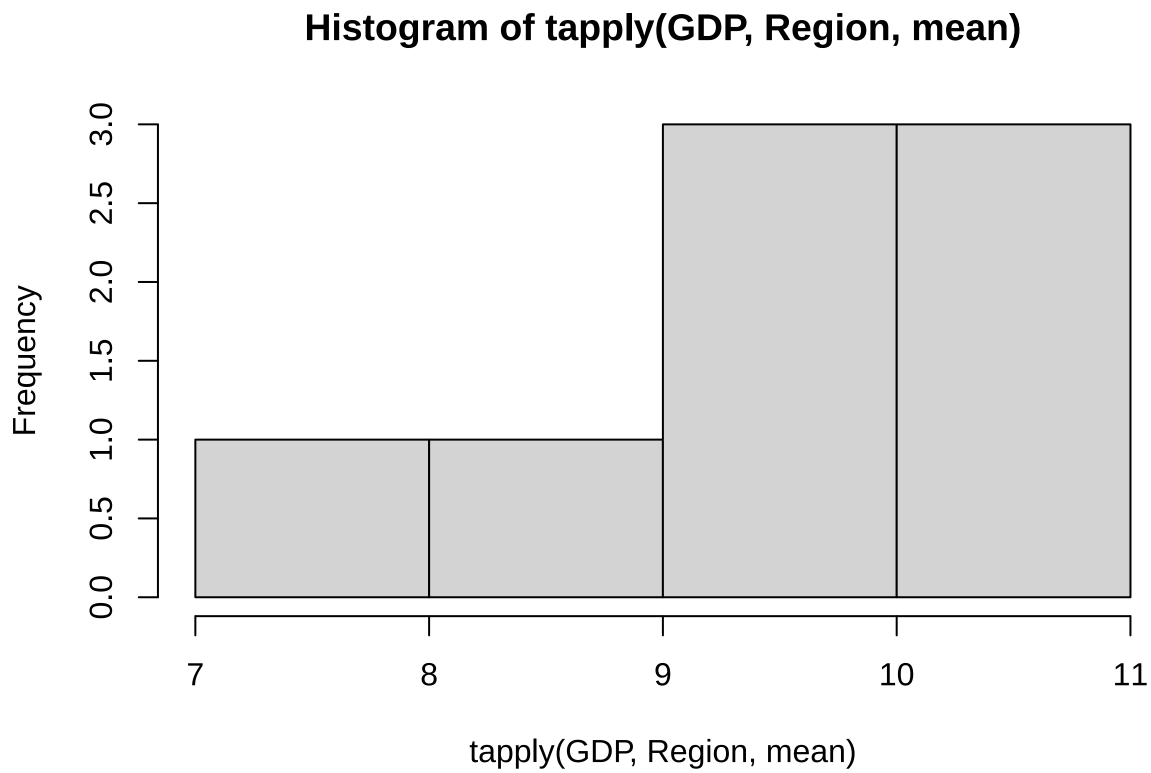
```
##                  country      GDP
## 79            Luxembourg 11.45400
## 117            Singapore 11.35669
## 60               Ireland 11.11744
## 138 United Arab Emirates 11.11682
## 70                Kuwait 11.09027
## 101               Norway 11.07906
## 126          Switzerland 10.95798
## 53             Hong Kong 10.93409
## 140        United States 10.90091
## 113         Saudi Arabia 10.80050
```

Calculons la moyenne des GPD par région du monde

```r
Region = data$Region
tapply(GDP,Region, mean)
```

```
##       Australia and New Zealand      Central and Eastern Europe
##                       10.600119                        9.611540
##                    Eastern Asia     Latin America and Caribbean
##                              NA                        9.305069
## Middle East and Northern Africa                   North America
##                              NA                       10.796625
##              Southeastern Asia                   Southern Asia
##                        9.204281                        8.353837
##              Sub-Saharan Africa                  Western Europe
##                        7.862872                       10.683490
```

```r
hist(tapply(GDP,Region,mean))
```



STATISTIQUE GÉNÉRALE ET INFERENTIELLE

Nous allons maintenant étudier les variables catégorielles "Région" :

```
table(Region)
```

```
## Region
##       Australia and New Zealand      Central and Eastern Europe
##                               2                              29
##                    Eastern Asia     Latin America and Caribbean
##                               6                              21
## Middle East and Northern Africa                   North America
##                              18                               2
##              Southeastern Asia                   Southern Asia
##                               8                               6
##              Sub-Saharan Africa                  Western Europe
##                              35                              20
```

```
tableregion = table(Region)
```

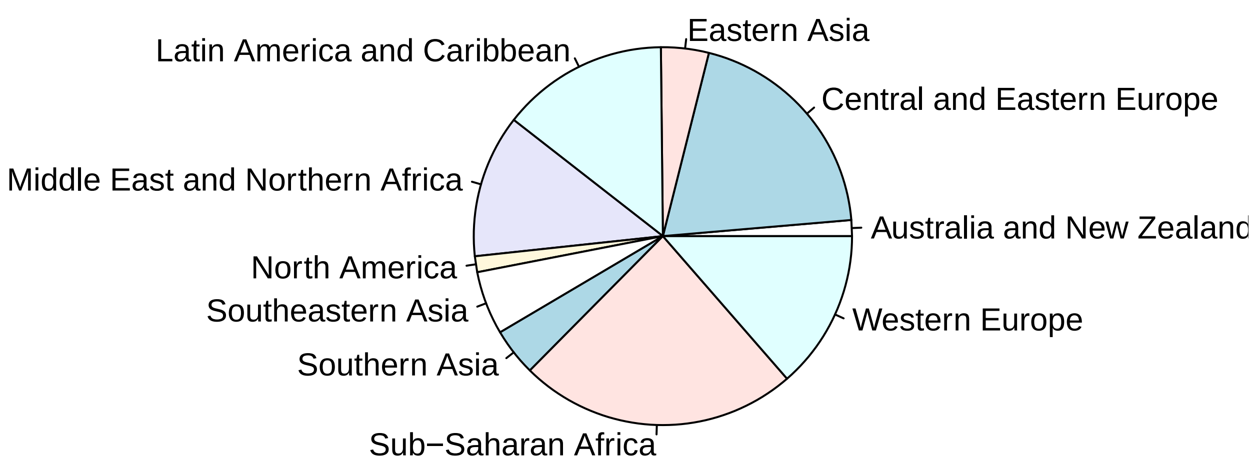
```
round(prop.table(tableregion),2)
```

```
## Region
```

```
##       Australia and New Zealand      Central and Eastern Europe
##                            0.01                            0.20
##                    Eastern Asia     Latin America and Caribbean
##                            0.04                            0.14
## Middle East and Northern Africa                   North America
##                            0.12                            0.01
##               Southeastern Asia                  Southern Asia
##                            0.05                            0.04
##              Sub-Saharan Africa                  Western Europe
##                            0.24                            0.14
```

La région la plus representée est l'afrique sub-saherienne Les régions les moins représentées sont l'australie et la nouvelle zelande ainsi que l'amérique du nord.

```
pie(tableregion)
```



Nous allons étudier les variable qualitative "Health"

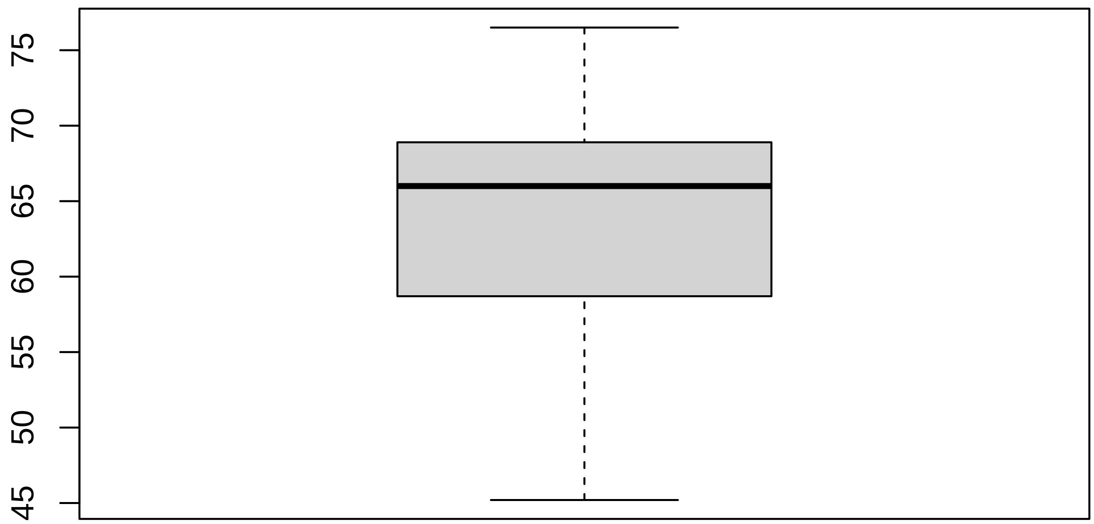
```
Health = data$Health
summary(Health, na.rm = TRUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   45.20   58.70   66.00   64.19   68.90   76.50       2
```

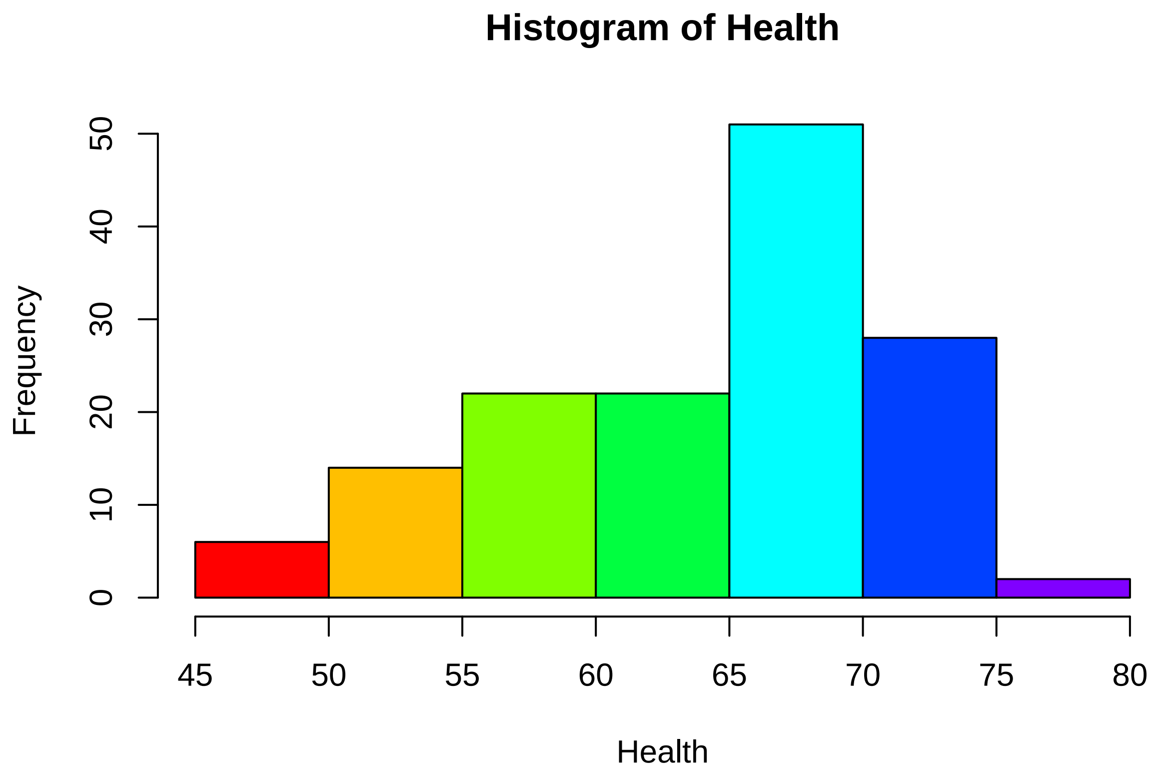
```r
mean(Health, na.rm = TRUE)
```

```
## [1] 64.19094
```

```r
boxplot(Health)
```




```r
hist(Health, col = rainbow(8))
```



Conclusion : Nous remarquons que l'esperance de vie moyenne se situe entre 65 ans et 70 ans et de loin.

Nous allons étudier les variable quantitatives "Happiness"

```
Happiness = data$Happiness
```

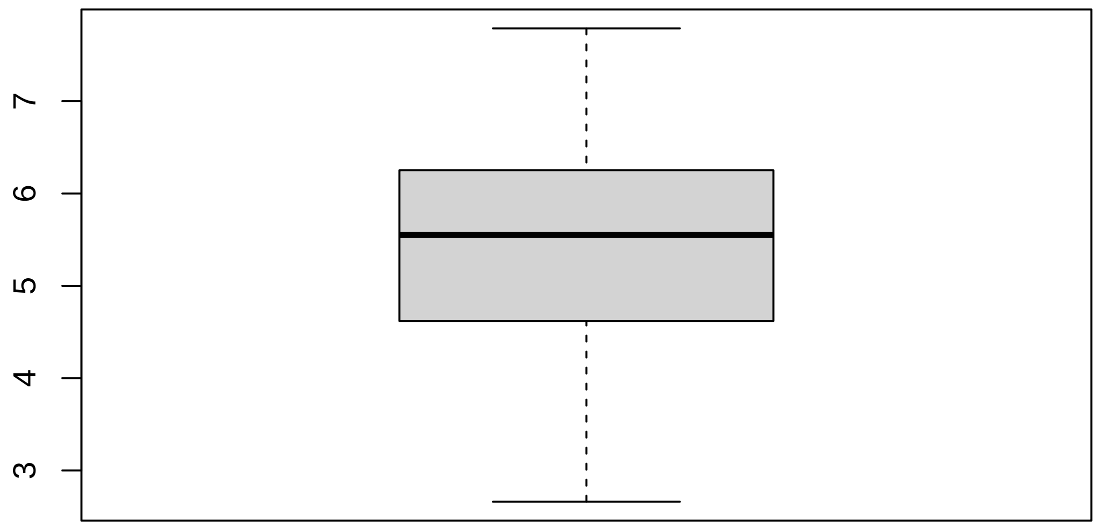
```
summary(Happiness) #sommaire
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.662   4.619   5.553   5.460   6.252   7.788
```

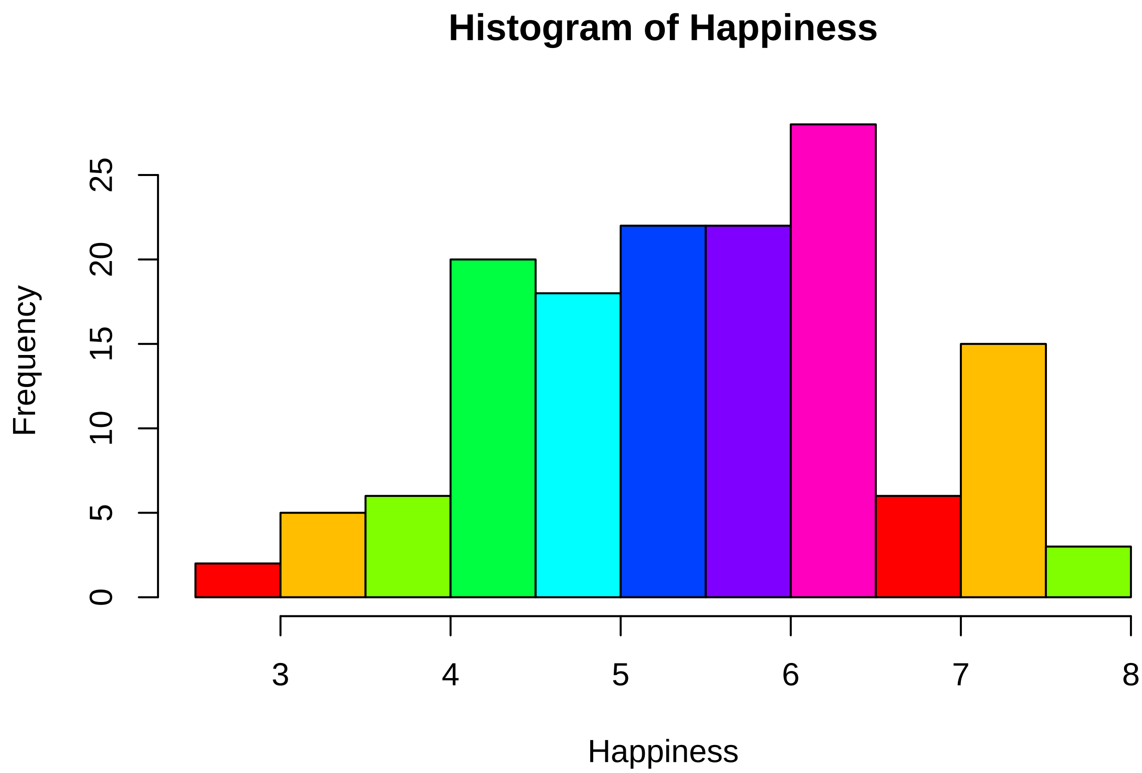
```
mean(Happiness)
```

```
## [1] 5.460421
```

```
boxplot(Happiness)
```




```
hist(Happiness, col=rainbow(8))
```



la moyenne des personnes heureuses se situe autour de 5,4.

Nous allons étudier la variable "Corruption"

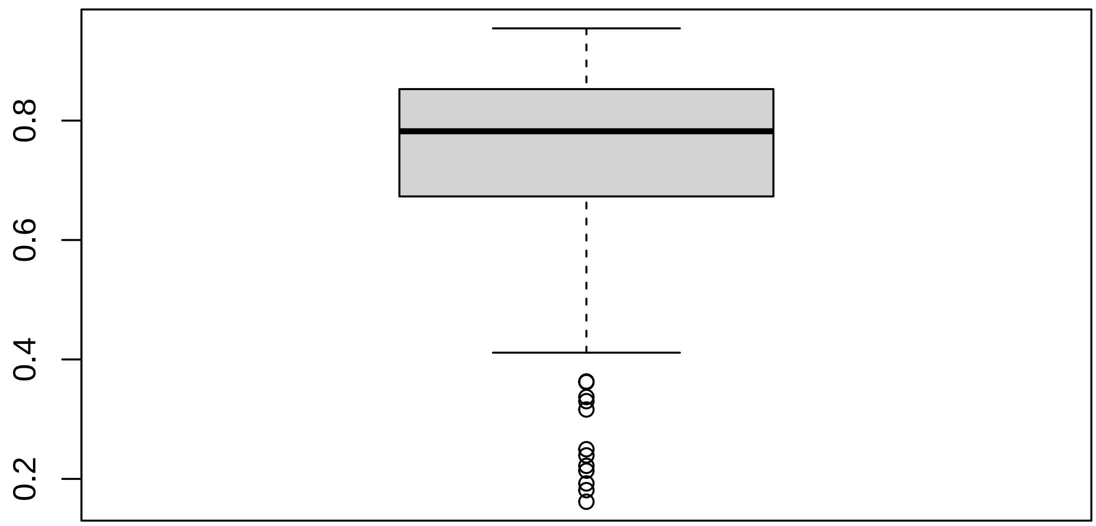
```
corruption = data$Corruption
summary(corruption)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.1618  0.6730  0.7821  0.7292  0.8521  0.9544      11
```

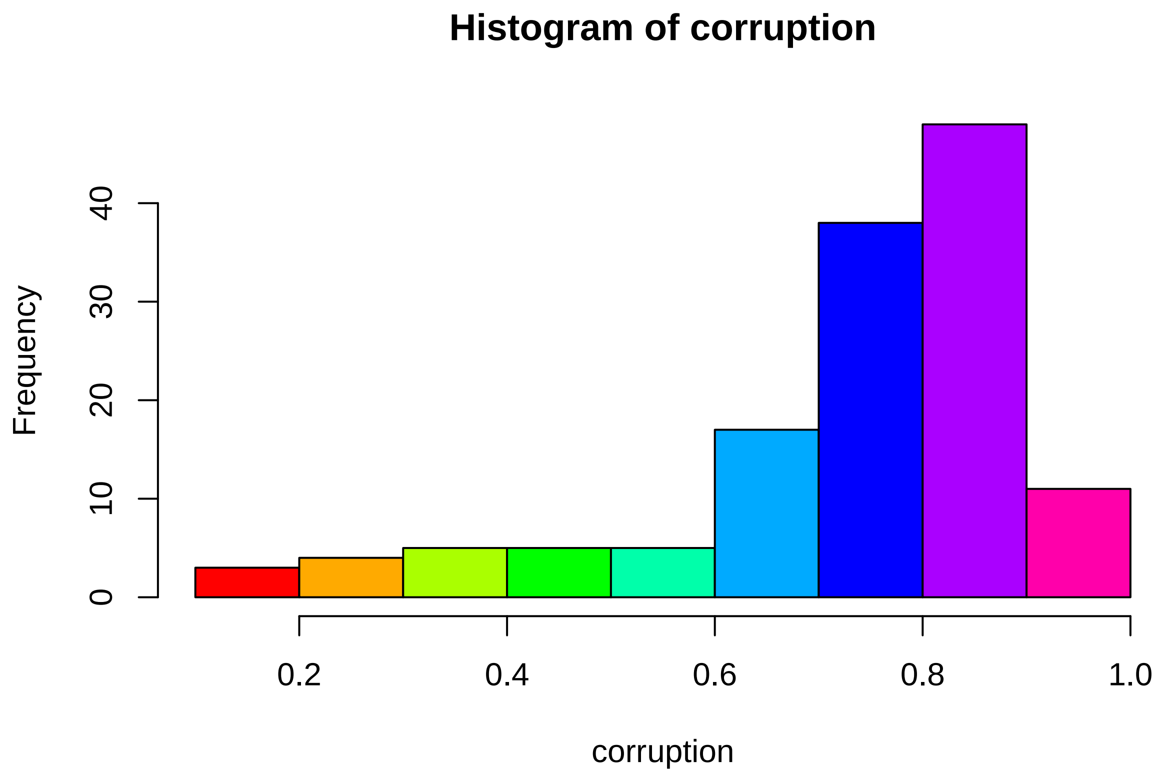
```
mean(corruption, na.rm = TRUE) #moyenne
```

```
## [1] 0.7291972
```

```
boxplot(corruption)
```



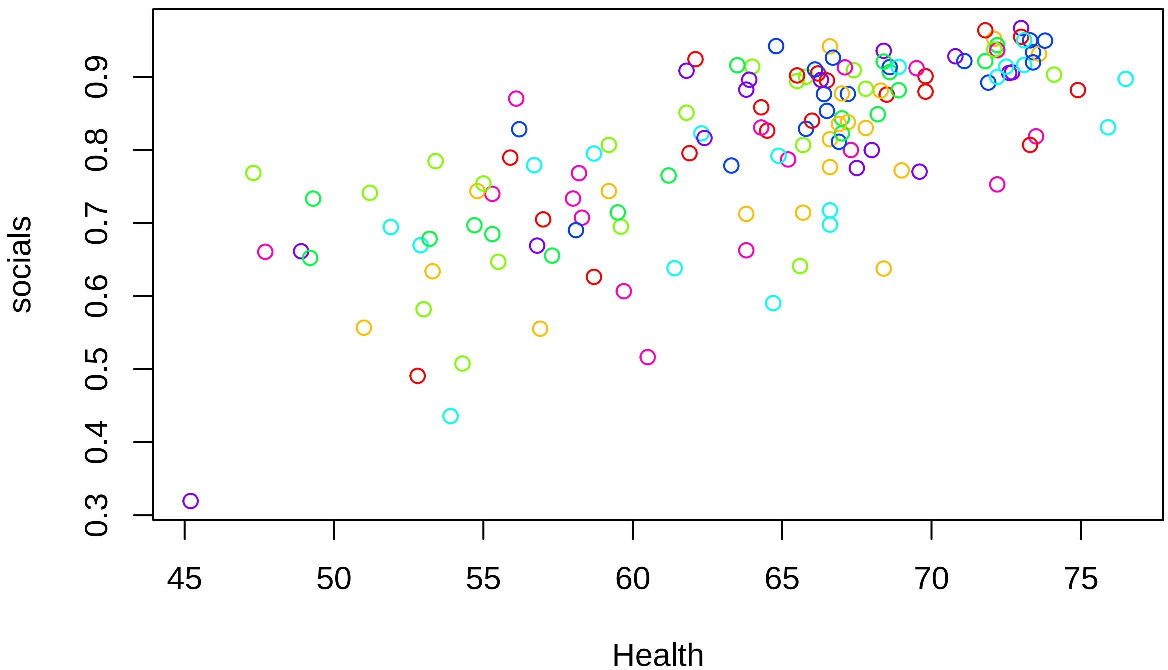

```
hist(corruption, col = rainbow(9))
```



Nous remarquons que la moyenne de la corruption est de 72%

Nous allons étudier l'association des variables "Health et"Social support".

```
socials = data$SocialSupport
table2 = table(socials,Health)
plot(socials~Health, col=rainbow(8))
```



Nous remarquons que plus le support social est important plus l'esperance de vie augmente.

```
cor.test(socials, Health, use="pairwise.complete.obs")
```

```
##
##  Pearson's product-moment correlation
##
## data:  socials and Health
## t = 12.541, df = 142, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6368830 0.7943199
## sample estimates:
##       cor
## 0.724936
```

Pvalue plus petit que 5% donc nous pouvons rejeter l hypothese de normalite donc les prix ne sont pas issues du hasard.

Nous allons étudier les associations entre la variable Nous allons étudier les variable catégorielle "Region" et la variable quantitative "Corruption"

```
table3 = table(Region, corruption)
summary(table3)
```

```
## Number of cases in table: 136
```

```
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 1224, df = 1215, p-value = 0.4224
##  Chi-squared approximation may be incorrect
```

Nous pouvons accepter l'hypothèse d'indépendance car p value est supérieur à 5%.

```
resultat = lm(corruption~factor(Region))
analyse<-anova(resultat)
analyse
```

```
## Analysis of Variance Table
##
## Response: corruption
##                 Df Sum Sq  Mean Sq F value    Pr(>F)
## factor(Region)   9 1.7225 0.191385  8.4286 8.774e-10 ***
## Residuals      126 2.8610 0.022707
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

les 3 étoiles associées au facteur region montrent que la région à un effet sur les valeurs de corruption.

Nous allons étudier les associations la variables catégorielle "Health" et quantitative "Happiness".

```
table4 = table(Health, Happiness)
chisq.test(Health, Happiness)
```

```
## Warning in chisq.test(Health, Happiness): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  Health and Happiness
## X-squared = 15805, df = 15696, p-value = 0.2684
```

Nous pouvons accepter l'hypothèse d'indépendance car p value est supérieur à 5%.

```
resultat2 = lm(Happiness~factor(Health))
analyse <-anova(resultat2)
analyse
```

```
## Analysis of Variance Table
##
## Response: Happiness
##                 Df  Sum Sq Mean Sq F value    Pr(>F)
## factor(Health) 109 173.764  1.5942  3.7812 1.324e-05 ***
## Residuals       35  14.756  0.4216
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

les 3 étoiles associées au facteur Health montrent que la variable Health à un effet sur les valeurs Happiness.

MODELE DE RÉGRESSION

Nous allonS étudier comment Hapiness dépend de la variable Région.

```r
table5 = table(Region, Happiness)
chisq.test(Happiness, Region)
```

```
## Warning in chisq.test(Happiness, Region): Chi-squared approximation may be
## incorrect
```

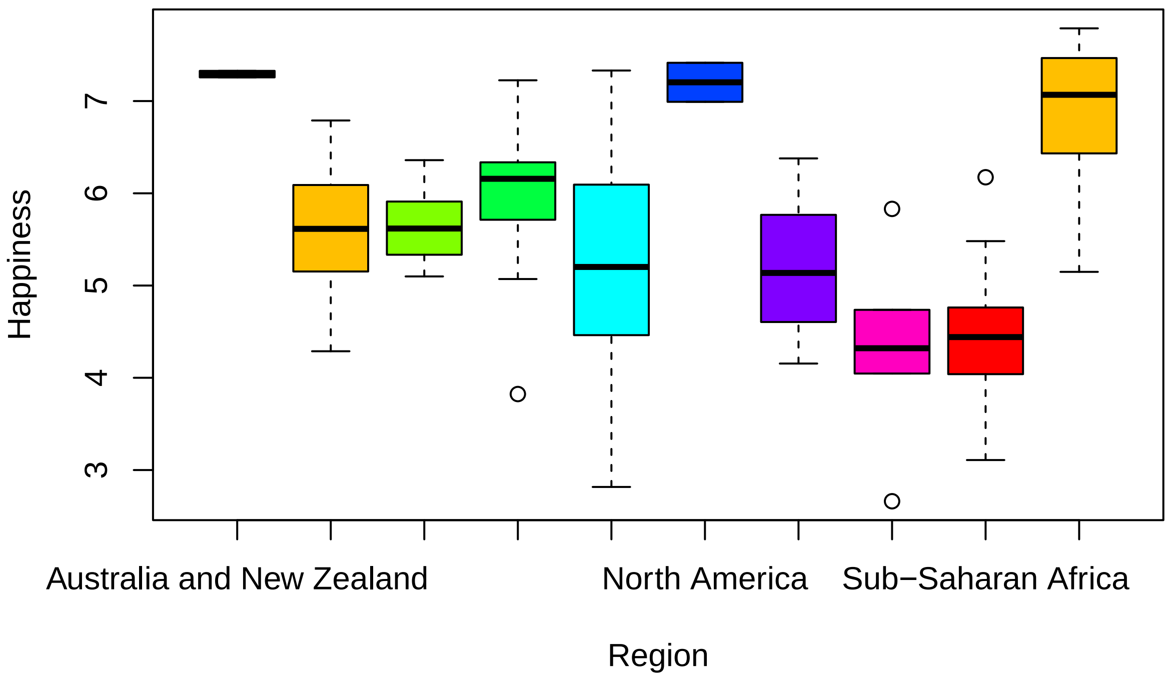
```
##
##  Pearson's Chi-squared test
##
## data:  Happiness and Region
## X-squared = 1323, df = 1314, p-value = 0.4254
```

Nous pouvons accepter l'hypothèse d'indépendance car p value est supérieur à 5%.

```r
tapply(Happiness, Region, "mean")
```

```
##       Australia and New Zealand      Central and Eastern Europe
##                        7.292110                        5.579568
##                    Eastern Asia     Latin America and Caribbean
##                        5.656567                        6.002161
## Middle East and Northern Africa                   North America
##                        5.149707                        7.203314
##              Southeastern Asia                   Southern Asia
##                        5.193576                        4.319351
##              Sub-Saharan Africa                  Western Europe
##                        4.397149                        6.891960
```

```r
boxplot(Happiness~Region, col= rainbow(8))
```




```
resultat3 = lm(Happiness~factor(Region))
analyse <-anova(resultat3)
analyse
```

```
## Analysis of Variance Table
##
## Response: Happiness
##                 Df  Sum Sq Mean Sq F value    Pr(>F)
## factor(Region)   9 110.266 12.2518  21.046 < 2.2e-16 ***
## Residuals      137  79.755  0.5822
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Nous observons que la région la plus heureuse est Australia and New Zealand et la moins heureuse est ma région Sub-saharan Africa. les 3 étoiles associées au facteur Region montrent que la variable Region à un effet sur les valeurs Happiness.

Nous allons étudier les variables "Freedom" et "Happiness"

```
Freedom = data$Freedom
table6 = table(Freedom, Happiness)
chisq.test(Freedom, Happiness)
```

```
## Warning in chisq.test(Freedom, Happiness): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  Freedom and Happiness
## X-squared = 21170, df = 21025, p-value = 0.2392
```

Nous pouvons accepter l'hypothèse d'indépendance car p value est supérieur à 5%.

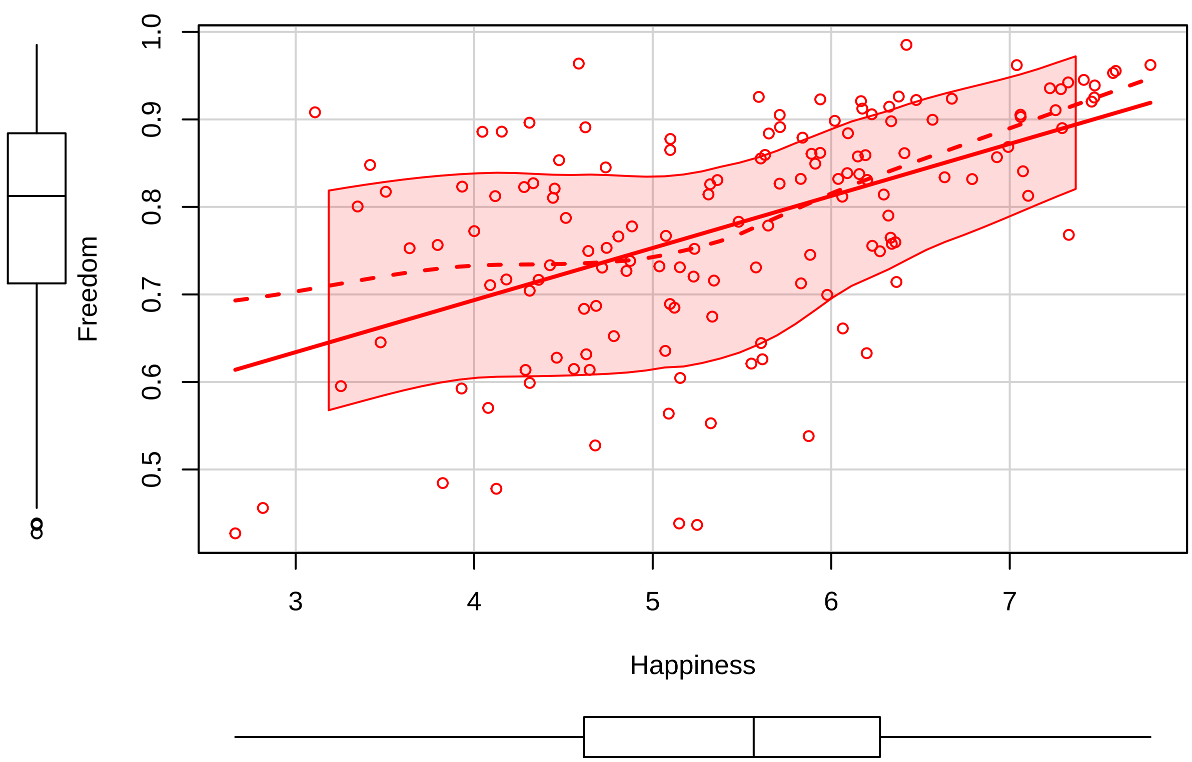
```r
library(car)
```

```
## Le chargement a nécessité le package : carData
```

```
##
## Attachement du package : 'car'
```

```
## L'objet suivant est masqué depuis 'package:dplyr':
##
##     recode
```

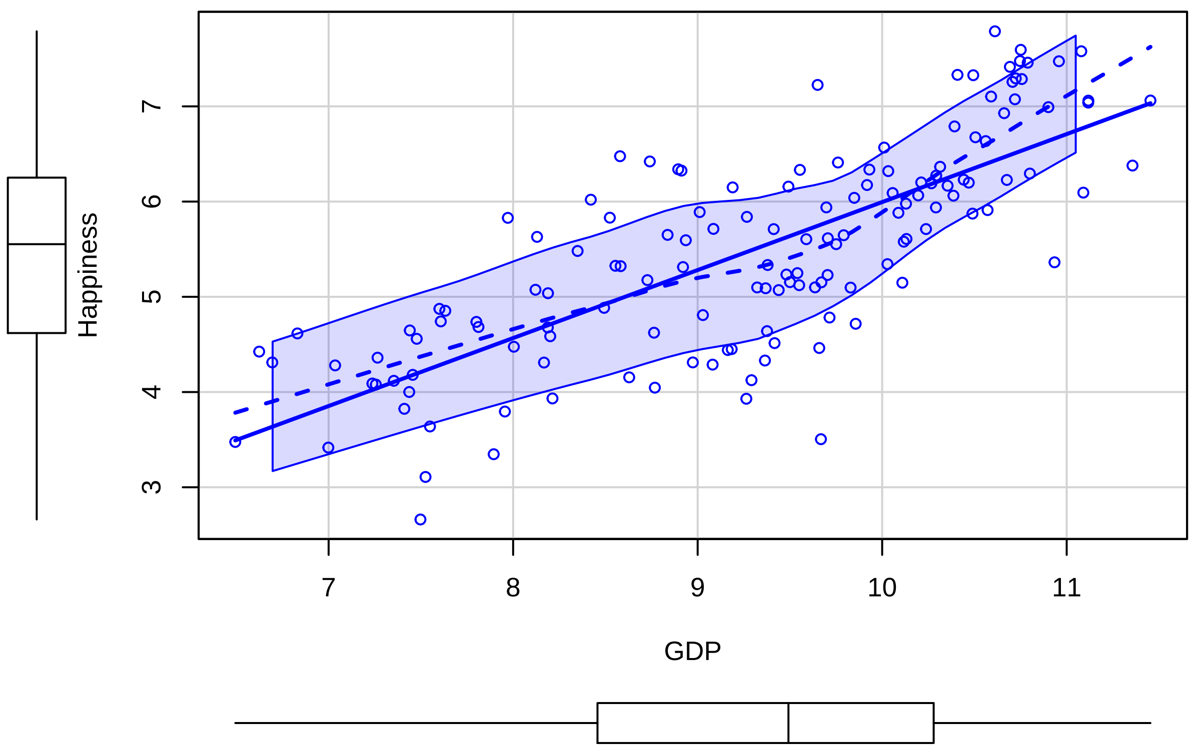
```r
scatterplot(Freedom~Happiness, col = rainbow(7))
```



Nous observons que plus la variable Freedom augmente plus la variable happiness est grande.

Nous allons prédire Happiness en fonction du GDP

```r
table7 = table(GDP, Happiness)
resultat4 <- lm(Happiness~log(GDP))
scatterplot(GDP, Happiness)
```




```r
summary(resultat4)
```

```
##
## Call:
## lm(formula = Happiness ~ log(GDP))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.28842 -0.48940  0.04027  0.50902  1.44357
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -8.4307     1.0131  -8.322 6.74e-14 ***
## log(GDP)      6.2693     0.4553  13.770  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7307 on 141 degrees of freedom
##   (4 observations effacées parce que manquantes)
## Multiple R-squared:  0.5735, Adjusted R-squared:  0.5705
## F-statistic: 189.6 on 1 and 141 DF,  p-value: < 2.2e-16
```

```r
predict(resultat4, interval = "prediction") #prediction
```

```
## Warning in predict.lm(resultat4, interval = "prediction"): predictions on current data refer to _futu

##          fit      lwr      upr
## 1   4.199397 2.738019 5.660774
## 2   5.600986 4.151392 7.050579
## 3   5.710019 4.260175 7.159863
## 4   5.909256 4.458517 7.359994
## 5   5.400532 3.950958 6.850106
## 6   6.432854 4.977086 7.888621
## 7   6.443089 4.987185 7.898993
## 8   5.794947 4.344790 7.245103
## 9   6.414741 4.959212 7.870270
## 10  4.735673 3.282079 6.189267
## 11  5.846619 4.396223 7.297016
## 12  6.406097 4.950680 7.861514
## 13  4.311077 2.851651 5.770502
## 14  5.229869 3.779861 6.679876
## 15  5.595892 4.146306 7.045478
## 16  5.793306 4.343157 7.243455
## 17  5.718907 4.269035 7.168779
## 18  5.896660 4.445995 7.347325
## 19  4.151092 2.688816 5.613368
## 20  4.761552 3.308232 6.214872
## 21  4.700120 3.246135 6.154105
## 22  6.424512 4.968855 7.880169
## 23  3.298460 1.815074 4.781846
## 24  4.182598 2.720912 5.644285
## 25  6.025526 4.574006 7.477047
## 26  5.772486 4.322422 7.222550
## 27  5.678060 4.228306 7.127813
## 28  4.980929 3.529548 6.432311
## 29  3.489193 2.011384 4.967003
## 30  5.781607 4.331506 7.231707
## 31  6.022851 4.571351 7.474351
## 32  6.242498 4.789007 7.695990
## 33  6.246040 4.792511 7.699569
## 34  6.458882 5.002764 7.915001
## 35  5.741610 4.291662 7.191558
## 36  5.527514 4.077994 6.977034
## 37  5.525482 4.075962 6.975001
## 38  5.270440 3.820573 6.720307
## 39  6.184971 4.732067 7.637875
## 40  4.164247 2.702219 5.626275
## 41  6.376736 4.921693 7.831779
## 42  6.347115 4.892436 7.801794
## 43  5.823489 4.373205 7.273773
## 44  4.077665 2.613962 5.541368
## 45  5.471528 4.022013 6.921044
## 46  6.440413 4.984545 7.896282
## 47  4.873880 3.421637 6.326122
## 48  6.073079 4.621184 7.524975
```

```
## 49  5.283009 3.833181 6.732837
## 50  4.284613 2.824741 5.744485
## 51  4.125970 2.663214 5.588725
## 52  4.927473 3.475682 6.379264
## 53  6.564674 5.107033 8.022316
## 54  6.126099 4.673748 7.578450
## 55  6.456384 5.000300 7.912468
## 56  5.180721 3.730511 6.630930
## 57  5.565123 4.115575 7.014671
## 58  5.914234 4.463466 7.365002
## 59  5.787395 4.337270 7.237519
## 60  6.668932 5.209635 8.128228
## 61  6.255694 4.802061 7.709326
## 62  6.292393 4.838355 7.746432
## 63  4.752248 3.298830 6.205665
## 64  5.352026 3.902371 6.801682
## 65  6.353178 4.898426 7.807931
## 66  5.364130 3.914498 6.813761
## 67  6.059833 4.608045 7.511621
## 68  4.609049 3.153981 6.064117
## 69  5.475073 4.025559 6.924588
## 70  6.653593 5.194549 8.112636
## 71  4.706638 3.252726 6.160550
## 72  5.177451 3.727227 6.627675
## 73  6.085298 4.633301 7.537295
## 74  5.683645 4.233877 7.133413
## 75  4.570896 3.115340 6.026452
## 76  3.422404 1.942697 4.902111
## 77  5.872747 4.422215 7.323279
## 78  6.185859 4.732946 7.638772
## 79  6.855907 5.393265 8.318550
## 80  5.670854 4.221119 7.120589
## 81  3.993837 2.528414 5.459260
## 82  3.767439 2.296887 5.237991
## 83  4.290932 2.831167 5.750697
## 84  6.313950 4.859664 7.768236
## 85  4.751530 3.298105 6.204955
## 86  5.953221 4.502209 7.404232
## 87  5.852896 4.402468 7.303324
## 88  5.026011 3.574944 6.477079
## 89  5.603102 4.153505 7.052699
## 90  5.817492 4.367236 7.267748
## 91  5.288783 3.838971 6.738594
## 92  3.800327 2.330564 5.270091
## 93  5.080400 3.629673 6.531127
## 94  5.457124 4.007602 6.906645
## 95  4.447884 2.990613 5.905154
## 96  6.480786 5.024365 7.937206
## 97  6.306918 4.852714 7.761123
## 98  5.044318 3.593370 6.495267
## 99  3.615409 2.141022 5.089797
## 100 5.046650 3.595716 6.497583
## 101 6.647252 5.188312 8.106192
## 102 5.003713 3.552493 6.454932
```

```
## 104 6.011635 4.560218 7.463052
## 105 5.403622 3.954052 6.853192
## 106 5.625044 4.175409 7.074678
## 107 5.299413 3.849632 6.749193
## 108 6.136088 4.683646 7.588529
## 109 6.152108 4.699518 7.604697
## 110 6.040322 4.588688 7.491956
## 111 6.077893 4.625958 7.529828
## 112 4.222316 2.761353 5.683278
## 113 6.487609 5.031093 7.944126
## 114 4.456964 2.999827 5.914100
## 115 5.716363 4.266499 7.166227
## 116 3.977723 2.511958 5.443488
## 117 6.802414 5.340778 8.264050
## 118 6.198724 4.745684 7.651764
## 119 6.223280 4.769990 7.676570
## 120 5.628202 4.178562 7.077842
## 121 6.304471 4.850295 7.758648
## 123 6.276038 4.822183 7.729893
## 124 5.593326 4.143744 7.042909
## 125 6.462204 5.006041 7.918368
## 126 6.578357 5.120507 8.036207
## 128 4.583431 3.128037 6.038825
## 129 4.522898 3.066698 5.979098
## 130 5.812288 4.362056 7.262521
## 131 4.001952 2.536699 5.467204
## 132 6.169929 4.717171 7.622688
## 133 5.544371 4.094841 6.993902
## 134 6.086912 4.634902 7.538923
## 135 5.816694 4.366441 7.266946
## 136 4.148418 2.686092 5.610745
## 137 5.326095 3.876382 6.775807
## 138 6.668581 5.209291 8.127872
## 139 6.364479 4.909588 7.819370
## 140 6.545620 5.088265 8.002976
## 141 5.961222 4.510158 7.412286
## 142 5.161114 3.710815 6.611413
## 143 5.643069 4.193399 7.092739
## 144 5.151730 3.701386 6.602074
## 146 4.770755 3.317530 6.223980
## 147 4.242508 2.781904 5.703111
```

Nous allons effetuer une régression multiple entre "GDP", "hapiness" et "freedom"

```r
model <- lm(Happiness~Freedom+GDP)
summary(model)
```

```
##
## Call:
## lm(formula = Happiness ~ Freedom + GDP)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.30473 -0.43231  0.07818  0.48686  1.18467
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.40290    0.48199  -4.985 1.81e-06 ***
## Freedom      2.61068    0.46401   5.626 9.77e-08 ***
## GDP          0.62874    0.04801  13.095  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6508 on 139 degrees of freedom
##   (5 observations effacées parce que manquantes)
## Multiple R-squared:  0.6663, Adjusted R-squared:  0.6615
## F-statistic: 138.7 on 2 and 139 DF,  p-value: < 2.2e-16
```

Nous observons que Freedom et GDP sont très significatif

```
summary(model)$coefficient
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -2.4028970  0.4819863 -4.985405 1.811737e-06
## Freedom      2.6106754  0.4640065  5.626377 9.770679e-08
## GDP          0.6287376  0.0480122 13.095371 4.993847e-26
```

```
confint(model) #intervalle de confiance
```

```
##                 2.5 %     97.5 %
## (Intercept) -3.355870 -1.4499244
## Freedom      1.693252  3.5280986
## GDP          0.533809  0.7236663
```

Créeons une carte du monde avec la variable hapiness

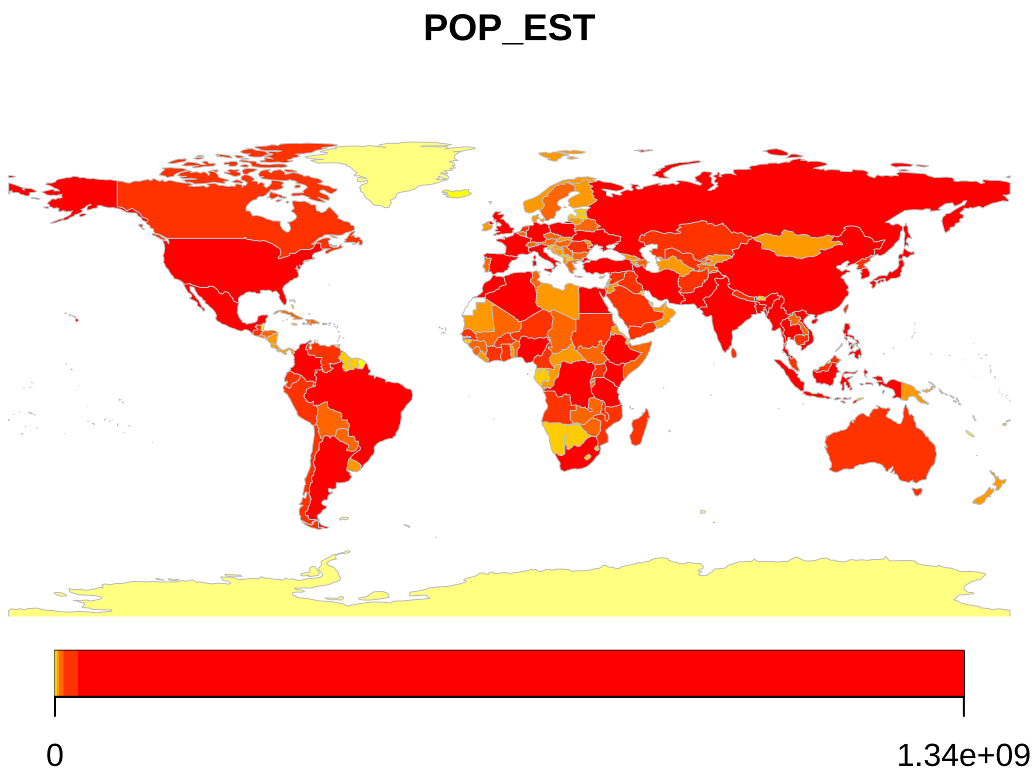
```
library(rworldmap)
```

```
## Le chargement a nécessité le package : sp

## ### Welcome to rworldmap ###

## For a short introduction type :   vignette('rworldmap')
```

```
mapCountryData(mapRegion = Happiness)
```

```
## using example data because no file specified in rwmCheckAndLoadInput

## Warning in if (mapRegion != "world") {: la condition a une longueur > 1 et seul
## le premier élément est utilisé

## Warning in if (mapRegion %in% map@data$ADMIN) {: la condition a une longueur > 1
## et seul le premier élément est utilisé
```

```
## Warning in if (mapRegion == "world") {: la condition a une longueur > 1 et seul
## le premier élément est utilisé

## Warning in if (mapRegion == "eurasia" | mapRegion == "Eurasia") {: la condition
## a une longueur > 1 et seul le premier élément est utilisé

## Warning in if (mapRegion == "africa" | mapRegion == "Africa") {: la condition a
## une longueur > 1 et seul le premier élément est utilisé

## Warning in if (mapRegion == "latin america" | mapRegion == "Latin America") {:
## la condition a une longueur > 1 et seul le premier élément est utilisé

## Warning in if (mapRegion == "north america" | mapRegion == "North America") {:
## la condition a une longueur > 1 et seul le premier élément est utilisé

## Warning in if (mapRegion == "uk" | mapRegion == "UK") {: la condition a une
## longueur > 1 et seul le premier élément est utilisé

## Warning in if (mapRegion == "oceania" | mapRegion == "Oceania") {: la condition
## a une longueur > 1 et seul le premier élément est utilisé

## Warning in if (mapRegion == "asia" | mapRegion == "Asia") {: la condition a une
## longueur > 1 et seul le premier élément est utilisé

## Warning in if (mapRegion == "europe" | mapRegion == "Europe") {: la condition a
## une longueur > 1 et seul le premier élément est utilisé

## Warning in setMapExtents(mapRegion): The mapRegion you
## specified(2.661718130111694.639548301696785.248912334442146.039330005645754.287736415863047.257037639
## needs to be a country name from getMap()$NAME or one of : (eurasia africa latin
## america north america uk oceania asia ).Plotting whole world

## Warning in if (aspect == "variable" & mapRegion != "world") aspect <-
## ifelse(is.na(proj4string(mapToPlot)) || : la condition a une longueur > 1 et
## seul le premier élément est utilisé
```




```
hist((Happiness))
```

LOI DE PROBALITÉ : BONUS

La variable Hapiness semble suivre une loi poisson. La variable Hapiness semble suivre une loi uniforme

```
library(e1071)
skewness(Happiness) #coefficient de skewness
```

```
## [1] -0.06702301
```

L'asymètrie est de -0.06702301 elle est négative donc elle penche vers la droite

```
kurtosis(Happiness) #coefficient de kurtosis
```

```
## [1] -0.6461513
```

l'excès de kutosis de Happiness est de -0.6461513 la distribution est platykurtique. Ce qui est cohérent car son histogramme n'est pas en forme de cloche.

MACHINE LEARNING : BONUS

```
library(ggpubr)
```

```
## Le chargement a nécessité le package : ggplot2
```

```r
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
# Calculer k-means avec k = 27
set.seed(123)
cluster <-kmeans(scale(Happiness),27,nstart = 25)
cluster
```

```
## K-means clustering with 27 clusters of sizes 2, 1, 7, 3, 6, 5, 6, 8, 2, 5, 3, 12, 6, 7, 5, 6, 10, 3,
##
## Cluster means:
##          [,1]
## 1  -2.3853054
## 2   2.0404535
## 3  -1.0057415
## 4  -1.3202632
## 5   1.6009321
## 6  -0.5435903
## 7   0.5270514
## 8   0.1074502
## 9   1.2258832
## 10 -0.6566386
## 11  0.8551710
## 12 -0.3072614
## 13 -0.8757071
## 14 -0.7419077
## 15  0.1982801
## 16  1.3977693
## 17  0.7643406
## 18  0.4306314
## 19 -0.1395106
## 20  0.3758171
## 21 -1.8489111
## 22  1.7872156
## 23 -1.4970838
## 24  0.3267648
## 25  1.0218446
## 26 -1.1810495
## 27  0.6382429
##
## Clustering vector:
##   [1]  1 14 19  7  3  5  5 12 27  3  8  9  6 15 12 21 17 12 14 14 12 22 21 14 17
##  [26] 12 27  6  3  5 19  7  9 22  8 24  4 17 18 26  2 25  6 26 13 16  8 12 17  6
##  [51] 23  7 19  7 22 26 12 10 13 16  5 27 12 20 20  6 20 13 27  7  8 14 18 12 23
##  [76] 13 15 17 16 19 26 21 10 25 10 27 11 19 19  8 19  3 26 13 10 22  5 11 14 19
## [101] 22 24 14 25 15 15  8 27 15  7  8 21 17 10 12 26 17 17 27 13 20  1 27  3  5
## [126] 22 17 24 21 18  3 27 26  8 19  4  3 16 16 16 17 11 12 12 21  4 23
##
## Within cluster sum of squares by cluster:
##  [1] 9.218241e-03 0.000000e+00 3.351225e-03 2.475571e-03 6.391180e-03
##  [6] 5.827246e-03 3.138244e-03 1.202394e-02 7.398987e-03 2.693726e-03
## [11] 1.921559e-03 1.419549e-02 3.698552e-03 4.491242e-03 3.691198e-03
```

```
## [16] 5.388466e-03 7.069316e-03 7.860722e-04 1.551256e-02 5.708248e-04
## [21] 8.604187e-02 1.934105e-02 1.534101e-02 4.692838e-05 4.575693e-03
## [26] 9.680223e-03 5.160868e-03
##  (between_SS / total_SS =  99.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Le nombre optimal de cluster est 27

#erreur sur ce code pour la visualisattion :

fviz_cluster(cluster, data = Happiness[, 4], palette = c("#2E9FDF", "#00AFBB", "#E7B800"), geom = "point", ellipse.type = "convex", ggtheme = theme_bw())

Merci pour ce projet, je n'ai pas pu aller plus loin

Gary DJIGO