Initiation à la statistique avec R, code et compléments chapitre 6

Frédéric Bertrand et Myriam Maumy-Bertrand

11 décembre 2018

#Chapitre 6  
#page 260  
require(BioStatR)

## Loading required package: BioStatR

glycine.blanche<-subset(Mesures,subset=(Mesures$espece=="glycine blanche"))  
mean(glycine.blanche$taille)

## [1] 14.77222

#page 261  
var(glycine.blanche$taille)

## [1] 10.91412

#page 262  
(var(glycine.blanche$taille))\*((length(glycine.blanche$taille)-1)/  
 length(glycine.blanche$taille))

## [1] 10.71201

glycine.blanche<-subset(Mesures5,subset=(Mesures5$espece=="glycine blanche"))  
  
#page 263  
effectif.cumule<-cumsum(table(glycine.blanche$graines))  
effectif.cumule

## 1 2 3 4 5 6 7   
## 7 26 37 47 52 53 54

37/54

## [1] 0.6851852

#page 264  
qnorm(0.975)

## [1] 1.959964

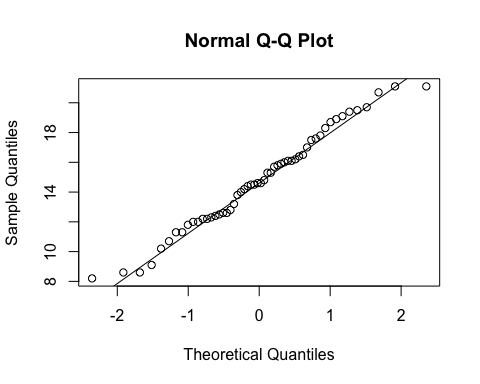
#page 265  
glycine.blanche<-subset(Mesures,subset=(Mesures$espece=="glycine blanche"))  
shapiro.test(glycine.blanche$taille)

##   
## Shapiro-Wilk normality test  
##   
## data: glycine.blanche$taille  
## W = 0.97977, p-value = 0.4906

#page 239  
length((glycine.blanche$taille))

## [1] 54

qqnorm(glycine.blanche$taille)  
qqline(glycine.blanche$taille)

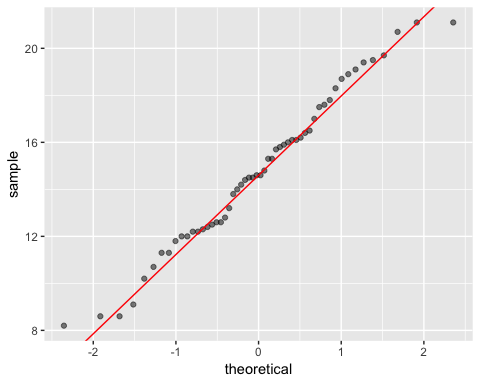


pdf("figch61A.pdf")  
qqnorm(glycine.blanche$taille)  
qqline(glycine.blanche$taille)  
dev.off()

## quartz\_off\_screen   
## 2

#argument: un dataframe et le nom d'une variable  
gg\_qqplot(glycine.blanche,"taille")

## 1st quartile : -0.674489750196082   
## 3rd quartile : 0.674489750196082   
## Intercept : 14.6   
## Slope : 3.37292004710024



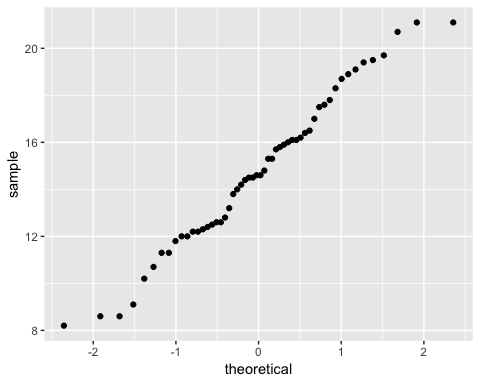
library(ggplot2)  
pdf("figch61B.pdf")  
gg\_qqplot(glycine.blanche,"taille")

## 1st quartile : -0.674489750196082   
## 3rd quartile : 0.674489750196082   
## Intercept : 14.6   
## Slope : 3.37292004710024

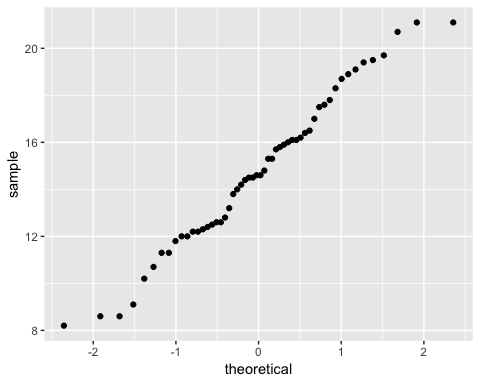
dev.off()

## quartz\_off\_screen   
## 2

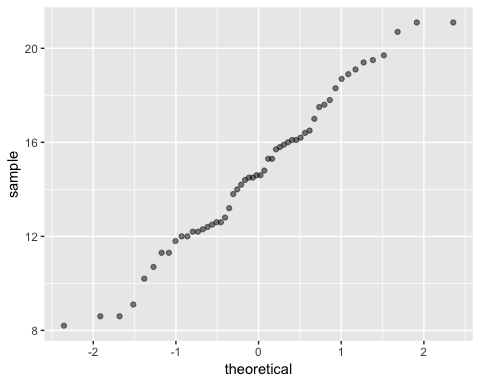
#En plus : autre manière de construire le diagramme quantile-quantile  
#basé sur la loi normale centrée et réduite  
ggplot(glycine.blanche, aes(sample = taille)) + stat\_qq()



ggplot(glycine.blanche, aes(sample = taille)) + geom\_point(stat = "qq")



#ou avec le fonction précédente et l'option qq.line=FALSE  
gg\_qqplot(glycine.blanche,"taille",qq.line=FALSE)

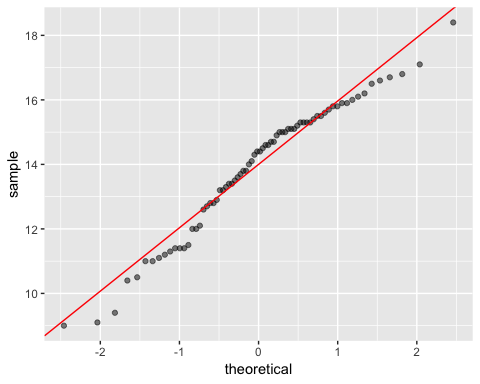


#page 267  
lauriers.roses<-subset(Mesures,subset=(Mesures$espece=="laurier rose"))  
shapiro.test(lauriers.roses$taille)

##   
## Shapiro-Wilk normality test  
##   
## data: lauriers.roses$taille  
## W = 0.96457, p-value = 0.04021

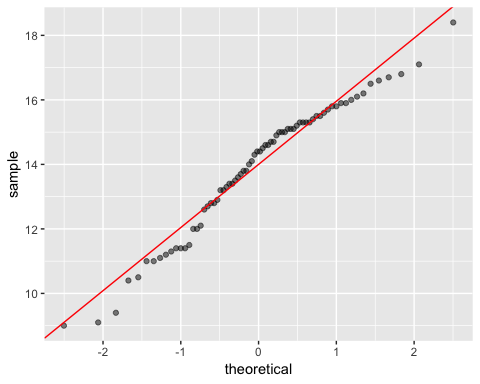
#pas issu d'une loi normal au risque alpha=5%  
gg\_qqplot(lauriers.roses,"taille")

## 1st quartile : -0.674489750196082   
## 3rd quartile : 0.674489750196082   
## Intercept : 14   
## Slope : 1.96444793951992

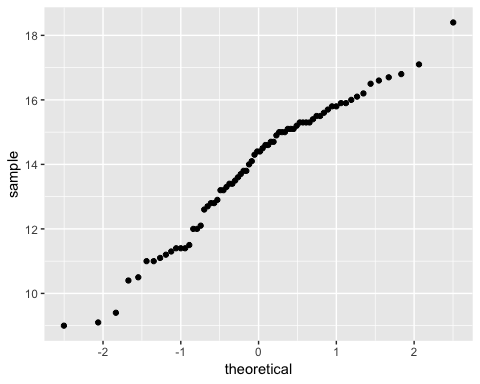


#page 268  
#essayons un qqplot avec une autre loi, ici Student (car dist = qt) dont on estime les ddl  
if(!("MASS" %in% rownames(installed.packages()))){install.packages("MASS")}  
library(MASS)  
params <- as.list(fitdistr(lauriers.roses$taille, "t")$estimate)  
gg\_qqplot(lauriers.roses,"taille",qt,list(df=params$df))

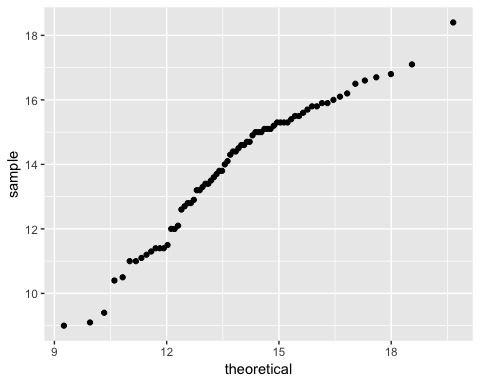
## 1st quartile : -0.676873231220387   
## 3rd quartile : 0.676873231220387   
## Intercept : 14   
## Slope : 1.95753050776001



#En plus : autre manière de construire le diagramme quantile-quantile  
#basé sur la loi de student  
ggplot(lauriers.roses, aes(sample = taille)) + stat\_qq(distribution = stats::qt,  
 dparams = list(df=params[[3]]))

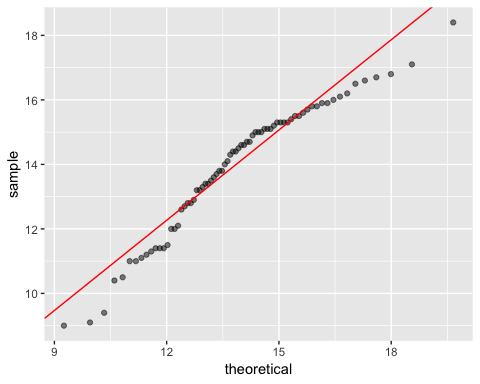


#En plus essayons un qqplot avec une loi gamma  
params <- as.list(fitdistr(lauriers.roses$taille,"gamma")$estimate)  
ggplot(lauriers.roses, aes(sample = taille)) + stat\_qq(distribution = stats::qgamma,   
 dparams = params)

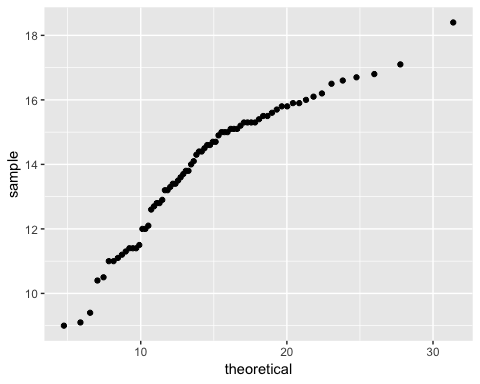


#avec la droite  
gg\_qqplot(lauriers.roses,"taille",qgamma,params)

## 1st quartile : 12.4360988717797   
## 3rd quartile : 15.2778275963289   
## Intercept : 1.07795284855338   
## Slope : 0.932530954523277

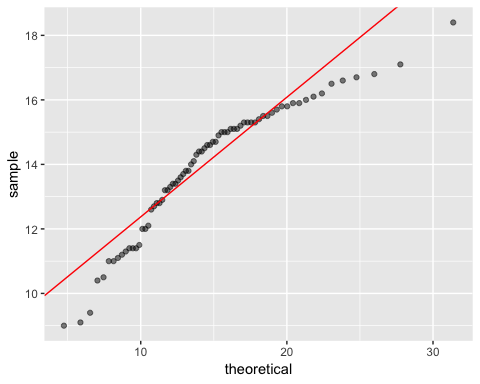


#essayons un qqplot avec une loi du chi-deux  
params <- list(df=fitdistr(lauriers.roses$taille,"chi-squared",start=list(df=5),  
 method="Brent",lower=1,upper=40)$estimate)  
ggplot(lauriers.roses, aes(sample = taille)) + stat\_qq(distribution = qchisq,  
 dparams = params)



#avec la droite  
gg\_qqplot(lauriers.roses,"taille",qchisq,params)

## 1st quartile : 10.8119220538675   
## 3rd quartile : 17.9552427488241   
## Intercept : 8.66403721864573   
## Slope : 0.370975924666377



if(!("gridExtra" %in% rownames(installed.packages()))){install.packages("gridExtra")}  
library(gridExtra)  
params <- as.list(fitdistr(lauriers.roses$taille, "t")$estimate)  
p1=gg\_qqplot(lauriers.roses,"taille",qt,list(df=params$df))

## 1st quartile : -0.676873231220387   
## 3rd quartile : 0.676873231220387   
## Intercept : 14   
## Slope : 1.95753050776001

params <- list(df=fitdistr(lauriers.roses$taille,"chi-squared",start=list(df=5),  
 method="Brent",lower=1,upper=40)$estimate)  
p2=gg\_qqplot(lauriers.roses,"taille",qchisq,params)

## 1st quartile : 10.8119220538675   
## 3rd quartile : 17.9552427488241   
## Intercept : 8.66403721864573   
## Slope : 0.370975924666377

pdf("fig61Cggplot")  
grid.arrange(p1, p2, nrow = 1)  
dev.off()

## quartz\_off\_screen   
## 2

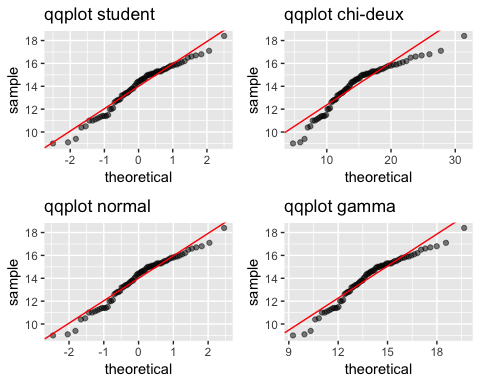
#En plus : graphique avec les quatre qqplots  
p0=gg\_qqplot(lauriers.roses,"taille")+ggtitle("qqplot normal")

## 1st quartile : -0.674489750196082   
## 3rd quartile : 0.674489750196082   
## Intercept : 14   
## Slope : 1.96444793951992

params <- as.list(fitdistr(lauriers.roses$taille,"gamma")$estimate)  
p3=gg\_qqplot(lauriers.roses,"taille",qgamma,params)+ggtitle("qqplot gamma")

## 1st quartile : 12.4360988717797   
## 3rd quartile : 15.2778275963289   
## Intercept : 1.07795284855338   
## Slope : 0.932530954523277

grid.arrange(p1+ggtitle("qqplot student"), p2+ggtitle("qqplot chi-deux"), p0, p3, nrow=2)



(moyenne<-mean(glycine.blanche$taille))

## [1] 14.77222

#page 269  
(quantile<-qt(0.975,53))

## [1] 2.005746

(ecart.type<-sd(glycine.blanche$taille))

## [1] 3.303652

moyenne-quantile\*(ecart.type/sqrt(length(glycine.blanche$taille)))

## [1] 13.8705

moyenne+quantile\*(ecart.type/sqrt(length(glycine.blanche$taille)))

## [1] 15.67395

t.test(glycine.blanche$taille)

##   
## One Sample t-test  
##   
## data: glycine.blanche$taille  
## t = 32.859, df = 53, p-value < 2.2e-16  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 13.87050 15.67395  
## sample estimates:  
## mean of x   
## 14.77222

#page 271  
glycine.blanche<-subset(Mesures,subset=(Mesures$espece=="glycine blanche"))  
shapiro.test(glycine.blanche$taille)

##   
## Shapiro-Wilk normality test  
##   
## data: glycine.blanche$taille  
## W = 0.97977, p-value = 0.4906

length(glycine.blanche$taille)

## [1] 54

#page 272  
(variance<-var(glycine.blanche$taille))

## [1] 10.91412

qchisq(0.975,53)

## [1] 75.00186

qchisq(0.025,53)

## [1] 34.77633

((length(glycine.blanche$taille)-1)\*variance)/qchisq(0.975,53)

## [1] 7.712453

((length(glycine.blanche$taille)-1)\*variance)/qchisq(0.025,53)

## [1] 16.63339

#page 273  
binom.test(x=5,n=10,p=0.5,alternative=c("two.sided","less","greater"),conf.level=0.95)

##   
## Exact binomial test  
##   
## data: 5 and 10  
## number of successes = 5, number of trials = 10, p-value = 1  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.187086 0.812914  
## sample estimates:  
## probability of success   
## 0.5

#page 274  
binom.ci(x=5,n=10,conf.level=0.95,method="exact")

## PointEst Lower Upper  
## 0.5 0.187086 0.812914

prop.test(x=5,n=10,p=0.5,alternative=c("two.sided","less","greater"),conf.level=0.95)

##   
## 1-sample proportions test without continuity correction  
##   
## data: 5 out of 10, null probability 0.5  
## X-squared = 0, df = 1, p-value = 1  
## alternative hypothesis: true p is not equal to 0.5  
## 95 percent confidence interval:  
## 0.2365931 0.7634069  
## sample estimates:  
## p   
## 0.5

#page 275  
binom.ci(x=5,n=10,conf.level=0.95,method="Wilson")

## PointEst Lower Upper  
## 0.5 0.2365931 0.7634069

binom.ci(x=5,n=10,conf.level=0.95,method="Wald")

## PointEst Lower Upper  
## 0.5 0.1901025 0.8098975

#page 283  
#Exercice 6.1  
#1)  
toxine<-c(1.2,0.8,0.6,1.1,1.2,0.9,1.5,0.9,1.0)  
str(toxine)

## num [1:9] 1.2 0.8 0.6 1.1 1.2 0.9 1.5 0.9 1

mean(toxine)

## [1] 1.022222

sd(toxine)

## [1] 0.2635231

#2)  
t.test(toxine)

##   
## One Sample t-test  
##   
## data: toxine  
## t = 11.637, df = 8, p-value = 2.708e-06  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 0.8196604 1.2247840  
## sample estimates:  
## mean of x   
## 1.022222

#page 284  
#4)  
variance<-var(toxine)  
((length(toxine)-1)\*variance)/qchisq(0.975,8)

## [1] 0.03168349

((length(toxine)-1)\*variance)/qchisq(0.025,8)

## [1] 0.2548735

sqrt(((length(toxine)-1)\*variance)/qchisq(0.975,8))

## [1] 0.1779986

#page 285  
sqrt(((length(toxine)-1)\*variance)/qchisq(0.025,8))

## [1] 0.50485

#Exercice 6.3  
#page 286  
#1)  
lambda\_n<-(1\*11+2\*41+3\*27+4\*16+5\*10+6\*2+7\*3)/110  
lambda\_n

## [1] 2.918182

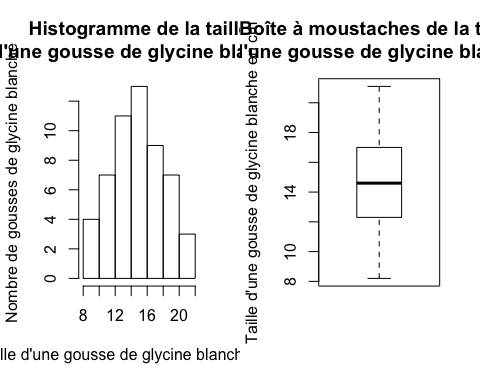
#2)  
echantillon<-rep(0:8,c(0,11,41,27,16,10,2,3,0))  
echantillon

## [1] 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [36] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [71] 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5  
## [106] 6 6 7 7 7

poi.ci(echantillon)

## PointEst 95% LCI 95% UCI  
## [1,] 2.918182 2.607644 3.255524

#Problème 6.1  
#page 287  
library(BioStatR)  
#1)  
glycine<-subset(Mesures,subset=(Mesures$espece=="glycine blanche"))  
#2)  
layout(t(1:2))  
histo<-hist(glycine$taille,ylab="Nombre de gousses de glycine blanche",  
 main="Histogramme de la taille\n d'une gousse de glycine blanche",  
 xlab="Taille d'une gousse de glycine blanche en cm")  
boxplot(glycine$taille,ylab="Taille d'une gousse de glycine blanche en cm",  
 main="Boîte à moustaches de la taille\n d'une gousse de glycine blanche")



pdf("chap5fig62.pdf")  
layout(t(1:2))  
histo<-hist(glycine$taille,ylab="Nombre de gousses de glycine blanche",  
 main="Histogramme de la taille\n d'une gousse de glycine blanche",  
 xlab="Taille d'une gousse de glycine blanche en cm")  
boxplot(glycine$taille,ylab="Taille d'une gousse de glycine blanche en cm",  
 main="Boîte à moustaches de la taille\n d'une gousse de glycine blanche")  
dev.off()

## quartz\_off\_screen   
## 2

#page 288  
#4)  
shapiro.test(glycine$taille)

##   
## Shapiro-Wilk normality test  
##   
## data: glycine$taille  
## W = 0.97977, p-value = 0.4906

#page 289  
length(glycine$taille)

## [1] 54

#5)  
classes<-histo$breaks  
classes

## [1] 8 10 12 14 16 18 20 22

effectifs<-histo$counts  
effectifs

## [1] 4 7 11 13 9 7 3

#6)  
mean(glycine$taille)

## [1] 14.77222

#page 290  
sd(glycine$taille)

## [1] 3.303652

#7)  
t.test(glycine$taille)

##   
## One Sample t-test  
##   
## data: glycine$taille  
## t = 32.859, df = 53, p-value < 2.2e-16  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 13.87050 15.67395  
## sample estimates:  
## mean of x   
## 14.77222

#8)  
15.67395-13.87050

## [1] 1.80345

#page 291  
1.80345/2

## [1] 0.901725

(8\*1.96/((15.67395-13.87050)/2))^2

## [1] 302.3736