Projet 1

Gilchrist

Illustration empirique de résultats théoriques clés

Q1

Le paramètre est la valeur réelle que l'on souhaite estimer.

L'estimateur est la méthode pour estimer ce paramètre.

L'estimation est le résultat de l'application de l'estimateur sur les données.

$\bar{Z_n} = \frac{1}{n}\sum_{i=1}^{n} Z_i$

$E(\bar{Z_n}) = E(\frac{1}{n}\sum_{i=1}^{n} Z_i)$

$= \frac{1}{n}\sum_{i=1}^{n} E(Z_i)$ par linéarité de l'espérance

$= \frac{1}{n} * n * E(Z_1)$ car de même loi

$E(\bar{Z_n}) = E(Z_1)$

$V(\bar{Z_n}) = V(\frac{1}{n}\sum_{i=1}^{n} Z_i)$

$= \frac{1}{n^2}\sum_{i=1}^{n} V(Z_i)$ par independance

$= \frac{1}{n^2} * n * V(Z_1)$

$V(\bar{Z_n}) = \frac{1}{n} * V(Z_1)$ car de même loi

Q2

l'erreur standard correspond à l'écart-type de $\bar{Z_n}$

l'estimateur est:

$\frac{\sqrt{\frac{1}{n-1}\sum_{i}^{n}(Z_i - \bar{Z_n})^2}}{\sqrt{n}}$

Q3

```r
# 10 realisation d'une loi exponentielle de paramètre   = 2

#set.seed(2) de garder la meme realisation
rexp(10,rate = 2)
```

```
 [1] 0.21637572 0.82720041 0.17908103 0.25183630 1.67435664 0.03550434
 [7] 0.24584590 0.01480166 0.03719376 0.07516066
```

```r
#a :
rnorm(10,mean=0, sd=1)
```

```
 [1] -0.6560525  1.2192529  0.2585224  2.0550090  0.3268893 -1.3036973
 [7] -1.6861987  0.6857671 -0.7204704  0.9439775
```

```r
alpha <- 2
beta <- 4
rgamma(10,shape = alpha, rate = beta)
```

```
 [1] 0.51272672 0.20583706 1.02914776 0.02690109 0.35683183 0.16246762
 [7] 0.09297424 0.29155838 0.62608976 0.39210851
```

```r
#b

## Distribution de loi exponentielle

d_exp <- rexp(20,2)
d_exp
```

```
 [1] 0.02336003 0.14132335 0.10305944 0.07737802 0.07550970 0.20732578
 [7] 0.37221604 0.54876137 0.58279749 0.28928851 0.41400022 0.59639118
[13] 0.36657797 0.05709763 0.13708772 0.14357822 0.62813959 0.09528387
[19] 0.05612462 1.87264377
```

```r
moy_empiride <- mean(d_exp)
moy_empiride
```

```
[1] 0.3393972
```

```r
erreur_standart <- sd(d_exp)/sqrt(20)
erreur_standart
```

```
[1] 0.09266147
```

```r
erreur_standart_b <-sqrt(var(d_exp))/sqrt(20)
erreur_standart_b
```

```
[1] 0.09266147
```

```r
## Distribution de loi Normale

d_normale <- rnorm(20,mean = 0, sd= 1)
d_normale
```

```
 [1]  1.6210979 -0.6149340 -1.4784632  0.2067315  0.2820035 -0.1250721
 [7] -0.1912022  0.8368718  0.7773812 -0.2808039 -0.6279029  1.2032676
[13] -1.3503150 -0.1922614 -0.6537205  0.7947002 -1.2789463 -0.2993549
[19]  1.3524133  0.3781514
```

```r
mean(d_normale)
```

```
[1] 0.0179821
```

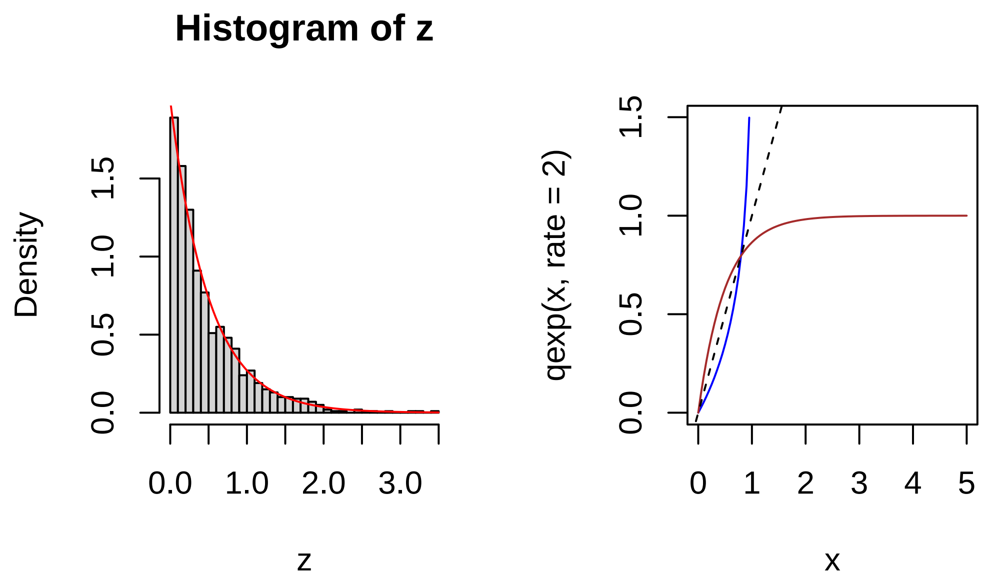
```r
sd(d_normale)/sqrt(20)
```

```
[1] 0.2000141
```

```r
#c
par(mfrow = c(1, 2))
n <- 1000
z <- rexp(n, rate = 2)
hist(z, freq = FALSE, breaks = 30)
curve(dexp(x, rate = 2), add = TRUE, col = "red")
curve(qexp(x, rate = 2), from = 0, to = 5, col = "blue")
```

```
Warning in qexp(x, rate = 2): Production de NaN
```

```r
curve(pexp(x, rate = 2), add = TRUE, col = "brown")
abline(0, 1, lty=2)
```



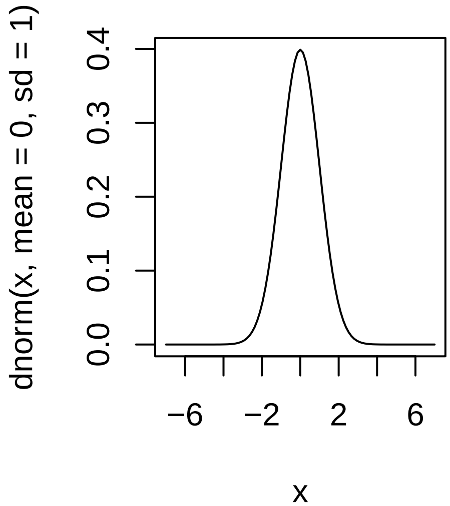

```r
# On part de la réalisation de 1000 observation suivant une loi exponentielle
#de paramètre 2. ET:
# On trace l'histogramme de cette realisation avec la fonction hist()
#Puis on trace successivement la densité, le quantile, et la fonction
# de repartition respectivement avec les fonction dexp ,qexp, pexp

qnorm(d_normale,mean = 0, sd =1)
```

```
Warning in qnorm(d_normale, mean = 0, sd = 1): Production de NaN
```

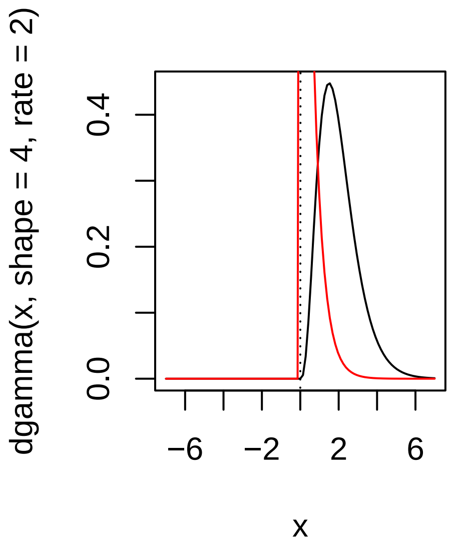
```
 [1]        NaN        NaN        NaN -0.8178147 -0.5769000        NaN
 [7]        NaN  0.9816823  0.7633785        NaN        NaN        NaN
[13]        NaN        NaN        NaN  0.8228388        NaN        NaN
[19]        NaN -0.3103396
```

```r
curve(dnorm(x,mean = 0, sd =1),from = -7, to = 7)
#verif de la convergence de la somme var exp vers une loi gamma
par(mfrow = c(1, 2))
```




```
curve(dgamma(x,shape = 4, rate =2),from = -7, to = 7)
curve(dexp(x, rate = 2), add = TRUE, col = "red")
abline(0, 40, lty=15)

#d
# mean(z <= 1.2) devrait se rapprocher de la fonction de repartition
# evaluer en 2 pour x= 1.2 et quantile(z, probs = 0.9 de valeur obtenue
# en evaluant la reciproque de la fonction de repartion en 0.9
```



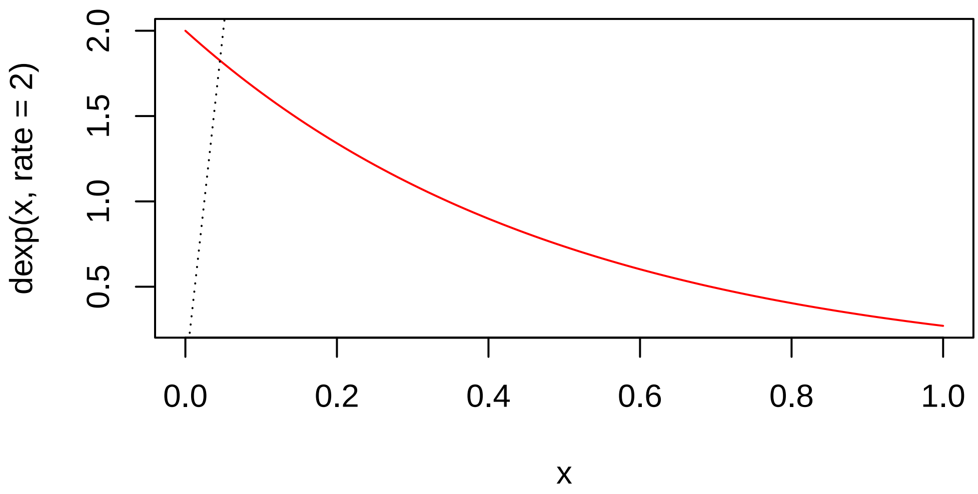
Q4

Loi Forte des grands nombre: Soit (Z_n) une suite de variable aléatoire integrable et I.i.d. Soit m leur esperance commune alors $\bar{Z_n} = \frac{1}{n} \sum \_i = 1nZ_i$ converge presque surement vers m

TCL : Soit (Z_n) une suite de variable aléatoire de carré integrable et I.i.d. Avec $m = E(Z_1)$ et $ ˆ2=v(Z_1) $, alors on a: $\frac{\sqrt{n}}{\epsilon}(\bar{(}Z_n - m)$ converge en loi vers une gaussienne centré réduite

Q5

```
curve(dexp(x, rate = 2),  col = "red")
abline(0, 40, lty=15)
```




```r
n=1000
z <- rexp(n, rate=2)
mean(z) # ce qui approxime à la moyenne théorique, ici 1/2
```

```
[1] 0.4908711
```

```r
var(z)/n #0.0002553287
```

```
[1] 0.0002283723
```

```r
1/(4*n) # 0.00025
```

```
[1] 0.00025
```

Evaluation

Exercice 3

```r
#1
moyenne <- function(x){
  cmpt = 0
  for (val in x){
    cmpt = cmpt +val
  }
  cmpt/length(x)
}
x <- 1:10
mean(x)
```

```
[1] 5.5
```

```r
moyenne(x)
```

```
[1] 5.5
```

```r
# 2
moyenne_b <- function(x){
  x <- x[!is.na(x)]
  cmpt = 0
  for (val in x){
    cmpt = cmpt +val
  }
  cmpt/length(x)
}
y <- c(1:10, NA)
mean(y,na.rm = TRUE)
```

```
[1] 5.5
```

```r
moyenne_b(y)
```

```
[1] 5.5
```

```r
# 3
indPremNeg <- function(x){
  for (i in 1:length(x)){
    if (x[i] < 0){
      return(i)
    }
  }
  return(0)
}

z <- c(1,2,-3,1)
v <-  c(1,2,3,1)
indPremNeg(z)
```

```
[1] 3
```

```r
which(z<0)
```

```
[1] 3
```

```r
# 4

premValPosCol <- function(df){
  v = c()
  for (i in 1 :ncol(df)){
    for (j in 1 :nrow(df)){
      if ( df[j,i] > 0){
        v = c(v,df[j,i])
        break
      }
      next
    }
    next
    v = c(v, NA)
  }
  v
}
u <- runif(10, min = -1, max = 1)
v <- runif(10, min = -1, max = 1)
d <- data.frame(u, v)
d
```

```
            u           v
1  -0.30813733  0.03031529
2   0.14317513  0.75227536
3   0.69112376  0.44774216
4   0.97843532  0.38999744
5   0.83840579 -0.08455445
6  -0.09301557 -0.02641467
7  -0.86163239 -0.68286800
8   0.18818431  0.56412475
9   0.19391642 -0.14168970
10  0.12935956 -0.53916676
```

```
premValPosCol(d)
```

```
[1] 0.14317513 0.03031529
```