

Solutions for PC5: Estimation and R

Octobre 5 2018

I Sampling random variables

Sampling random variables is very useful in statistics. In this part, we learn to sample from different distributions using a generator of uniform $\mathcal{U}(0,1)$. Such a generator is the cornerstone of simulation algorithms and every computer has one.

In statistics, sampling is used in two contexts. First, simulations are used to check statistical methods before applying it to real data. It is a way to control what happens when you know the truth, i.e. the population. Second, some inference methods are based on simulation for instance bootstrap methods, used in order to build confidence intervals, or MCMCs methods used to approximately sample from a posterior distribution.

The three following exercises will help you understand how you can simulate various random variables ‘runif’ of R. They are for illustrative purposes only. In the future, use existing functions, such as `rnorm`, `rexp`, `rpois`, if they exist. This will prevent errors and speed up your code.

Exercise 1: Discrete distribution

Let $p = (p_1, \dots, p_k) \in \Delta_k = \{p \in [0,1]^k : \sum_{i=1}^k p_i = 1\}$ and X is a discrete random variable such that $P(X = i) = p_i$ for all $i \leq k$. Let P_p be the distribution of X .

1. Propose an algorithm to sample one realization of P_p using `runif` only.

Let U be a random variable distributed from $\mathcal{U}(0,1)$. Let

$$Z = \begin{cases} 1 & \text{if } U \leq p_1 \\ i & \text{if } \sum_{j=1}^{i-1} p_j < U \leq \sum_{j=1}^i p_j \\ k & \text{if } \sum_{i=1}^{k-1} p_i < U \end{cases}$$

Then Z is distributed from P_p .

```
p<-c(0.2,0.5,0.3)
u<-runif(1,min=0,max=1)
x<-1+(u>p[1])+(u>p[1]+p[2])
x
```

```
## [1] 2
```

2. Write a function sampling n realizations of i.i.d. random variables from P_p .

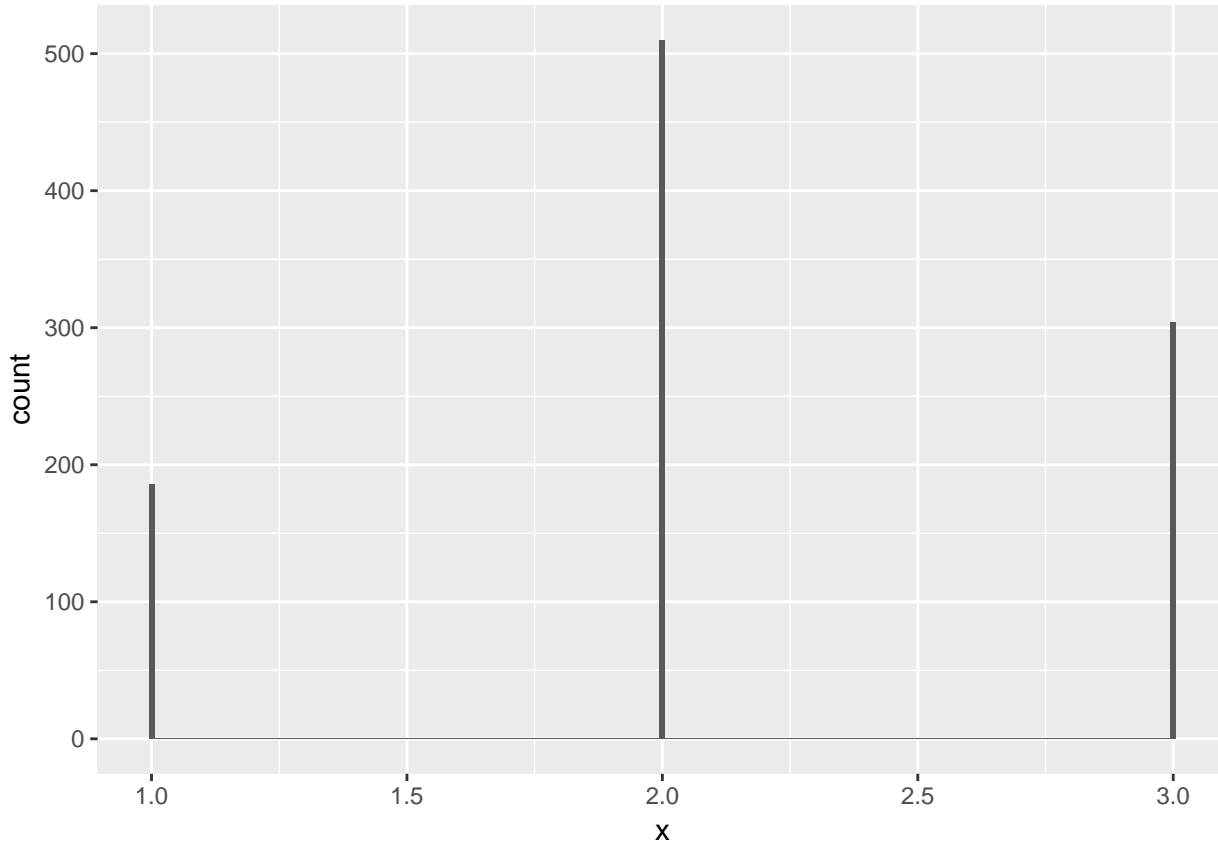
```
rdiscrete <- function(n,p){
  k<- length(p);
  u<-runif(n,min=0,max=1);
  x<-1;
  for (j in 1:(k-1)){
    x<-x+(u>sum(p[1:j]));
  }
  return(x)
}
```

3. How can you check that the algorithm is indeed sampling from P_p ?

We can check if the law of large number seems to be verified by sampling a lot of realizations. Then the proportion of realizations with value i should be around p_i .

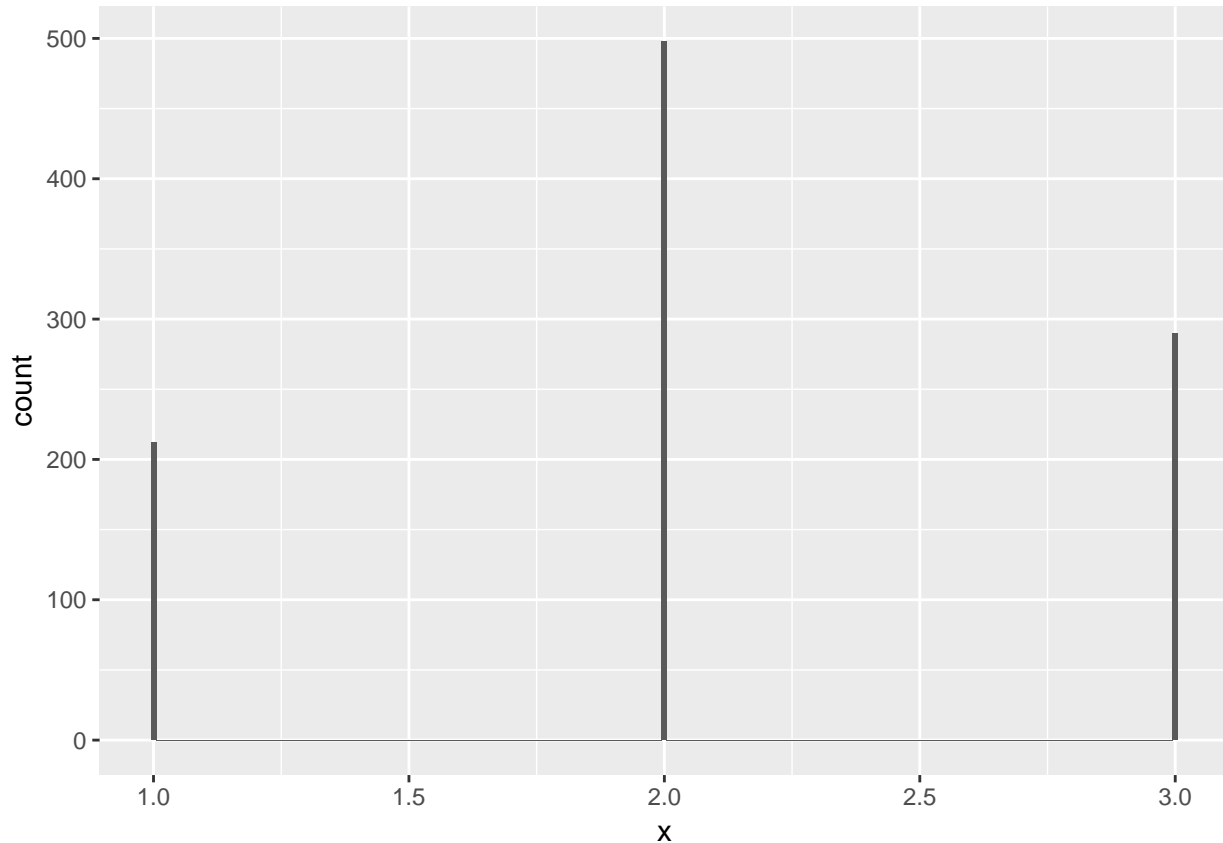
```
n <- 1000
p <- c(0.2,0.5,0.3)
data_discr1=data.frame(x=rdiscrete(n,p))
library(ggplot2)
ggplot(data=data_discr1, aes(x=x))+ geom_bar(binwidth = 0.01)
```

```
## Warning: `geom_bar()` no longer has a `binwidth` parameter. Please use
## `geom_histogram()` instead.
```



```
data_discr2=data.frame(x=rdiscrete(n,p))
library(ggplot2)
ggplot(data=data_discr2, aes(x=x))+ geom_bar(binwidth = 0.01)
```

```
## Warning: `geom_bar()` no longer has a `binwidth` parameter. Please use
## `geom_histogram()` instead.
```



Exercise 2: Simulations of random variables by inverting the cdf

1. Let $F : (a, b) \rightarrow (0, 1)$ be a one to one cdf with $-\infty \leq a \leq b \leq +\infty$, and U a random variable distributed from $\mathcal{U}(0, 1)$. What is the distribution of $F^{-1}(U)$, where F^{-1} is the inverse function of F ?

Let U be a random variable distributed from $\mathcal{U}(0, 1)$ and $Z = F^{-1}(U)$. We characterize the distribution of Z by computing its cdf. Let $t \in \mathbb{R}$, since a cdf is non-decreasing,

$$\mathbb{P}(Z \leq t) = \mathbb{P}(F^{-1}(U) \leq t) = \mathbb{P}(F(F^{-1}(U)) \leq F(t)) = \mathbb{P}(U \leq F(t)) = F(t).$$

Then the cdf of Z is F .

2. Propose an algorithm to sample from a Exponential distribution $\text{Exp}(\lambda)$ using a sampler of the uniform $\mathcal{U}(0, 1)$.

The cdf of the Exponential distribution $\text{Exp}(\lambda)$ is

$$t \mapsto F_{\text{Exp}(\lambda)}(t) = \mathbb{P}_{\text{Exp}(\lambda)}(X \leq t) = \int_0^t \lambda \exp(-\lambda x) dx = [-\exp(-\lambda x)]_{x=0}^t = 1 - \exp(-\lambda t).$$

This cdf is one to one from $(0, +\infty)$ in $(0, 1)$ and its inverse on $(0, 1)$ is $F^{-1}(u) = \frac{-\log(1-u)}{\lambda}$.

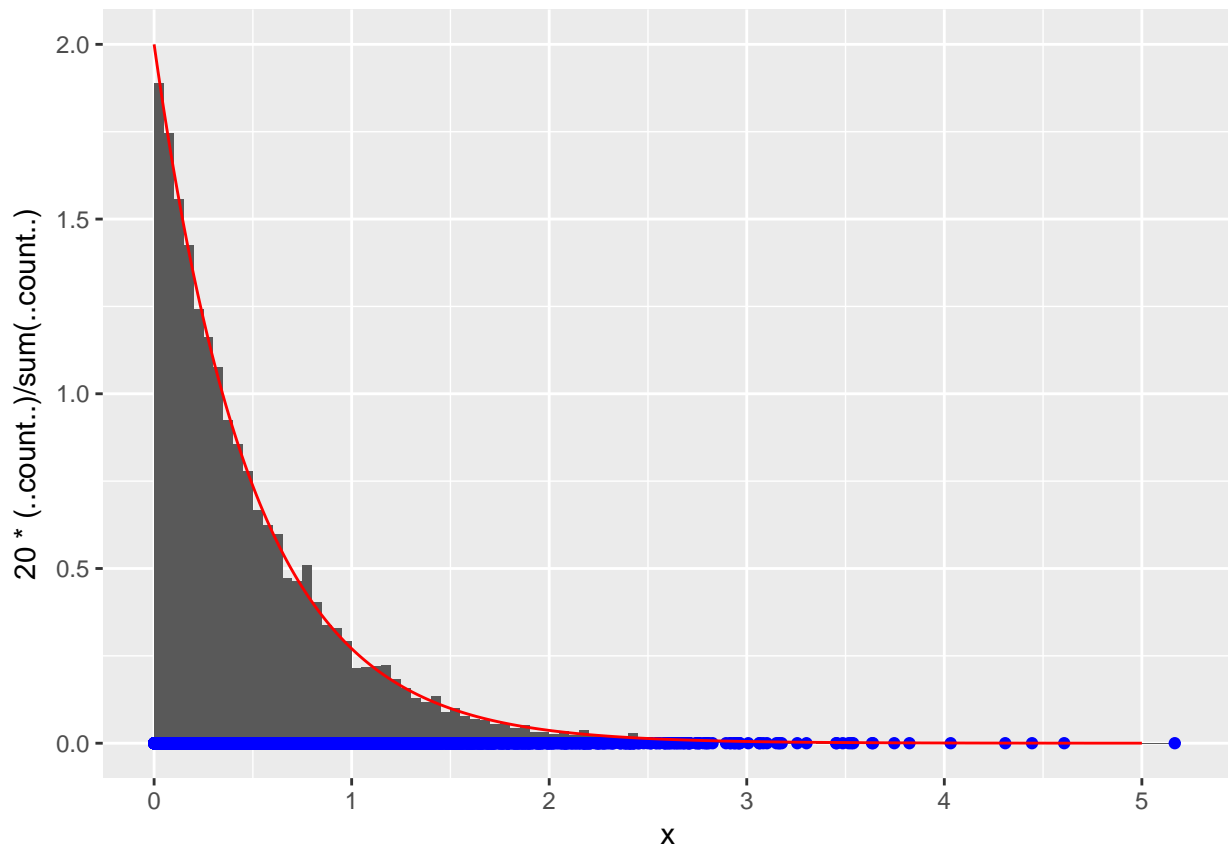
Then $Y = \frac{-\log(1-U)}{\lambda}$ and also $X = \frac{-\log(U)}{\lambda}$ are distributed from an Exponential distribution $\text{Exp}(\lambda)$.

```
rexponential <- function(n,lambda){
  u <- runif(n,min=0,max=1);
  x <- -1/lambda*log(u)
  return(x)
}
```

```

n <- 10000
lambda=2
x1 <- rexp(n,lambda)
data_exp1<-data.frame(x=x1,xsort=sort(x1), quant= qexp(1/(n+1)*(1:n), rate = lambda))
dens_exp<-data.frame(t=seq(0,5, by=0.001),exp_density=lambda*exp(-lambda*seq(0,5, by=0.001)))
ggplot(data=data_exp1, aes(x=x))+ geom_histogram(aes(y=20*(..count..)/sum(..count..)),binwidth=1/20,
boundary=0)+
  geom_point(aes(y=0),colour="blue") + geom_line(data=dens_exp, aes(x=t,y=exp_density),color="red")

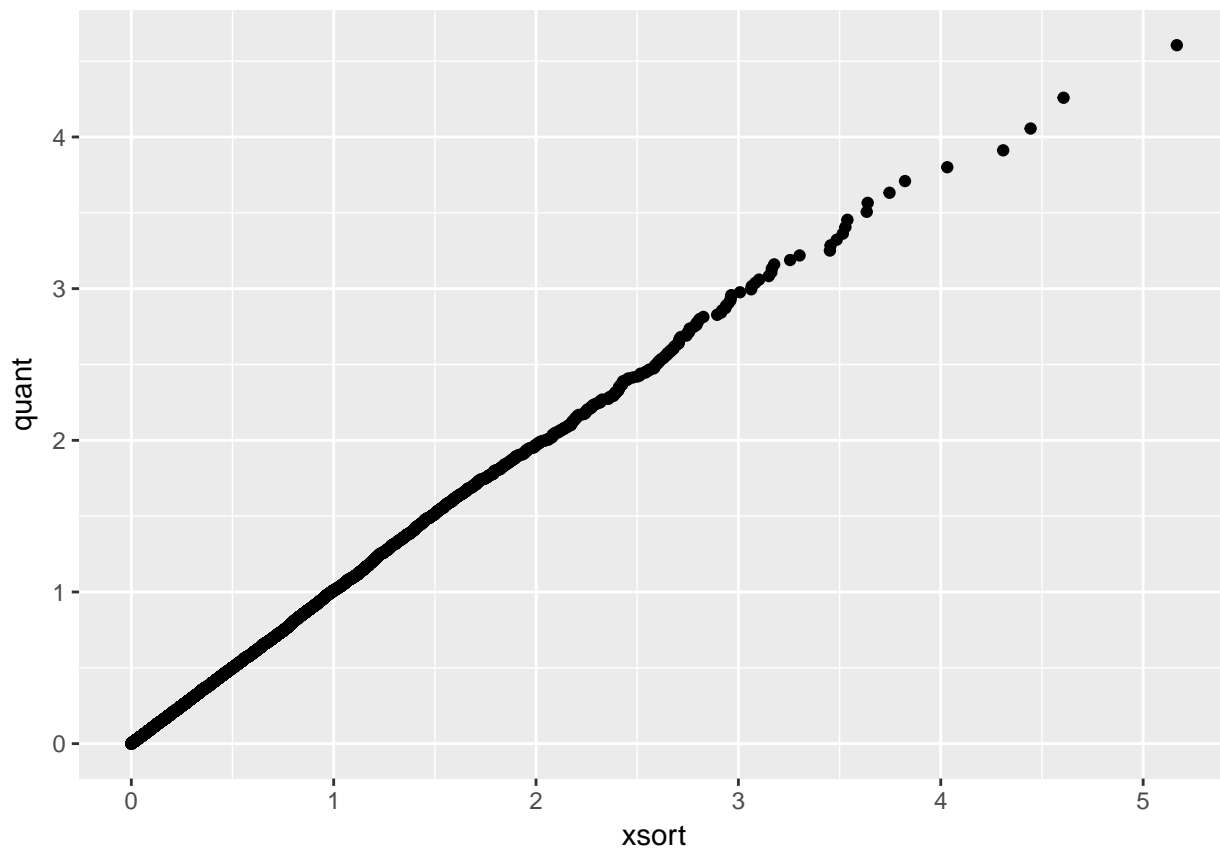
```



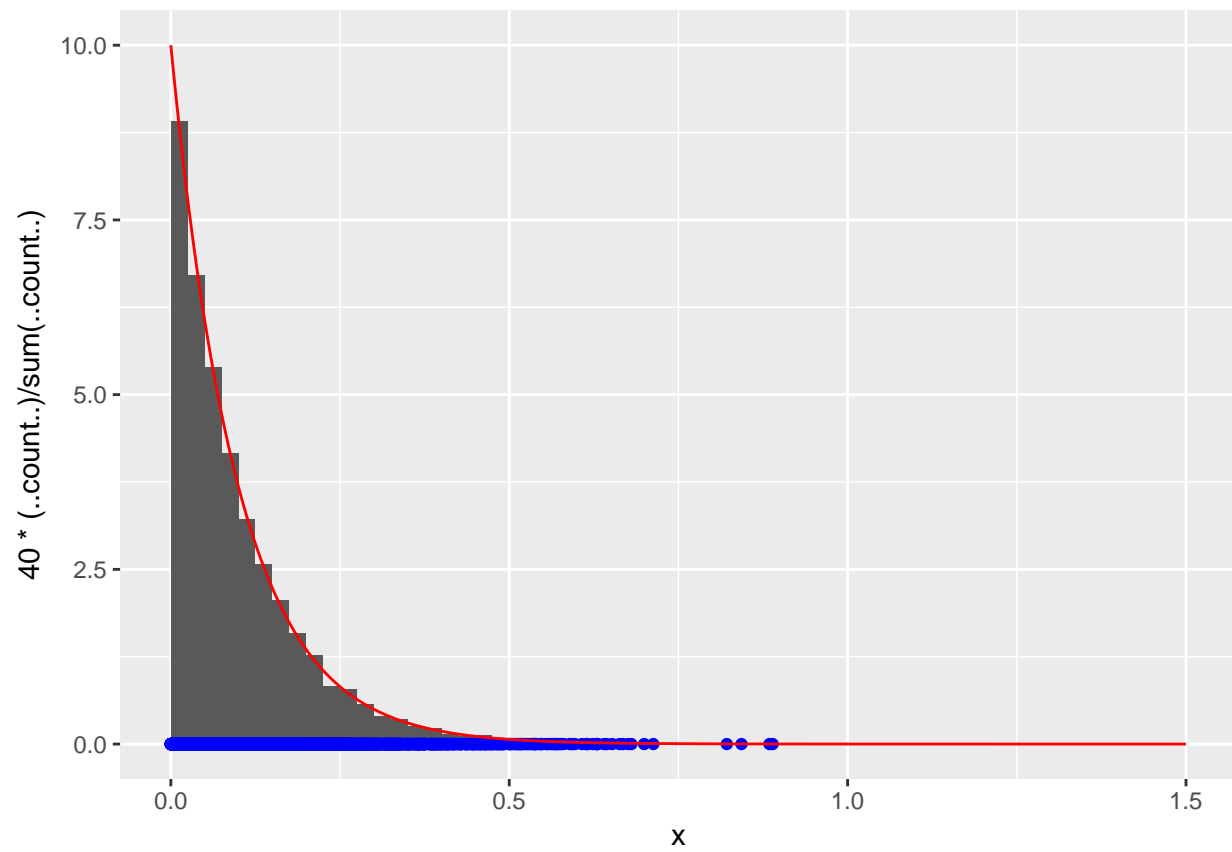
```

ggplot(data=data_exp1, aes(x=xsort,y=quant))+geom_point()

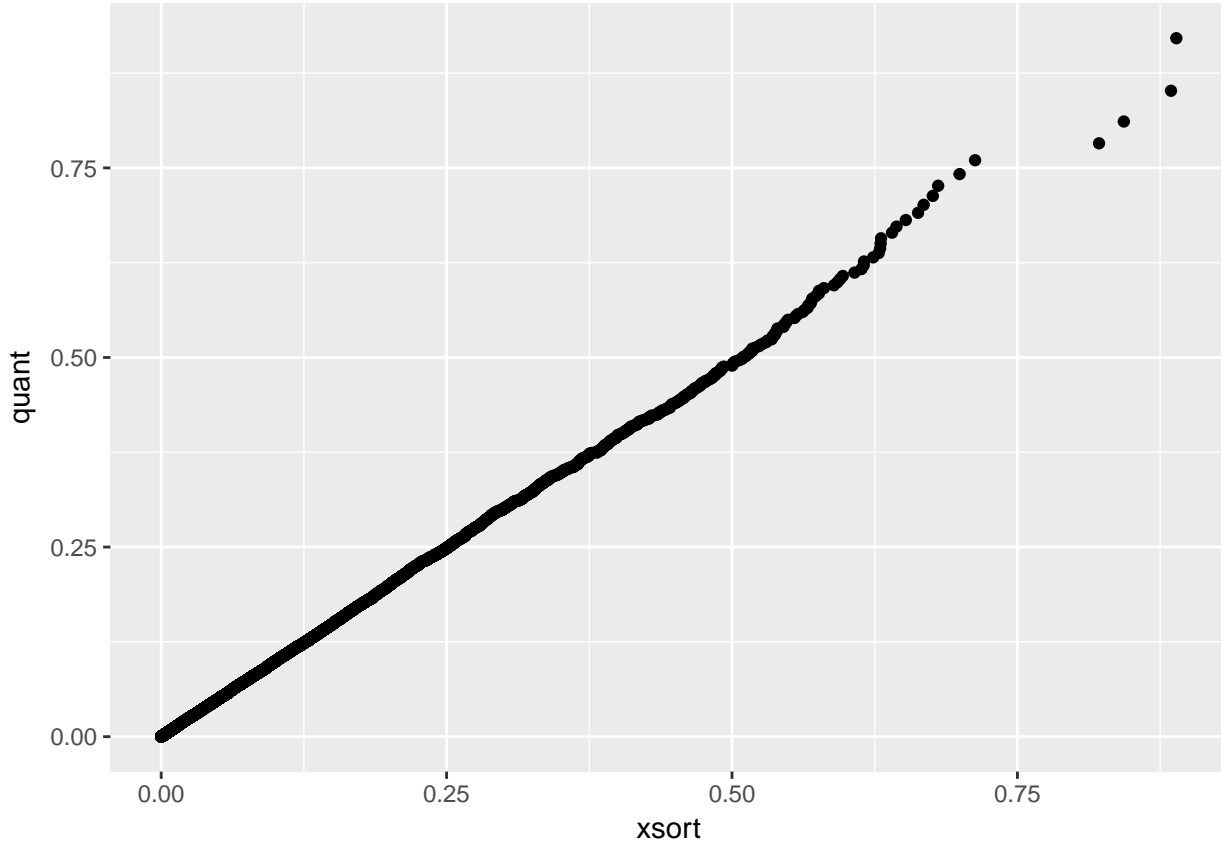
```



```
lambda=10
x2 <- rexponential(n,lambda)
data_exp2 <-data.frame(x=x2,xsort=sort(x2), quant= qexp(1/(n+1)*(1:n), rate = lambda))
dens_exp2<-data.frame(t=seq(0,1.5, by=0.001),exp_density=lambda*exp(-lambda*seq(0,1.5, by=0.001)))
ggplot(data=data_exp2, aes(x=x))+ geom_histogram(aes(y = 40*(..count..)/sum(..count..)),binwidth=1/40,
                                                boundary=0)+geom_point(aes(y=0), colour="blue")+
  geom_line(data=dens_exp2, aes(x=t,y=exp_density),color="red")
```



```
ggplot(data=data_exp2, aes(x=xsort,y=quant))+geom_point()
```



3. Propose an algorithm to sample from a Cauchy distribution using a sampler of the uniform $\mathcal{U}(0, 1)$. The density function of a Cauchy distribution with parameters $x_0 \in \mathbb{R}$ and $\gamma > 0$ is

$$x \in \mathbb{R} \mapsto \frac{1}{\pi\gamma} \frac{\gamma^2}{(x - x_0)^2 + \gamma^2}.$$

The cdf of the Cauchy distribution is

$$t \mapsto F_{\text{Cauchy}(x_0, \gamma)}(t) = \mathbb{P}_{\text{Cauchy}(x_0, \gamma)}(X \leq t) = \int_{-\infty}^t \frac{1}{\pi\gamma} \frac{\gamma^2}{(x - x_0)^2 + \gamma^2} dx = \int_{-\infty}^{\frac{t-x_0}{\gamma}} \frac{1}{\pi} \frac{1}{u^2 + 1} du$$

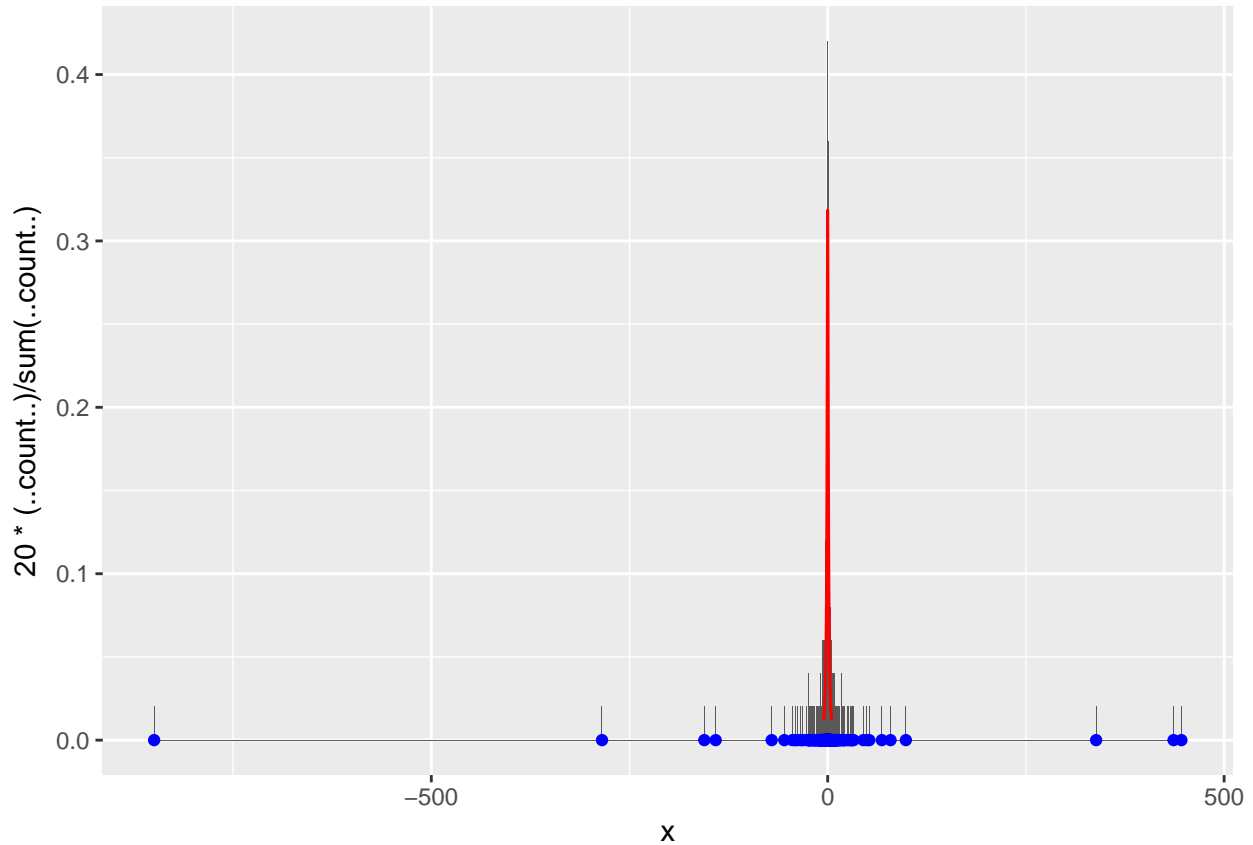
$$F_{\text{Cauchy}(x_0, \gamma)}(t) = \left[\arctan\left(\frac{x - x_0}{\gamma}\right) \right]_{x=-\infty}^{x=t} = \frac{1}{2} + \arctan\left(\frac{t - x_0}{\gamma}\right).$$

And the cdf defined on $(-\infty, +\infty)$ in $(0, 1)$ is one to ones with inverse $F^{-1}(u) = \gamma \tan(\pi(u - 1/2)) + x_0$. So that $\gamma \tan(\pi(U - 1/2)) + x_0$ is distributed from a Cauchy distribution with parameters x_0 and γ .

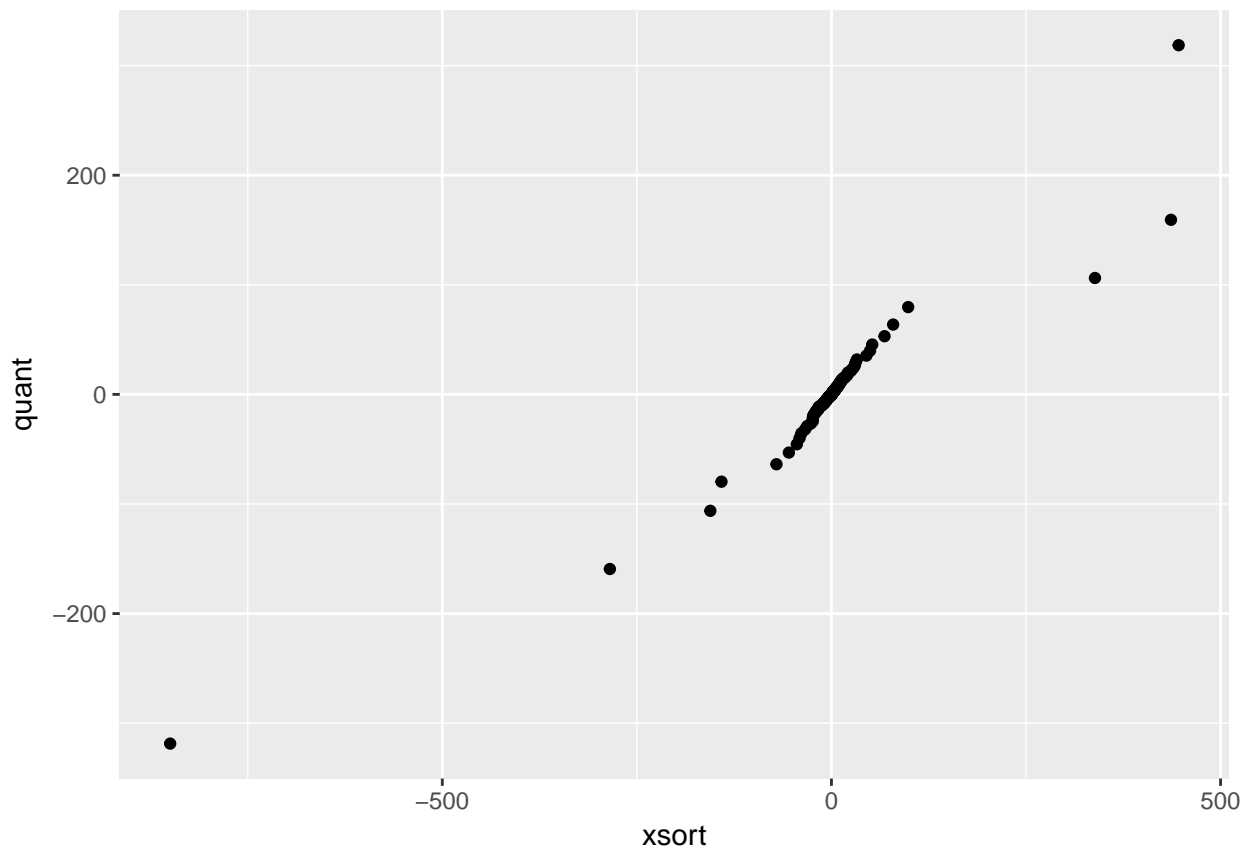
```
rcauchy <- function(n,x0,gamma){
  u <- runif(n,min=0,max=1);
  x <- x0 + gamma*tan(pi*(u-1/2))
  return(x)
}

n <- 1000
x0<-0
gamma <- 1
x1 <- rcauchy(n,x0,gamma)
data_cauchy1<-data.frame(x=x1,xsort=sort(x1), quant= qcauchy(1/(n+1)*(1:n), location=x0,scale=gamma))
```

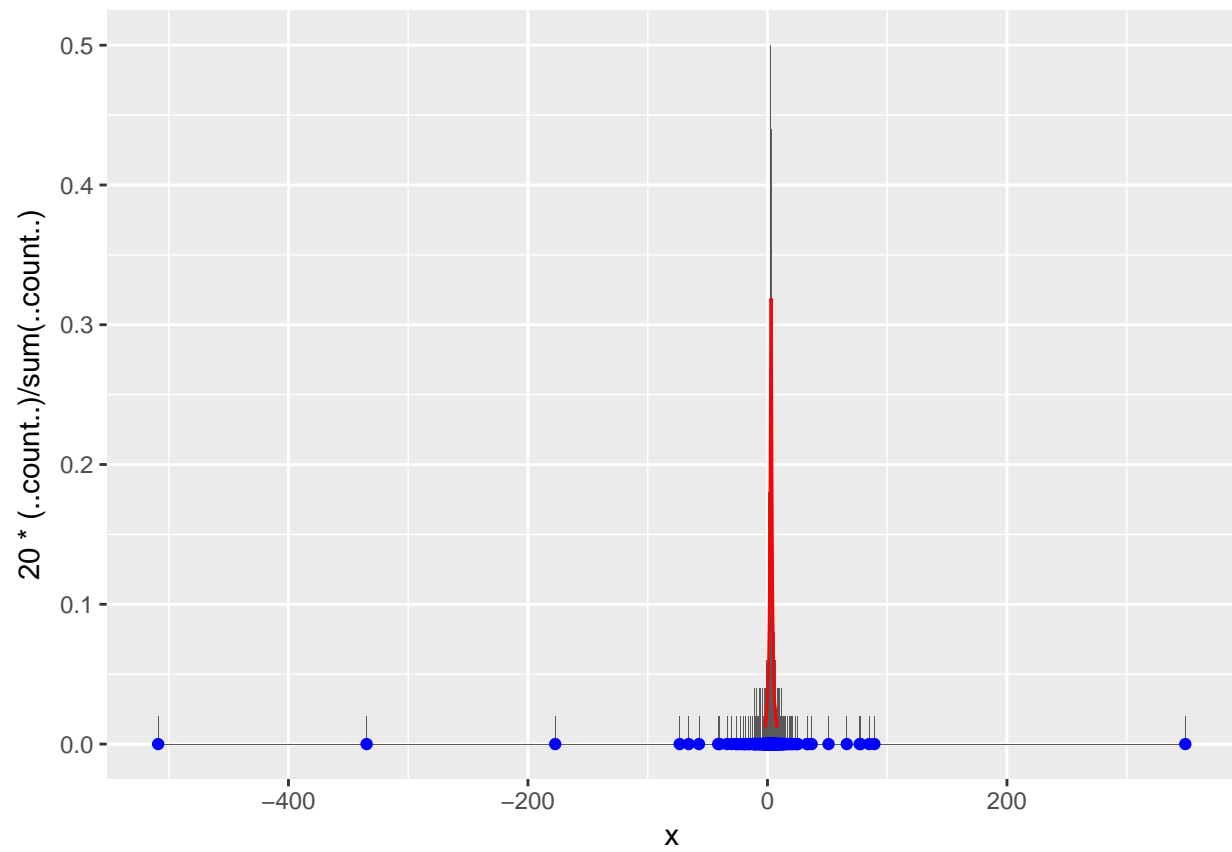
```
dens_cauchy<-data.frame(t=seq(-5,5, by=0.001),density=dcauchy(seq(-5,5, by=0.001),location=x0,
                                                                scale=gamma))
ggplot(data=data_cauchy1, aes(x=x))+ geom_histogram(aes(y = 20*(..count..)/sum(..count..)),
                                                    binwidth=1/20,boundary=0) +
  geom_point(aes(y=0), colour="blue")+ geom_line(data=dens_cauchy, aes(x=t,y=density),color="red")
```



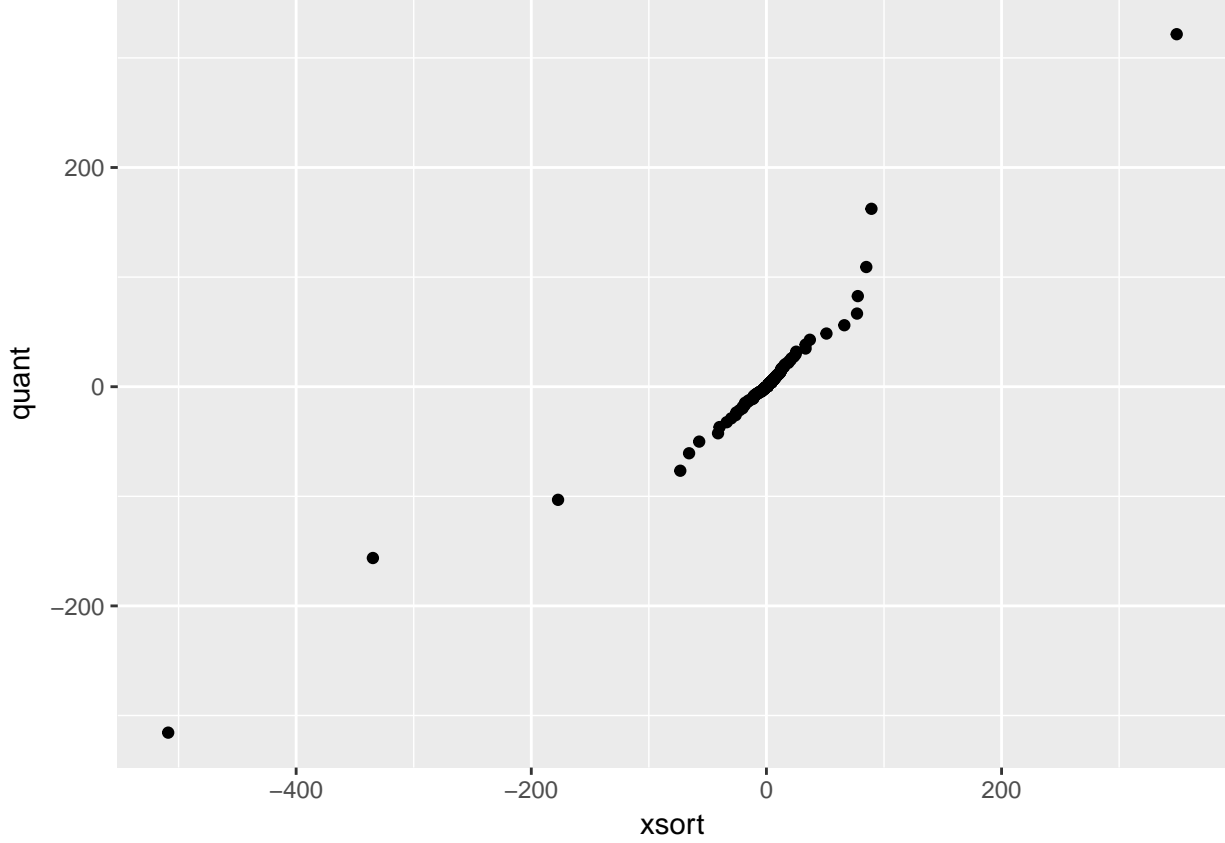
```
ggplot(data=data_cauchy1, aes(x=xsort,y=quant))+geom_point()
```

```
x0<-3
gamma <- 1
x2 <- rcauchy(n,x0,gamma)
data_cauchy2<-data.frame(x=x2,xsort=sort(x2), quant= qcauchy(1/(n+1)*(1:n), location=x0,
                                                             scale=gamma))
dens_cauchy2<-data.frame(t=seq(-2,8, by=0.001),density=dcauchy(seq(-2,8, by=0.001),
                                                                location=x0,scale=gamma))
ggplot(data=data_cauchy2, aes(x=x))+ geom_histogram(aes(y = 20*(..count..)/sum(..count..)),
                                                    binwidth=1/20,boundary=0) +
  geom_point(aes(y=0), colour="blue") + geom_line(data=dens_cauchy2,
                                                  aes(x=t,y=density),color="red")
```



```
ggplot(data=data_cauchy2, aes(x=xsort,y=quant))+geom_point()
```



Exercise 3: Simulation of Gaussian random variables and vectors using Box-Muller

1. Let U and V be two independent random variables distributed from the uniform distribution $\mathcal{U}(0, 1)$. Let $R = -\ln(U)$, $Z_1 = \sqrt{R}\cos(2\pi V)$ and $Z_2 = \sqrt{R}\sin(2\pi V)$. What is the distribution of R ? What is the distribution of the random vector (Z_1, Z_2) ?

Using Exercise 2, R is distributed from an exponential distribution $\text{Exp}(1/2)$.

Let $g : \mathbb{R}^2 \rightarrow \mathbb{R}$ be continuous and bounded.

$$\mathbb{E}(g(Z_1, Z_2)) = \mathbb{E}[g(\sqrt{R}\cos(2\pi V), \sqrt{R}\sin(2\pi V))] = \int_{\mathbb{R}_+ \times (0, 1)} g(\sqrt{r}\cos(2\pi v), \sqrt{r}\sin(2\pi v)) \exp(-r/2) dr dv$$

$$\mathbb{E}(g(Z_1, Z_2)) = \int_{\mathbb{R}_+ \times (0, 1)} g(s\cos(2\pi v), s\sin(2\pi v)) \exp(-\frac{s^2}{2}) s ds dv.$$

We use the polar/Cartesian change of variables:

$$h : (s, v) \in \mathbb{R}_+ \times (0, 1) \mapsto (s\cos(2\pi v), s\sin(2\pi v)) \in \mathbb{R}^2 \setminus \{(0, 0)\}$$

is one to one with Jacobian:

$$\begin{vmatrix} \cos(2\pi v) & \sin(2\pi v) \\ -2\pi s \sin(2\pi v) & 2\pi s \cos(2\pi v) \end{vmatrix} = 2\pi r.$$

So that

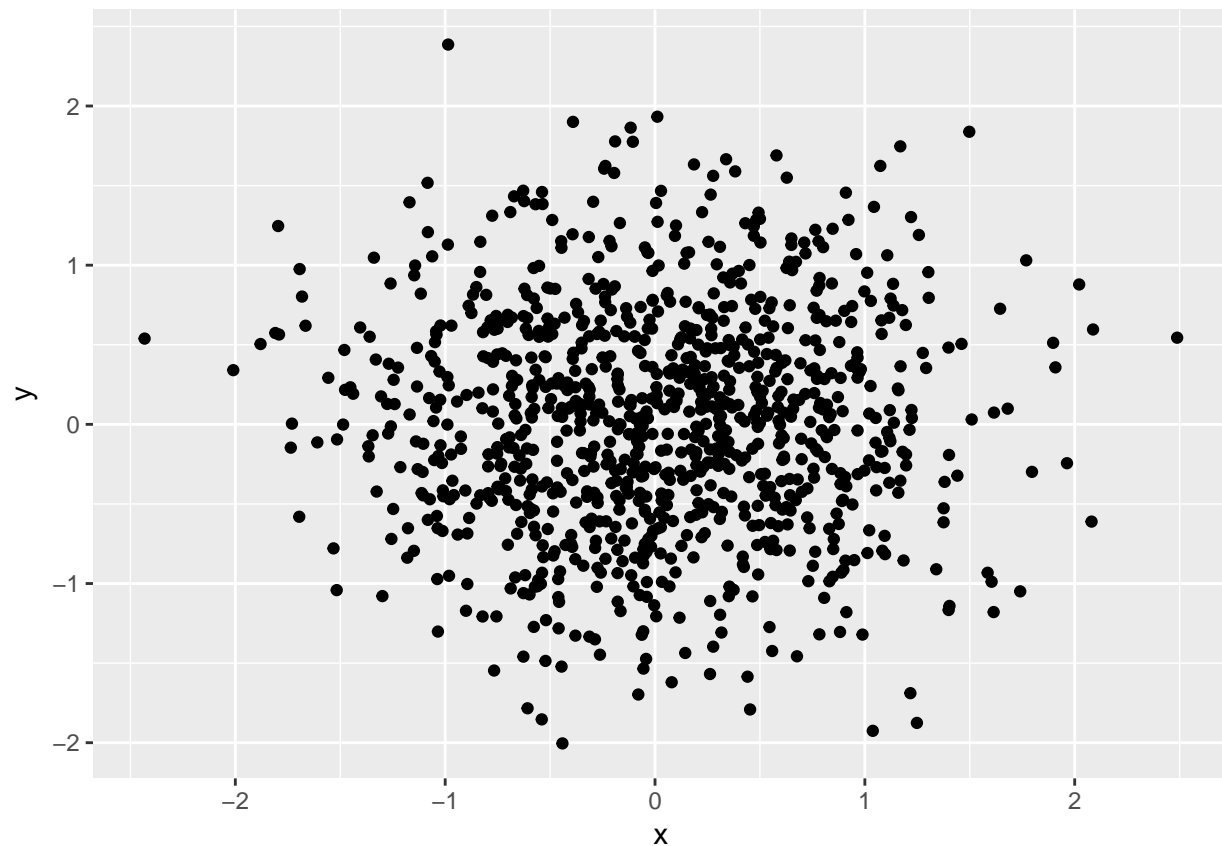
$$\mathbb{E}(g(Z_1, Z_2)) = \int_{\mathbb{R}^2 \setminus \{(0, 0)\}} g(x, y) \exp\left(-\frac{x^2 + y^2}{2}\right) \frac{1}{2\pi} dx dy = \int_{\mathbb{R}} \int_{\mathbb{R}} g(x, y) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right) dx dy.$$

and Z_1 and Z_2 are two independent standard Gaussian variables, equivalently $(Z_1, Z_2) \sim \mathcal{N}((0, 0), Id_2)$.

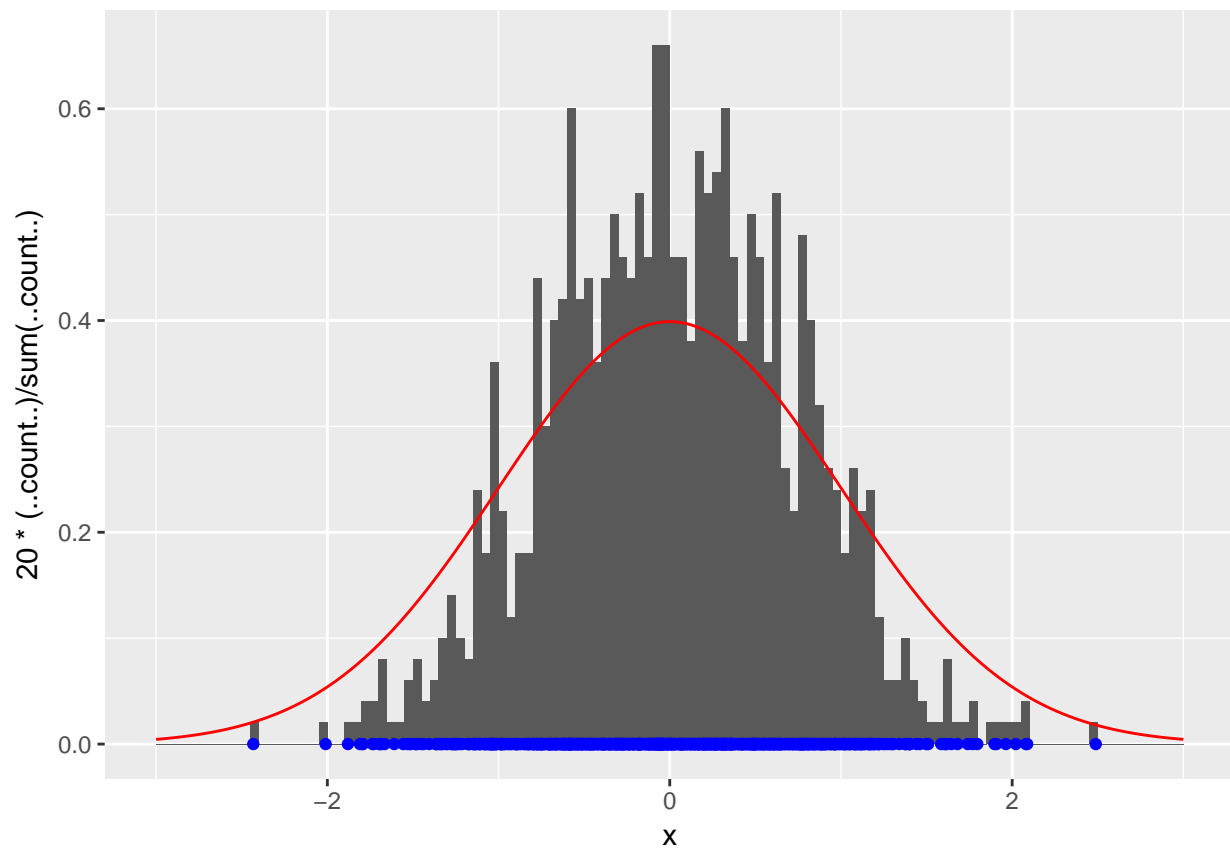
2. Deduce an algorithm to sample a realization of $(Z_1, Z_2)^T \sim \mathcal{N}((0,0)^T, Id_2)$ using a sampler of the uniform $\mathcal{U}(0, 1)$.

```
rGaussian <- function(n){
  u <- runif(n,min=0,max=1);
  v<- runif(n,min=0,max=1);
  x <- sqrt(-log(u))*cos(2*pi*v);
  y <- sqrt(-log(u))*sin(2*pi*v);
  return(data.frame(x=x,y=y))
}
```

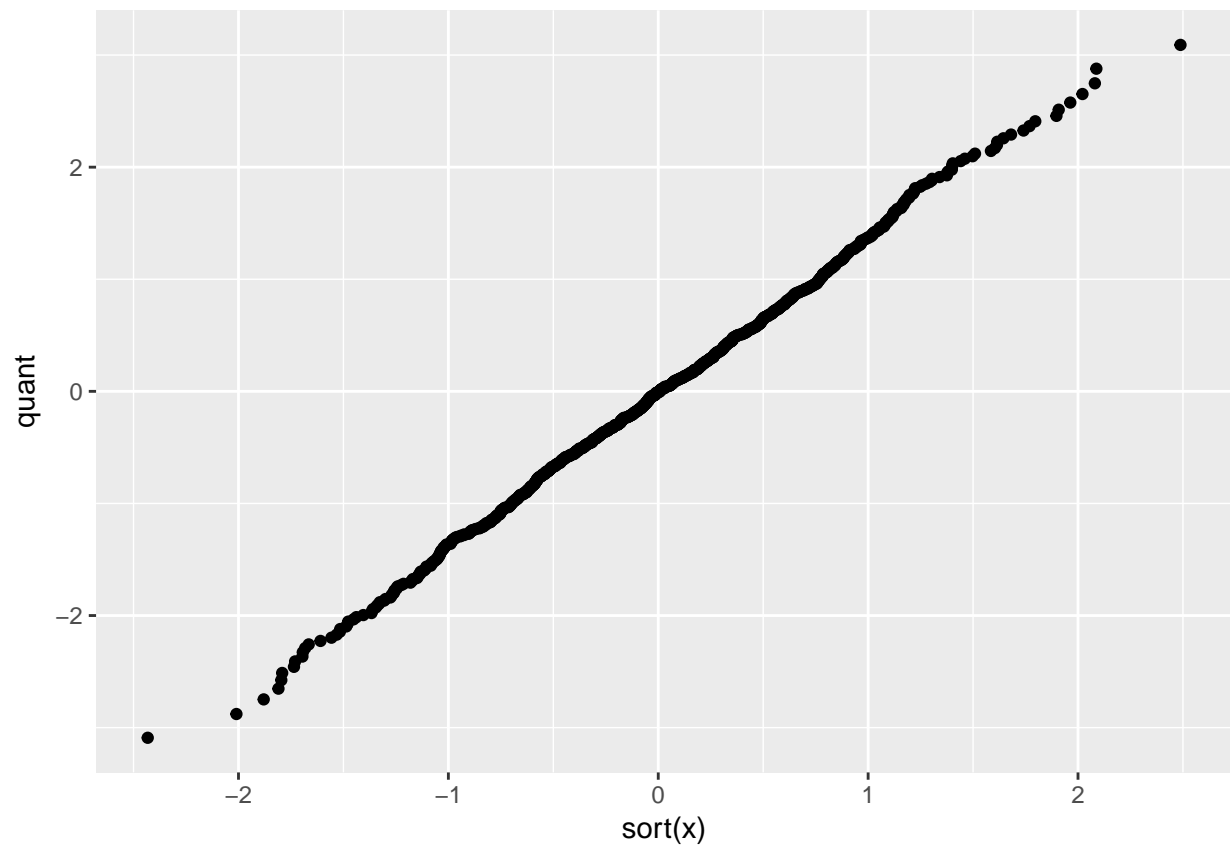
```
n <- 1000
data_Gaussian1 <- rGaussian(n)
ggplot(data=data_Gaussian1, aes(x=x,y=y))+ geom_point()
```



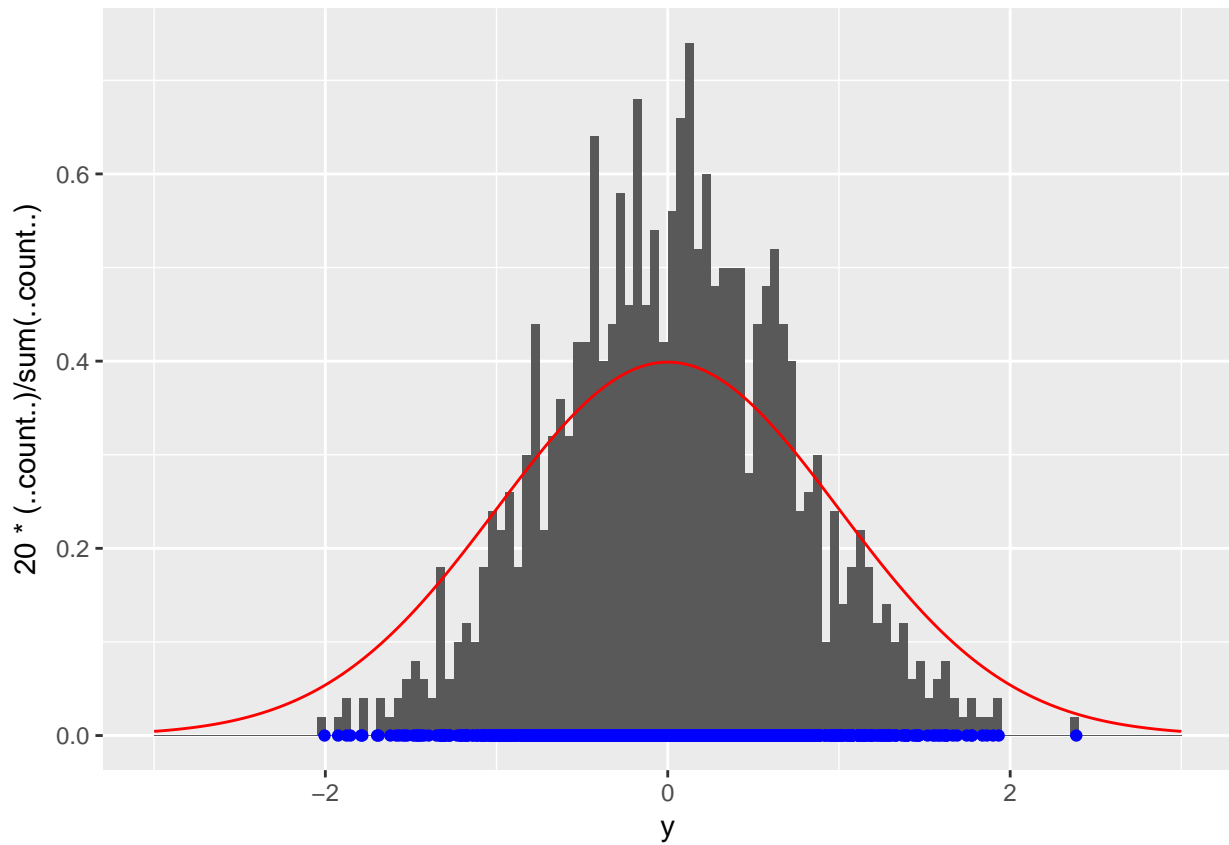
```
data_Gaussian1$quant <- qnorm(1/(n+1)*(1:n))
dens_gauss<-data.frame(t=seq(-3,3, by=0.001),density=dnorm(seq(-3,3, by=0.001)))
ggplot(data=data_Gaussian1, aes(x=x))+ geom_histogram(aes(y = 20*(..count..)/sum(..count..)),
  binwidth=1/20,boundary=0) +
  geom_point(aes(y=0), colour="blue")+ geom_line(data=dens_gauss, aes(x=t,y=density),color="red")
```



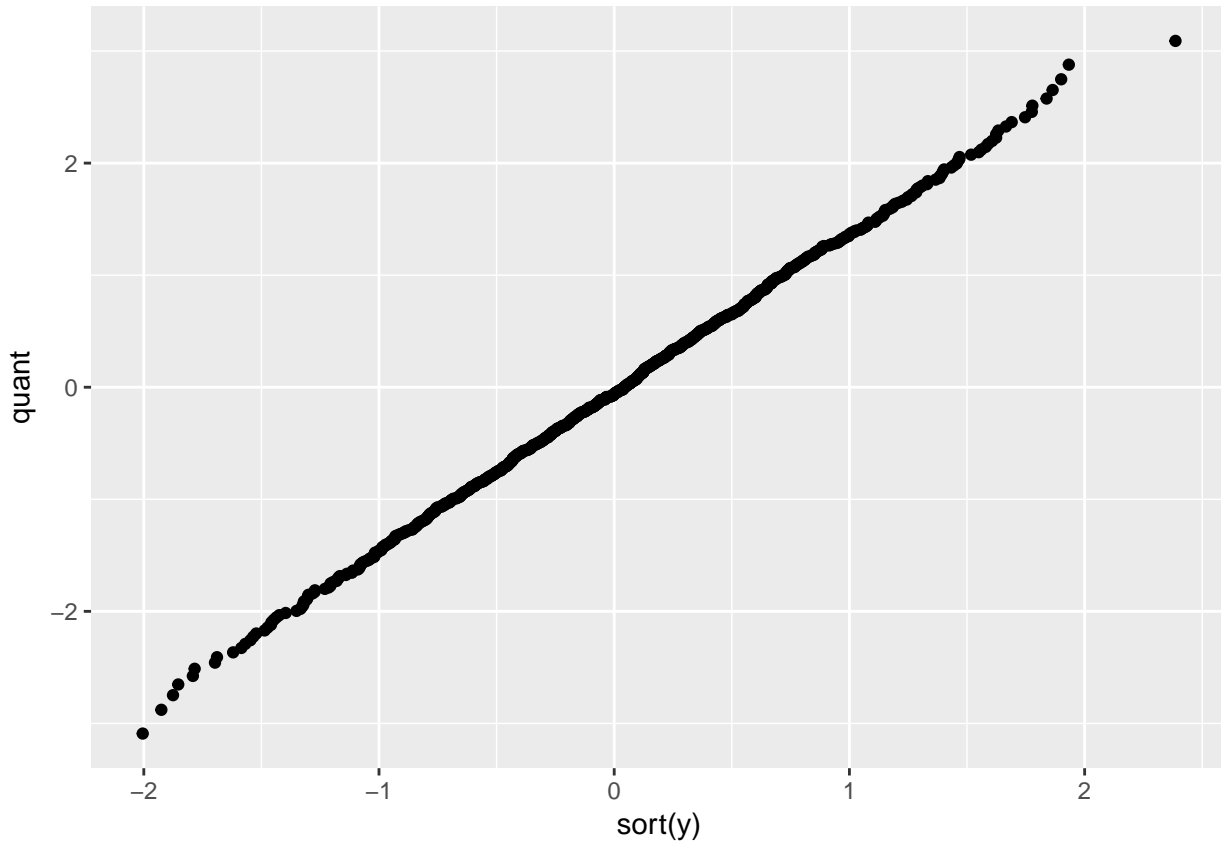
```
ggplot(data=data_Gaussian1, aes(x=sort(x),y=quant))+geom_point()
```



```
ggplot(data=data_Gaussian1, aes(x=y))+ geom_histogram(aes(y = 20*(..count..)/sum(..count..)),
                                                    binwidth=1/20,boundary=0) +
  geom_point(aes(y=0), colour="blue") + geom_line(data=dens_gauss, aes(x=t,y=density),color="red")
```



```
ggplot(data=data_Gaussian1, aes(x=sort(y),y=quant))+geom_point()
```



3. Propose an algorithm to sample realizations of a Gaussian vector $Z \sim \mathcal{N}(m, \Sigma)$ where $m \in \mathbb{R}^n$ and Σ is a positive symmetric matrix of size $n \times n$.

In the case of a Gaussian random variable (case where $n = 1$), define $Z = \sigma X + m$, where $X \sim \mathcal{N}(0, 1)$, $m \in \mathbb{R}$, $\sigma^2 > 0$. Then Z is distributed from a Gaussian distribution with mean m and variance σ^2 .

We can do the same in the multivariate case. We define the square root of a positive symmetric matrix as follows. Any positive matrix is diagonalizable so that $\Sigma = PDP^T$ where P is change of basis matrix and D is diagonal. Then the square root of Σ is the following positive symmetric matrix $\sqrt{\Sigma} = P\sqrt{D}P^T$, where \sqrt{D} is the diagonal matrix where the diagonal values are the square root of the diagonal value of D . One can check that $\sqrt{\Sigma} = \sqrt{\Sigma}^T$ and $\sqrt{\Sigma}\sqrt{\Sigma}^T = \Sigma$.

Let X be a standard Gaussian vector of size n and $Z = \sqrt{\Sigma}X + m$. Then Z is a Gaussian vector with mean m and covariance Σ .

```
library(expm)
```

```
## Loading required package: Matrix
```

```
##
```

```
## Attaching package: 'expm'
```

```
## The following object is masked from 'package:Matrix':
```

```
##
```

```
## expm
```

```
rGaussian_multivariate <- function(m,Sigma){
  d <- length(m);
  dGauss_standard<-rGaussian(m);
  x <- dGauss_standard$x;
```

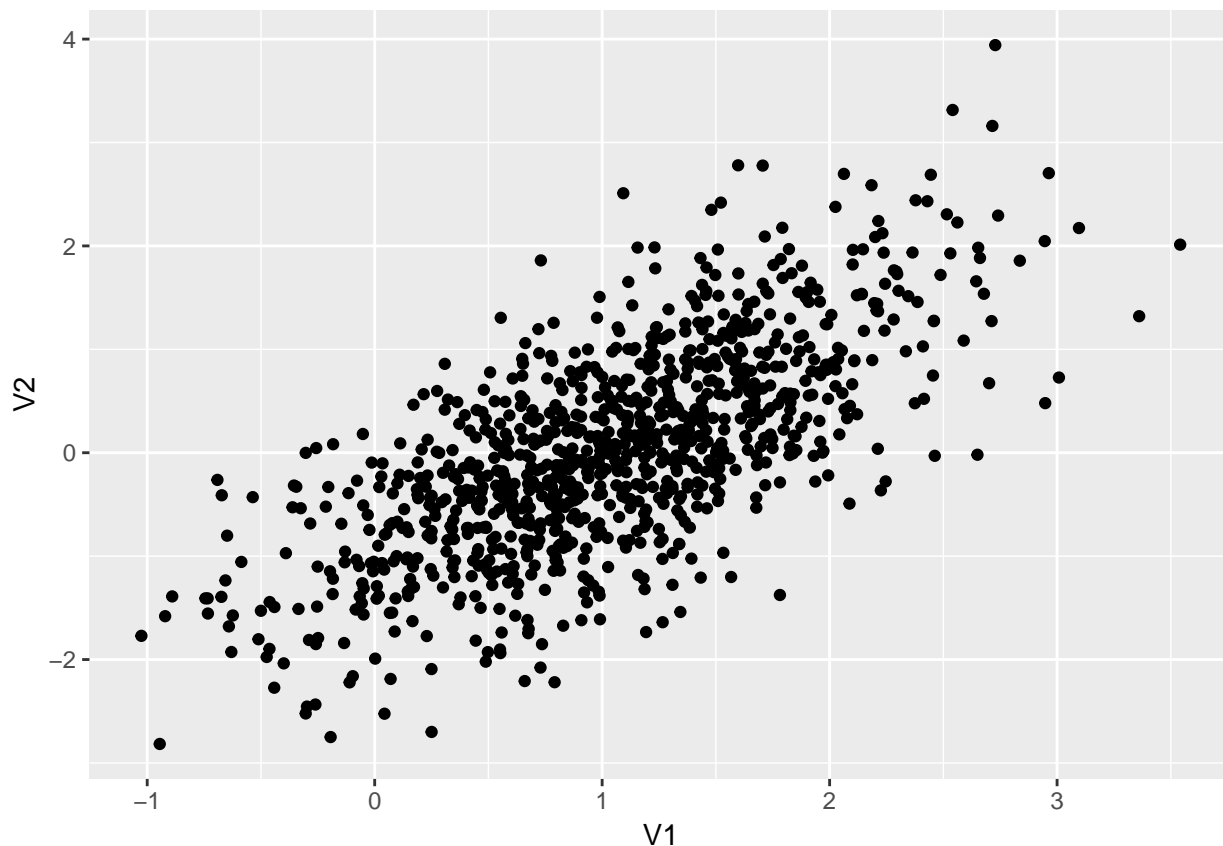


```

  z <- Re(sqrtm(Sigma))%*%x+m
  return(z)
}

m<- c(1,0)
Sigma=matrix(c(1,1,1,2), nrow=2, byrow=T)
n <- 1000
dgauss <-t(rGaussian_multivariate(m,Sigma))
for (i in 1:(n-1)){
  dgauss <- rbind(dgauss,t(rGaussian_multivariate(m,Sigma)))
}
data_Gaussian_mult <- as.data.frame(dgauss)
ggplot(data=data_Gaussian_mult, aes(x=V1,y=V2))+ geom_point()

```



Another class of algorithms is the rejection sampling. You can learn more on this method pages 92 and 93 of your reference book “Mathematical statistics and data analysis” by Rice.

Another class of algorithms is the rejection sampling. You can learn more on this method pages 92 and 93 of your reference book “Mathematical statistics and data analysis” by Rice.

II Estimation

In this part, we are going to illustrate the properties of some estimators using Monte-Carlo algorithms. In Monte-Carlo algorithms the expectation $\mathbb{E}_{X \sim P}(f(X))$ is approximating via $\frac{1}{n} \sum_{i=1}^n f(x_i)$ where (x_1, \dots, x_n) is a sample from P .

Exercise 4: Binomial distribution

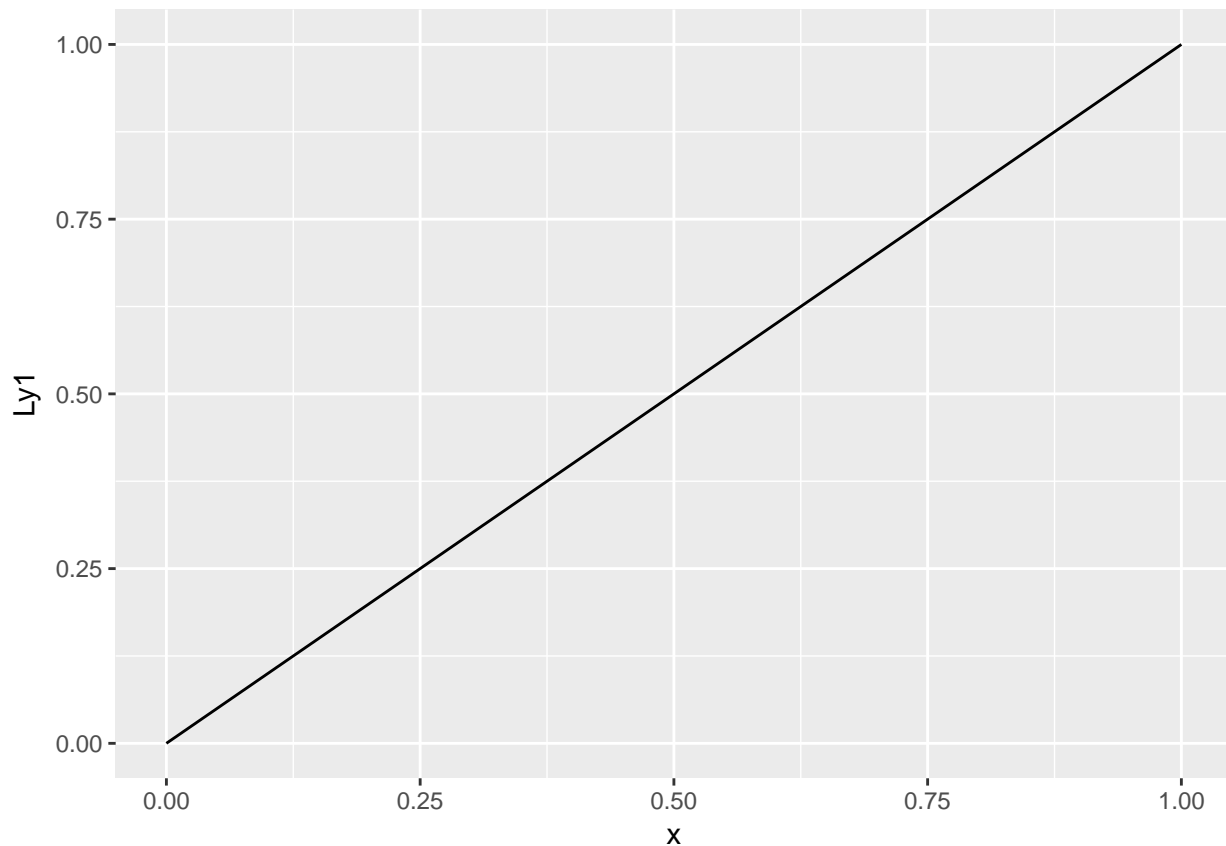
Let Y be distributed from a Bernoulli distribution with parameter p and let X be distributed from a binomial distribution $\text{Bin}(n, p)$ with $p \in (0, 1)$ and $n \in \mathbb{N}$.

1. Sample one realization y of $Y \sim \mathcal{B}(p)$ with $p = 0.4$. Plot the likelihood associated to this observed value y with respect to p . Plot the likelihood associated to 4 other observed values in the same graph. What are the maximum likelihood estimates in the five cases?

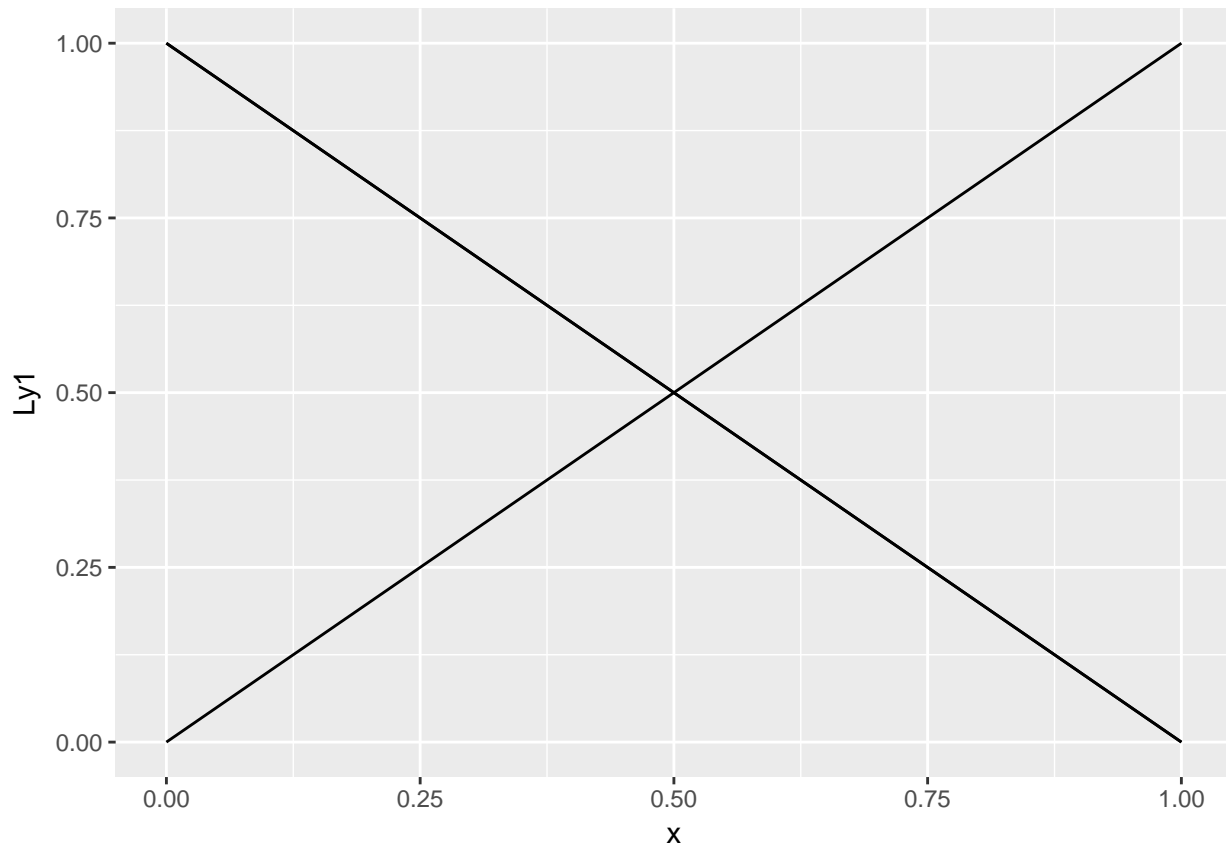
The likelihood is

$$\mathcal{L}(p; y) = p^y (1 - p)^{1-y}.$$

```
p <- 0.4
y1 <- rbinom(1,size=1,prob=p)
x <- seq(0,1,by=0.001)
Ly1 <- x^y1*(1-x)^(1-y1)
xy <- data.frame(x=x,Ly1=Ly1)
p1 <- ggplot(data=xy, aes(x=x,y=Ly1))+geom_line()
p1
```



```
y2<- rbinom(1,size=1,prob=p)
Ly2 <- x^y2*(1-x)^(1-y2)
xy$Ly2 <- Ly2
y3<- rbinom(1,size=1,prob=p)
Ly3 <- x^y3*(1-x)^(1-y3)
xy$Ly3 <- Ly3
p1+geom_line(data=xy, aes(x=x,y=Ly2))+geom_line(data=xy, aes(x=x,y=Ly3))
```



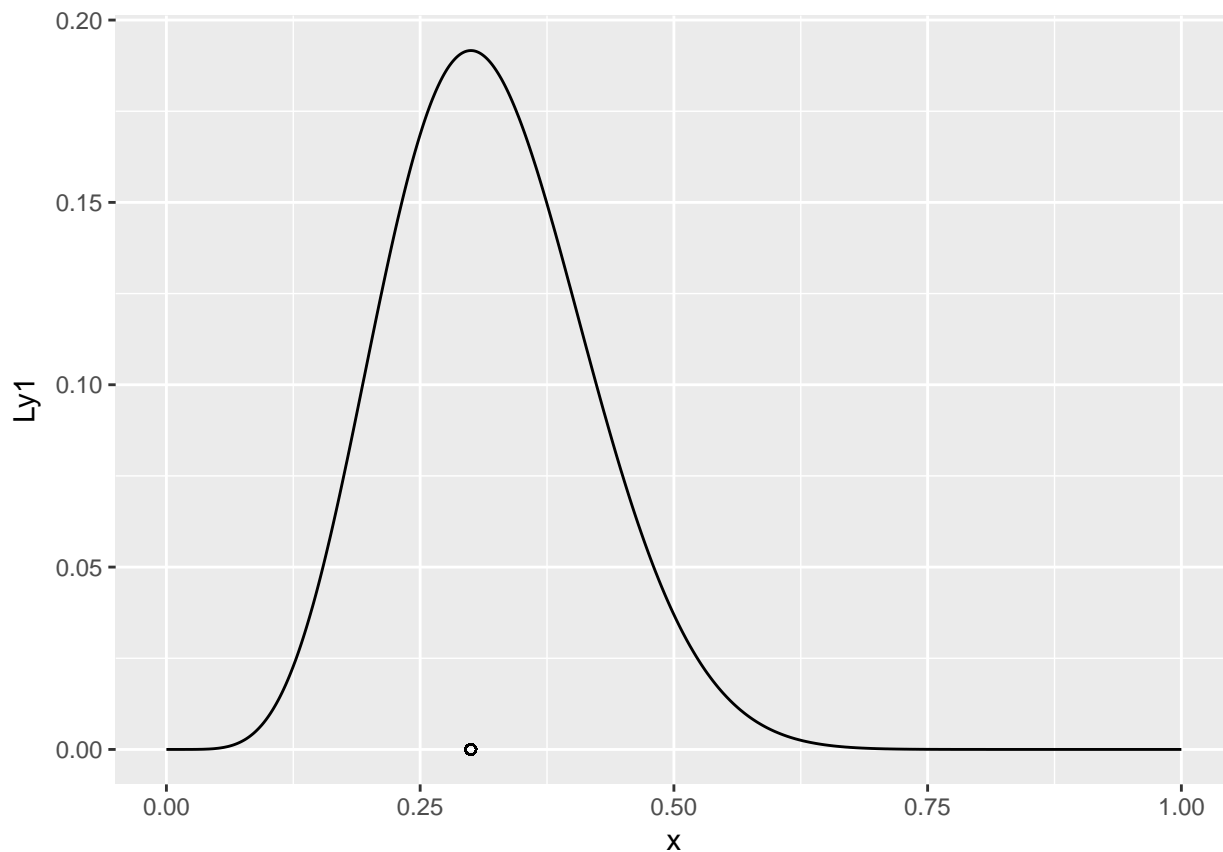
The maximum likelihood estimate equals 1 when $y = 1$ and 0 otherwise, i.e. the maximum likelihood estimator equals y .

2. Sample one realization x_1 of $X \sim \text{Bin}(n, p)$ with $p = 0.4$ and $n = 20$. Plot the likelihood associated to this observed value x . Plot the likelihood associated to 4 other observed values in the same graph. What are the maximum likelihood estimates in the five cases? In exercise 5 of PC3, 4, we obtained that the mle in this model is unique and equals $\hat{p}_{MLE} = X/n$.

The likelihood associated to x_1 is

$$p \mapsto \mathcal{L}(p; x_1) = \binom{n}{x_1} p^{x_1} * (1 - p)^{n-x_1}.$$

```
n <- 20
p <- 0.4
x <- seq(0,1,by=0.001)
x1 <- rbinom(1,n,p)
Ly1 <- choose(n, x1)*x^x1*(1-x)^(n-x1)
xy <- data.frame(x=x,Ly1=Ly1)
pl <- ggplot(data=xy, aes(x=x,y=Ly1))+geom_line()+ geom_point(aes(x=x1/n,y=0),shape=1)
pl
```



```

x2 <- rbinom(1,n,p)
Ly2 <- choose(n, x2)*x^x2*(1-x)^(n-x2)
xy$Ly2 <- Ly2
p1 <- p1+geom_line(data=xy, aes(x=x,y=Ly2),color="red")+ geom_point(aes(x=x2/n,y=0),color="red",shape=1)

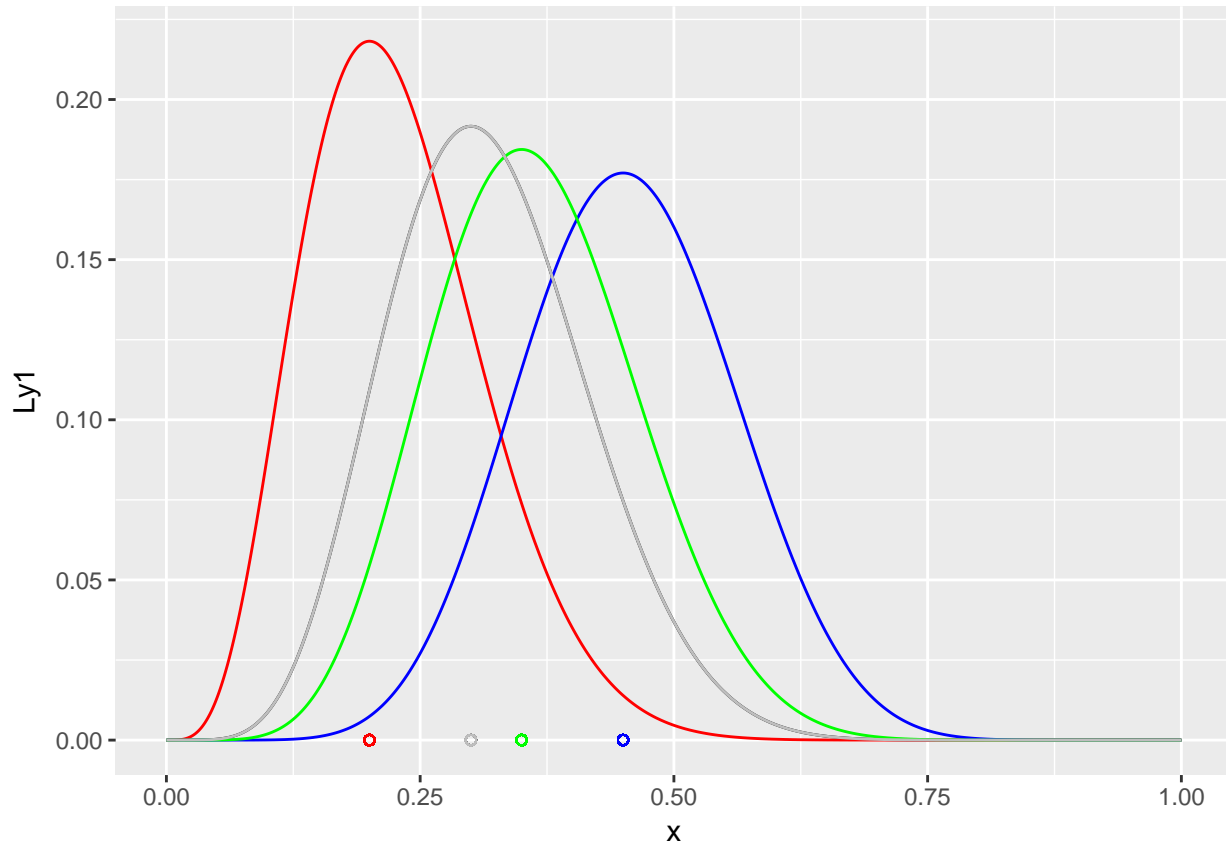
x3 <- rbinom(1,n,p)
Ly3 <- choose(n, x3)*x^x3*(1-x)^(n-x3)
xy$Ly3 <- Ly3
p1 <- p1 + geom_line(data=xy, aes(x=x,y=Ly3),color="blue")+ geom_point(aes(x=x3/n,y=0),color="blue",
                                                                    shape=1)

x4 <- rbinom(1,n,p)
Ly4 <- choose(n, x4)*x^x4*(1-x)^(n-x4)
xy$Ly4 <- Ly4
p1 <- p1 + geom_line(data=xy, aes(x=x,y=Ly4),color="green")+ geom_point(aes(x=x4/n,y=0),color="green",
                                                                    shape=1)

x5 <- rbinom(1,n,p)
Ly5 <- choose(n, x5)*x^x5*(1-x)^(n-x5)
xy$Ly5 <- Ly5
p1 <- p1 + geom_line(data=xy, aes(x=x,y=Ly5),color="grey")+ geom_point(aes(x=x5/n,y=0),color="grey",
                                                                    shape=1)

p1

```



```
c(x1/n,x2/n,x3/n,x4/n,x5/n)
```

```
## [1] 0.30 0.20 0.45 0.35 0.30
```

3. . We consider a Beta $\text{Beta}(2, 2)$ prior on p in $(0, 1)$. Plot the prior distribution and the posterior distribution associated to the realization x_1 . Plot the posterior distribution associated to 4 other observed values obtained in the previous question in the same graph. What are the maximum a posteriori estimates in the five cases?

In the case where the prior is uniform $\mathcal{U}(0, 1)$, the posterior distribution is the beta distribution $\text{Beta}(x_1 + 1, n - x_1 + 1)$ and the MAP estimator equals the MLE (see exercise 5 of PC3_4).

In the case of Beta $\text{Beta}(2, 2)$ prior, the posterior distribution is the beta distribution $\text{Beta}(x_1 + 2, n - x_1 + 2)$ and the MAP estimator equals $\hat{p}_{MAP} = \frac{x_1 + 1}{n + 2}$, the MAP estimate is then $\frac{x_1 + 1}{n + 2}$.

```
post1 <- dbeta(x, shape1=x1+2, shape2=n-x1+2)
xy$post1 <- post1
plpost <- ggplot(data=xy) + geom_line(data=xy, aes(x=x, y=post1), color="black", linetype = "dashed") +
  geom_point(aes(x=(x1+1)/(n+2), y=0), color="black", shape=3)

post2 <- dbeta(x, shape1=x2+2, shape2=n-x2+2)
xy$post2 <- post2
plpost <- plpost + geom_line(data=xy, aes(x=x, y=post2), color="red", linetype = "dashed") +
  geom_point(aes(x=(x2+1)/(n+2), y=0), color="red", shape=3)

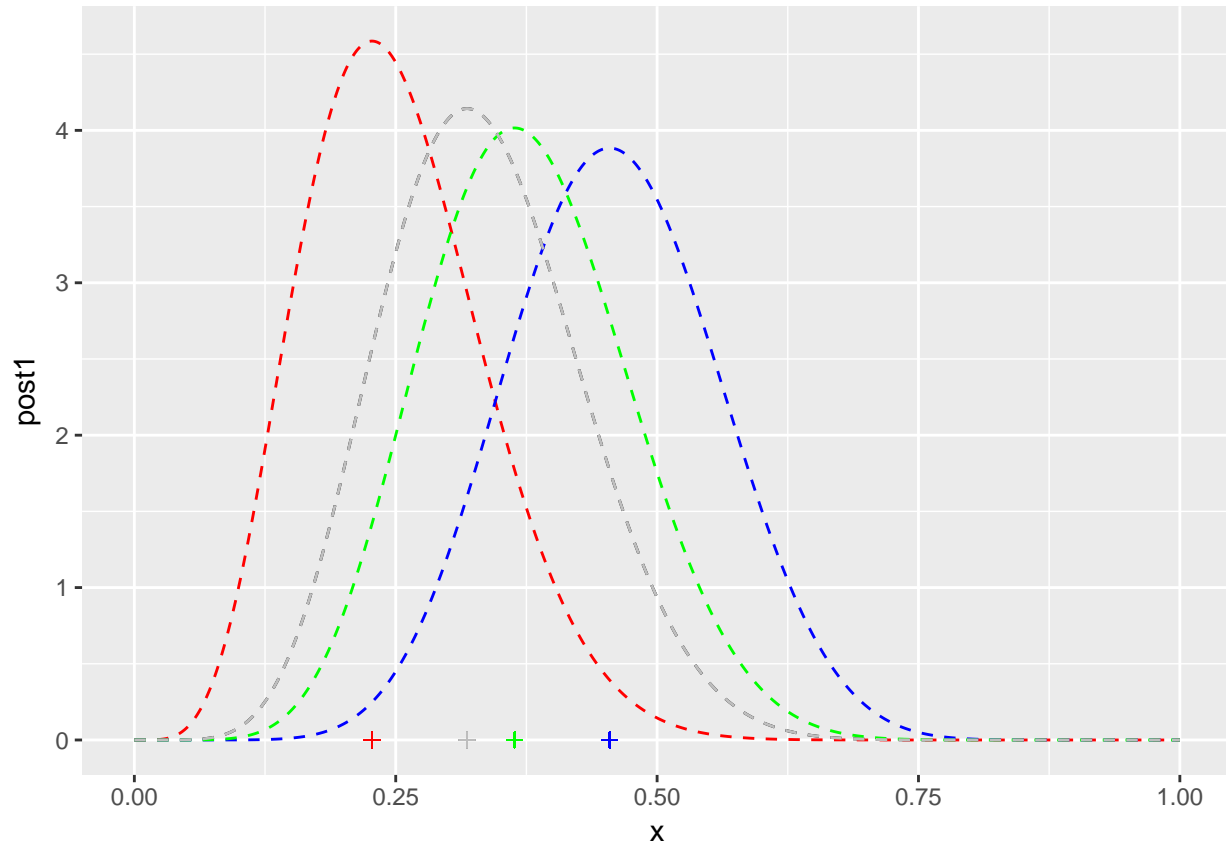
post3 <- dbeta(x, shape1=x3+2, shape2=n-x3+2)
xy$posty3 <- post3
plpost <- plpost + geom_line(data=xy, aes(x=x, y=post3), color="blue", linetype = "dashed") +
  geom_point(aes(x=(x3+1)/(n+2), y=0), color="blue", shape=3)
```

```

post4 <- dbeta(x,shape1=x4+2,shape2=n-x4+2)
xy$post4 <- post4
plpost <- plpost + geom_line(data=xy, aes(x=x,y=post4),color="green",linetype = "dashed") +
  geom_point(aes(x=(x4+1)/(n+2),y=0),color="green",shape=3)

post5 <- dbeta(x,shape1=x5+2,shape2=n-x5+2)
xy$post5 <- post5
plpost <- plpost + geom_line(data=xy, aes(x=x,y=post5),color="grey",linetype = "dashed") +
  geom_point(aes(x=(x5+1)/(n+2),y=0),color="grey",shape=3)
plpost

```



```

c((x1+1)/(n+2),(x2+1)/(n+2),(x3+1)/(n+2),(x4+1)/(n+2),(x5+1)/(n+2))

```

```
## [1] 0.3181818 0.2272727 0.4545455 0.3636364 0.3181818
```

```

c(x1/n,x2/n,x3/n,x4/n,x5/n)

```

```
## [1] 0.30 0.20 0.45 0.35 0.30
```

4. Compare the bias, variance and mse of the two previous estimators \hat{p}_{MLE} and \hat{p}_{MAP} thanks to simulations for $p \in \{0.4, 0.95\}$ and $n \in \{10, 20, 50\}$. Help: you can compare Monte-Carlo approximations of the variance and the bias with the theoretical ones obtained in PC3_4. Also plot histograms of the obtained estimates.

```

n <- 10
p <- 0.4
nMC <- 1000
x <- rbinom(nMC,n,p)

```

```

pmle <- 1/n*x
pmap <- 1/(n+2)*(x+1)
bias_approx_MC_MLE <- mean(pmle)-p
bias_approx_MC_MAP <- mean(pmap)-p
var_approx_MC_MLE <- (nMC-1)/nMC*var(pmle)
var_approx_MC_MAP <- (nMC-1)/nMC*var(pmap)
mse_approx_MC_MLE <- bias_approx_MC_MLE^2 + var_approx_MC_MLE
mse_approx_MC_MAP <- bias_approx_MC_MAP^2 + var_approx_MC_MAP
data_map_mle_binom <- data.frame(x=x,pest=pmle,est="MLE",n=rep(n,nMC),p=rep(p,nMC))
data_map_mle_binom <- rbind(data_map_mle_binom,data.frame(x=x,pest=pmap,est="MAP",n=rep(n,nMC),
p=rep(p,nMC)))
mse_binom <-data.frame(bias_MC=bias_approx_MC_MLE,bias=0,var_MC=var_approx_MC_MLE,var=p*(1-p)/n,
mse_MC=mse_approx_MC_MLE,mse=p*(1-p)/n,est=1,n=n,p=p)
mse_binom <- rbind(mse_binom,c(bias_approx_MC_MAP,(1-2*p)/(n+2),var_approx_MC_MAP,n*p*(1-p)/((n+2)^2),
mse_approx_MC_MAP,((n-4)*p*(1-p)+1)/((n+2)^2),2,n,p))

n <- 20
p <- 0.4
nMC <-1000
x <-rbinom(nMC,n,p)
pmle <- 1/n*x
pmap <- 1/(n+2)*(x+1)
bias_approx_MC_MLE <- mean(pmle)-p
bias_approx_MC_MAP <- mean(pmap)-p
var_approx_MC_MLE <- (nMC-1)/nMC*var(pmle)
var_approx_MC_MAP <- (nMC-1)/nMC*var(pmap)
mse_approx_MC_MLE <- bias_approx_MC_MLE^2 + var_approx_MC_MLE
mse_approx_MC_MAP <- bias_approx_MC_MAP^2 + var_approx_MC_MAP

data_map_mle_binom <- rbind(data_map_mle_binom,data.frame(x=x,pest=pmle,est="MLE",n=rep(n,nMC),
p=rep(p,nMC)))
data_map_mle_binom <- rbind(data_map_mle_binom,data.frame(x=x,pest=pmap,est="MAP",n=rep(n,nMC),
p=rep(p,nMC)))
mse_binom <- rbind(mse_binom,c(bias_approx_MC_MLE,0,var_approx_MC_MLE,p*(1-p)/n,mse_approx_MC_MLE,
p*(1-p)/n,1,n,p))
mse_binom <- rbind(mse_binom,c(bias_approx_MC_MAP,(1-2*p)/(n+2),var_approx_MC_MAP,n*p*(1-p)/((n+2)^2),
mse_approx_MC_MAP,((n-4)*p*(1-p)+1)/((n+2)^2),2,n,p))

n <- 50
p <- 0.4
nMC <-1000
x <-rbinom(nMC,n,p)
pmle <- 1/n*x
pmap <- 1/(n+2)*(x+1)
bias_approx_MC_MLE <- mean(pmle)-p
bias_approx_MC_MAP <- mean(pmap)-p
var_approx_MC_MLE <- (nMC-1)/nMC*var(pmle)
var_approx_MC_MAP <- (nMC-1)/nMC*var(pmap)
mse_approx_MC_MLE <- bias_approx_MC_MLE^2 + var_approx_MC_MLE
mse_approx_MC_MAP <- bias_approx_MC_MAP^2 + var_approx_MC_MAP
data_map_mle_binom <- rbind(data_map_mle_binom,data.frame(x=x,pest=pmle,est="MLE",n=rep(n,nMC),
p=rep(p,nMC)))
data_map_mle_binom <- rbind(data_map_mle_binom,data.frame(x=x,pest=pmap,est="MAP",n=rep(n,nMC),
p=rep(p,nMC)))

```

```

mse_binom <- rbind(mse_binom,c(bias_approx_MC_MLE,0,var_approx_MC_MLE,p*(1-p)/n,mse_approx_MC_MLE,
                               p*(1-p)/n,1,n,p))
mse_binom <- rbind(mse_binom,c(bias_approx_MC_MAP,(1-2*p)/(n+2),var_approx_MC_MAP,n*p*(1-p)/((n+2)^2),
                               mse_approx_MC_MAP,((n-4)*p*(1-p)+1)/((n+2)^2),2,n,p))

```

```

n <- 10
p <- 0.95
nMC <-1000
x <-rbinom(nMC,n,p)
pmle <- 1/n*x
pmap <- 1/(n+2)*(x+1)
bias_approx_MC_MLE <- mean(pmle)-p
bias_approx_MC_MAP <- mean(pmap)-p
var_approx_MC_MLE <- (nMC-1)/nMC*var(pmle)
var_approx_MC_MAP <- (nMC-1)/nMC*var(pmap)
mse_approx_MC_MLE <- bias_approx_MC_MLE^2 + var_approx_MC_MLE
mse_approx_MC_MAP <- bias_approx_MC_MAP^2 + var_approx_MC_MAP
data_map_mle_binom <- rbind(data_map_mle_binom,data.frame(x=x,pest=pmle,est="MLE",n=rep(n,nMC),
                                                         p=rep(p,nMC)))
data_map_mle_binom <- rbind(data_map_mle_binom,data.frame(x=x,pest=pmap,est="MAP",n=rep(n,nMC),
                                                         p=rep(p,nMC)))
mse_binom <- rbind(mse_binom,c(bias_approx_MC_MLE,0,var_approx_MC_MLE,p*(1-p)/n,mse_approx_MC_MLE,
                               p*(1-p)/n,1,n,p))
mse_binom <- rbind(mse_binom,c(bias_approx_MC_MAP,(1-2*p)/(n+2),var_approx_MC_MAP,n*p*(1-p)/((n+2)^2),
                               mse_approx_MC_MAP,((n-4)*p*(1-p)+1)/((n+2)^2),2,n,p))

```

```

n <- 20
p <- 0.95
nMC <-1000
x <-rbinom(nMC,n,p)
pmle <- 1/n*x
pmap <- 1/(n+2)*(x+1)
bias_approx_MC_MLE <- mean(pmle)-p
bias_approx_MC_MAP <- mean(pmap)-p
var_approx_MC_MLE <- (nMC-1)/nMC*var(pmle)
var_approx_MC_MAP <- (nMC-1)/nMC*var(pmap)
mse_approx_MC_MLE <- bias_approx_MC_MLE^2 + var_approx_MC_MLE
mse_approx_MC_MAP <- bias_approx_MC_MAP^2 + var_approx_MC_MAP
data_map_mle_binom <- rbind(data_map_mle_binom,data.frame(x=x,pest=pmle,est="MLE",n=rep(n,nMC),
                                                         p=rep(p,nMC)))
data_map_mle_binom <- rbind(data_map_mle_binom,data.frame(x=x,pest=pmap,est="MAP",n=rep(n,nMC),
                                                         p=rep(p,nMC)))
mse_binom <- rbind(mse_binom,c(bias_approx_MC_MLE,0,var_approx_MC_MLE,p*(1-p)/n,mse_approx_MC_MLE,
                               p*(1-p)/n,1,n,p))
mse_binom <- rbind(mse_binom,c(bias_approx_MC_MAP,(1-2*p)/(n+2),var_approx_MC_MAP,n*p*(1-p)/((n+2)^2),
                               mse_approx_MC_MAP,((n-4)*p*(1-p)+1)/((n+2)^2),2,n,p))

```

```

n <- 50
p <- 0.95
nMC <-1000
x <-rbinom(nMC,n,p)
pmle <- 1/n*x
pmap <- 1/(n+2)*(x+1)
bias_approx_MC_MLE <- mean(pmle)-p

```



```

bias_approx_MC_MAP <- mean(pmap)-p
var_approx_MC_MLE <- (nMC-1)/nMC*var(pmle)
var_approx_MC_MAP <- (nMC-1)/nMC*var(pmap)
mse_approx_MC_MLE <- bias_approx_MC_MLE^2 + var_approx_MC_MLE
mse_approx_MC_MAP <- bias_approx_MC_MAP^2 + var_approx_MC_MAP
data_map_mle_binom <- rbind(data_map_mle_binom,data.frame(x=x,pest=pmle,est="MLE",n=rep(n,nMC),
                                                         p=rep(p,nMC)))
data_map_mle_binom <- rbind(data_map_mle_binom,data.frame(x=x,pest=pmap,est="MAP",n=rep(n,nMC),
                                                         p=rep(p,nMC)))
mse_binom <- rbind(mse_binom,c(bias_approx_MC_MLE,0,var_approx_MC_MLE,p*(1-p)/n,mse_approx_MC_MLE,
                              p*(1-p)/n,1,n,p))
mse_binom <- rbind(mse_binom,c(bias_approx_MC_MAP,(1-2*p)/(n+2),var_approx_MC_MAP,n*p*(1-p)/((n+2)^2),
                              mse_approx_MC_MAP,((n-4)*p*(1-p)+1)/((n+2)^2),2,n,p))

```

```
mse_binom
```

```

##      bias_MC      bias      var_MC      var      mse_MC
## 1  0.005800000  0.000000000  0.0255263600  0.0240000000  0.025560000
## 2  0.021500000  0.016666667  0.0177266389  0.0166666667  0.018188889
## 3 -0.002300000  0.000000000  0.0122597100  0.0120000000  0.012265000
## 4  0.007000000  0.009090909  0.0101319917  0.0099173554  0.010180992
## 5 -0.000840000  0.000000000  0.0046880944  0.0048000000  0.004688800
## 6  0.003038462  0.003846154  0.0043344068  0.0044378698  0.004343639
## 7  0.005700000  0.000000000  0.0041075100  0.0047500000  0.004140000
## 8 -0.070250000 -0.075000000  0.0028524375  0.0032986111  0.007787500
## 9 -0.000850000  0.000000000  0.0024417775  0.0023750000  0.002442500
## 10 -0.041681818 -0.040909091  0.0020179979  0.0019628099  0.003755372
## 11  0.000000000  0.000000000  0.0009544000  0.0009500000  0.000954400
## 12 -0.017307692 -0.017307692  0.0008823964  0.0008783284  0.001181953
##      mse est  n  p
## 1  0.024000000  1 10 0.40
## 2  0.016944444  2 10 0.40
## 3  0.012000000  1 20 0.40
## 4  0.010000000  2 20 0.40
## 5  0.004800000  1 50 0.40
## 6  0.004452663  2 50 0.40
## 7  0.004750000  1 10 0.95
## 8  0.008923611  2 10 0.95
## 9  0.002375000  1 20 0.95
## 10 0.003636364  2 20 0.95
## 11 0.000950000  1 50 0.95
## 12 0.001177885  2 50 0.95

```

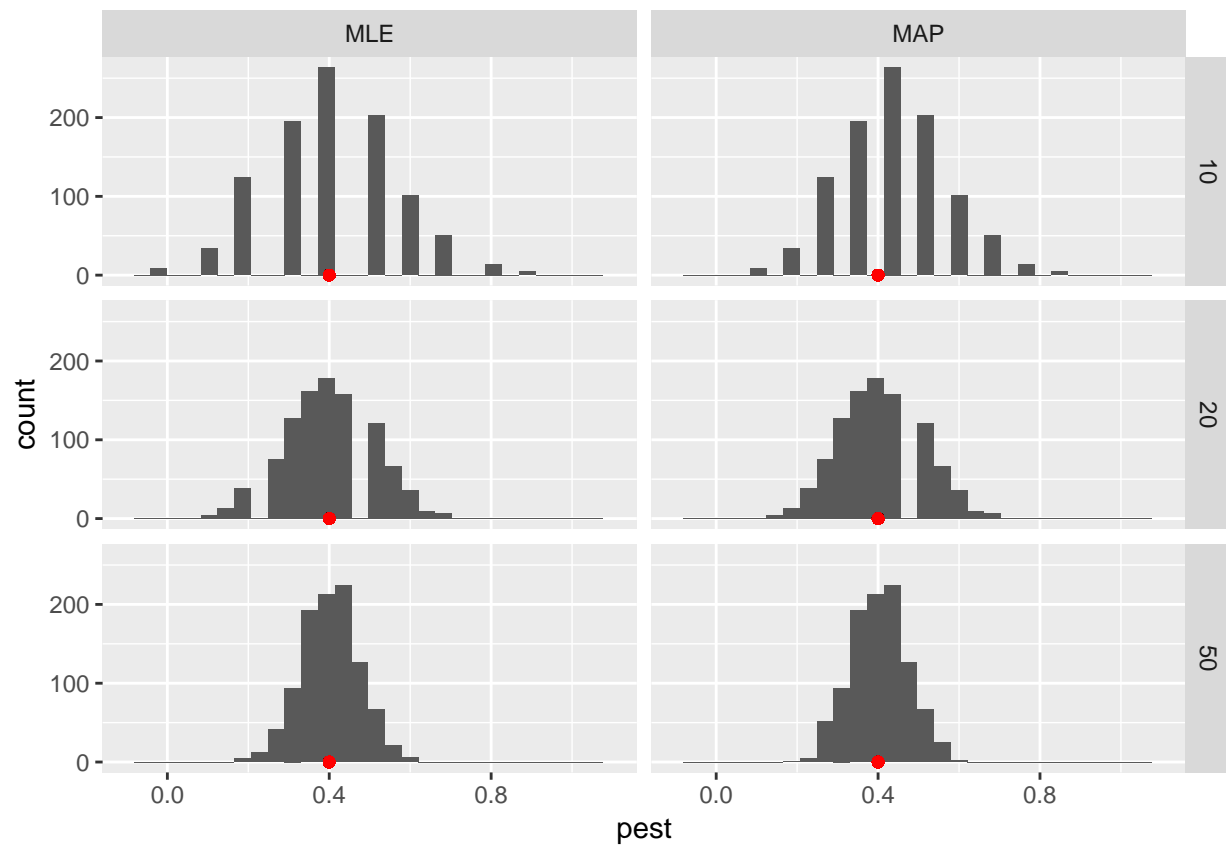
```

ggplot(data=subset(data_map_mle_binom, p<0.5))+geom_histogram(aes(x=pest),boundary=0) +
  scale_x_continuous(limits = c(-0.1,1.1)) +geom_point(aes(x=0.4,y=0),color="red")+facet_grid(n~est)

```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

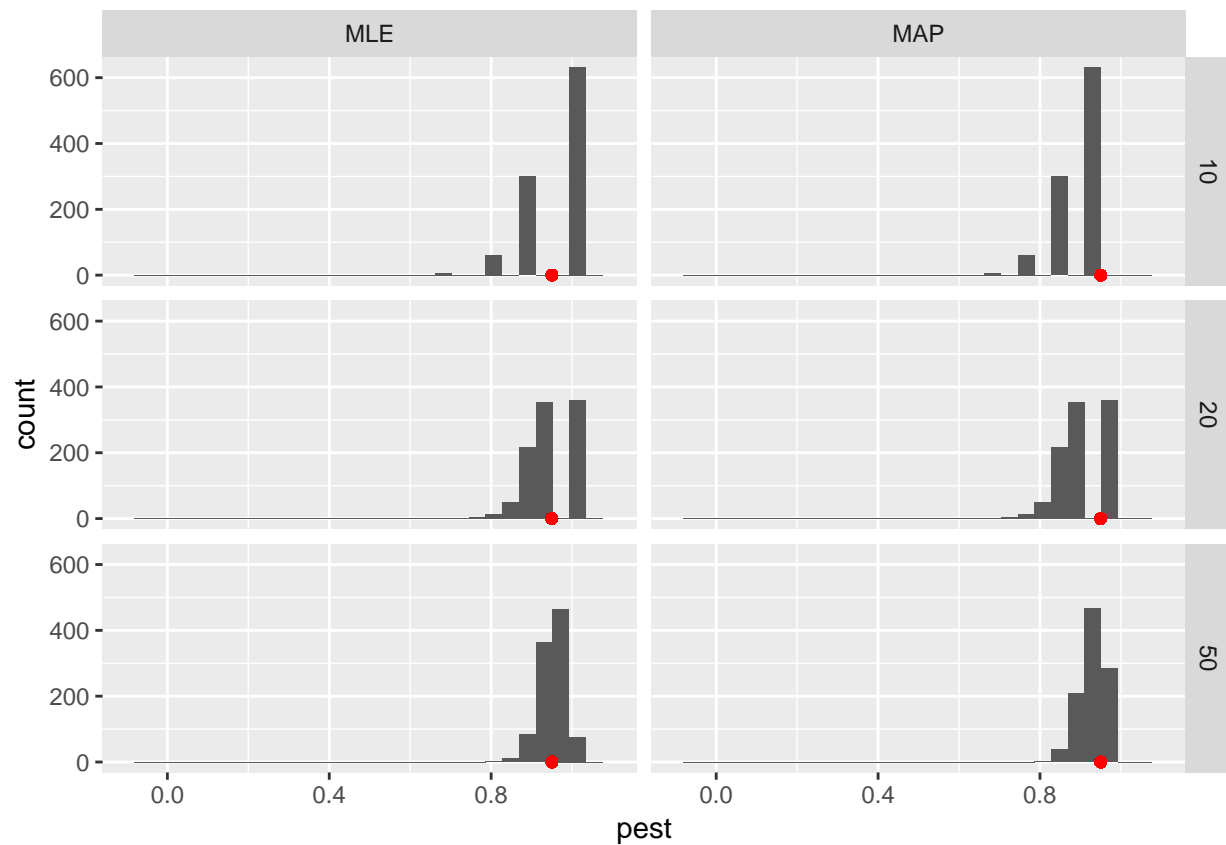
```
## Warning: Removed 6 rows containing missing values (geom_bar).
```



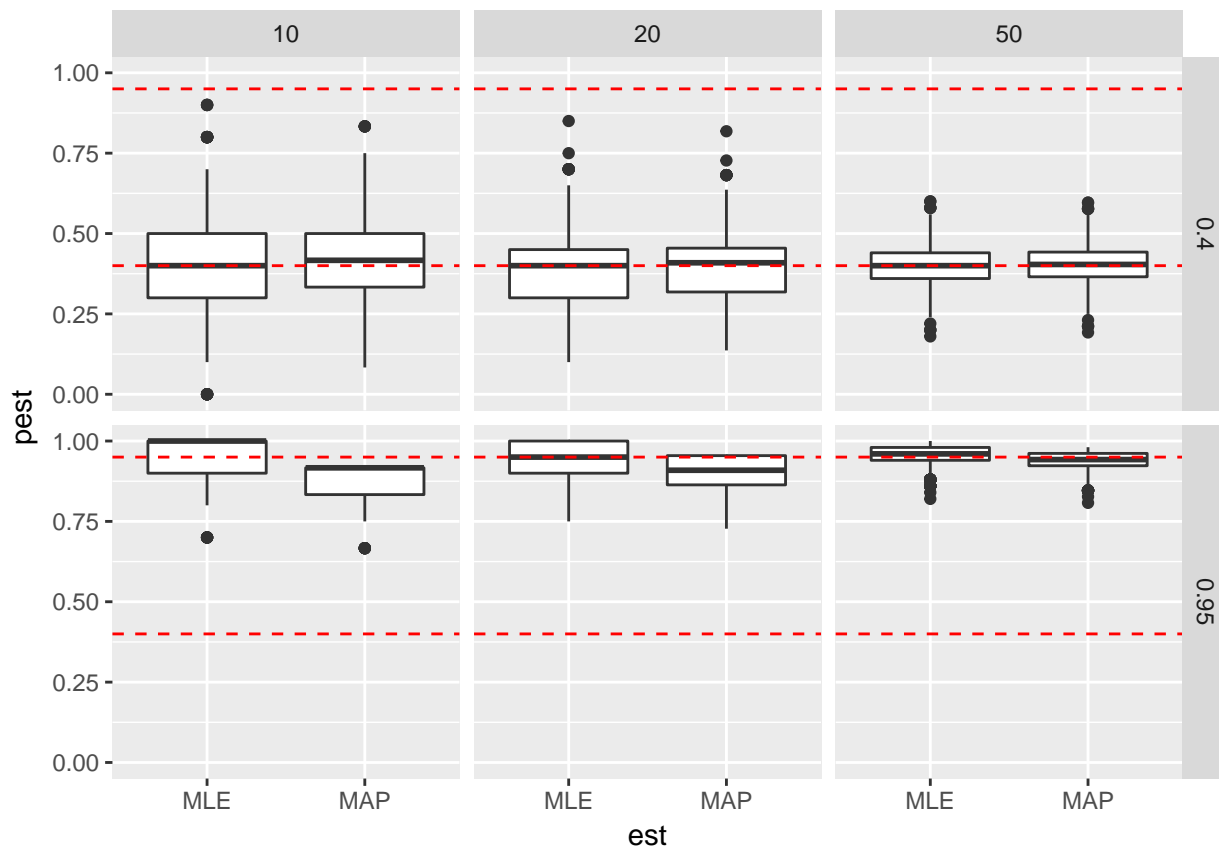
```
ggplot(data=subset(data_map_mle_binom, p>=0.5)) + geom_histogram(aes(x=pest),boundary=0) +
  scale_x_continuous(limits = c(-0.1,1.1)) + geom_point(aes(x=0.95,y=0),color="red")+facet_grid(n~est)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 6 rows containing missing values (geom_bar).
```



```
ggplot(data=data_map_mle_binom) + geom_boxplot(aes(x=est,y=pest,group=est)) + facet_grid(p~n) +
  geom_hline(yintercept=0.4,color="red",linetype="dashed")+geom_hline(yintercept=0.95,
    color="red",linetype="dashed")
```



Exercise 5: Gaussian distribution

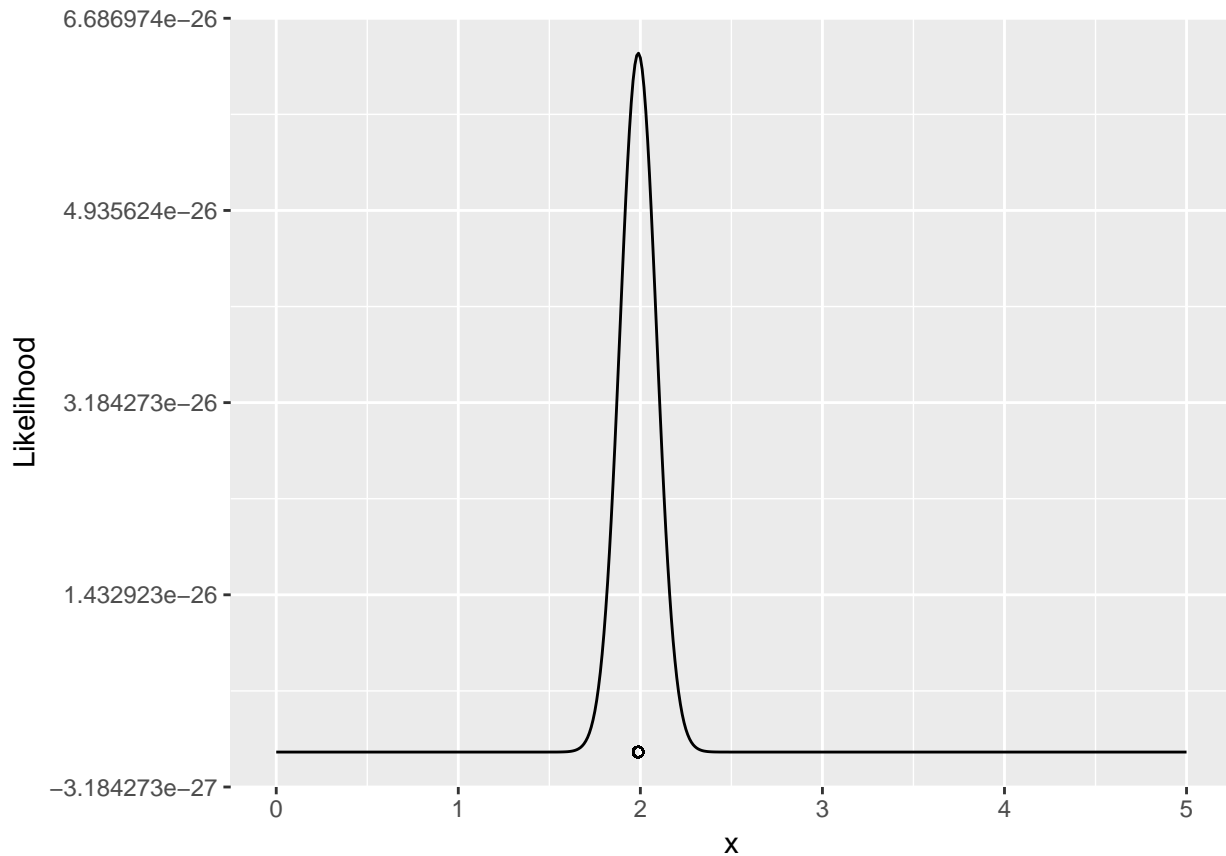
Let X_1, \dots, X_n be i.i.d. random variables distributed from $\mathcal{N}(m, 1)$. Fix $n = 100$ and $m = 2$.

1. Sample a realization (x_1, \dots, x_n) of the vector (X_1, \dots, X_n) . Plot the likelihood associated to this sample with respect to m .

The likelihood is:

$$m \mapsto \mathcal{L}(m, x_1, \dots, x_n) = \left(\frac{1}{\sqrt{2 * \pi}} \right)^n \exp \left(- \sum_{i=1}^n \frac{(x_i - m)^2}{2} \right)$$

```
n<-100
m<-2
M=seq(0,5,by=0.01)
sampleGauss <- rnorm(n,mean=2,sd=1)
Likelihood<-M
for (i in 1:length(M)){
  Likelihood[i] <- exp(-1/2*sum((sampleGauss-M[i])^2))
}
likeli_gauss <- data.frame(x=M,Likelihood=Likelihood)
pl <-ggplot(data=likeli_gauss, aes(x=x,y=Likelihood))+geom_line() +
  geom_point(aes(x=mean(sampleGauss),y=0),shape=1)
pl
```

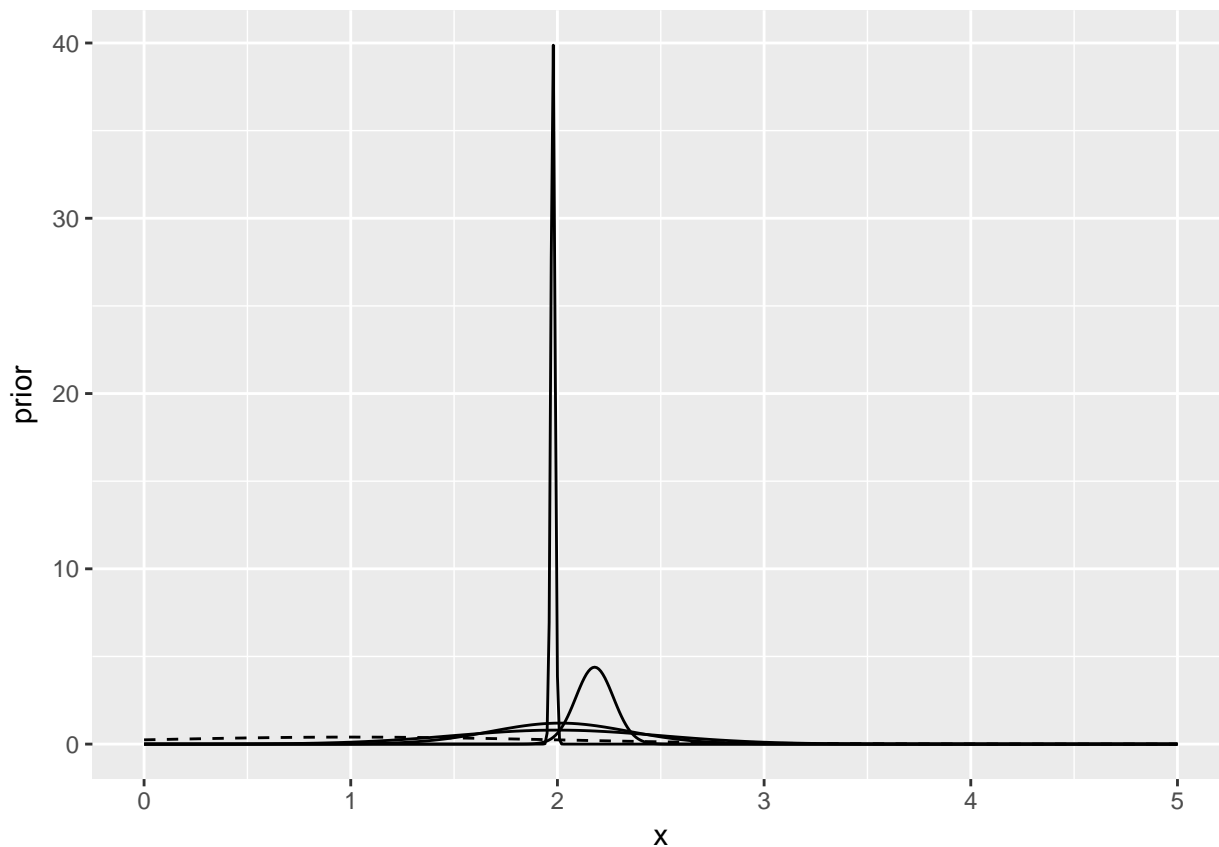


2. We consider a Gaussian prior $\Pi = \mathcal{N}(1, 1)$ on m . Plot the prior distribution and the posterior distributions associated to $\Pi(\cdot|x_1)$, $\Pi(\cdot|x_1, x_2)$, $\Pi(\cdot|x_1, \dots, x_{10})$ and $\Pi(\cdot|x_1, \dots, x_n)$.

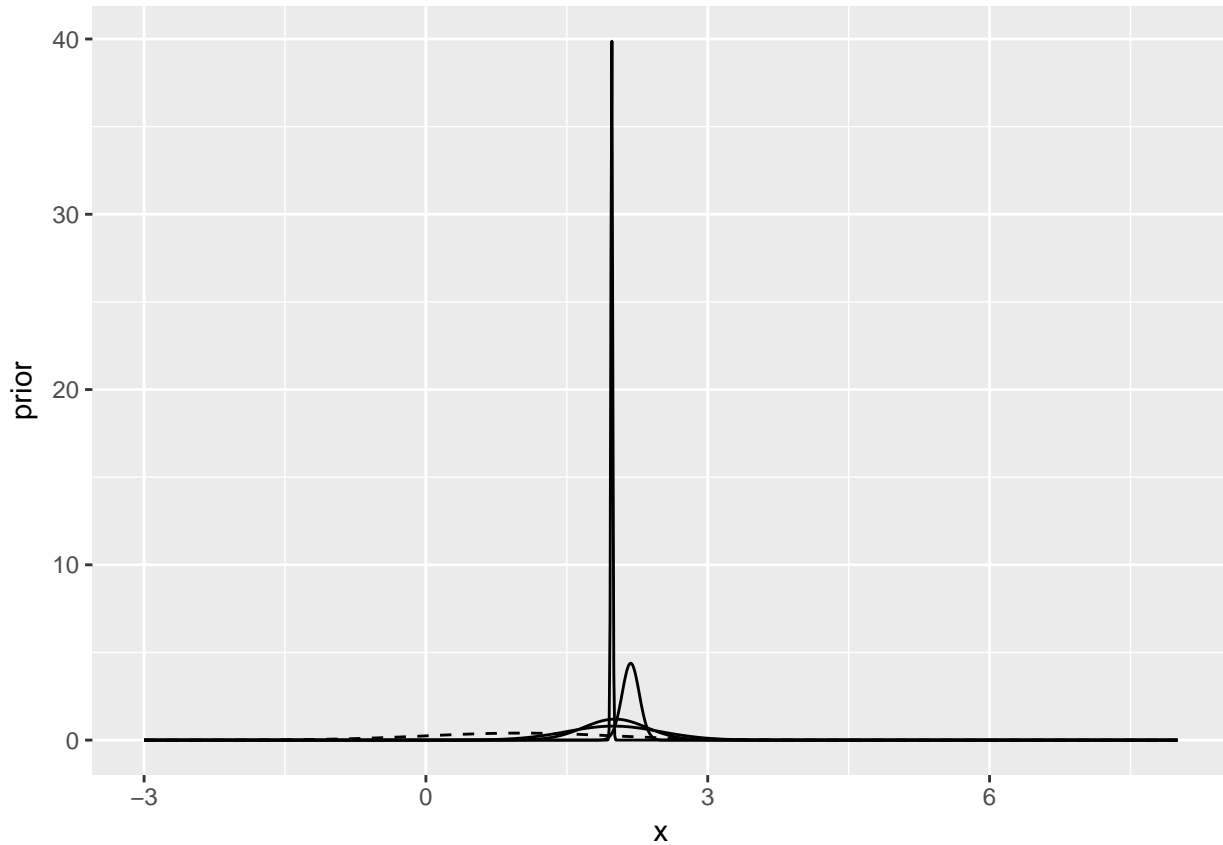
The posterior distribution $\Pi(\cdot|x_1, x_k)$ is the Normal distribution $\mathcal{N}\left(\frac{\frac{1}{k} \sum_{i=1}^k x_i + \frac{1}{k}}{1 + \frac{1}{k}}, \frac{1}{1+k}\right)$.

```
data_post_Gauss <- data.frame(x=M, prior=dnorm(M, 1, 1), post_x1=dnorm(M, (sampleGauss[1]+1)/(1+1), 1/2),
                             post_x2=dnorm(M, (mean(sampleGauss[1:2])+1/2)/(1+1/2), 1/3),
                             post_x10=dnorm(M, (mean(sampleGauss[1:10])+1/10)/(1+1/10), 1/11),
                             post_xn=dnorm(M, mean=(mean(sampleGauss)+1/n)/(1+1/n), sd=1/(n+1)) )

ggplot(data=data_post_Gauss)+geom_line(aes(x=x, y=prior), linetype="dashed") +
  geom_line(aes(x=x, y=post_x1)) + geom_line(aes(x=x, y=post_x2)) +
  geom_line(aes(x=x, y=post_x10)) + geom_line(aes(x=x, y=post_xn))
```



```
tau<-1
mu <- 1
M<-seq(-3,8,by=0.01)
data_post_Gauss <- data.frame(x=M,prior=dnorm(M,mu,tau),
                             post_x1=dnorm(M, (sampleGauss[1]+mu/tau)/(1+1/tau) ,tau/(1+tau)),
                             post_x2=dnorm(M, (mean(sampleGauss[1:2])+mu/(2*tau))/(1+1/(2*tau)) ,
                                             tau/(1+tau*2)),
                             post_x10=dnorm(M, (mean(sampleGauss[1:10])+mu/(10*tau))/(1+1/(10*tau)) ,
                                              tau/(1+tau*10)),
                             post_xn=dnorm(M,mean=(mean(sampleGauss)+mu/(n*tau))/(1+1/(n*tau)),
                                             sd=tau/(1+tau*n) ) )
ggplot(data=data_post_Gauss)+geom_line(aes(x=x,y=prior),linetype="dashed") +
  geom_line(aes(x=x,y=post_x1)) + geom_line(aes(x=x,y=post_x2)) +
  geom_line(aes(x=x,y=post_x10)) + geom_line(aes(x=x,y=post_xn))
```



3. We consider the following estimators $\hat{m}_1 = X_1$, $\hat{m}_2 = \frac{1}{3}(X_1 + X_2 + X_3)$, $\hat{m}_3 = \frac{1}{n} \sum_{i=1}^n X_i$, $\hat{m}_4 = \frac{\hat{m}_3 + 1/n}{1 + 1/n}$ and $\hat{m}_5 = \hat{m}_3 + \frac{10}{n}$. Give the bias and variance of these five estimators. Sample 1000 samples (x_1, \dots, x_{100}) and plot histograms of the five associated estimates.

	Bias	Variance	MSE
\hat{m}_1	0	1	1
\hat{m}_2	0	1/3	1/3
\hat{m}_3	0	1/n	1/n
\hat{m}_4	$(1 - m)/(n + 1)$	$n/(n + 1)^2$	$((1 - m)^2 + n)/(n + 1)^2$
\hat{m}_5	10/n	1/n	11/n

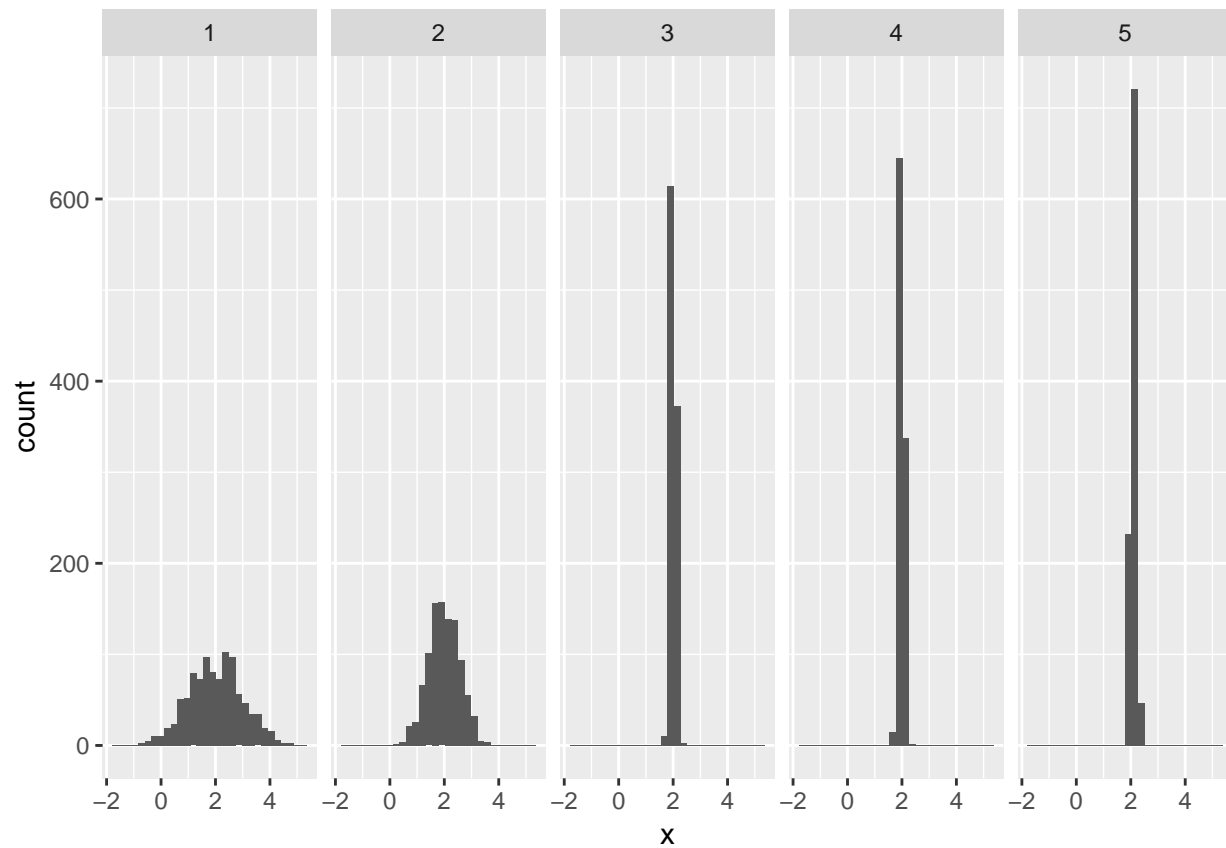
```

m1 <- NULL
m2 <- NULL
m3 <- NULL
m4 <- NULL
m5 <- NULL
for (i in 1:1000){
  xi<-rnorm(n,mean=2,sd=1);
  m1<-c(m1, xi[1]);
  m2<-c(m2, mean(xi[1:3]));
  m3i<- mean(xi);
  m3<-c(m3, m3i);
  m4<-c(m4, (m3i+1/n)/(1+1/n));
  m5<-c(m5, m3i+10/n);
}
data_gauss_MC<- data.frame(x=c(m1,m2,m3,m4,m5),y=c(rep(1,1000),rep(2,1000),rep(3,1000),rep(4,1000),
rep(5,1000)))

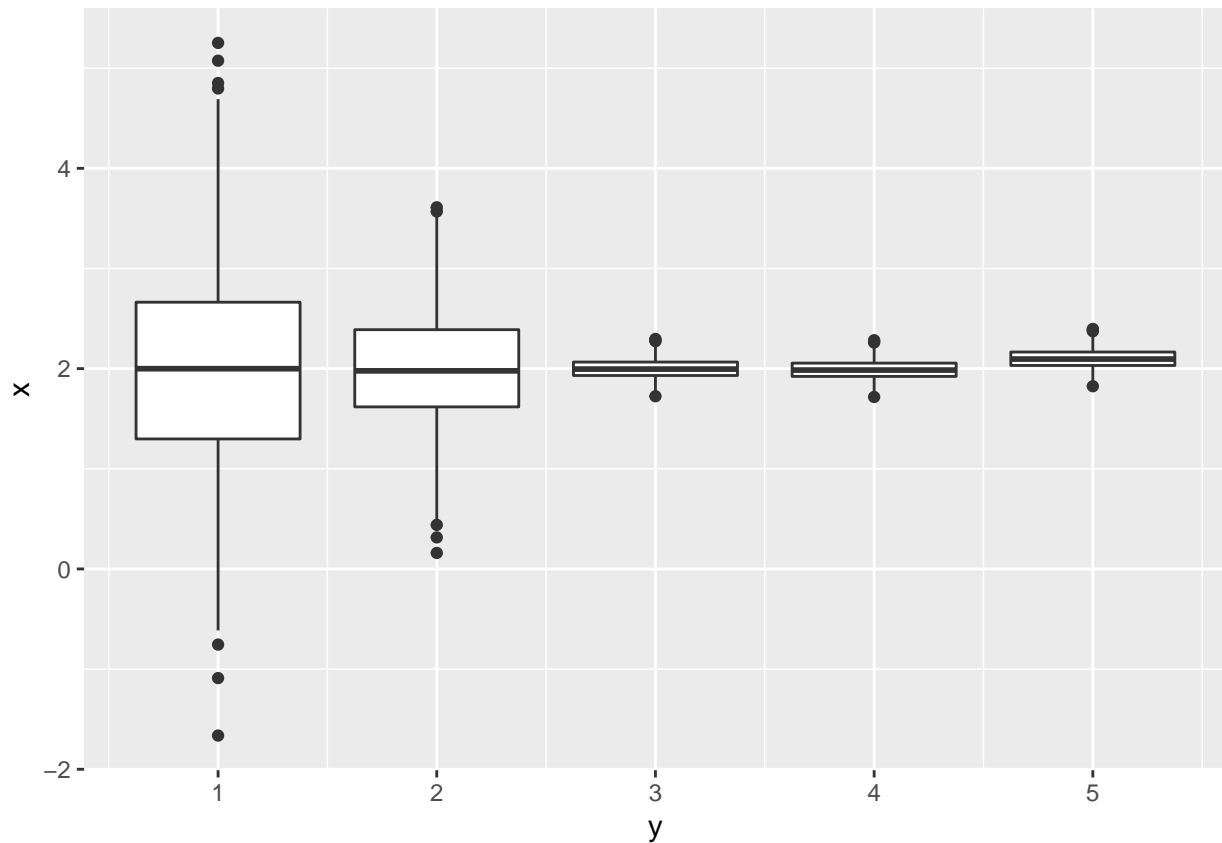
```

```
ggplot(data=data_gauss_MC, aes(x=x))+geom_histogram() + facet_grid(. ~ y)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

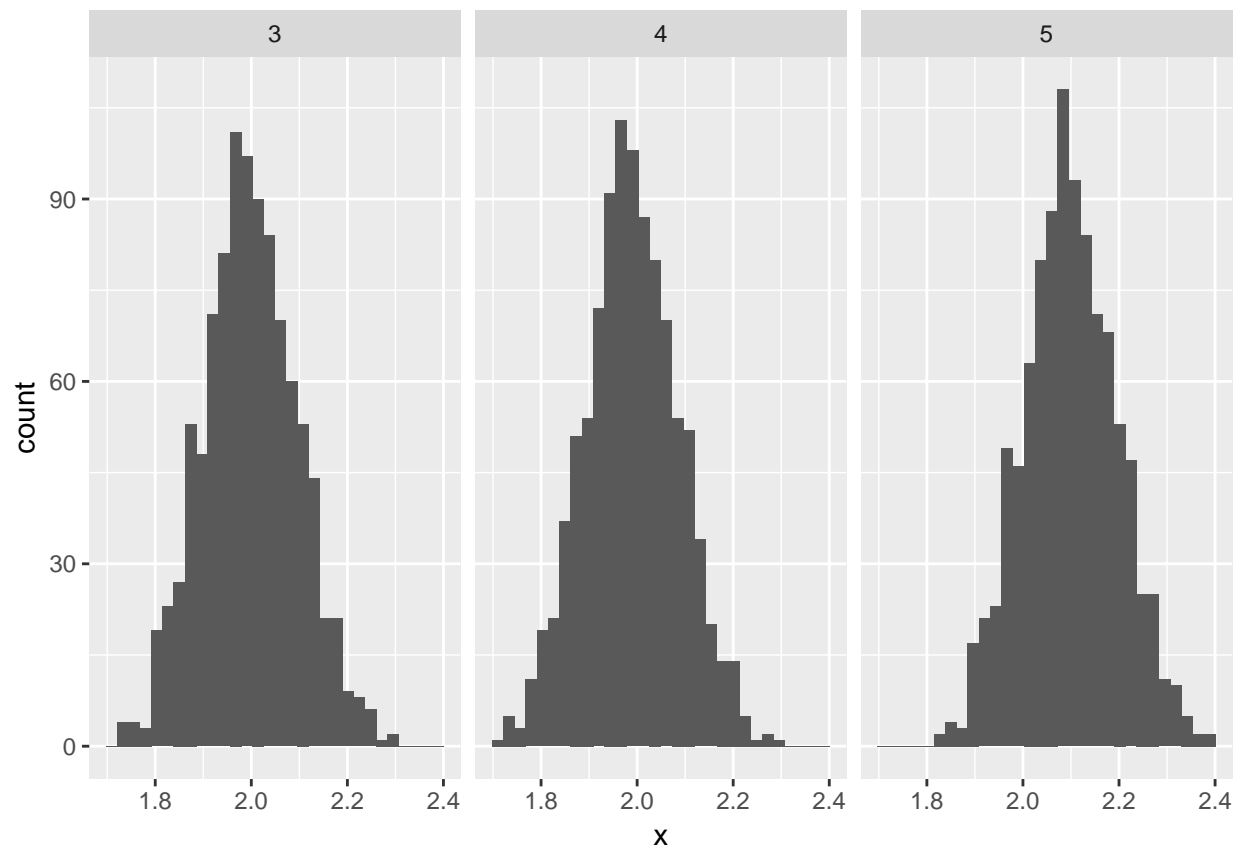


```
ggplot(data=data_gauss_MC, aes(x=y, y=x, group=y)) + geom_boxplot()
```

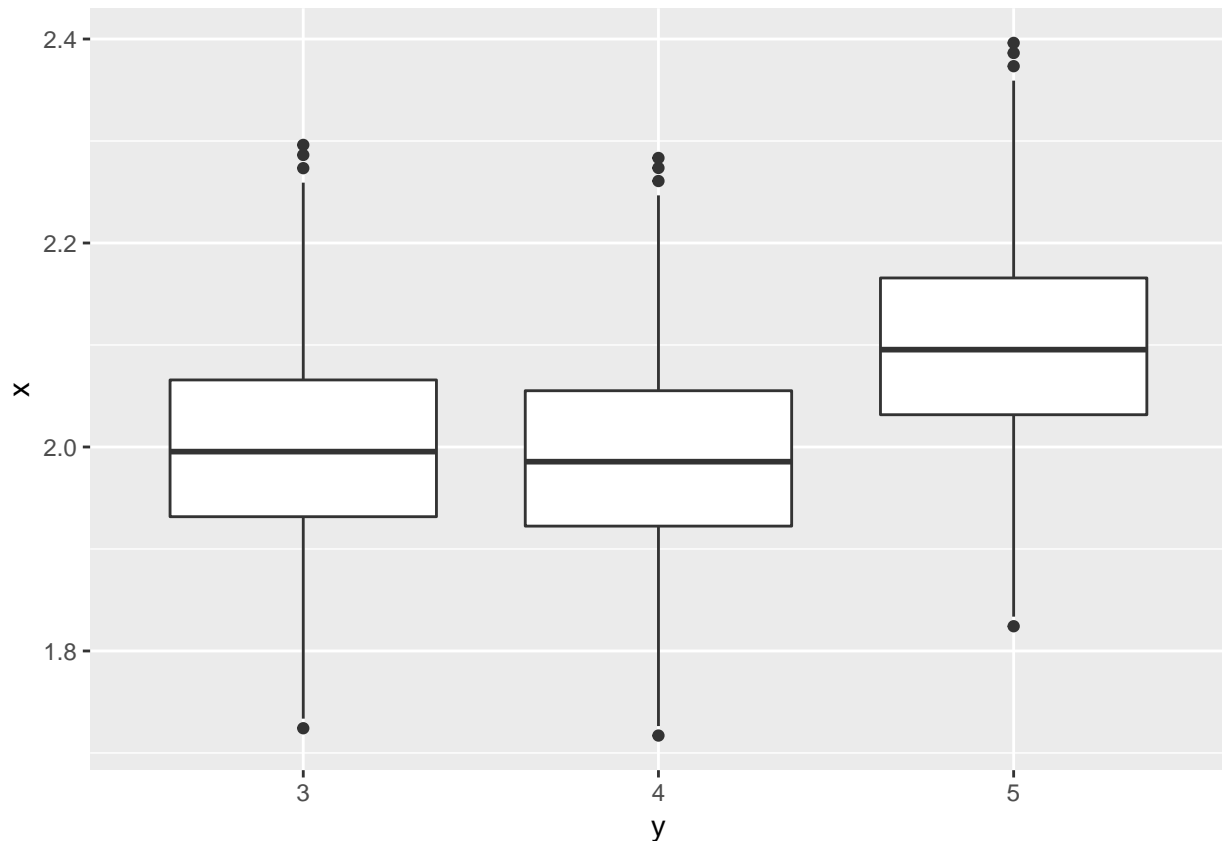



```
data_gauss_MC<- data.frame(x=c(m3,m4,m5),y=as.factor(c(rep(3,1000),rep(4,1000),rep(5,1000))))
ggplot(data=data_gauss_MC, aes(x=x))+geom_histogram()+ facet_grid(. ~ y)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(data=data_gauss_MC,aes(x=y,y=x,group=y)) + geom_boxplot()
```



Exercise 6: Exponential distribution

Let X_1, \dots, X_n be i.i.d. random variables distributed from $\text{Exp}(\lambda)$. In exercise 1 of PC3,4, we have obtained the following estimators using the method of moments:

$$\hat{\lambda}_1 = \frac{n}{\sum_{i=1}^n X_i}, \quad \hat{\lambda}_2 = \sqrt{\frac{2}{\frac{1}{n} \sum_{i=1}^n X_i^2}} \quad \text{and} \quad \hat{\lambda}_{3,t_0} = \frac{\log\left(\frac{1}{n} \sum_{i=1}^n 1_{(t_0, +\infty)}(X_i)\right)}{t_0}.$$

1. Compare these estimators thanks to simulations.

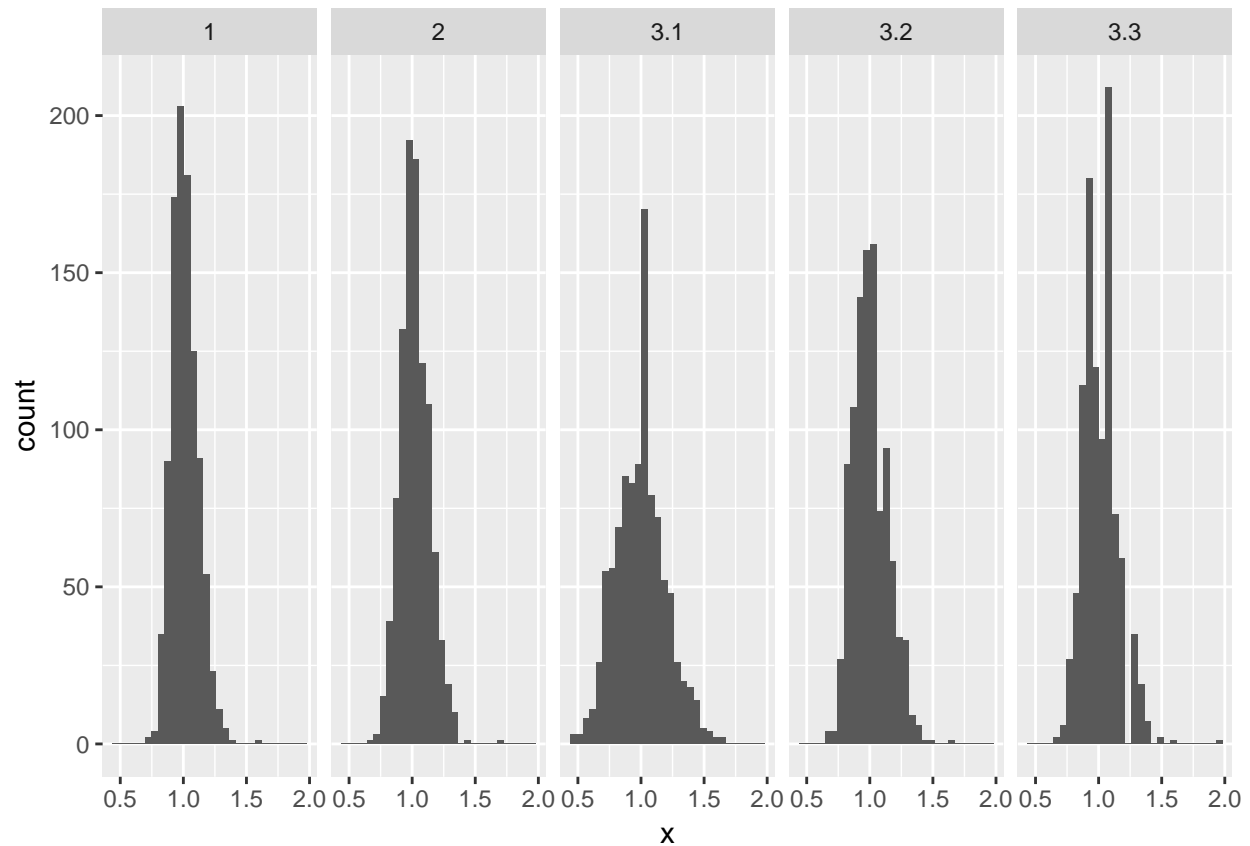
```
lambda<-1
n<-100
t01<-0.3
t02<-1
t03<-2
l1 <- NULL
l2 <- NULL
l3 <- NULL
l31<-NULL
l32<-NULL
l33<-NULL
for (i in 1:1000){
  xi<-rexp(n,rate=lambda);
  l1<-c(l1, 1/mean(xi));
  l2<-c(l2, sqrt(2/mean(xi^2)));
  l31<-c(l31, -log(mean(xi>t01))/t01);
  l32<-c(l32, -log(mean(xi>t02))/t02);
```

```

133<-c(133, -log(mean(xi>t03))/t03);
}
data_exp_MC<- data.frame(x=c(11,12,131,132,133),y=as.factor(c(rep(1,1000),rep(2,1000),rep(3.1,1000),
                                                                rep(3.2,1000),rep(3.3,1000))))
ggplot(data=data_exp_MC, aes(x=x))+geom_histogram()+ facet_grid(. ~ y)

```

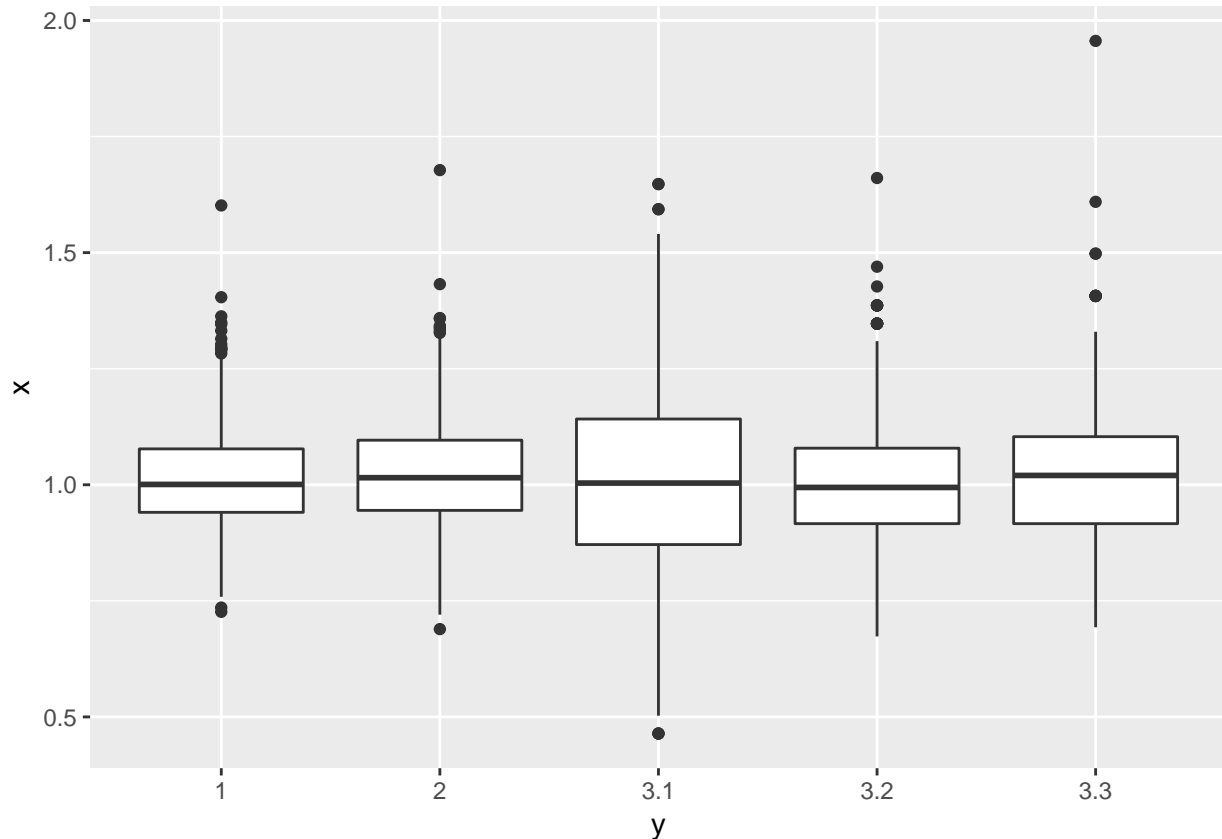
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



```

ggplot(data=data_exp_MC,aes(x=y,y=x,group=y)) + geom_boxplot()

```



```
biais_var_mse <-data.frame(biais=c(mean(l1)-lambda , mean(l2)-lambda,mean(l31)-lambda ,
                                mean(l32)-lambda , mean(l33)-lambda) ,
                           var=c( var(l1) , var(l2),var(l31) , var(l32),var(l33) ) ,
                           mse= c((mean(l1)-lambda)^2+var(l1) , (mean(l2)-lambda)^2+var(l2) ,
                                (mean(l31)-lambda)^2+var(l31) , (mean(l32)-lambda)^2+var(l32) ,
                                (mean(l33)-lambda)^2+var(l33)))
```

```
biais_var_mse
```

```
##      biais      var      mse
## 1  0.011640154 0.01110071 0.01123620
## 2  0.022216004 0.01357468 0.01406823
## 3 -0.002181591 0.03906817 0.03907293
## 4  0.008313292 0.01881343 0.01888254
## 5  0.018074919 0.01897645 0.01930315
```

To compare the estimators, we need to compare their behaviors for all the values of lambda since we don't know from which parameter the observations come from (statistics assumptions). We can check the distribution and approximate the mse of the estimates with Monte-Carlo simulations for different values of lambda, for instance in a grid.

Another way to compare these estimators could be to study their asymptotic behaviours. Using the law of large number, we can prove that these estimators are consistent, i.e.

$$\hat{\lambda} \rightarrow \lambda \quad \text{in } P_{\lambda} - \text{probability, for all } \lambda > 0.$$

A more precise asymptotic study can be obtained using the Delta-method (related to the CLT) or the asymptotic theory on MLEs in regular models (see Section 8.5.2 of Rice's book).