

## Assignment 2

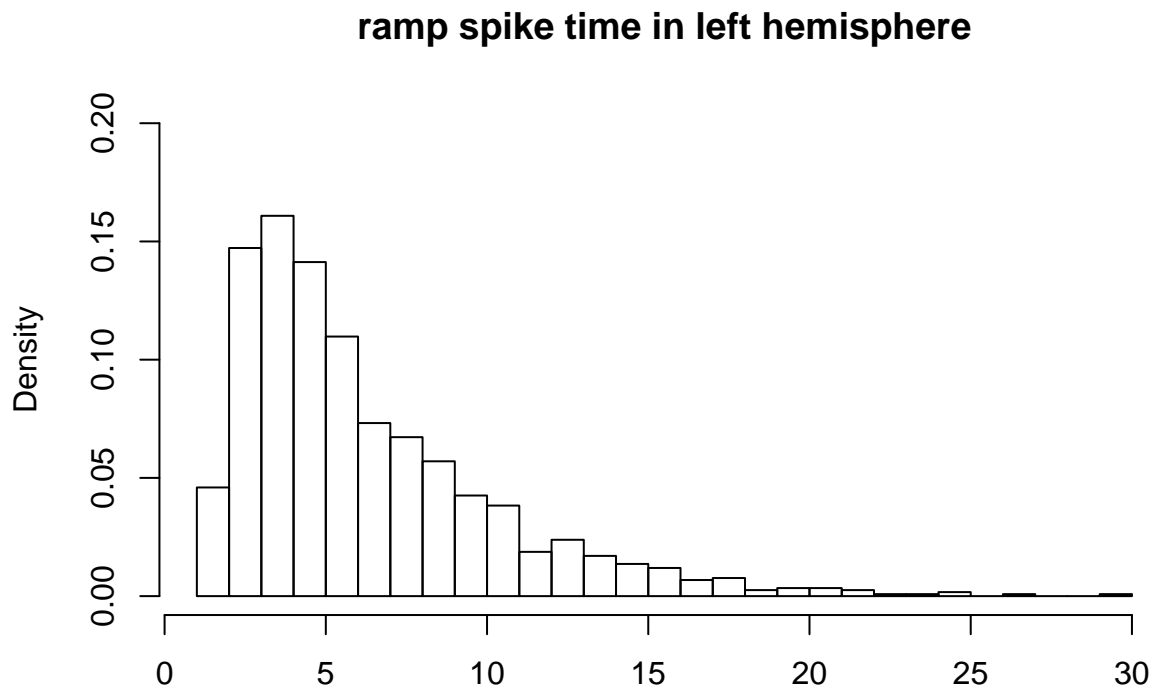
### 1. Brain cell dataset

4.1: plot separately histograms for the distribution of the ramp spike time variable in the left and right hemisphere

```
left_raw <- cells$ef__peak_t_ramp[cells$specimen__hemisphere == "left"]
left <- left_raw[!is.na(left_raw)] # removing NA values again to avoid missing values error

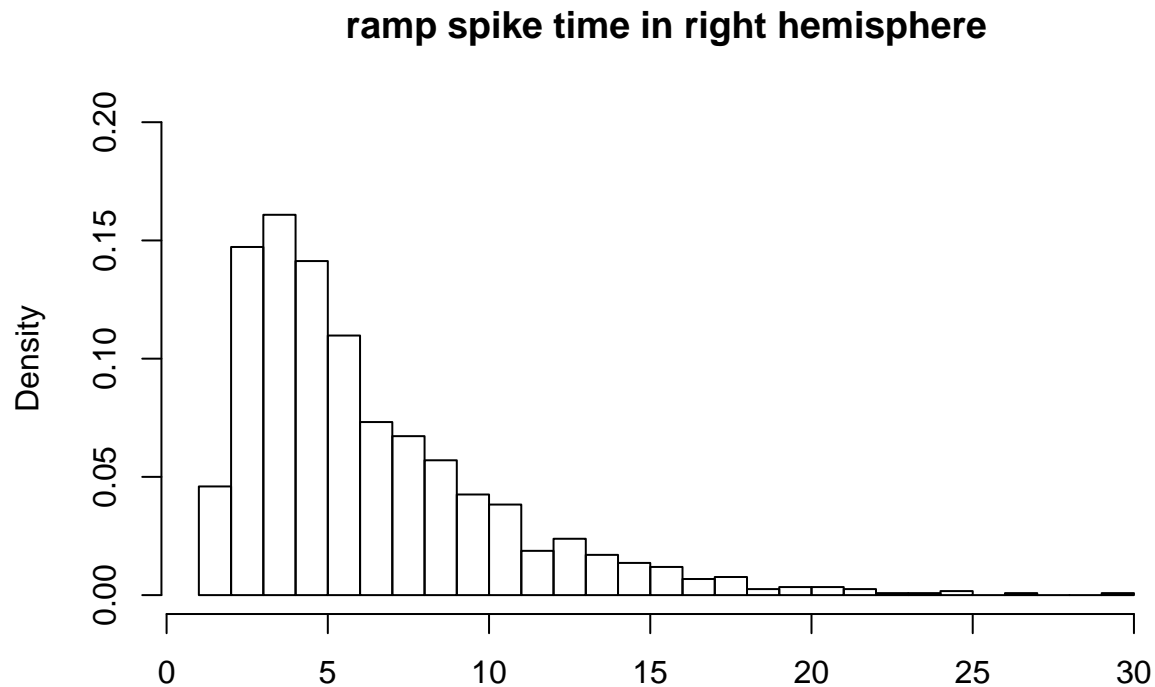
right_raw <- cells$ef__peak_t_ramp[cells$specimen__hemisphere == "right"]
right <- right_raw[!is.na(right_raw)]

hist(left,
      breaks = 40,
      proba = T,
      main = "ramp spike time in left hemisphere", xlab = "", ylim = c(0,0.2))
```



```
hist(left,
      breaks = 40,
```

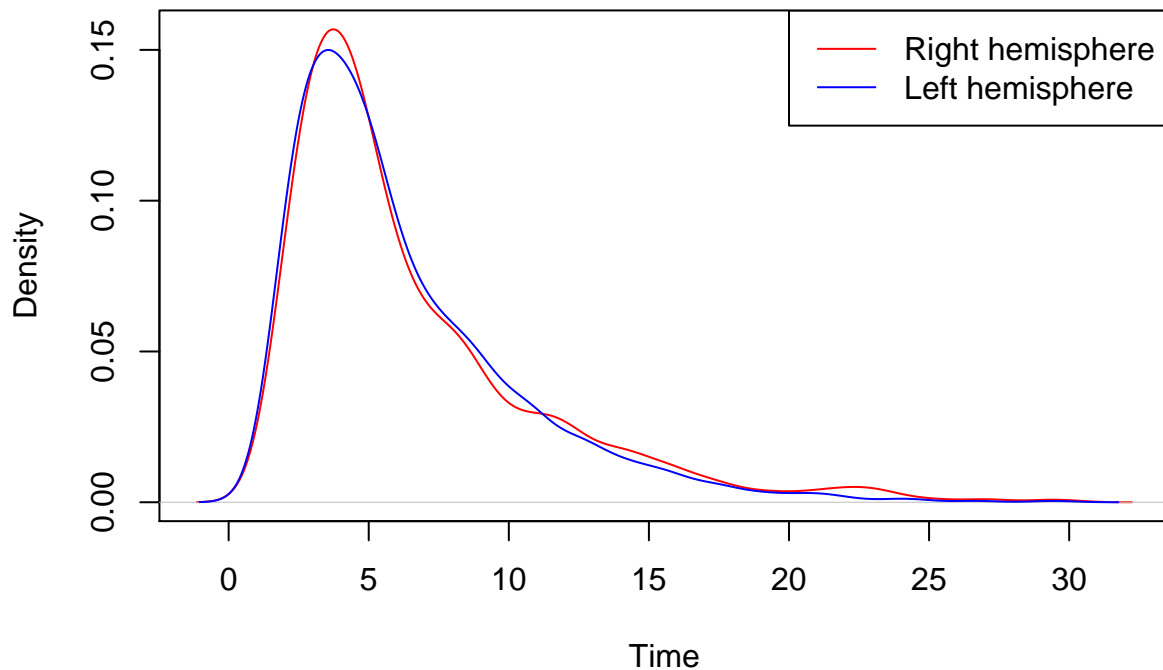
```
proba = T,
main = "ramp spike time in right hemisphere", xlab = "", ylim = c(0,0.2))
```



4.2: plot together kernel density estimations for the ramp spike time variable in the left and right hemisphere.

```
plot(density(right),
     col = "red",
     main = "Kernel density estimation for the ramp spike time",
     xlab = "Time")
lines(density(left),
      col = "blue")
legend("topright",
      legend = c("Right hemisphere", "Left hemisphere"),
      col = c("red", "blue"),
      lty = 1)
```

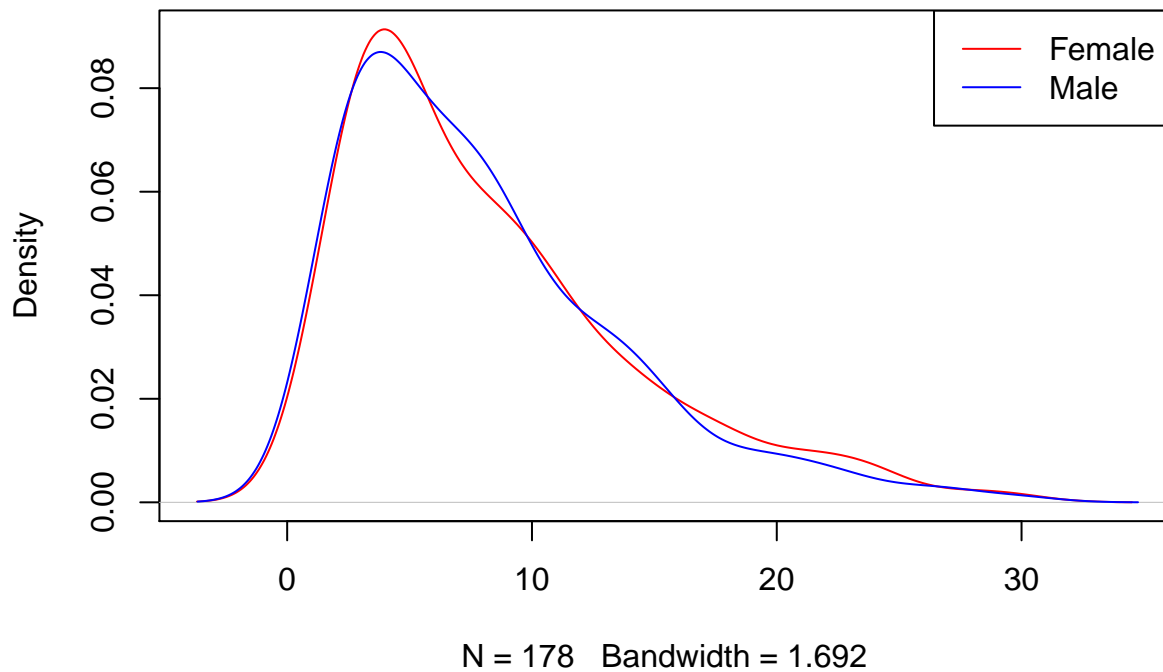
### Kernel density estimation for the ramp spike time



4.3: plot together kernel density estimations for the ramp spike time variable for males and females.

```
female_raw <-  
  cells$ef__peak_t_ramp[cells$donor__species == "Homo Sapiens" &  
    cells$donor__sex == "Female"]  
female <- female_raw[!is.na(female_raw)]  
  
male_raw <-  
  cells$ef__peak_t_ramp[cells$donor__species == "Homo Sapiens" &  
    cells$donor__sex == "Male"]  
male <- male_raw[!is.na(male_raw)]  
  
plot(density(female),  
  col = "red",  
  main = "Kernel Density Estimations for males and females")  
lines(density(male),  
  col = "blue")  
legend("topright",  
  legend = c("Female", "Male"),  
  col = c("red", "blue"),  
  lty = 1)
```

## Kernel Density Estimations for males and females



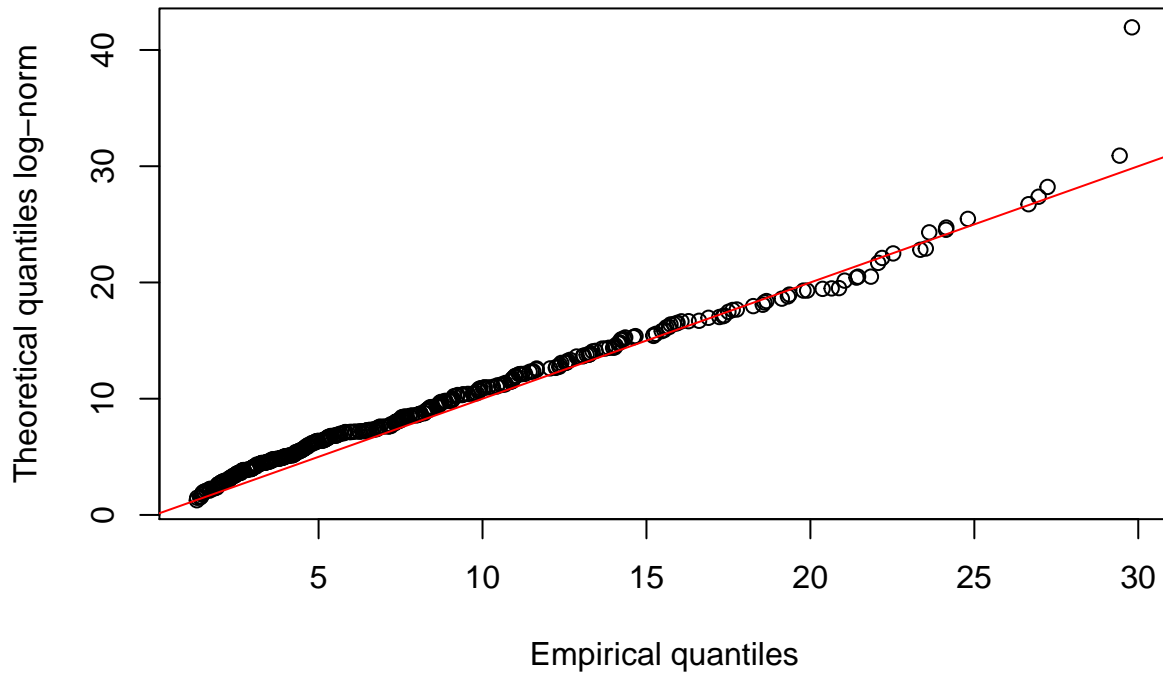
### Ex 5

#### 5.1: Q-Q plot of ramp spike time distribution for humans, between empirical quantiles and theoretical ones (dlnorm distribution)

```
humans_raw <- cells$ef__peak_t_ramp[cells$donor__species == "Homo Sapiens"]
humans <- humans_raw[!is.na(humans_raw)]

qqplot(humans,      # qqplot: two "data set" as arg and align quantiles
        rlnorm(ppoints(humans), sdlog = 0.6, meanlog = 2),
        main = "Q-Q Plot, ramp spike time d. and log-normal d.",
        xlab = "Empirical quantiles",
        ylab = "Theoretical quantiles log-norm")
abline(0, 1,
       col = 'red')
```

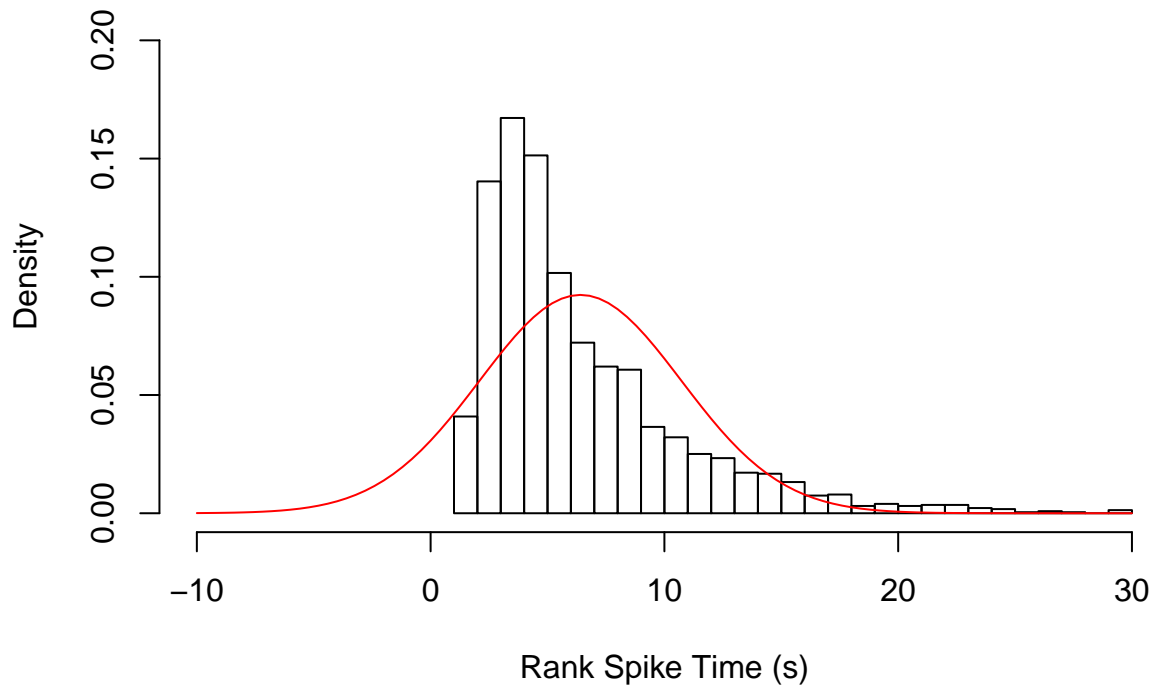
### Q-Q Plot, ramp spike time d. and log-normal d.



5.2: Plot the Gaussian density on top of the histogram of the ramp spike time with better parameter values

```
allcells <- cells$ef__peak_t_ramp[!is.na(cells$ef__peak_t_ramp)]
hist(allcells,
     breaks = 40,
     ylim = c(0, 0.2),
     xlim = c(-10, 30),
     probability = TRUE,
     main = "Rank Spike Time frequencies for all cells and d.norm estimation",
     xlab = "Rank Spike Time (s)")
curve(dnorm(x, mean = 6.41, sd = 4.32),
     col = "red",
     add = TRUE)
```

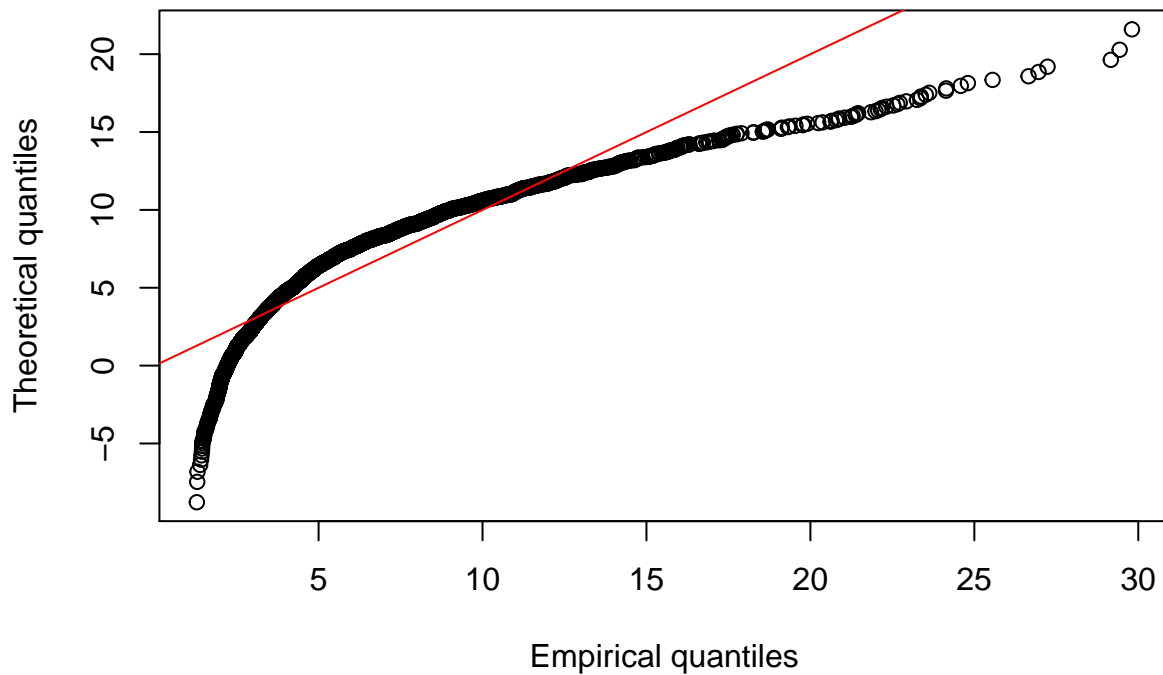
## Rank Spike Time frequencies for all cells and d.norm estimation



> Our data doesn't have a left tail and the peak of the histogram is much larger than the curve of the normal distribution.

### 5.3: Q-Q plot for the ramp spike time against the Gaussian distribution with these new parameter values

```
theoretical_spike_q <- qnorm(ppoints(allcells), sd = 4.32, mean = 6.41)
plot(sort(allcells), theoretical_spike_q,
     xlab = "Empirical quantiles",
     ylab = "Theoretical quantiles")
abline(0, 1,
       col = "red")
```



We can visually observe that our data don't follow a Gaussian distribution since several quantiles in the tails don't match in the diagonal.

## 2. Empirical mean and variance

6.1: Sample  $n$  values of  $X \sim \text{Bi}(\text{parameter size} = 100, \text{prob} = 0.3)$  for  $n = 10, 100, 1000, 10000$ .

```
n = 100      # n. of experiments (size or i.e. nb of coin tosses)
p = 0.3      # p. of success (i.e. head)
EM = c()
EV = c()
Esd = c()
rep = c()
for (x in c(10, 100, 1000, 10000)){      # x = nb of sampling repetitions
  sample <- rbinom(x, size = n, prob = p)
  EM <- c(EM, mean(sample))              # empirical mean
  EV <- c(EV, var(sample))               # empirical variance
  Esd <- c(Esd, sd(sample))              # empirical standard deviation
  rep <- c(rep, x)                       # n. of sampling repetition
}
mean <- n * p                            # true or theoretical mean
var <- n * p * (1-p)                     # true or theoretical variance
sd <- sqrt(var)                          # true or theoretical standard deviation
```

```
data.frame(EM, mean, EV, var, Esd, sd,
           row.names = rep)
```

```
##           EM mean      EV var      Esd      sd
## 10      29.6000    30 20.93333    21 4.575296 4.582576
## 100     30.8900    30 20.82616    21 4.563569 4.582576
## 1000    29.9670    30 19.42333    21 4.407191 4.582576
## 10000   29.9796    30 20.65905    21 4.545223 4.582576
```

**6.2:** Write a function to repeat the sampling and estimation for  $n = 1000$  and plot.

```
samp_est_f <- function(nb_rep){
  nb_trials = 100
  p = 0.3
  EM = c()
  EV = c()
  Esd = c()
  for (i in 1:1000){
    sample <- rbinom(nb_rep, size = nb_trials, prob = p)
    EM <- c(EM, mean(sample)) # that's an easy way to "append" values to a vector
  }

  hist(EM,
       probability = TRUE,
       breaks = 50) # histogram of the distribution of the empirical means

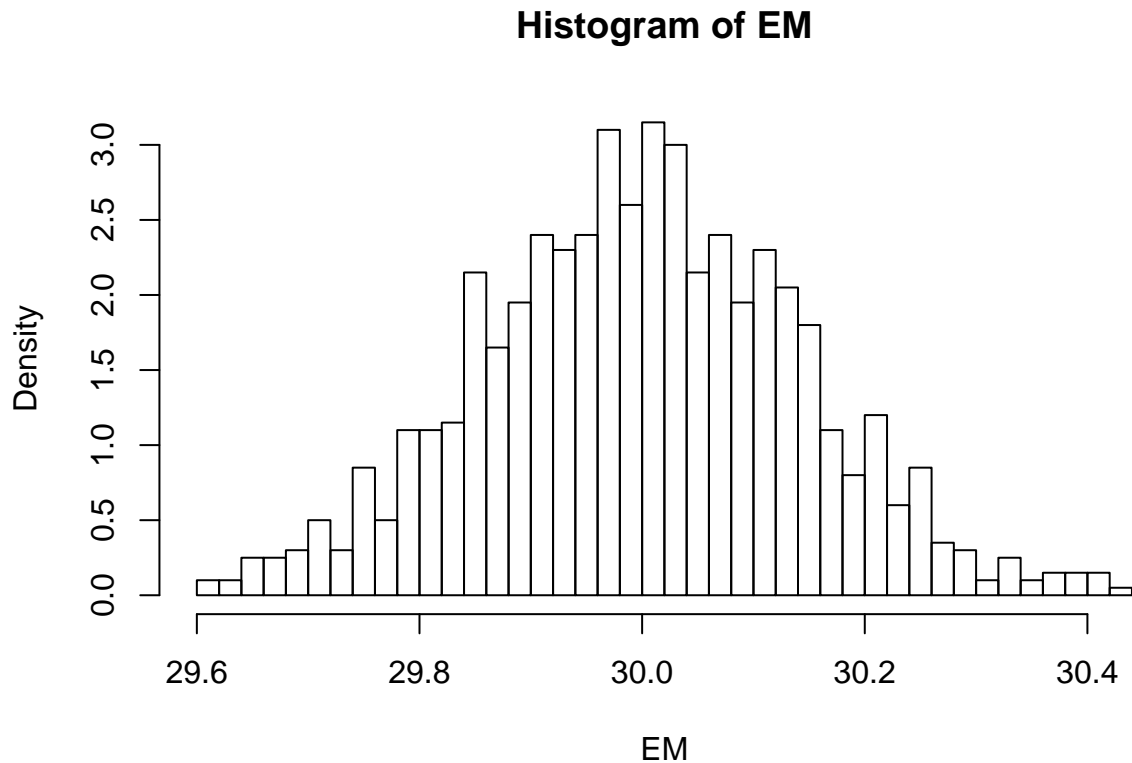
  data = list() # list to store the output of the function

  data$EM <- EM # means distributions for 6.4
  data$sample <- sample # the last sample generated
  data$rep <- nb_rep # n. of repetition
  data$var_EM <- var(EM) # (emp.) variance of empirical/sample means
  data$sd_EM <- sd(EM) # (emp.) standard deviation of the empirical/sample means

  cat("rep:", data$rep, # cat() to print numbers and strings in the same line
      ", var_EM:", data$var_EM,
      ", sd_EM:", data$sd_EM)
  return(data)
}

data <- samp_est_f(1000)
```





```
## rep: 1000 , var_EM: 0.02079311 , sd_EM: 0.1441982
```

### 6.3: Compute the standard error of the mean

```
SEM_f <- function(x){
  SEM <- (sd(data$sample) / sqrt(length(data$sample) - 1))
  return(SEM)
}
data$SEM <- SEM_f(1000)
cat("sd_EM:", data$sd_EM, ", SEM:", data$SEM)
```

```
## sd_EM: 0.1441982 , SEM: 0.1420409
```

```
# SEM = ESTIMATOR of the error between the sample means and the population mean
# sd_EM = standard deviation of the sample means
```

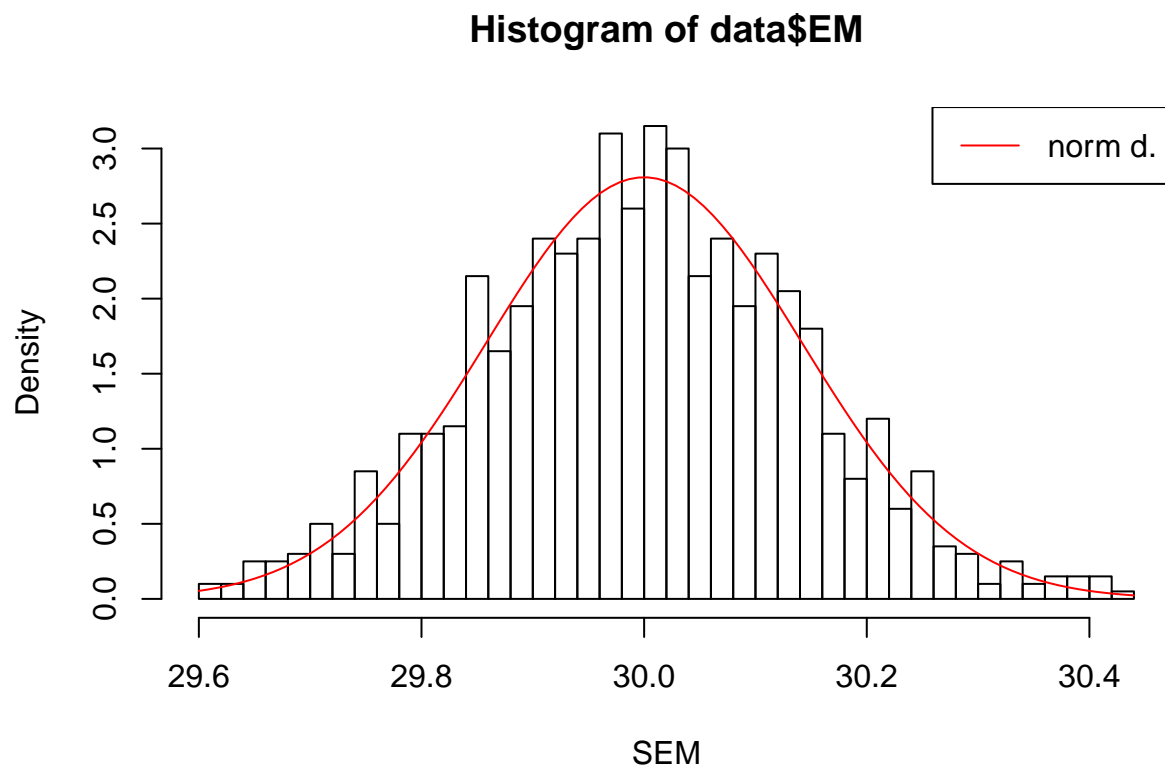
### 6.4: draw the density of Gaussian distribution with parameters mean = $100 \times 0.3$ and sd equal to the sem

```
plot_f <- function(x){
  hist(data$EM,
    probability = TRUE,
    breaks = 50,
    xlab = "SEM")
  curve(dnorm(x, mean = 100 * 0.3, sd = data$SEM),
    col = "red",
```

```

    add = TRUE)
  legend("topright",
        legend = "norm d.",
        col = "red",
        lty = 1)
}
plot_f(1000)

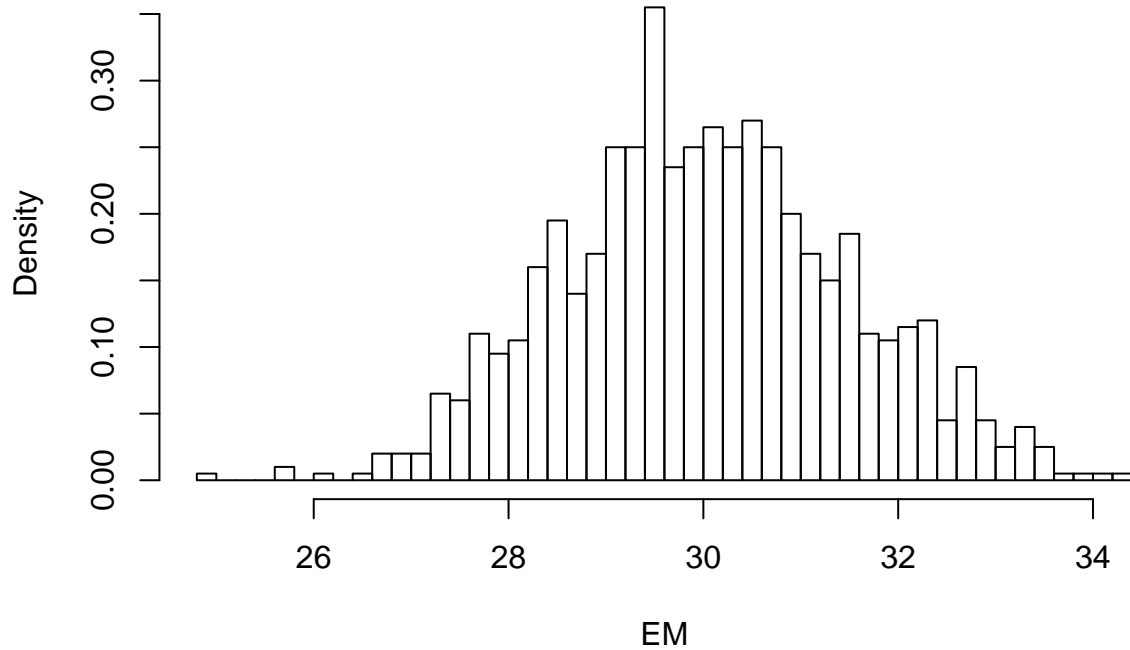
```



check the result for different values of x:

```
data_10 <- samp_est_f(10)
```

## Histogram of EM



```
## rep: 10 , var_EM: 2.135361 , sd_EM: 1.461287
```

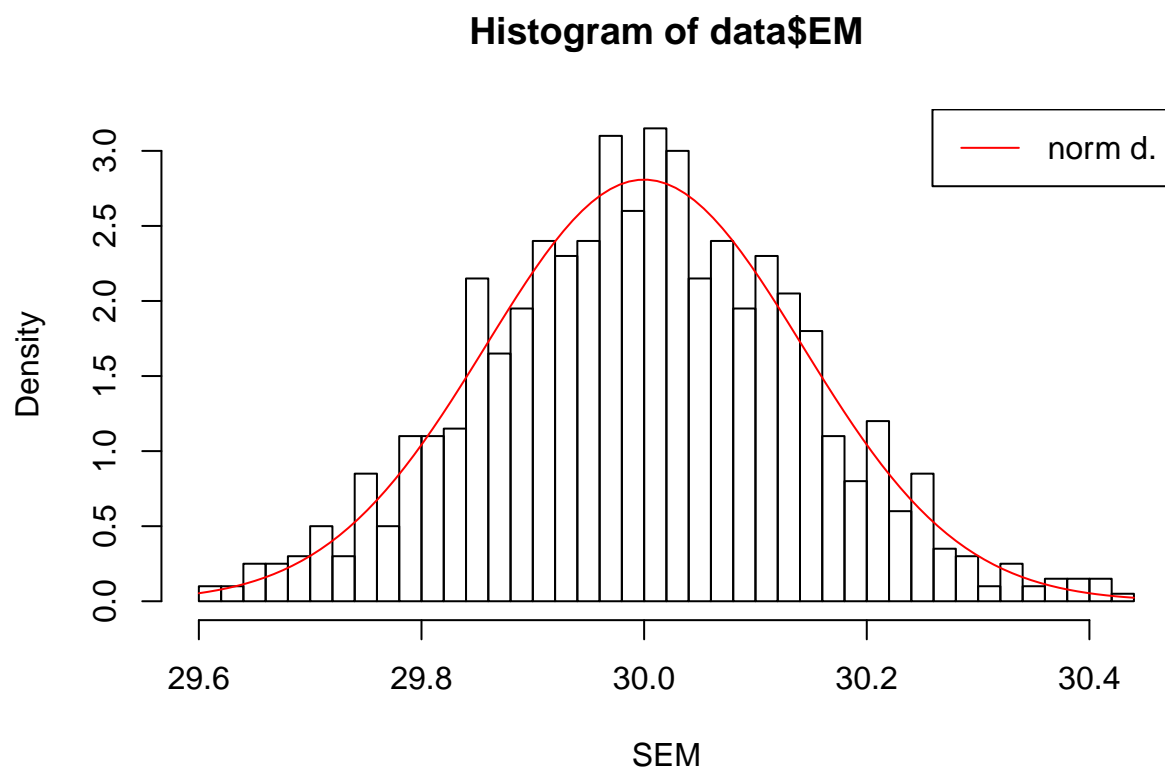
```
data_10$SEM <- SEM_f(10)
cat("rep:", data_10$rep,
    ", var_EM:", data_10$var_EM,
    ", sd_EM:", data_10$sd_EM)
```

```
## rep: 10 , var_EM: 2.135361 , sd_EM: 1.461287
```

```
cat("sd_EM:", data_10$sd_EM, ", SEM:", data_10$SEM)
```

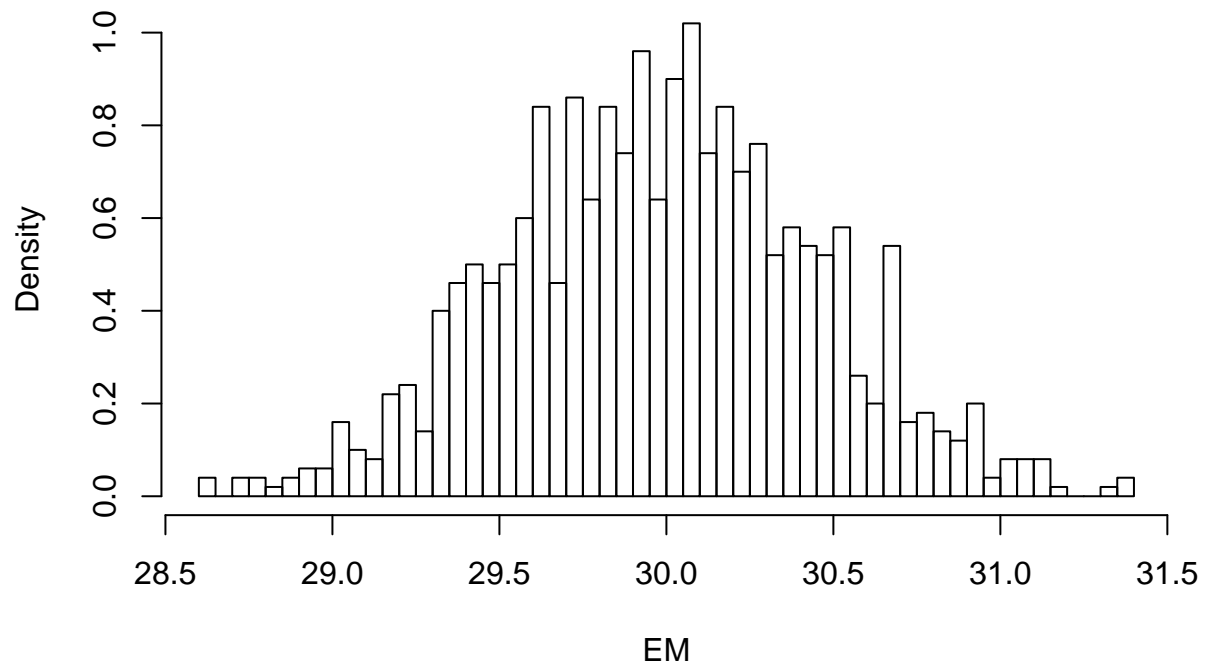
```
## sd_EM: 1.461287 , SEM: 0.1420409
```

```
plot_f(10)
```



```
data_100 <- samp_est_f(100)
```

## Histogram of EM



```
## rep: 100 , var_EM: 0.2151068 , sd_EM: 0.4637961
```

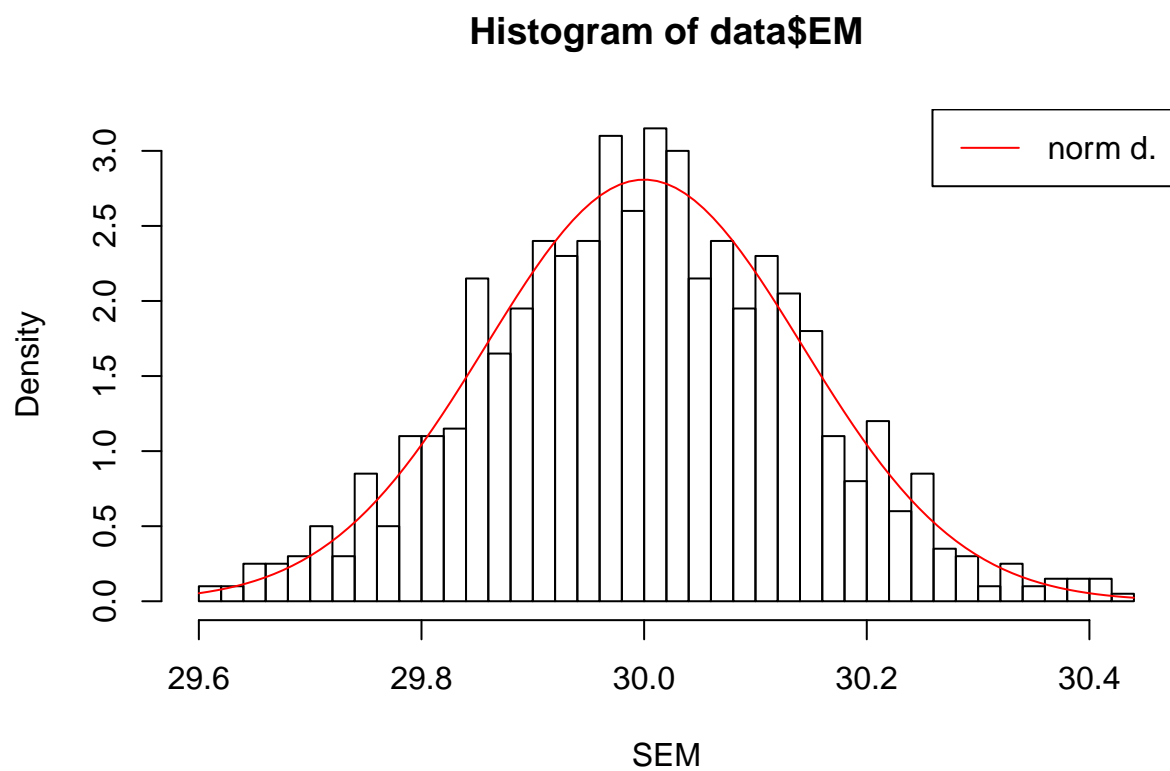
```
data_100$SEM <- SEM_f(100)
cat("rep:", data_100$rep,
    ", var_EM:", data_100$var_EM,
    ", sd_EM:", data_100$sd_EM)
```

```
## rep: 100 , var_EM: 0.2151068 , sd_EM: 0.4637961
```

```
cat("sd_EM:", data_100$sd_EM, ", SEM:", data_100$SEM)
```

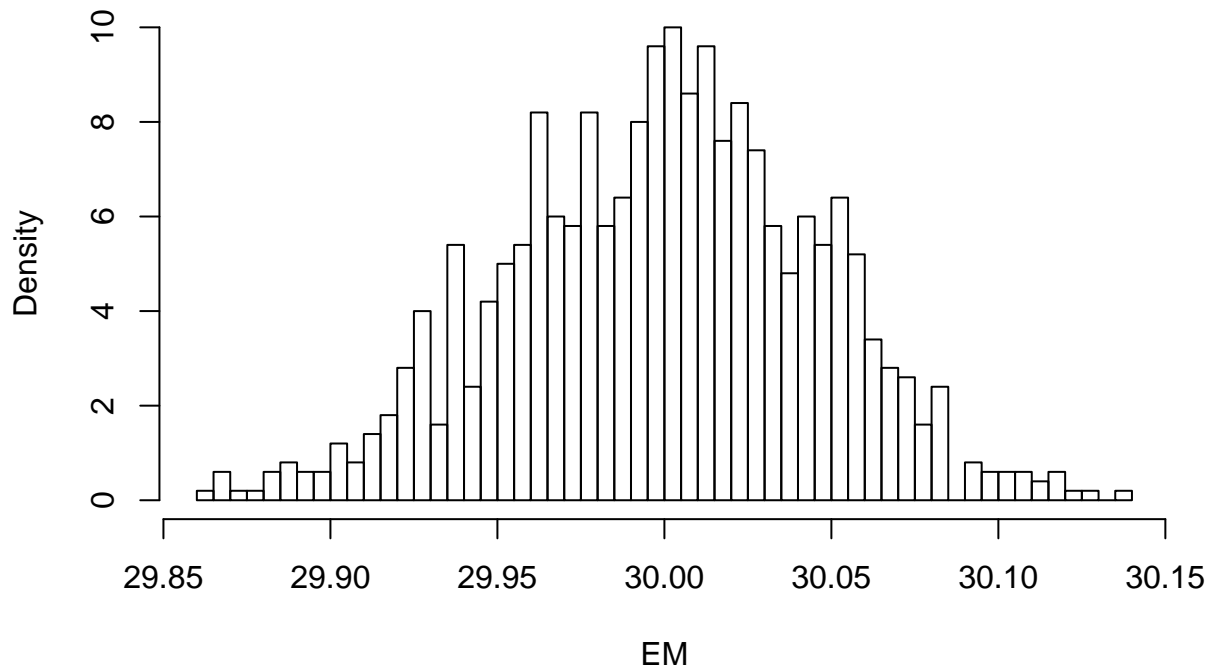
```
## sd_EM: 0.4637961 , SEM: 0.1420409
```

```
plot_f(100)
```



```
data_10000 <- samp_est_f(10000)
```

## Histogram of EM



```
## rep: 10000 , var_EM: 0.0021458 , sd_EM: 0.04632278
```

```
data_10000$SEM <- SEM_f(10000)
cat("rep:", data_10000$rep,
    ", var_EM:", data_10000$var_EM,
    ", sd_EM:", data_10000$sd_EM)
```

```
## rep: 10000 , var_EM: 0.0021458 , sd_EM: 0.04632278
```

```
cat("sd_EM:", data_10000$sd_EM, ", SEM:", data_10000$SEM)
```

```
## sd_EM: 0.04632278 , SEM: 0.1420409
```

```
plot_f(10000)
```

