DSE_Project1_Part2_Fall2022

Sumaiya Uddin

2022-10-04

Getting Started

```
#setting seed

set.seed(29071995)
```

Load packages

In this lab we will explore the data using the `dplyr` package and visualize it using the `ggplot2` package for data visualization. The data can be found in the companion package for this course, `statsr`.

Let's load the packages.

```
library(statsr)
library(dplyr)
library(ggplot2)
library(plyr)
```

Dataset

This Data set was selected from kaggle.com. This data set is useful to researchers studying the dependency between medical insurance cost and consumers. I choose this data set to see is there anything that effects our medical insurance charges. I was hoping may be age or other any medical issues may reasons of high costs. And i surprisingly found some correlations.

Now let's read our data set.

```
# Importing Dataset and read

library(readr)
insurance <- read_csv("~/CCNY/Statistics/project1/part2/archive/insurance.csv")
```

```
## Rows: 1338 Columns: 7
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (3): sex, smoker, region
## dbl (4): age, bmi, children, charges
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(insurance)
```

```
## # A tibble: 6 × 7
##     age sex      bmi children smoker region    charges
##   <dbl> <chr>  <dbl>    <dbl> <chr>  <chr>       <dbl>
## 1    19 female  27.9        0 yes    southwest  16885.
## 2    18 male    33.8        1 no     southeast   1726.
## 3    28 male    33          3 no     southeast   4449.
## 4    33 male    22.7        0 no     northwest  21984.
## 5    32 male    28.9        0 no     northwest   3867.
## 6    31 female  25.7        0 no     southeast   3757.
```

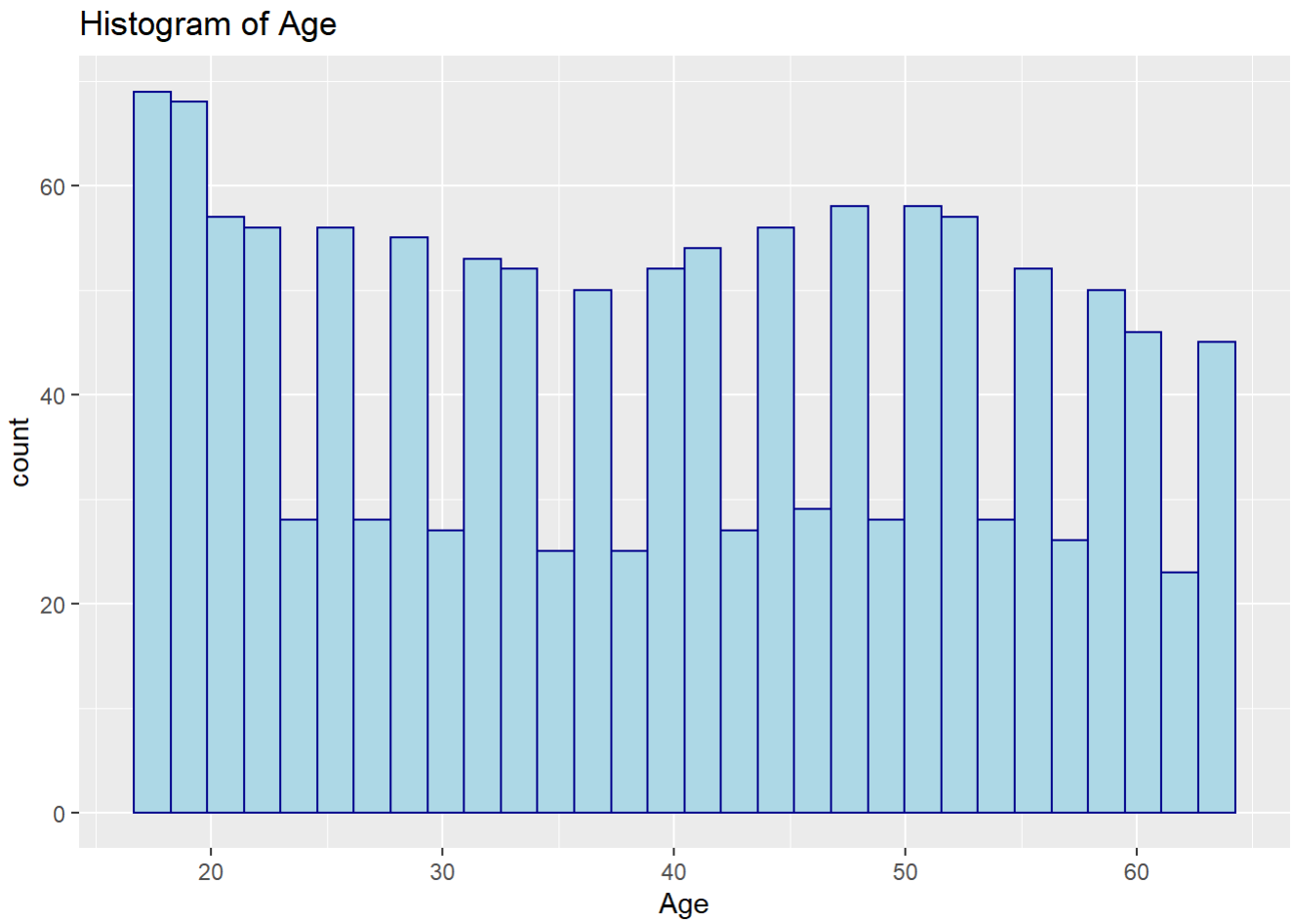
We can see that in Data set there are 7 different variables, some categorical and some numerical.

2. Data Visualization

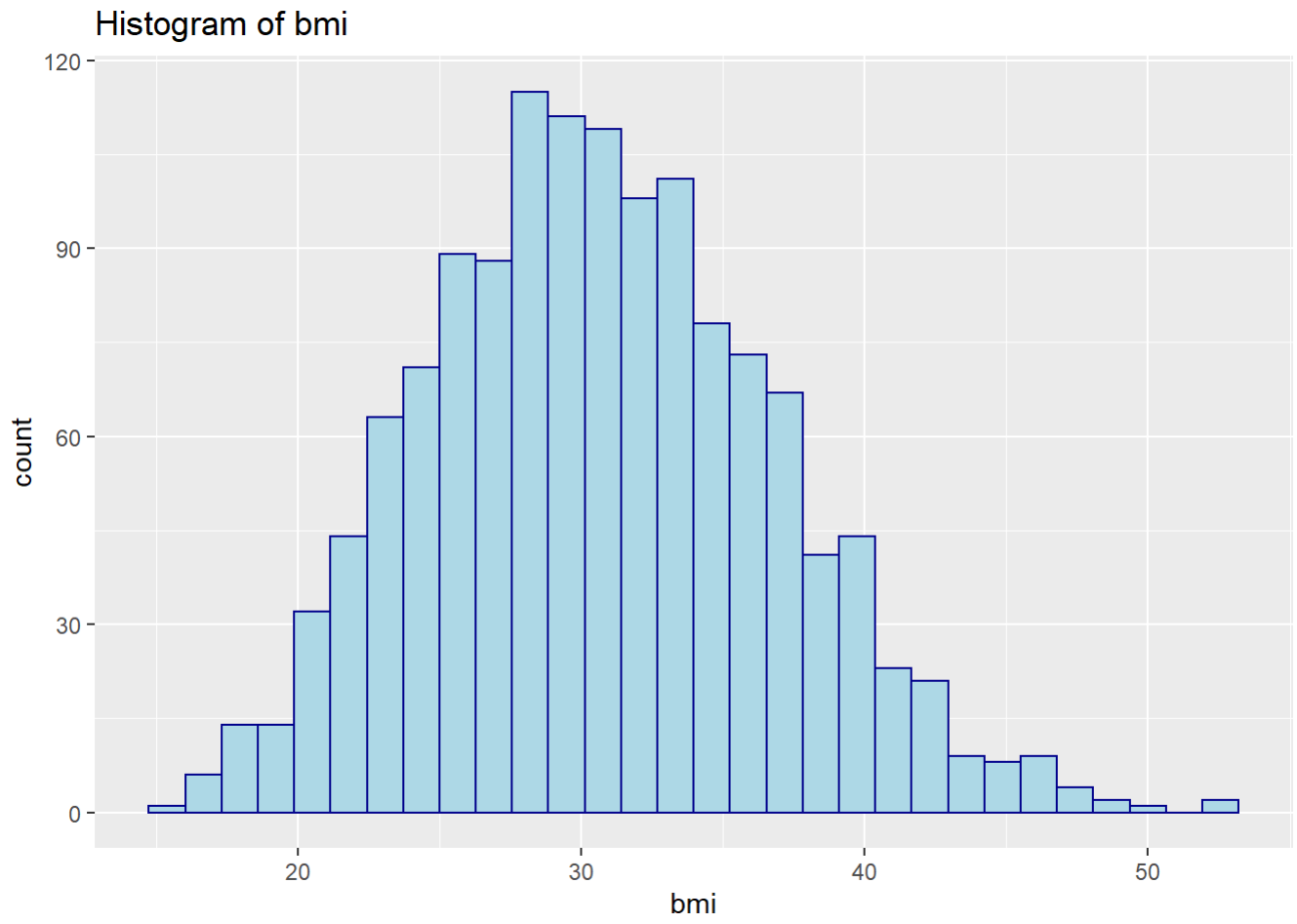
a). From Data set we will choose 3 different variables and for each variables we will plot Histogram, QQ-plot and Cumulative relative frequency. And will try to find which of the variable is approximately normal.

Chosen variables are = age, bmi and charges.

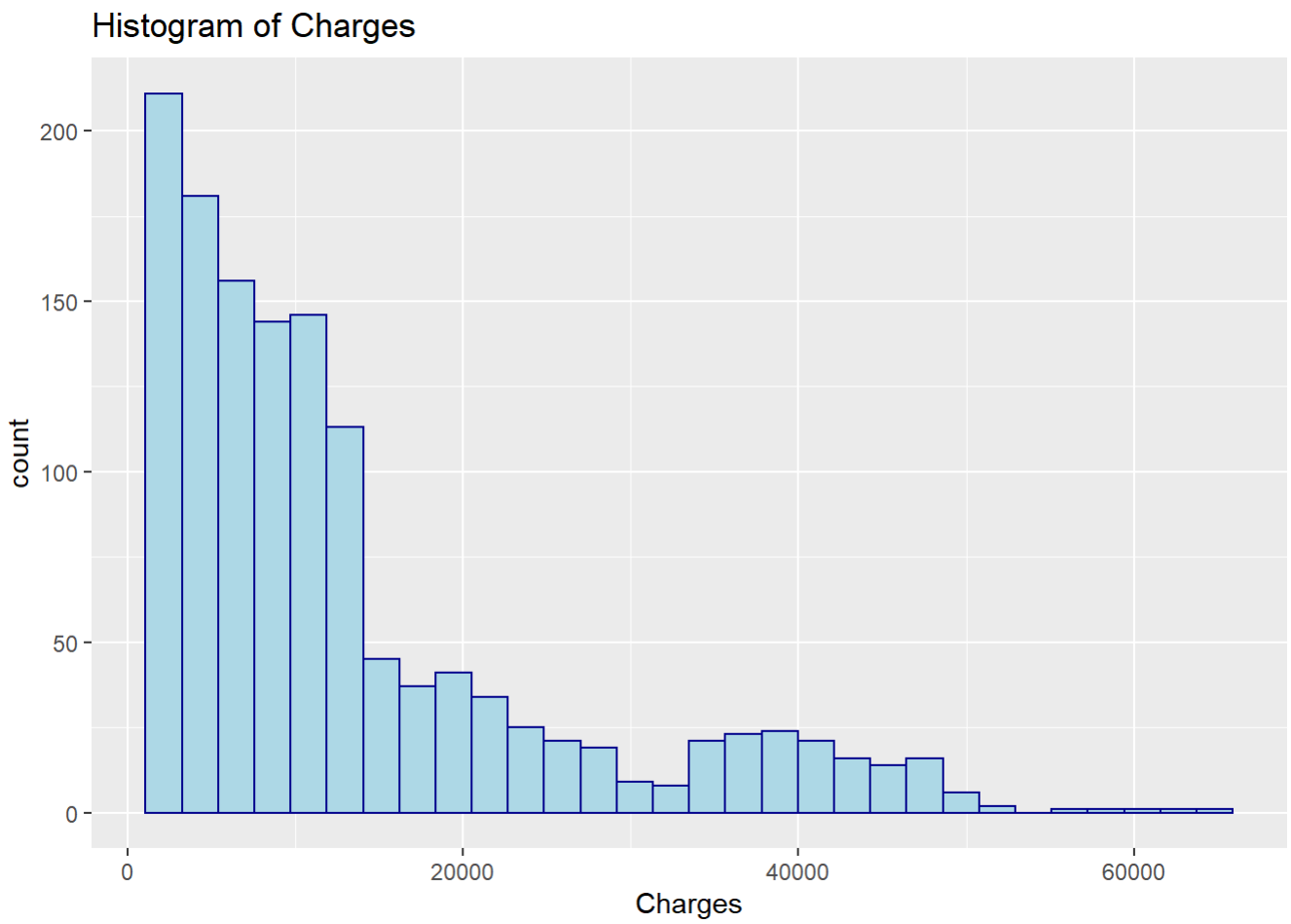
```
# Ploting Histogram for age, bim and charges

age_hist = qplot(x = insurance$age, geom = "histogram", xlab = "Age", ylab = "count", main = "Hi
stogram of Age", col = I("darkblue"), fill = I("lightblue"), bins= 30)
age_hist
```




```
bmi_hist = qplot(x = insurance$bmi, geom = "histogram", xlab = "bmi", ylab = "count", main = "Hi
stogram of bmi", col = I("darkblue"), fill = I("lightblue"), bins= 30)
bmi_hist
```

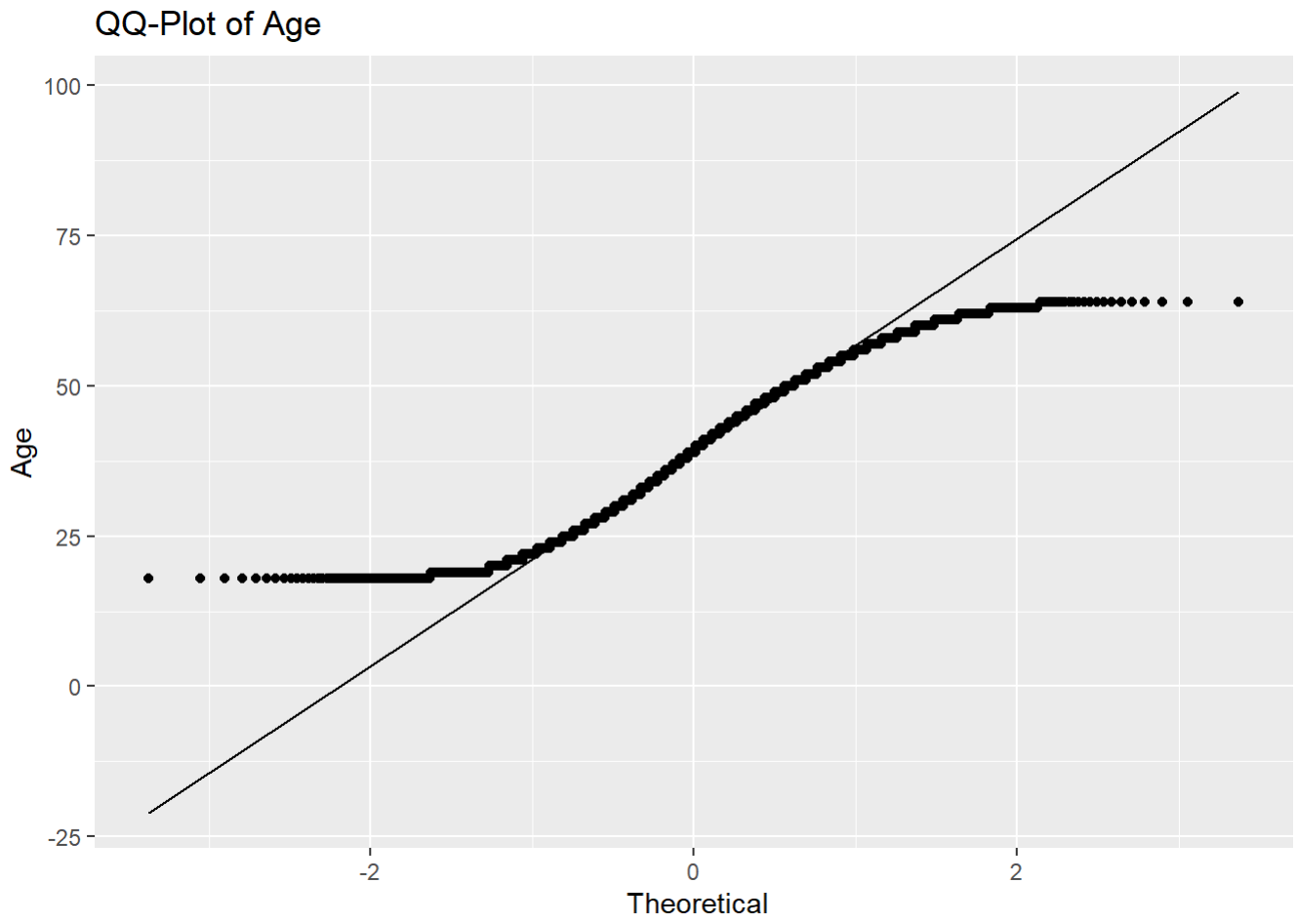



```
charges_hist = qplot(x = insurance$charges, geom = "histogram", xlab = "Charges", ylab = "count"
, main = "Histogram of Charges", col = I("darkblue"), fill = I("lightblue"), bins= 30)
charges_hist
```

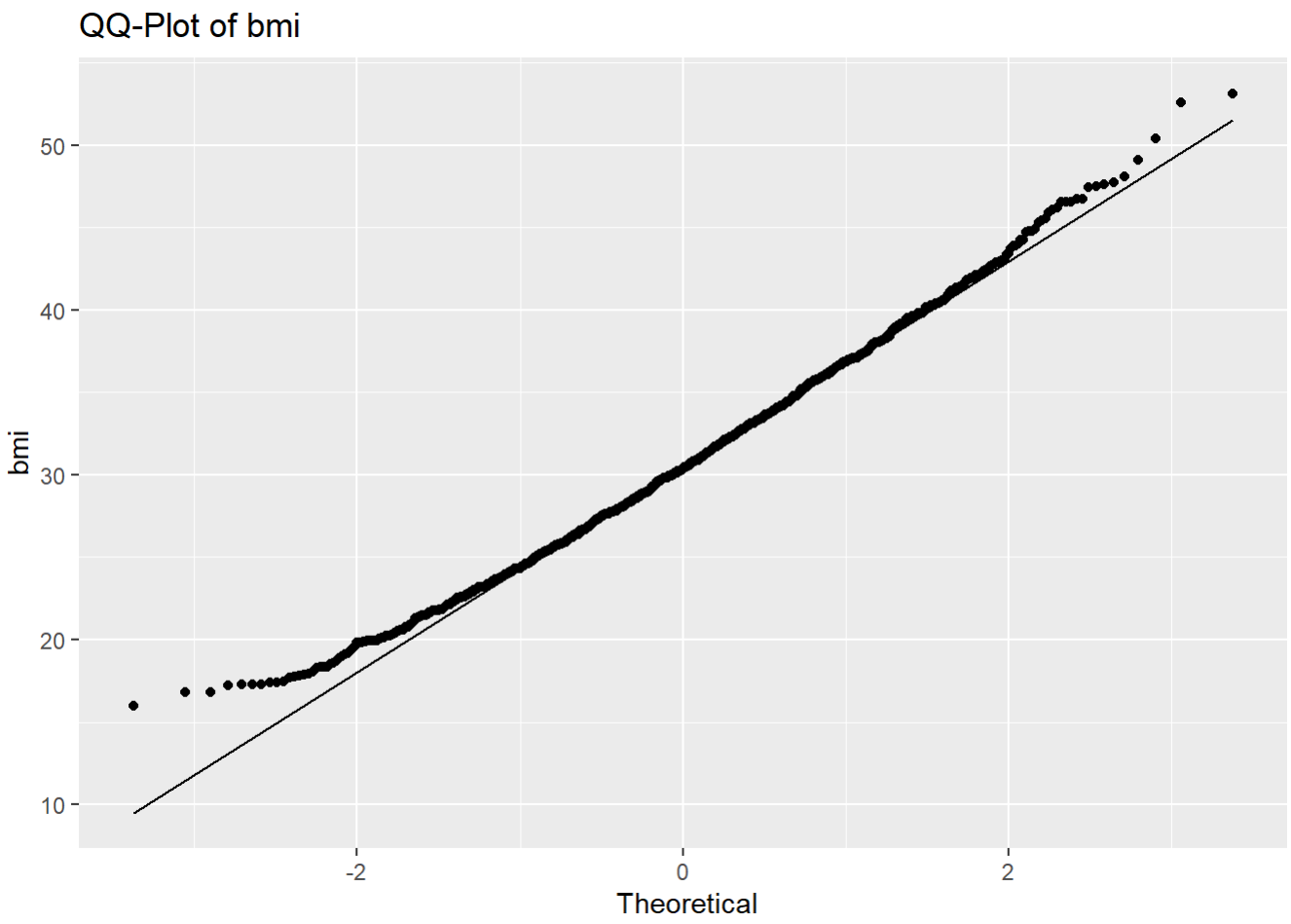


From this graphs we can assume histogram of bmi is normally distributed, Because it's roughly Bell-shaped.

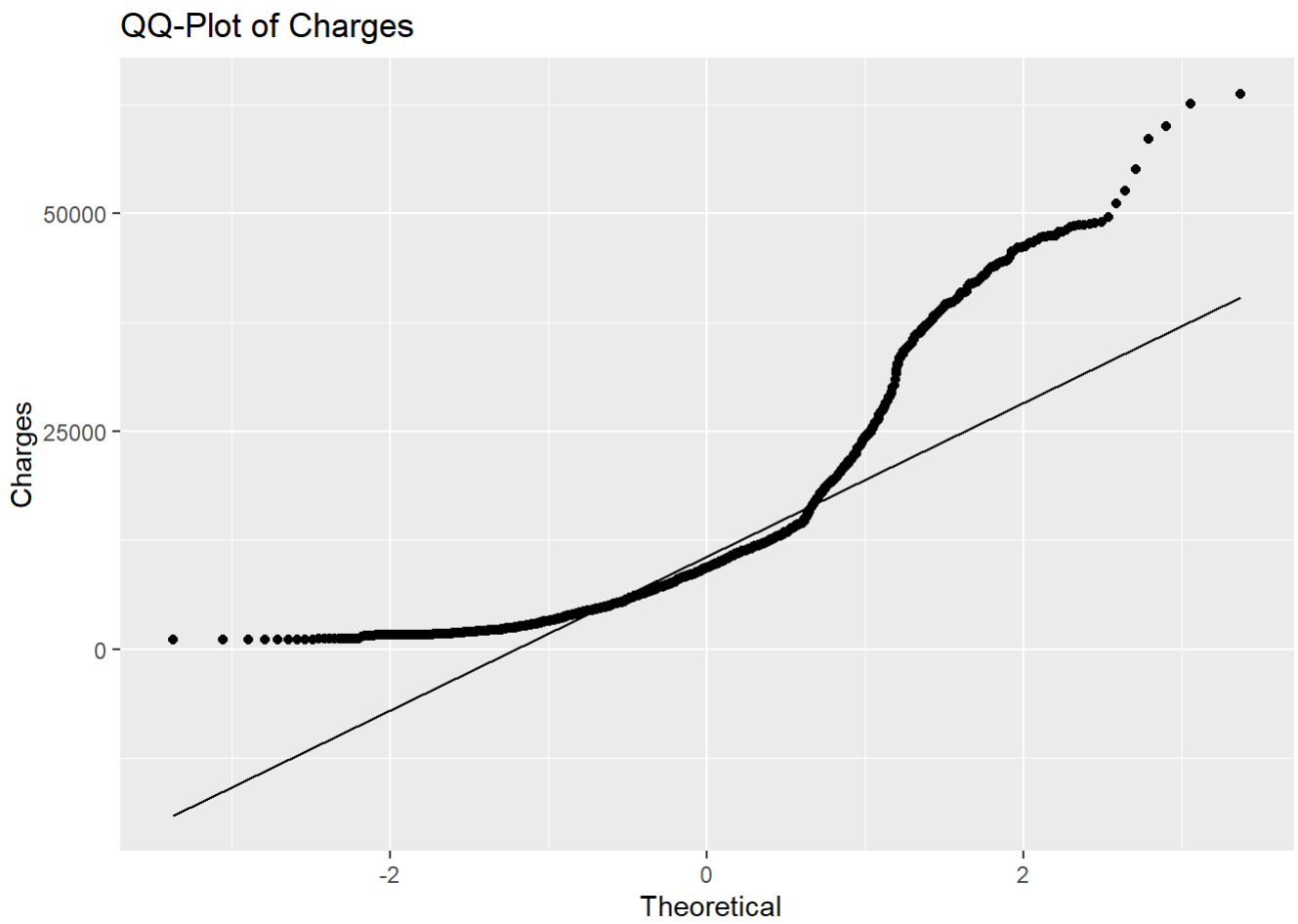
```
# Plotting QQ-plot for age,bmi and charges

age_qq <- ggplot(data = insurance, aes(sample=age)) + stat_qq() + stat_qq_line() + labs(title =
"QQ-Plot of Age", x=" Theoretical", y= "Age")
age_qq
```




```
bmi_qq <- ggplot(data = insurance, aes(sample=bmi)) + stat_qq() + stat_qq_line() + labs(title =
"QQ-Plot of bmi", x=" Theoretical", y= "bmi")
bmi_qq
```



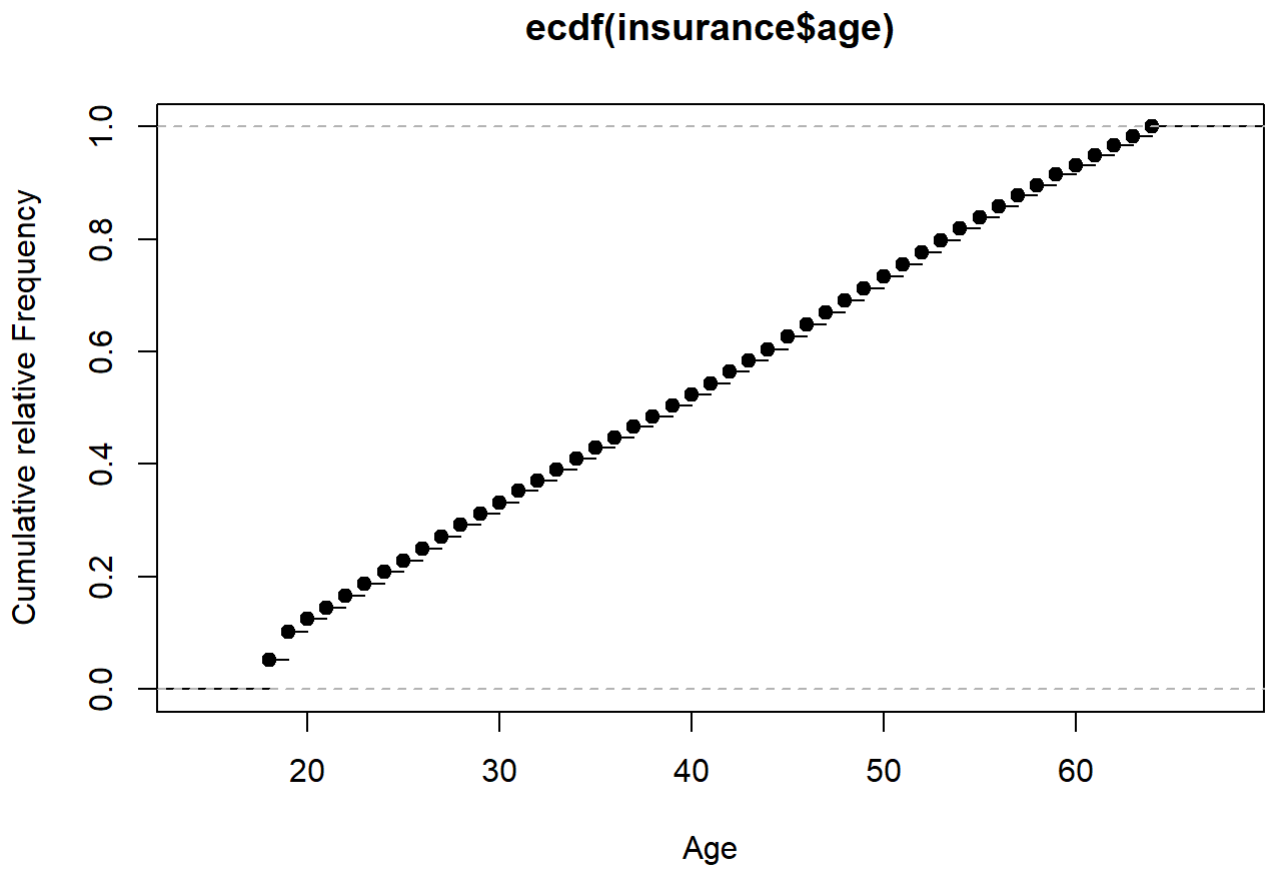

```
charges_qq <- ggplot(data = insurance, aes(sample=charges)) + stat_qq() + stat_qq_line() + labs
(title = "QQ-Plot of Charges", x=" Theoretical", y= "Charges")
charges_qq
```



The points in the bmi QQ-plot roughly fall along a straight diagonal line. So we can assume bmi of data set is normally distributed.

Now we are going to find cumulative relative frequency graph for those variables. For that we can create an interpolate function with the built-in function ecdf. Then we plot right away.

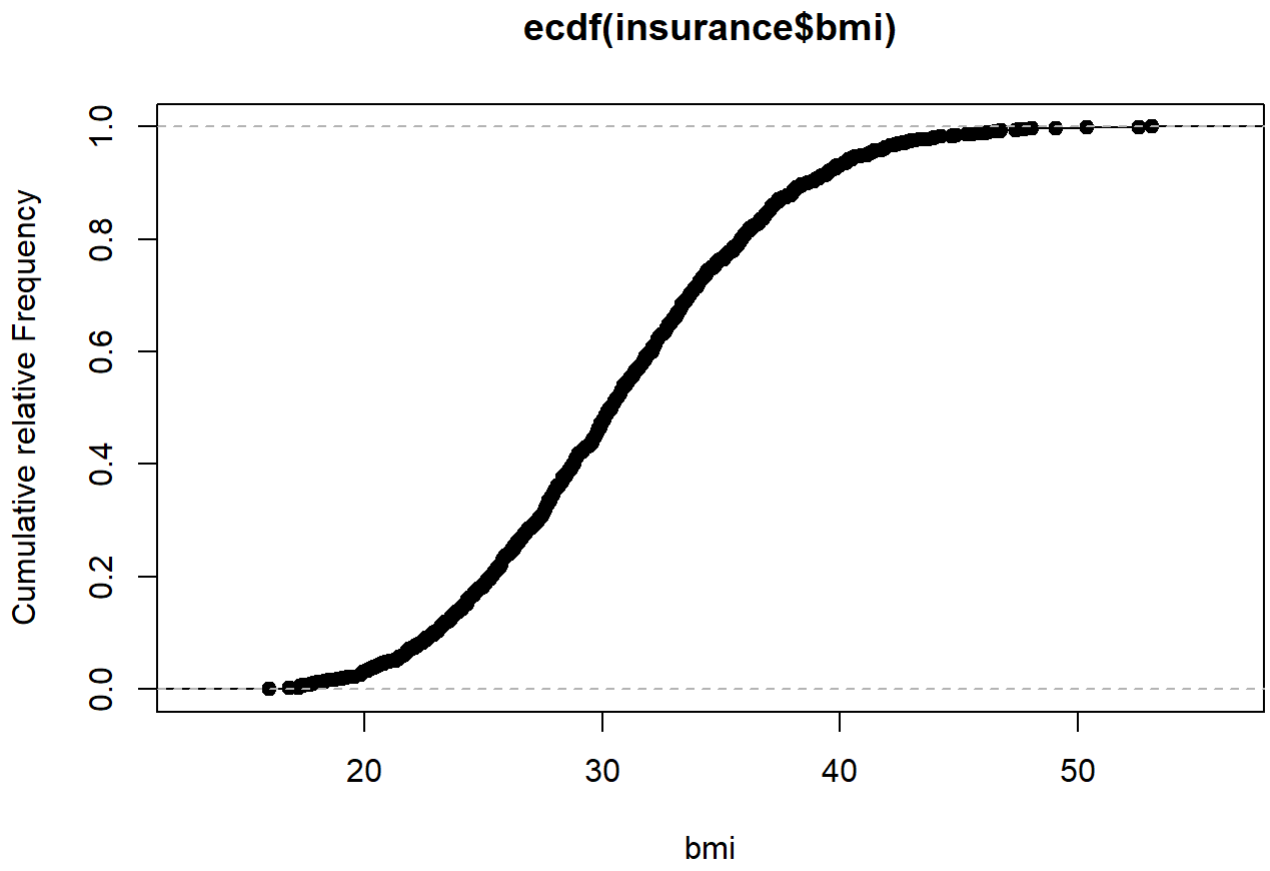
```
age_fn <- ecdf(insurance$age)
age_crf = plot(age_fn, xlab = "Age", ylab = "Cumulative relative Frequency")
```




```
age_crf
```

```
## NULL
```

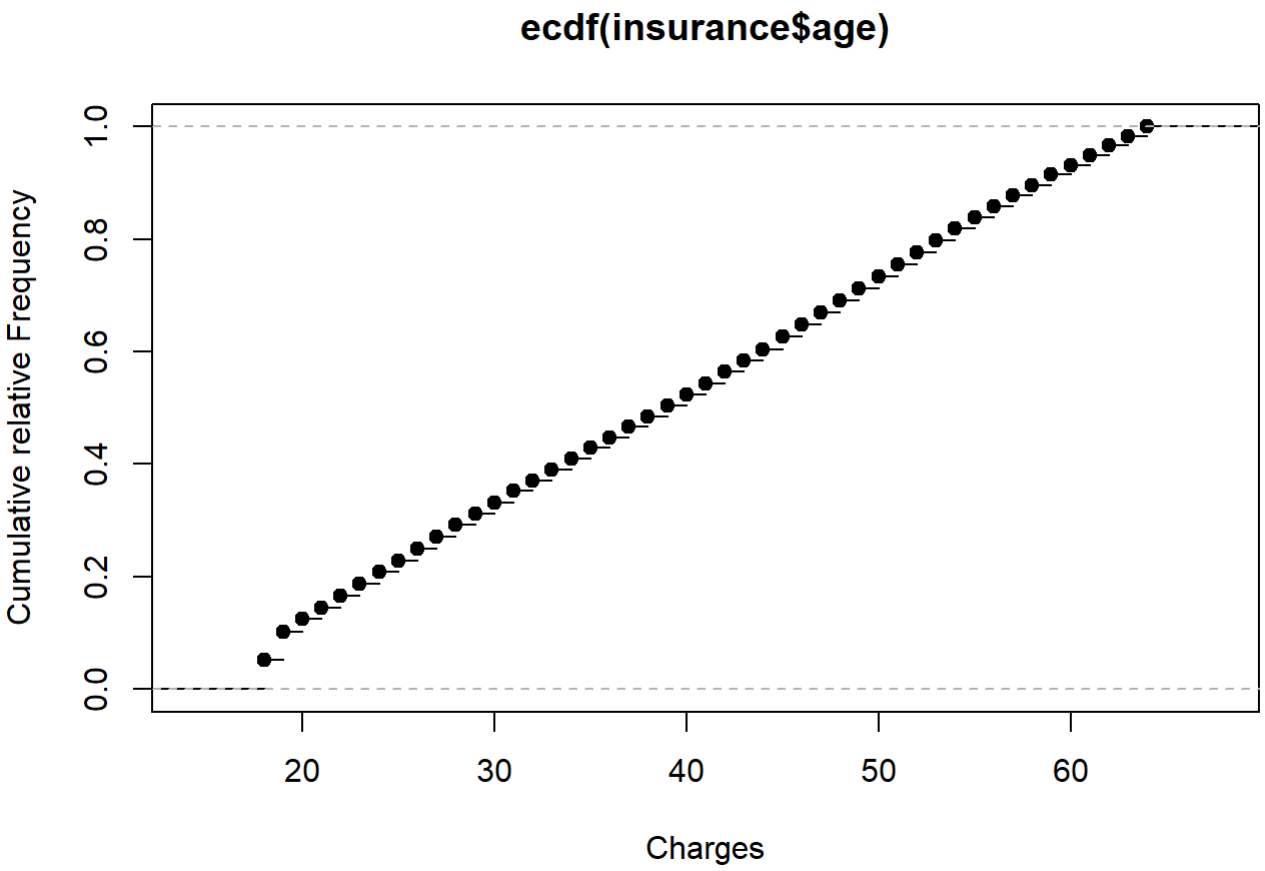
```
bmi_fn <- ecdf(insurance$bmi)
bmi_crf = plot(bmi_fn, xlab = "bmi", ylab = "Cumulative relative Frequency")
```




```
bmi_crf
```

```
## NULL
```

```
charges_fn <- ecdf(insurance$charges)
charges_crf = plot(age_fn, xlab = "Charges", ylab = "Cumulative relative Frequency")
```



b). Now we will check are any of the histograms of variables is normally distributed.

```
shapiro.test(insurance$age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  insurance$age
## W = 0.9447, p-value < 2.2e-16
```

```
shapiro.test(insurance$bmi)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  insurance$bmi
## W = 0.99389, p-value = 2.605e-05
```

```
shapiro.test(insurance$charges)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  insurance$charges
## W = 0.81469, p-value < 2.2e-16
```

From histogram and QQ-plot we assumed that bmi of data set is normally distributed. But after formal statistical test we can see none of variables are normally distributed. Because none of them have greater p-value than α = 0.05

3. Point Estimates

a). Computing sample mean, median, mode, variance and standard deviation of Numerical variables of data set.

```
# Creating subset of numerical variables.

numeric_subset <- subset(insurance, select = c("age","bmi","children", "charges"))
head(numeric_subset)
```

```
## # A tibble: 6 × 4
##     age   bmi children charges
##   <dbl> <dbl>    <dbl>   <dbl>
## 1    19  27.9        0  16885.
## 2    18  33.8        1   1726.
## 3    28  33          3   4449.
## 4    33  22.7        0  21984.
## 5    32  28.9        0   3867.
## 6    31  25.7        0   3757.
```

```
# Computing Mean for each columns.
colMeans(numeric_subset)
```

```
##          age          bmi     children      charges
##    39.207025    30.663397     1.094918 13270.422265
```

```
#Computing median for each columns.
apply(numeric_subset, 2, FUN = median)
```

```
##      age      bmi children  charges
##   39.000   30.400    1.000 9382.033
```

```
# Computing mode for each columns.
# Creating Mode function.

mode <- function(x) {
  ux <- unique(x)
  tab <- tabulate(match(x, ux))
  ux[tab == max(tab)]
}

apply(insurance, 2, mode)
```

```
##         age         sex         bmi    children      smoker      region
##        "18"      "male"    "32.300"         "0"        "no" "southeast"
##     charges
## " 1639.563"
```

```
# Computing variance for each columns.
var(numeric_subset, na.rm = TRUE)
```

```
##                   age          bmi     children      charges
## age      1.974014e+02 9.362337e+00   0.71930275 5.087480e+04
## bmi      9.362337e+00 3.718788e+01   0.09379476 1.464730e+04
## children 7.193027e-01 9.379476e-02   1.45321275 9.926742e+02
## charges  5.087480e+04 1.464730e+04 992.67419705 1.466524e+08
```

```
# Computing standard deviation for each variables.
sapply(numeric_subset, sd)
```

```
##          age          bmi     children      charges
##    14.049960     6.098187     1.205493 12110.011237
```

b). Now we are going to Determine the proportion of the data values that lies within 1.5 IQR. For that we have to remove outliers from data set.

```
# Finding first(Q1) and third(Q3) quartiles.
# Finding interguartile range

IQR <- quantile(insurance$bmi, 0.75) - quantile(insurance$bmi, 0.25)

# Finding lower and upper limit

lower <- quantile(insurance$bmi, 0.25) - 1.5*IQR
upper <- quantile(insurance$bmi, 0.75) + 1.5*IQR

# Removeing outliers

no_outliers <- insurance %>% filter(insurance$bmi < lower | insurance$bmi > upper) %>%
  select(age, bmi, charges)
no_outliers
```

```
## # A tibble: 9 × 3
##     age   bmi charges
##   <dbl> <dbl>   <dbl>
## 1    58  49.1  11381.
## 2    46  48.1   9433.
## 3    47  47.5   8084.
## 4    54  47.4  63770.
## 5    23  50.4   2438.
## 6    37  47.6  46114.
## 7    22  52.6  44501.
## 8    52  47.7   9749.
## 9    18  53.1   1163.
```

Confidence Interval

c). Assuming population variance is the same as sample variance, constructing 95% confidence interval for the population mean based on the entire data set and on the portion determined before.

For that we are here going to use z-value method.

```
# Taking randon sample size n.
n <- 500

z_star_95 <- qnorm(0.975)
z_star_95
```

```
## [1] 1.959964
```

```
# CI for bmi for Data set
bmi_z <- insurance %>%
  summarise(lower = mean(bmi) - z_star_95 * (sd(bmi) / sqrt(n)),
            upper = mean(bmi) + z_star_95 * (sd(bmi) / sqrt(n)))
bmi_z
```

```
##      lower    upper
## 1 30.12888 31.19792
```

```
# CI for bmi for no outliers data poportion
bmi_no_z <- no_outliers %>%
  summarise(lower = mean(bmi) - z_star_95 * (sd(bmi) / sqrt(n)),
            upper = mean(bmi) + z_star_95 * (sd(bmi) / sqrt(n)))
bmi_no_z
```

```
##      lower    upper
## 1 49.08012 49.47321
```

```
# CI for age for Data set
age_z <- insurance %>%
  summarise(lower = mean(age) - z_star_95 * (sd(age) / sqrt(n)),
            upper = mean(age) + z_star_95 * (sd(age) / sqrt(n)))
age_z
```

```
##      lower    upper
## 1 37.97551 40.43854
```

```
# CI for age for no outliers data poportion
age_no_z <- no_outliers %>%
  summarise(lower = mean(age) - z_star_95 * (sd(age) / sqrt(n)),
            upper = mean(age) + z_star_95 * (sd(age) / sqrt(n)))
age_no_z
```

```
##      lower    upper
## 1 38.33231 41.00103
```

```
# CI for charges for Data set
charges_z <- insurance %>%
  summarise(lower = mean(charges) - z_star_95 * (sd(charges) / sqrt(n)),
            upper = mean(charges) + z_star_95 * (sd(charges) / sqrt(n)))
charges_z
```

```
##      lower    upper
## 1 12208.95 14331.89
```

```
# CI for charges for no outliers data poportion
charges_no_z <- no_outliers %>%
  summarise(lower = mean(charges) - z_star_95 * (sd(charges) / sqrt(n)),
            upper = mean(charges) + z_star_95 * (sd(charges) / sqrt(n)))
charges_no_z
```

```
##     lower    upper
## 1 19824.7 23871.73
```

From results it's clearly showing confidence interval for the population mean based on the entire data set and on the portion determined in b) are totally different from each other.

d. Assuming population variance is unknown, construct 95% confidence interval for the population mean. Is it different from CI computed in part c?

Because population variance is unknown now we are only use no outlieres proportion here to compare. And we here use t-value method.

```
# Finding CI using t-value
t.test(no_outliers$age)
```

```
##
##  One Sample t-test
##
## data:  no_outliers$age
## t = 7.8169, df = 8, p-value = 5.157e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  27.96497 51.36836
## sample estimates:
## mean of x
##  39.66667
```

```
t.test(no_outliers$bmi)
```

```
##
##  One Sample t-test
##
## data:  no_outliers$bmi
## t = 65.928, df = 8, p-value = 3.117e-12
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  47.55308 51.00025
## sample estimates:
## mean of x
##  49.27667
```

```
t.test(no_outliers$charges)
```

```
##
##  One Sample t-test
##
## data:  no_outliers$charges
## t = 2.8392, df = 8, p-value = 0.02184
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##   4102.966 39593.464
## sample estimates:
## mean of x
##  21848.22
```

From results it's clearly showing when population variance is unknown confidence interval for the population mean is different from when it's known.

4. Modeling

In this part of project we are going to choose appropriate theoretical distribution for each of our variables. And we will estimate parameters of the theoretical distribution.

In visualization part we have seen histogram, cumulative relative frequency and QQ-plot. After observing all of them we can choose Discrete Uniform distribution for 'age' variable, gamma distribution fro 'bmi' and Gamma distribution for 'charges'.

Now we will estimate parameters for each of the theoretical distribution using our variables. And display the estimated theoretical distribution and relative frequency for the each variables.

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library("EnvStats")
```

```
##
## Attaching package: 'EnvStats'
```

```
## The following object is masked from 'package:MASS':
##
##     boxcox
```

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

```
## The following objects are masked from 'package:stats':
##
##     predict, predict.lm
```

```
## The following object is masked from 'package:base':
##
##     print.default
```

```
library(fitdistrplus)
```

```
## Loading required package: survival
```

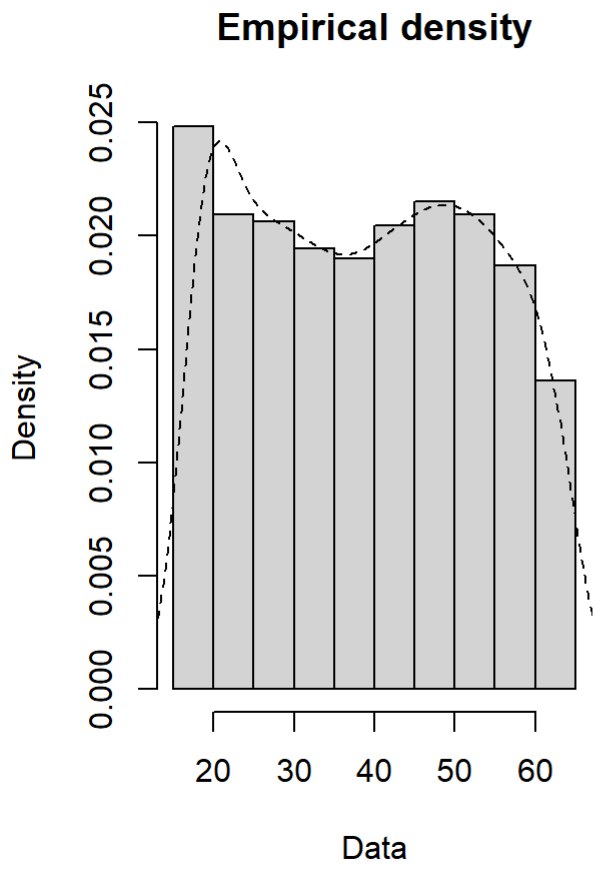
```
# Fitting age data to uniform distribution.
fit_age <- fitdistr(insurance$age, densfun = "poisson")
fit_age
```

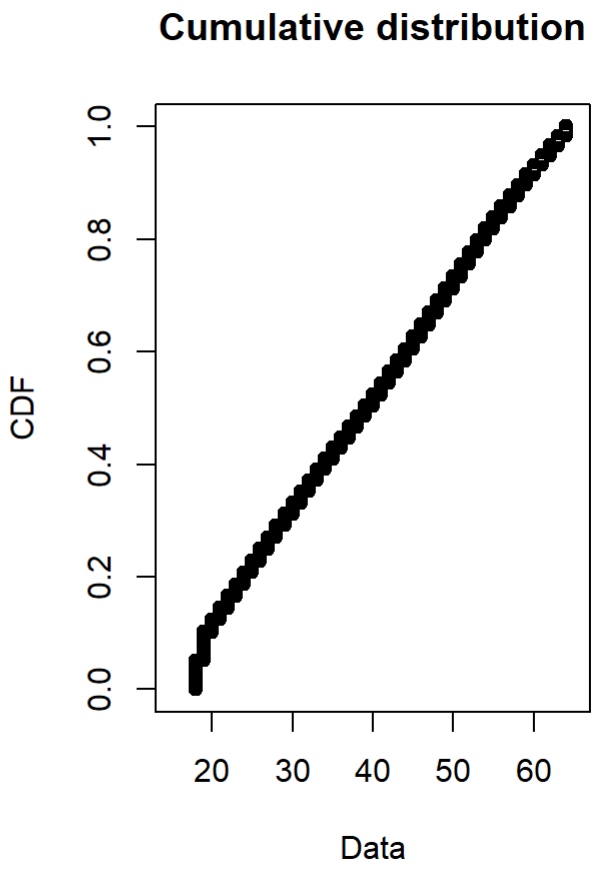
```
##      lambda
##   39.2070254
##  ( 0.1711803)
```

```
# Estimating parameters for Uniform Distribuition.
eunif(insurance$age, method = "mle")
```

```
## $distribution
## [1] "Uniform"
##
## $sample.size
## [1] 1338
##
## $parameters
## min max
##  18  64
##
## $n.param.est
## [1] 2
##
## $method
## [1] "mle"
##
## $data.name
## [1] "insurance$age"
##
## $bad.obs
## [1] 0
##
## attr(,"class")
## [1] "estimate"
```

```
# Histogram
plotdist(insurance$age, histo = TRUE, demp = TRUE)
```






```
# Fitting bmi data to gamma distribution.
fit_bmi <- fitdistr(insurance$bmi, densfun = "gamma")
```

```
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
```

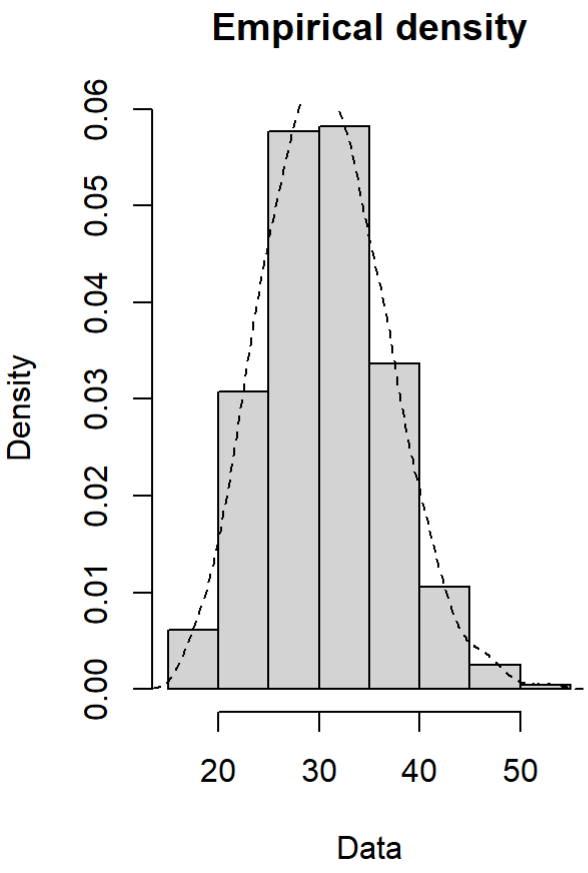
```
fit_bmi
```

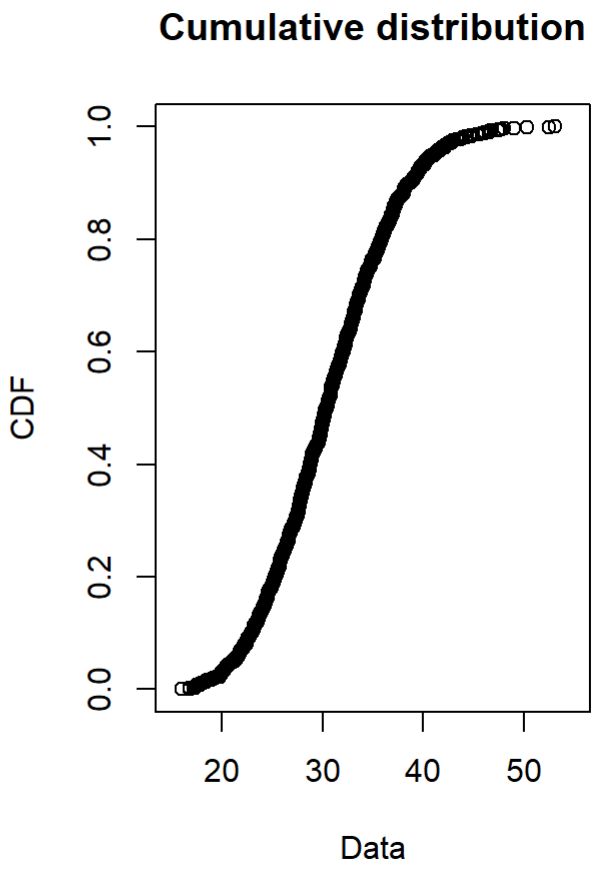
```
##       shape         rate
##   25.01207141    0.81569827
##  ( 0.96060454) ( 0.03164308)
```

```
# Estimating parameters for Gamma Distribuition.
egamma(insurance$bmi, method = "mle")
```

```
## $distribution
## [1] "Gamma"
##
## $sample.size
## [1] 1338
##
## $parameters
##     shape     scale
## 25.302519  1.211871
##
## $method
## [1] "MLE"
##
## $data.name
## [1] "insurance$bmi"
##
## $bad.obs
## [1] 0
##
## attr(,"class")
## [1] "estimate"
```

```
# Histogram
plotdist(insurance$bmi, histo = TRUE, demp = TRUE)
```





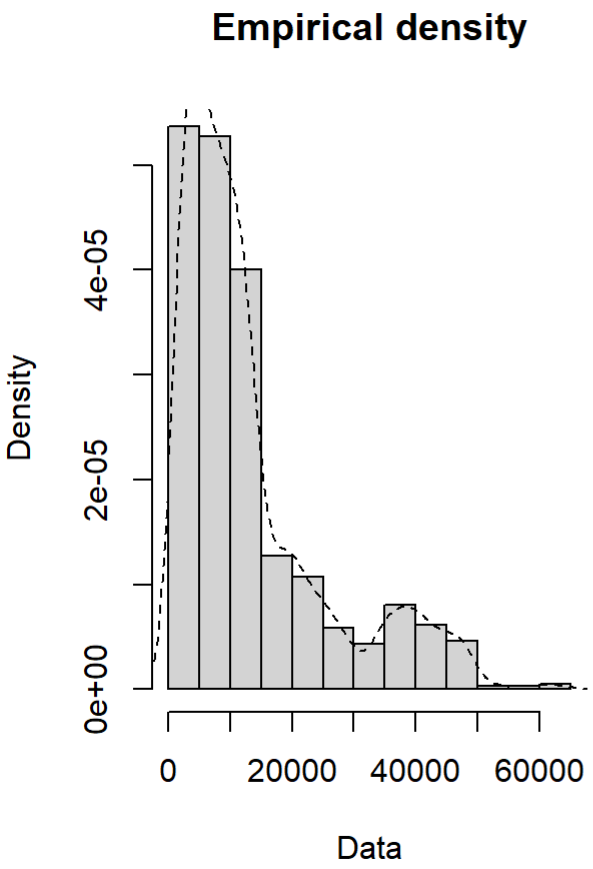
```
# Fitting charges data to gamma distribution.
fit_charges <- fitdistr(insurance$charges, densfun = "exponential")
fit_charges
```

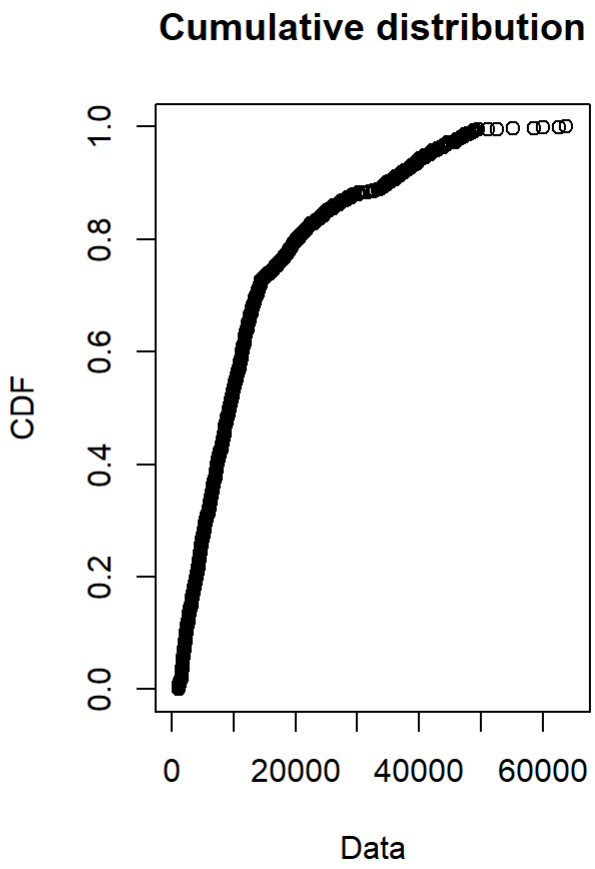
```
##        rate
##   7.535555e-05
##  (2.060095e-06)
```

```
# Estimating parameters for exponential Distribuiton.
eexp(insurance$charges, method = "mle")
```

```
## $distribution
## [1] "Exponential"
##
## $sample.size
## [1] 1338
##
## $parameters
##         rate
## 7.535555e-05
##
## $n.param.est
## [1] 1
##
## $method
## [1] "mle/mme"
##
## $data.name
## [1] "insurance$charges"
##
## $bad.obs
## [1] 0
##
## attr(,"class")
## [1] "estimate"
```

```
# Histogram
plotdist(insurance$charges, histo = TRUE, demp = TRUE)
```



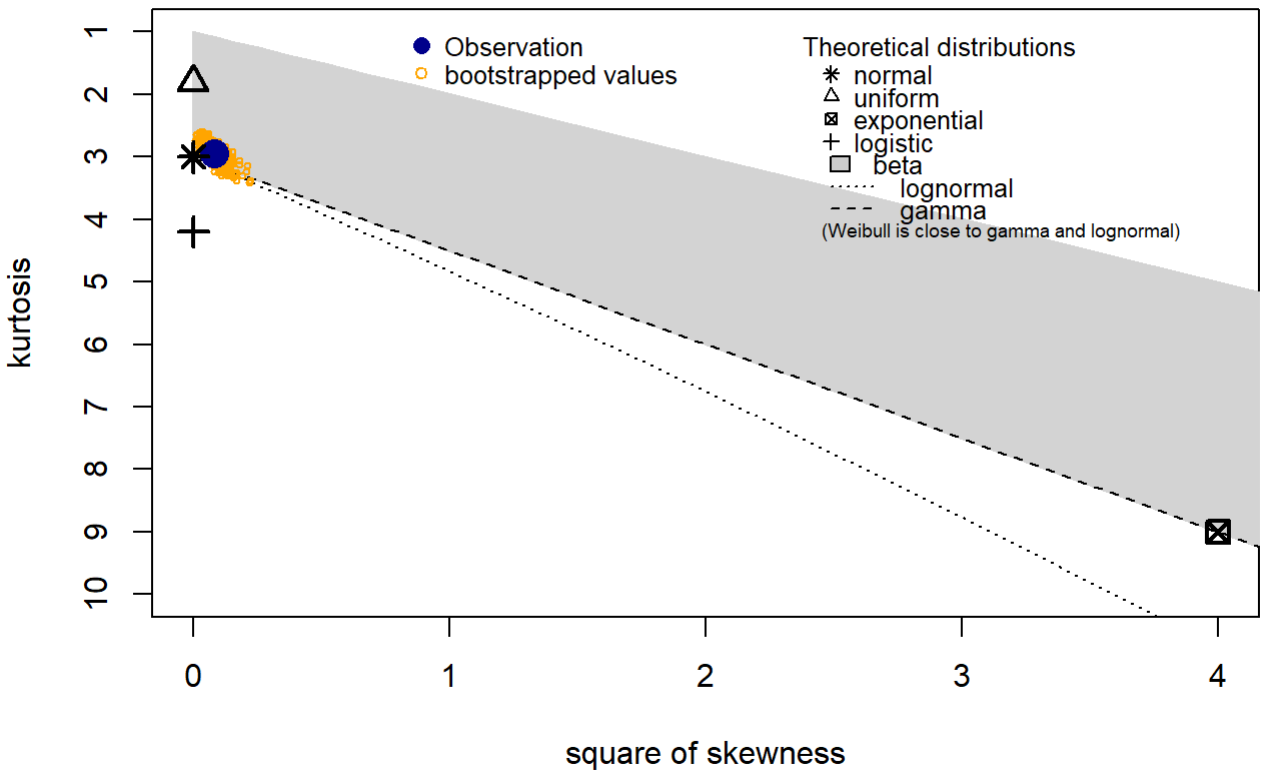


b). to choose which statistical distribution are best fit for data we must use Distribution fitting. If we choose wrong distribution our calculation will be wrong, and this is why our data must fit one of the theoretical distribution. Also to find parameterized mathematical function we must have to know which distribution is perfect for our Data. Parameterized function will calculate the probability of any observation from the sample space. And from distribution we can present our data graphically. By knowing which distribution are accurate for our data we can make inference about overall population.

c). Now we are going to check Could the data fit more than one theoretical distributions or not. For that we are going to follow Fitting Distribution process for 'bmi' variable.

Now we will plot Cullen and Frey graph to assess the potential fit of the data in terms of skewness and kurtosis.

```
# Cullen and Frey graph.
descdist(insurance$bmi, discrete = FALSE, boot = 500)
```



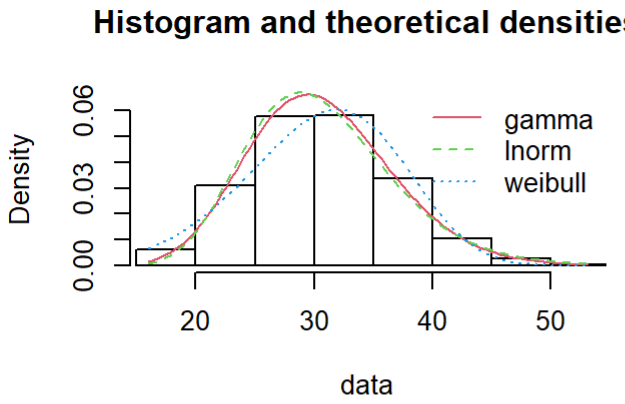

```
## summary statistics
## ------
## min:  15.96   max:  53.13
## median:  30.4
## mean:  30.6634
## estimated sd:  6.098187
## estimated skewness:  0.2840471
## estimated kurtosis:  2.949268
```

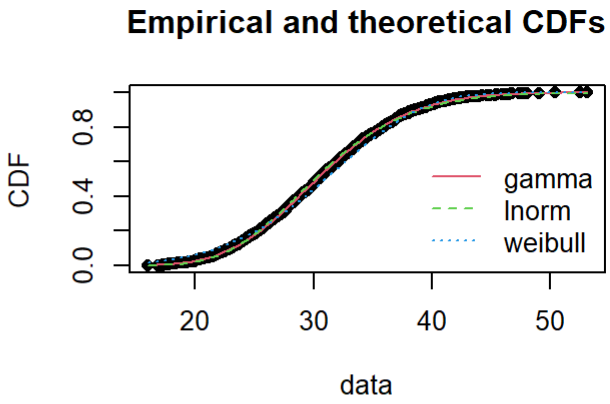
```
# Fitting data

dists <- c("gamma","lnorm","weibull")
fit <- list()
for (i in 1:length(dists))
  fit[[i]] <- fitdist(insurance$bmi, dists[i])

for (i in 1:length(dists))
  print(summary(fit[[i]]))
```

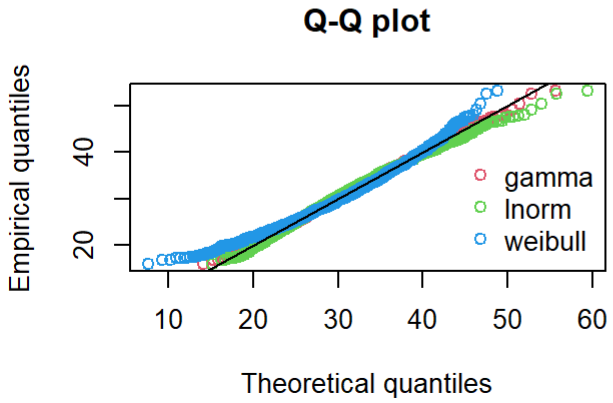
```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## shape 25.017624 0.96081162
## rate   0.815914 0.03165116
## Loglikelihood:  -4306.857   AIC:  8617.713   BIC:  8628.111
## Correlation matrix:
##           shape      rate
## shape 1.0000000 0.9900242
## rate  0.9900242 1.0000000
##
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters :
##          estimate  Std. Error
## meanlog 3.4029458 0.005533719
## sdlog   0.2024162 0.003912501
## Loglikelihood:  -4314.321   AIC:  8632.641   BIC:  8643.039
## Correlation matrix:
##         meanlog sdlog
## meanlog       1     0
## sdlog         0     1
##
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## shape  5.331338  0.1071856
## scale 33.173019  0.1801183
## Loglikelihood:  -4356.627   AIC:  8717.253   BIC:  8727.651
## Correlation matrix:
##           shape     scale
## shape 1.0000000 0.3288224
## scale 0.3288224 1.0000000
```

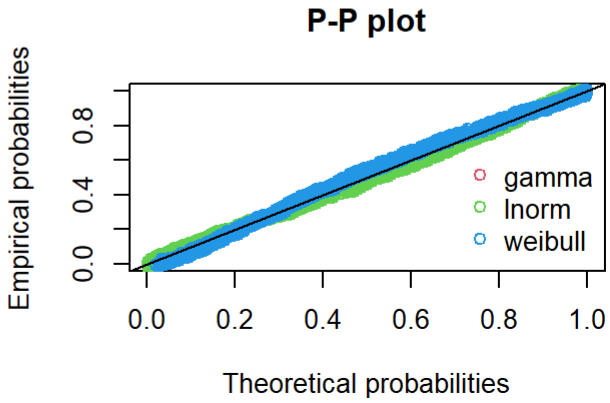
Now using output to produce PDF's, CDF's, Q-Q plots and P-P plots.

```
par(mfrow= c(2,2))
plot.legend <- dists
denscomp(fit, legendtext = plot.legend)
cdfcomp(fit, legendtext = plot.legend)
qqcomp(fit, legendtext = plot.legend)
ppcomp(fit, legendtext = plot.legend)
```










```
# Goodness of fit
gofstat(fit, fitnames = c("gamma", "lnorm", "weibull"))
```

```
## Goodness-of-fit statistics
##                                 gamma      lnorm    weibull
## Kolmogorov-Smirnov statistic 0.01895978 0.03224282 0.04298374
## Cramer-von Mises statistic   0.07584272 0.28301195 0.88293803
## Anderson-Darling statistic   0.47839594 1.76864652 6.51604342
##
## Goodness-of-fit criteria
##                                 gamma    lnorm  weibull
## Akaike's Information Criterion 8617.713 8632.641 8717.253
## Bayesian Information Criterion 8628.111 8643.039 8727.651
```

From this result we can see Gamma distribution seems to be best fit among the candidates we chose to explore.

And from here we can say that we can fit the data more than one theoretical distribution But there will be only one which will fit best.

And this the reason why our data must fit one of the theoretical distributions. and we can primarily choose any distribution but, we must go through fitting distributions process to make sure our data actually fitting best.

e). From all results and visualizations we can say each distributions fitted well.

5. Use a random number generator

(CLT) Use a random number generator to pick N samples of size n from your original data. Use N = 10, 100, 500 and n = 2,5,10. For each choice of N and n:

a. Compute the average X *.

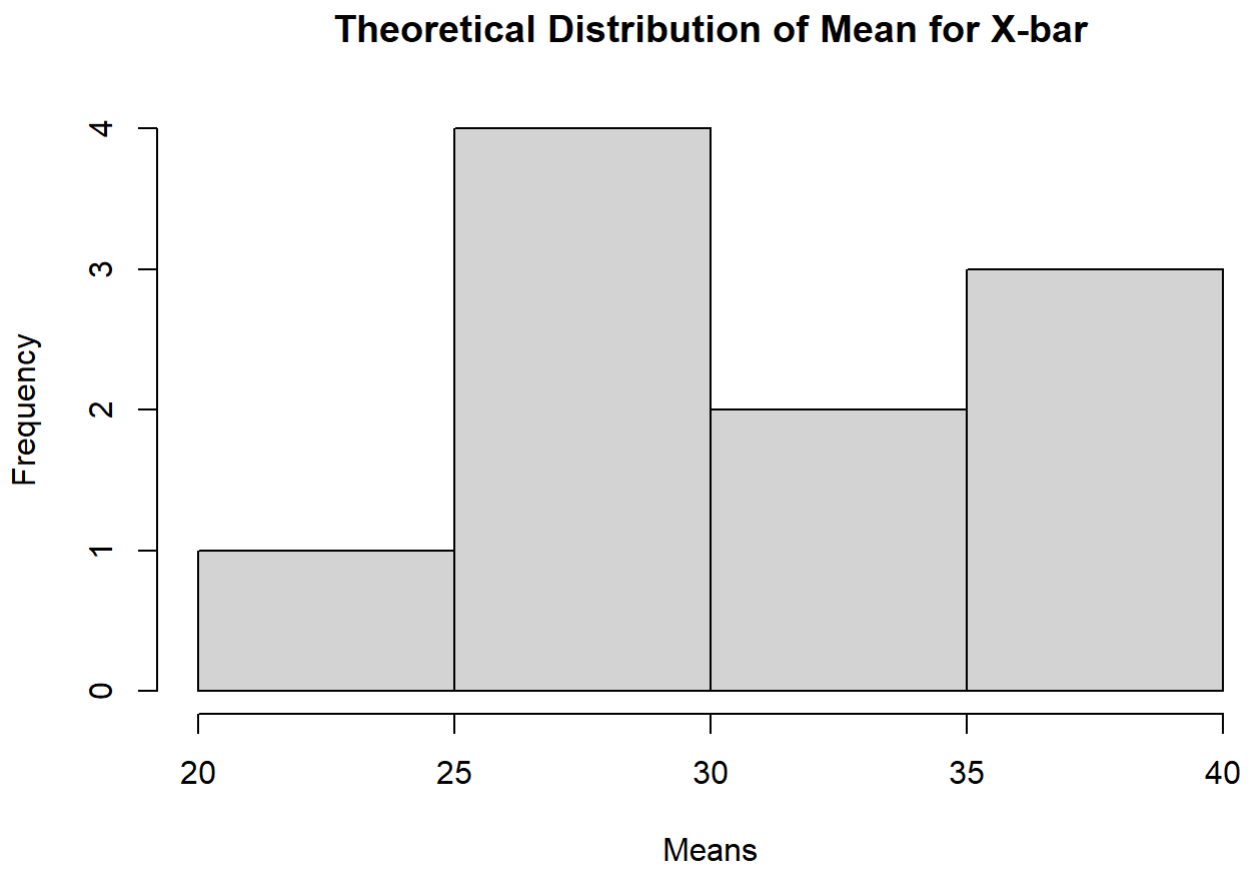
b. Base this on the mean and standard deviation from your original data, state the approximate theoretical distribution of X *.

c. Construct a histogram displaying your data.

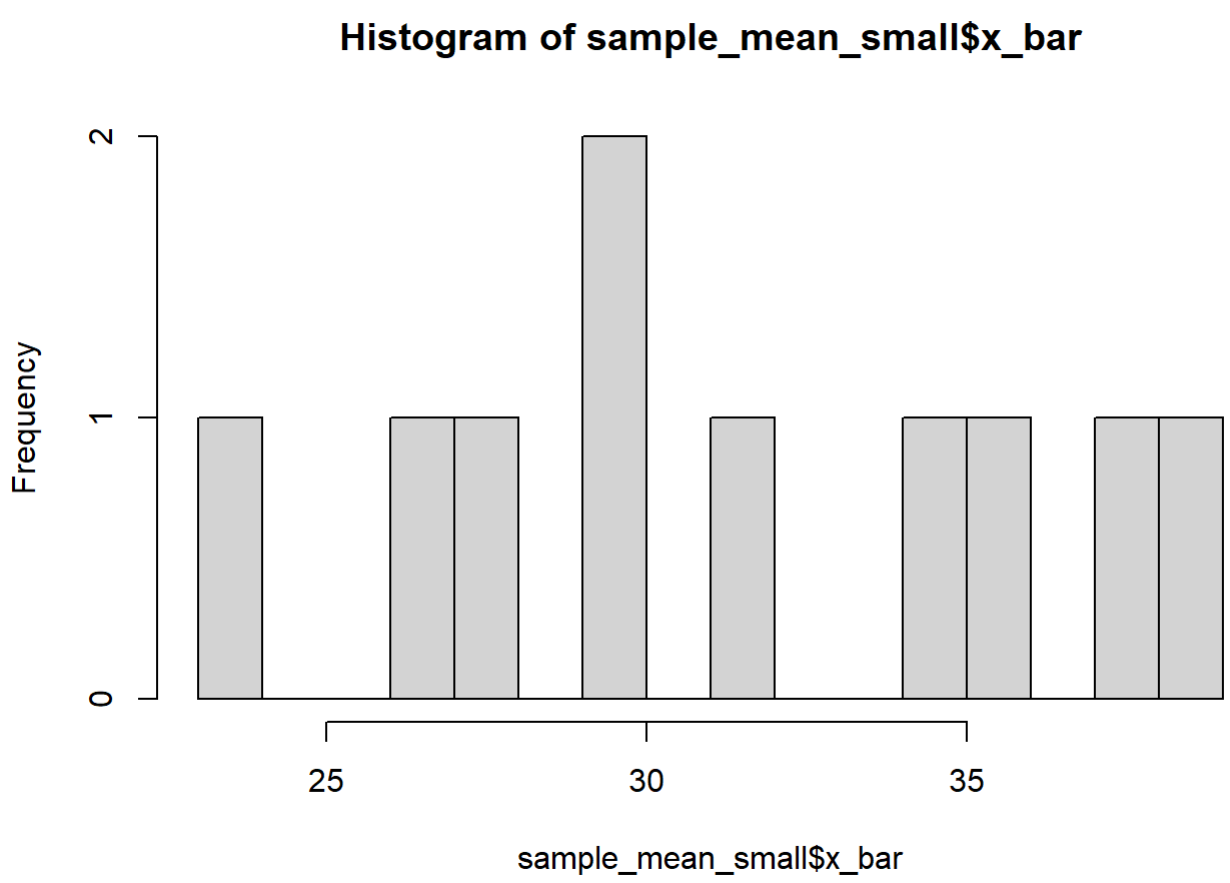
d. Draw the graph of the theoretical distribution of X * and compare the relative frequencies to the probabilities. Are the values close?

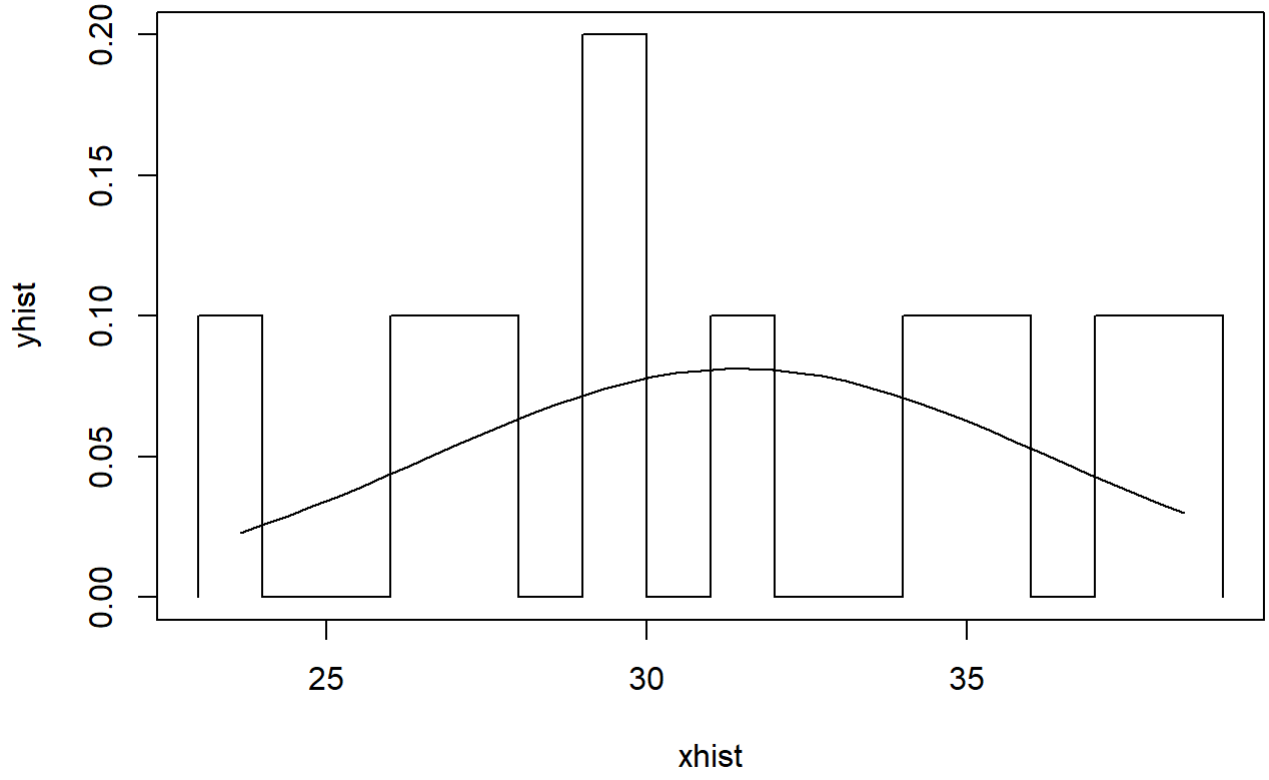
e. Does it appear that the data of averages fit the distribution of X * well?

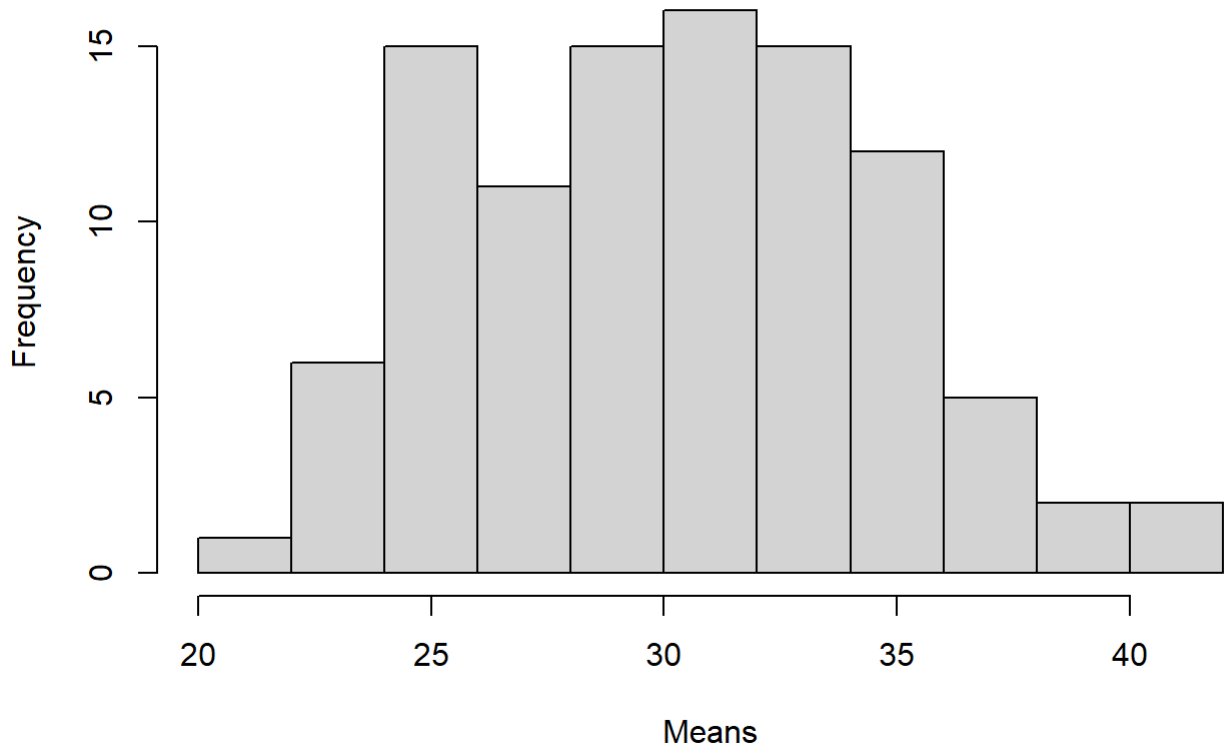
```
library("dplyr")
# Computing x_bar and ploting histogram
for (j in c(10,100,500)){
  sample_mean_small <- vector(mode = "numeric", length = 0)
  for (i in 1:j){
    sample_mean <- insurance %>%
                sample_n(size = 2) %>%
                summarise(x_bar = mean(bmi))
    sample_mean_small <- rbind(sample_mean_small, sample_mean)
  }
  hist(sample_mean_small$x_bar, main= 'Theoretical Distribution of Mean for X-bar',xlab= 'Means'
)
  print(mean(sample_mean_small$x_bar))

  h <- hist(sample_mean_small$x_bar, breaks = 20)
xhist <- c(min(h$breaks),h$breaks)
yhist <- c(0, h$density,0)
xfit<- seq(min(sample_mean_small$x_bar),max(sample_mean_small$x_bar),length=40)
yfit<- dnorm(xfit, mean = mean(sample_mean_small$x_bar), sd=sd(sample_mean_small$x_bar))
plot(xhist,yhist,type='s')
lines(xfit,yfit)
}
```


Theoretical Distribution of Mean for X-bar
Frequency
0
1
2
3
4
20
25
30
35
40
Means

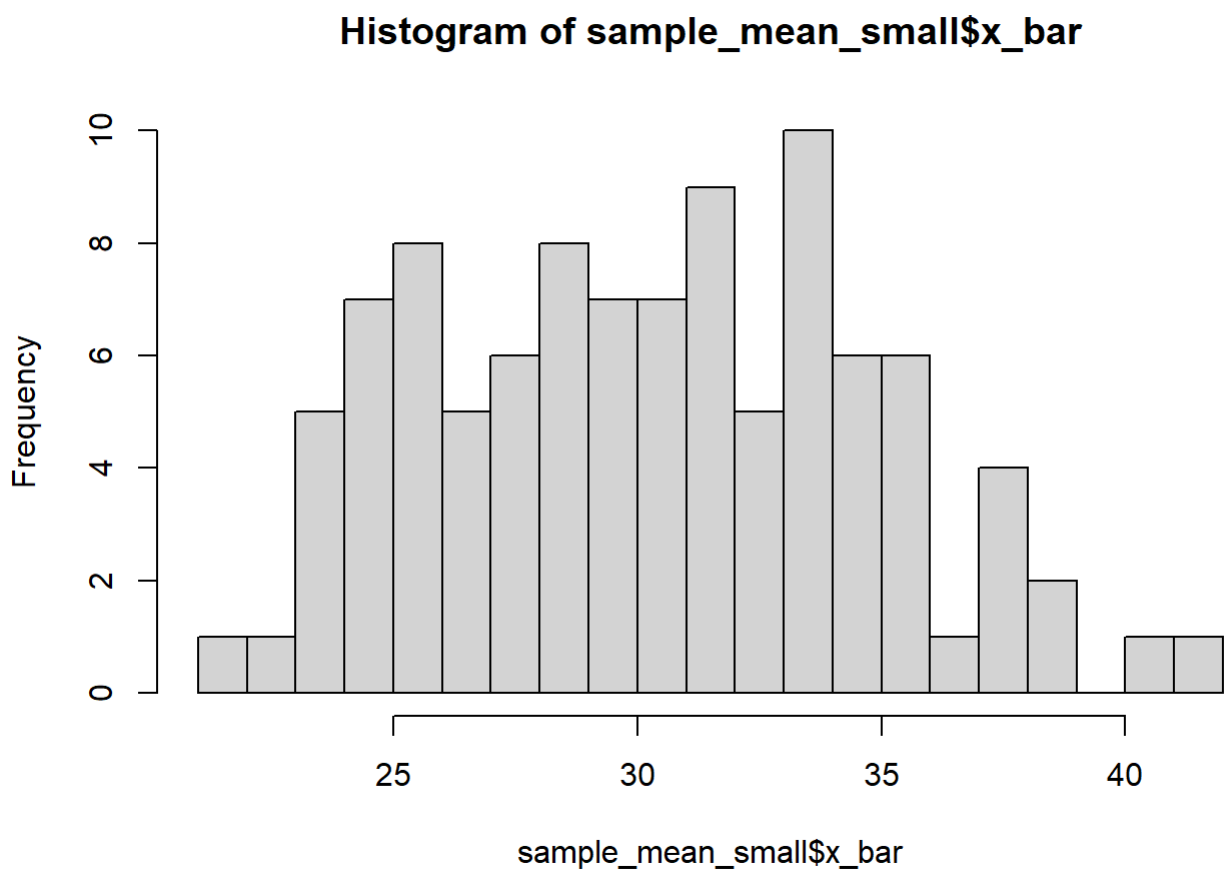
```
## [1] 31.45325
```

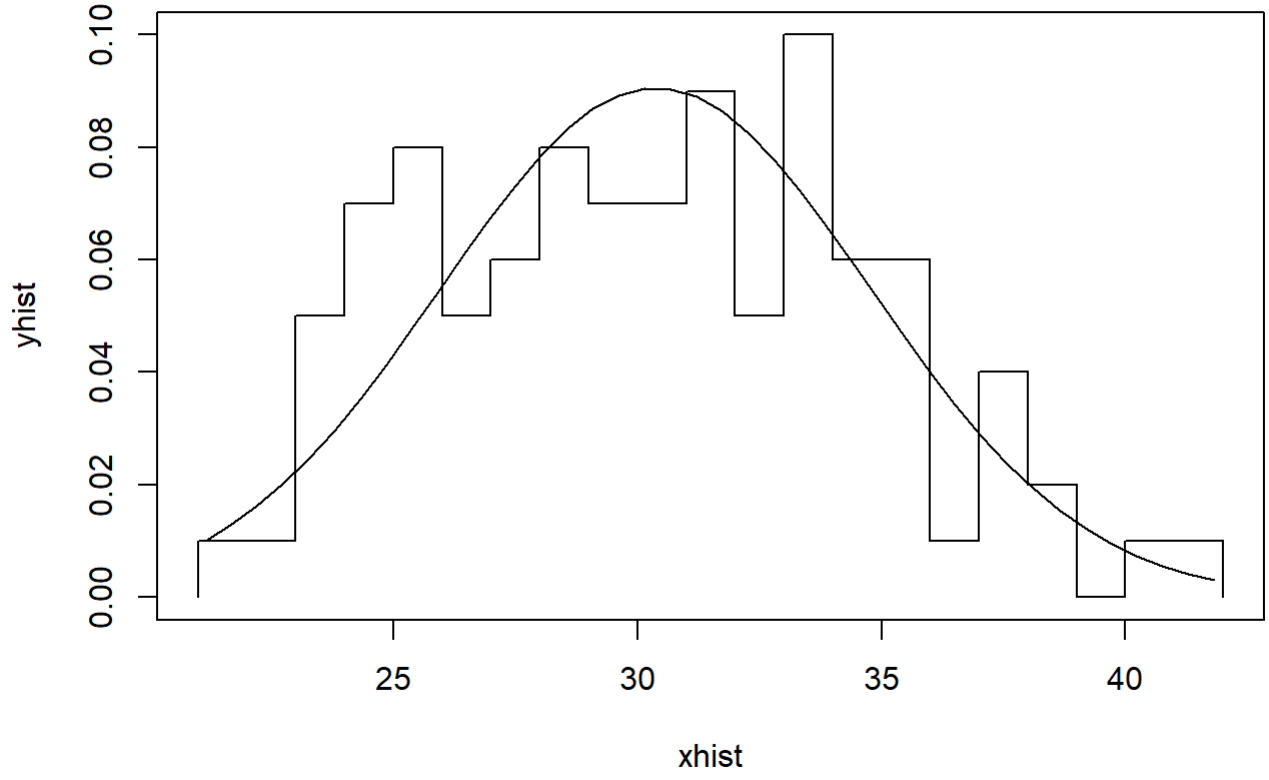

Histogram of sample_mean_small$x_bar
Frequency
0
1
2
25
30
35
sample_mean_small$x_bar

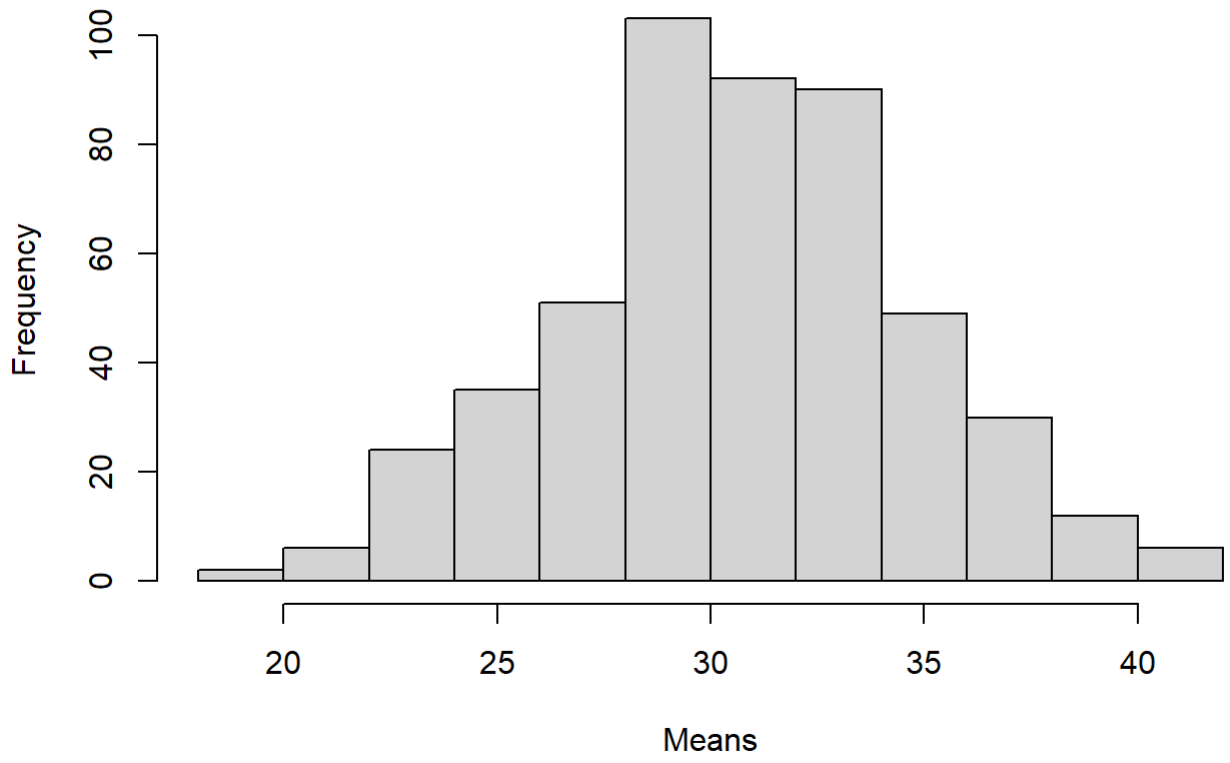

yhist
0.00
0.05
0.10
0.15
0.20
25
30
35
xhist


Theoretical Distribution of Mean for X-bar
Frequency
0
5
10
15
20
25
30
35
40
Means

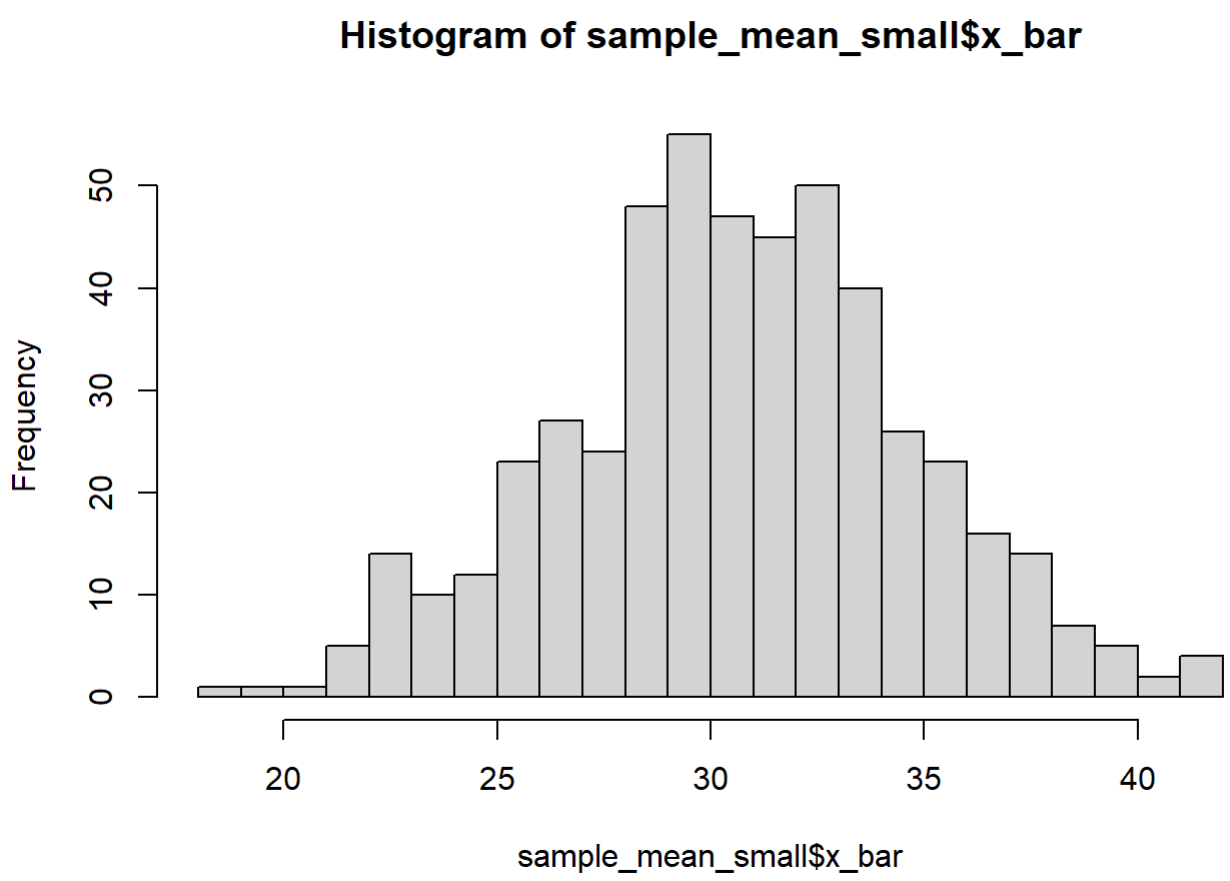
```
## [1] 30.36995
```

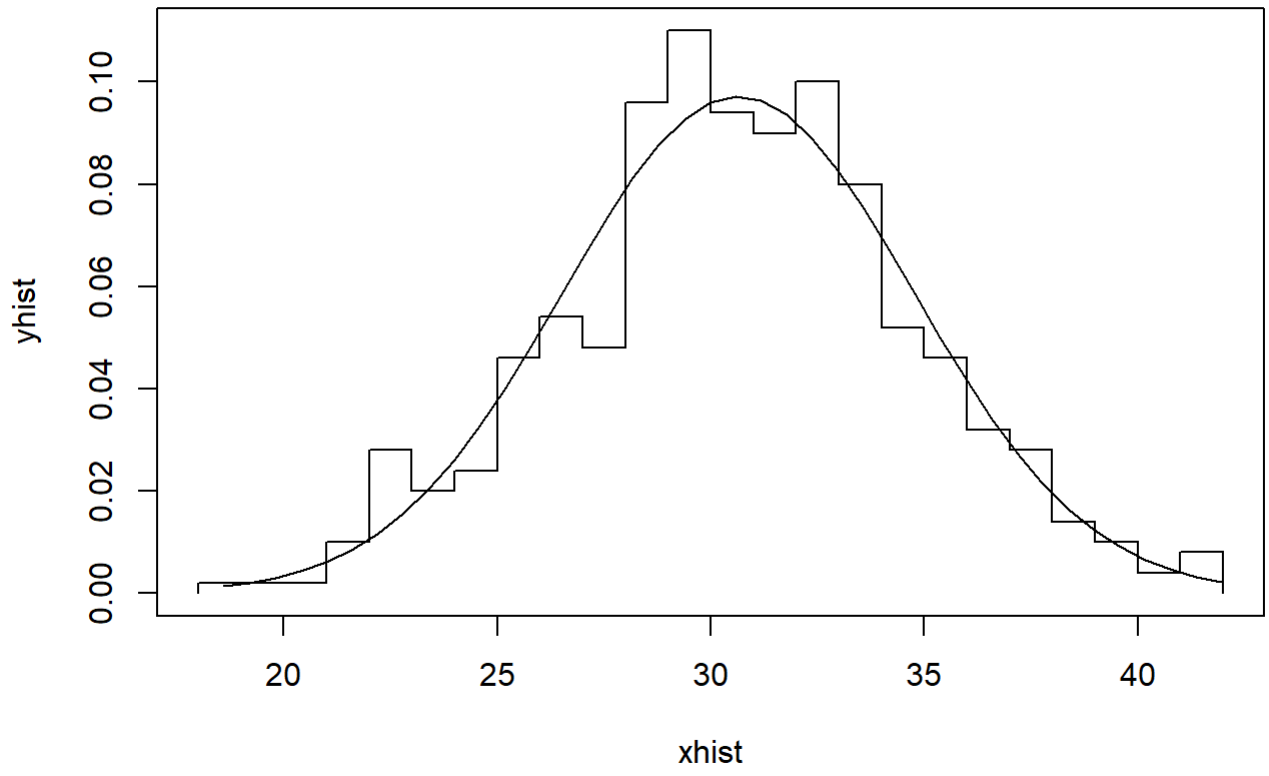

Histogram of sample_mean_small$x_bar
Frequency
0
2
4
6
8
10
25
30
35
40
sample_mean_small$x_bar


yhist
0.00
0.02
0.04
0.06
0.08
0.10
25
30
35
40
xhist

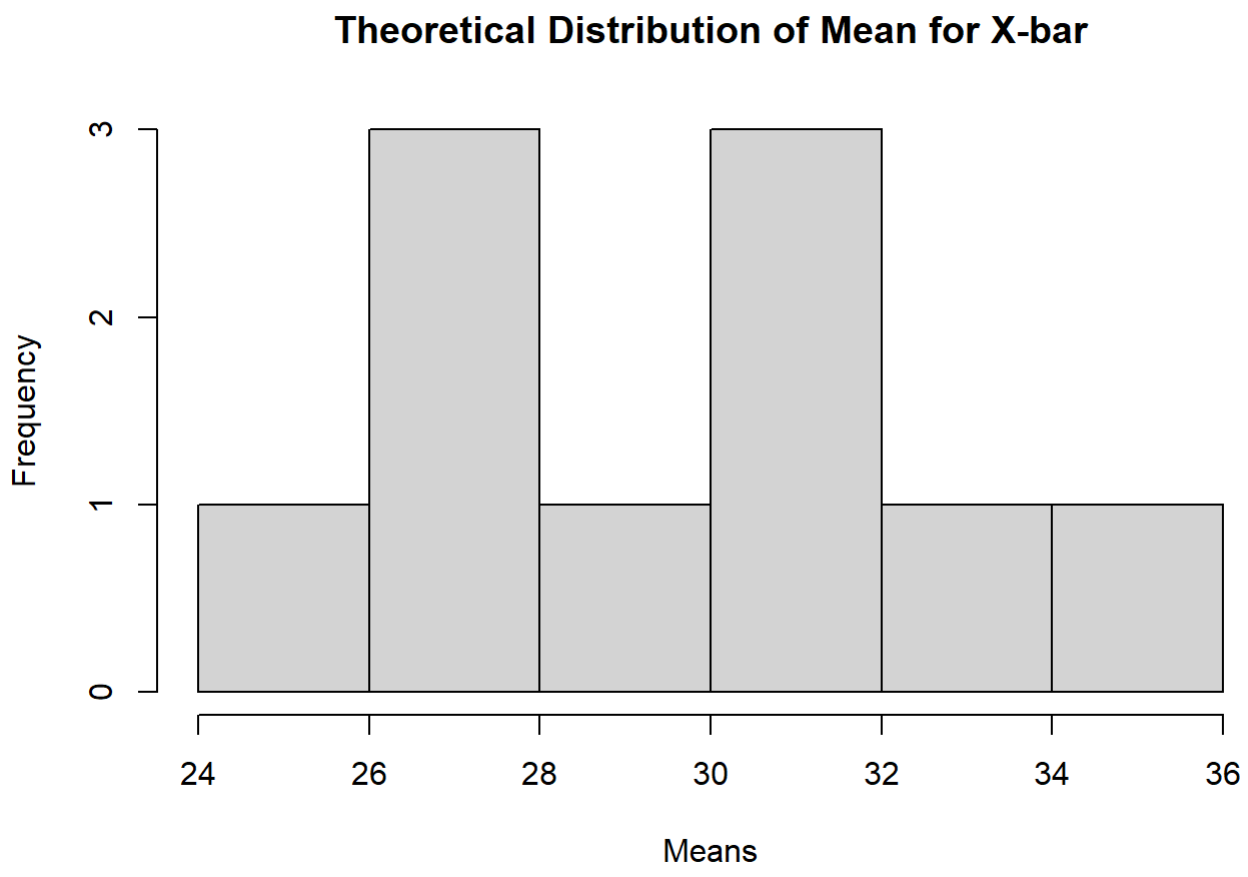

Theoretical Distribution of Mean for X-bar
Frequency
0
20
40
60
80
100
20
25
30
35
40
Means

```
## [1] 30.64592
```

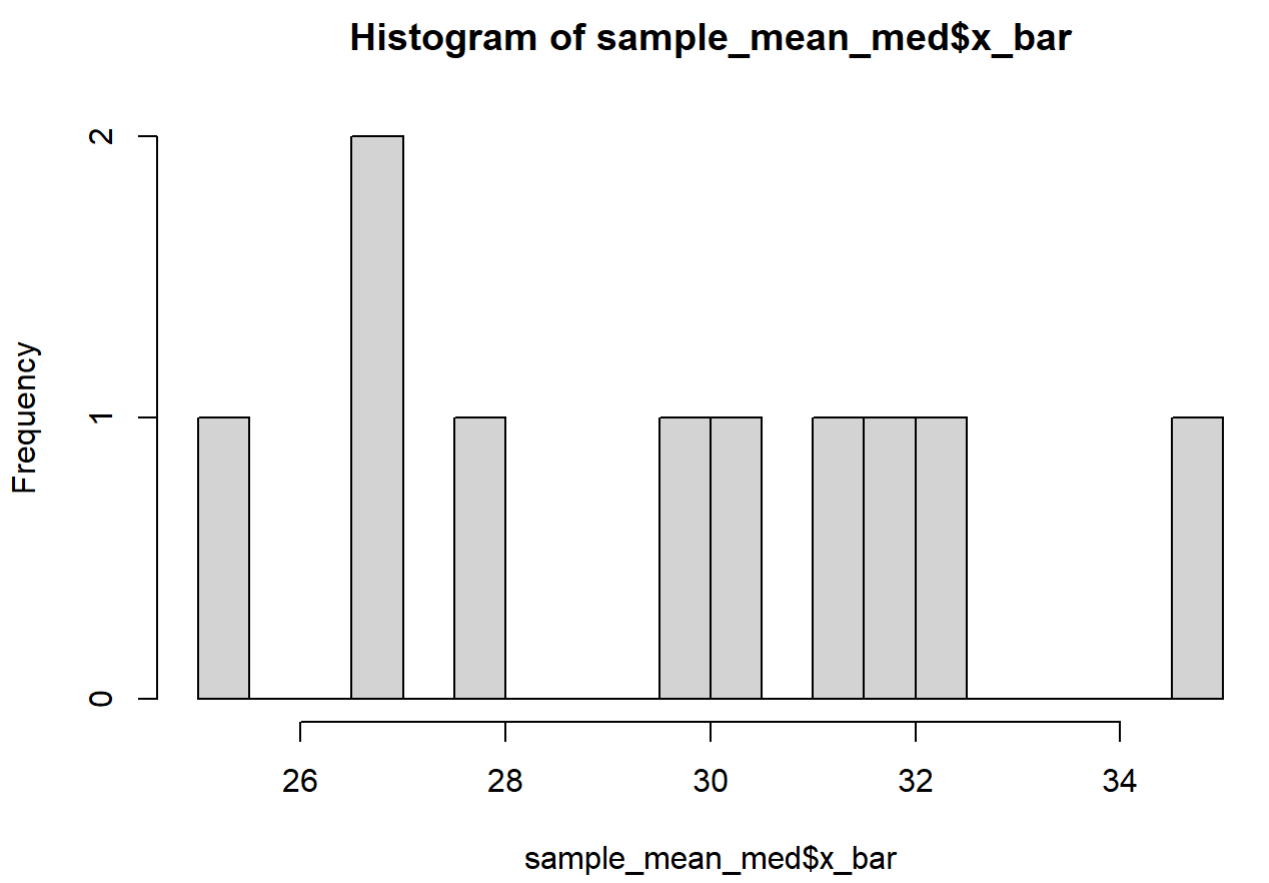

Histogram of sample_mean_small$x_bar
Frequency
0
10
20
30
40
50
20
25
30
35
40
sample_mean_small$x_bar

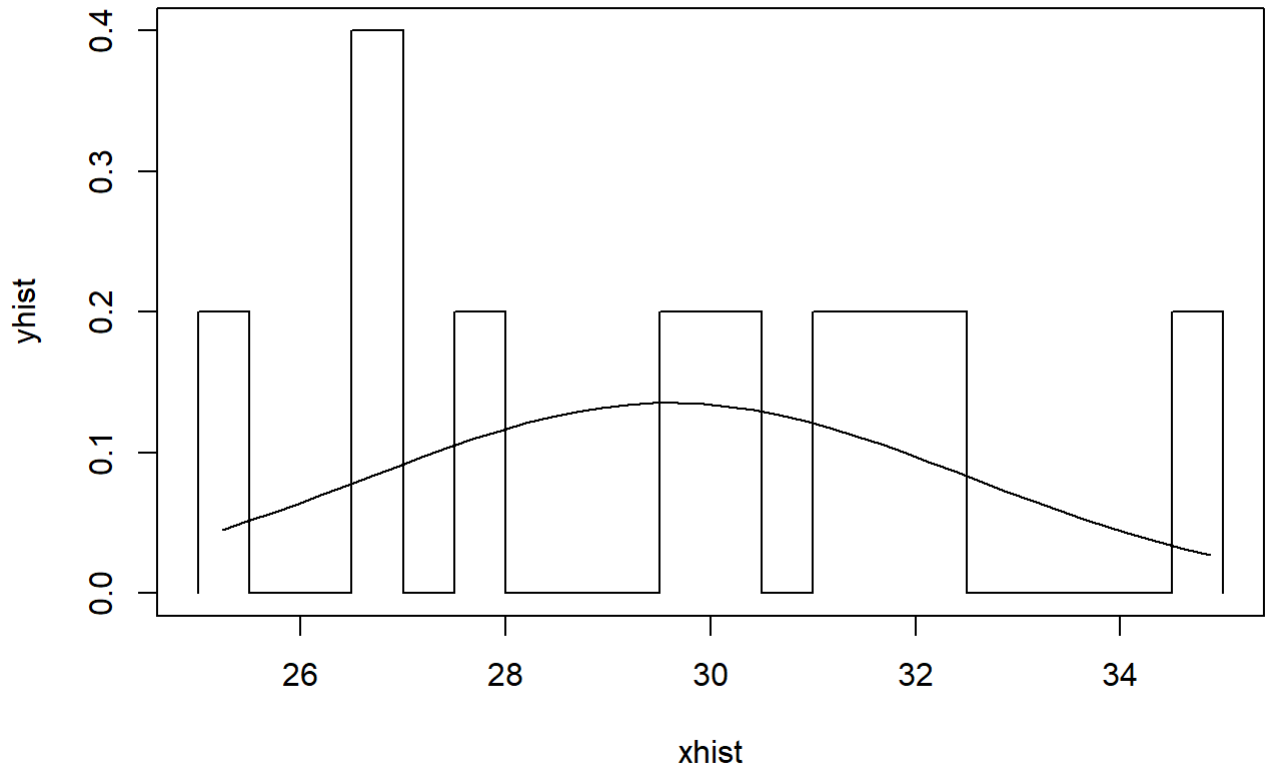

yhist
0.00
0.02
0.04
0.06
0.08
0.10
20
25
30
35
40
xhist

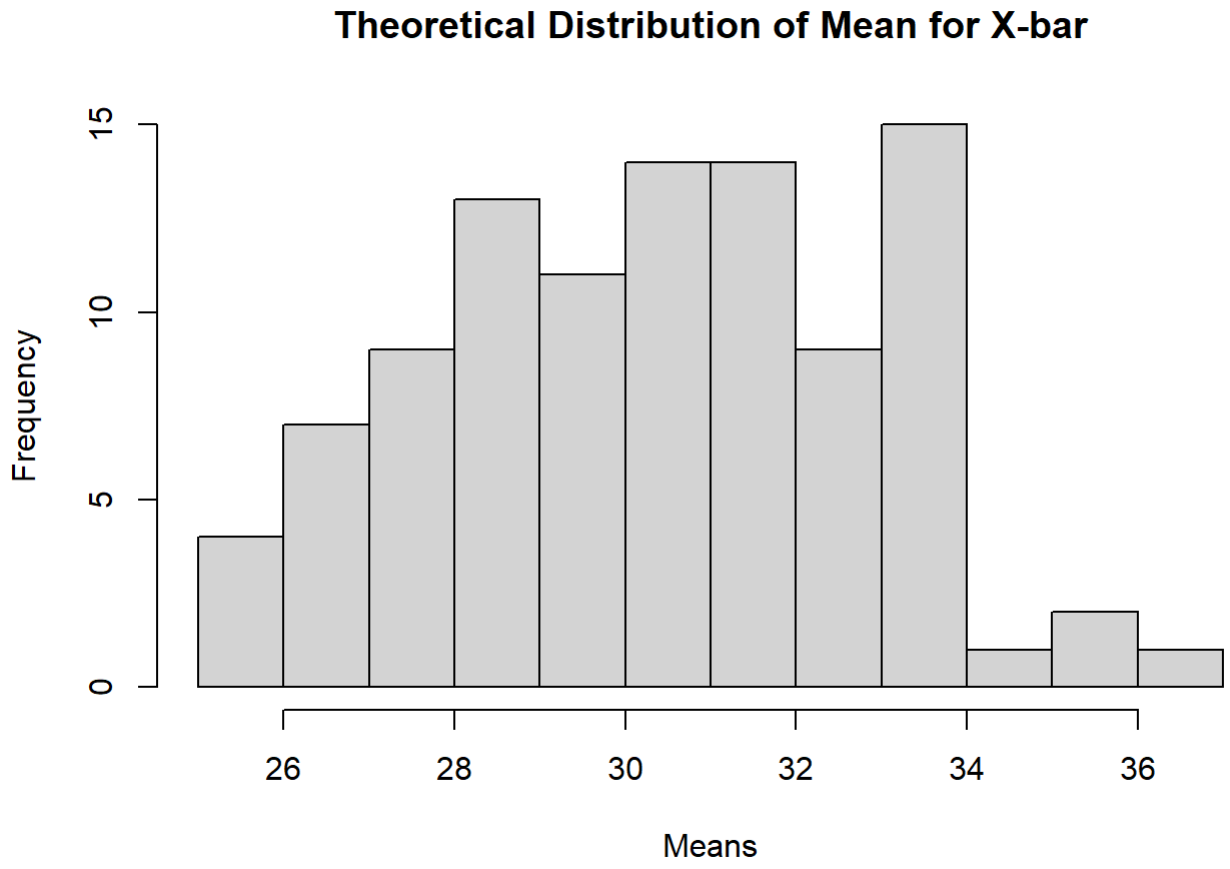
```
for (j in c(10,100,500)){
  sample_mean_med <- vector(mode = "numeric", length = 0)
  for (i in 1:j){
    sample_mean <- insurance %>%
                sample_n(size = 5) %>%
                summarise(x_bar = mean(bmi))
    sample_mean_med <- rbind(sample_mean_med, sample_mean)
  }
  hist(sample_mean_med$x_bar, main= 'Theoretical Distribution of Mean for X-bar',xlab= 'Means')
  print(mean(sample_mean_med$x_bar))

  h <- hist(sample_mean_med$x_bar, breaks = 20)
xhist <- c(min(h$breaks),h$breaks)
yhist <- c(0, h$density,0)
xfit<- seq(min(sample_mean_med$x_bar),max(sample_mean_med$x_bar),length=40)
yfit<- dnorm(xfit, mean = mean(sample_mean_med$x_bar), sd=sd(sample_mean_med$x_bar))
plot(xhist,yhist,type='s')
lines(xfit,yfit)
}
```



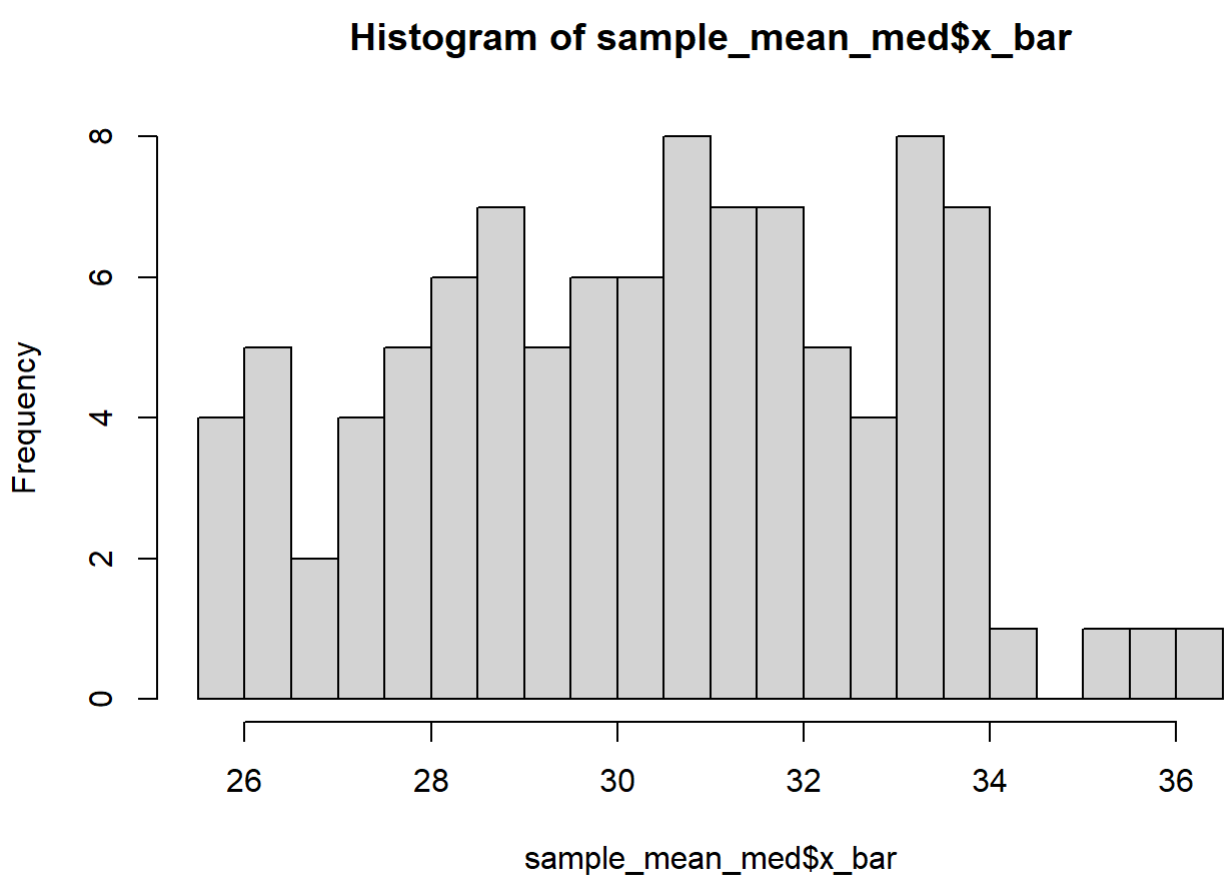

```
## [1] 29.5954
```

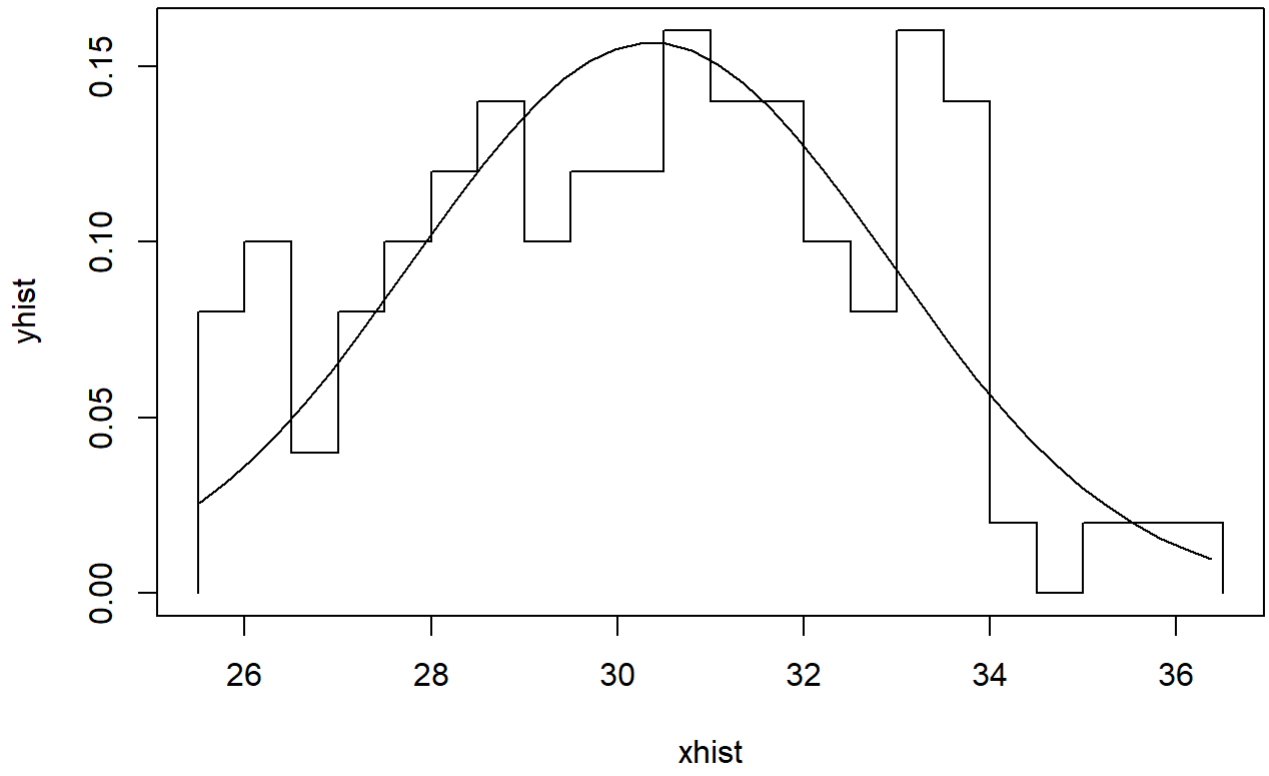

Histogram of sample_mean_med$x_bar
Frequency
0
1
2
26
28
30
32
34
sample_mean_med$x_bar

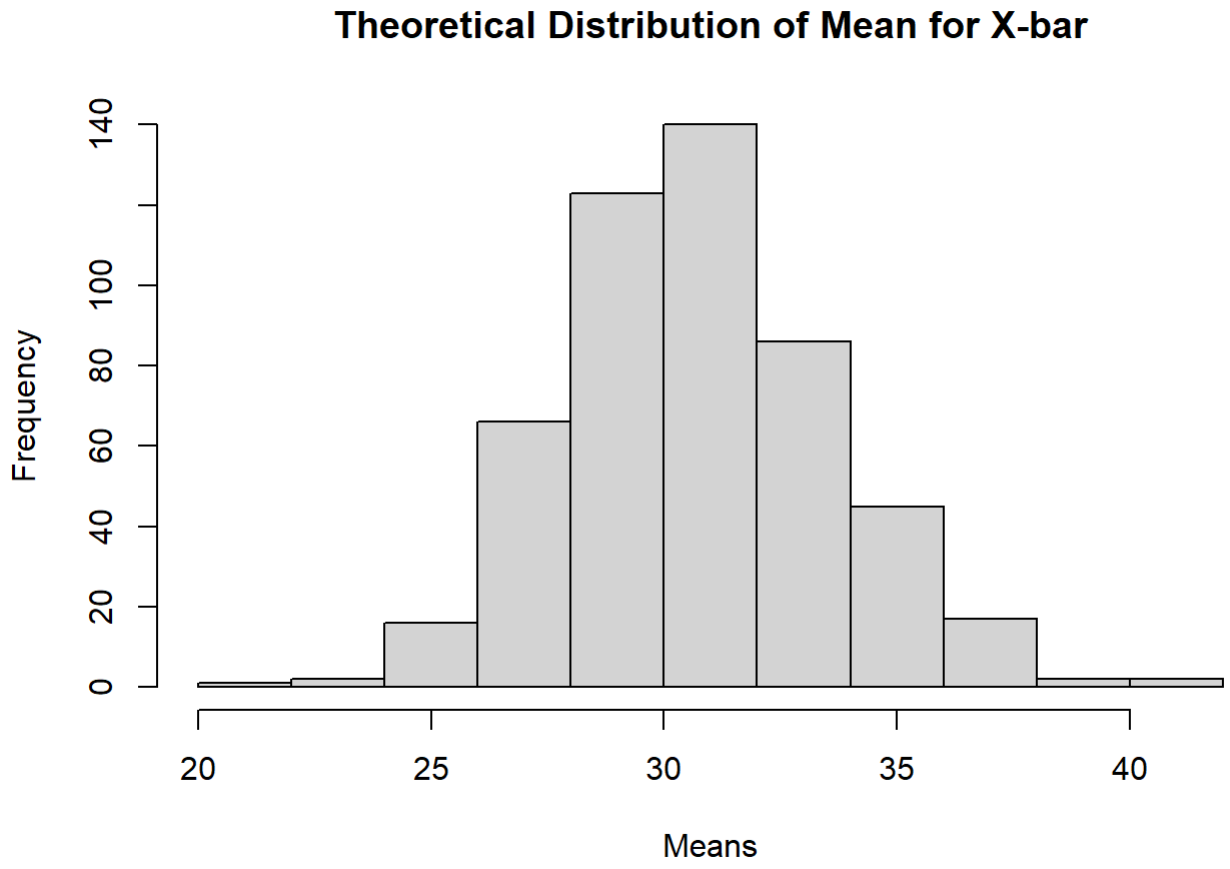

yhist
0.0
0.1
0.2
0.3
0.4
26
28
30
32
34
xhist


Theoretical Distribution of Mean for X-bar
Frequency
0
5
10
15
26
28
30
32
34
36
Means

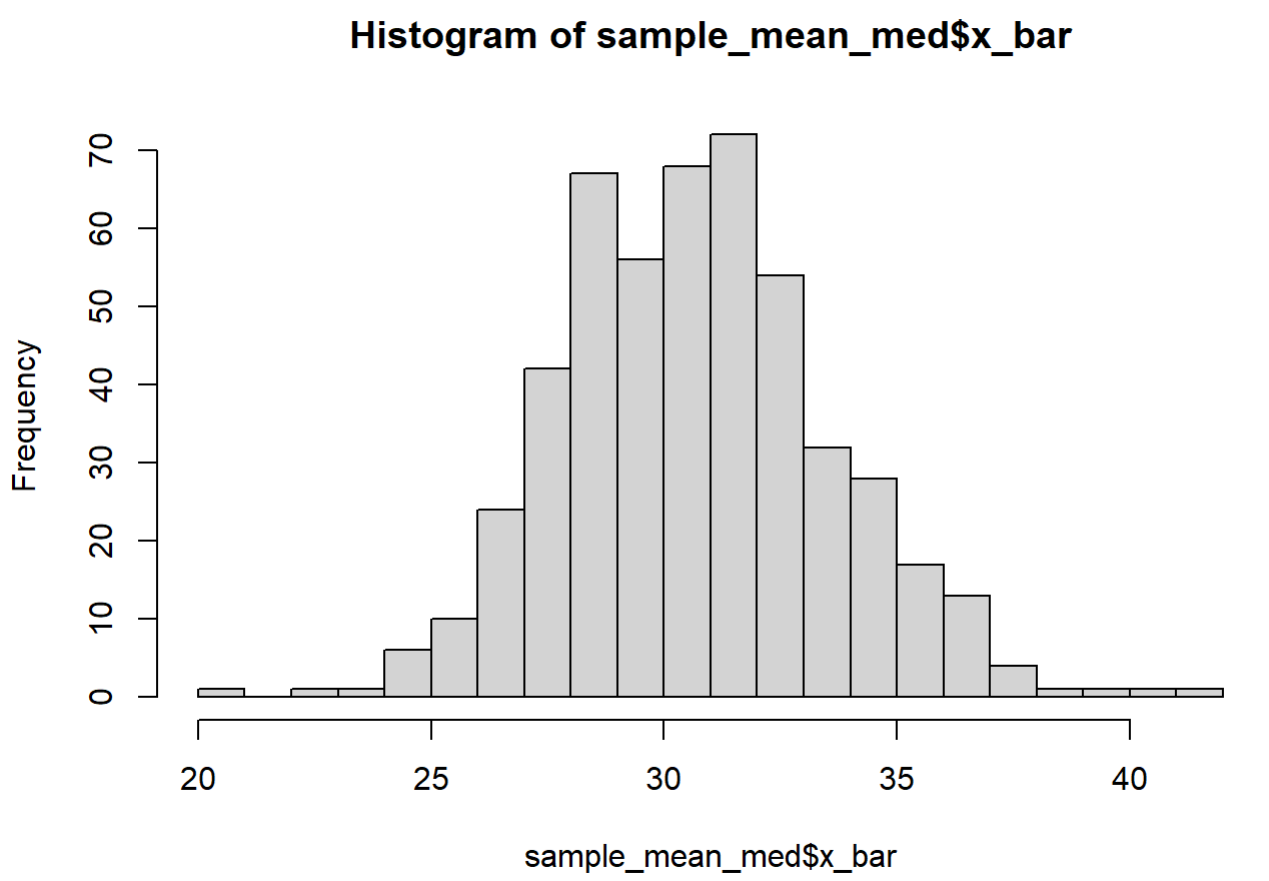
```
## [1] 30.35954
```

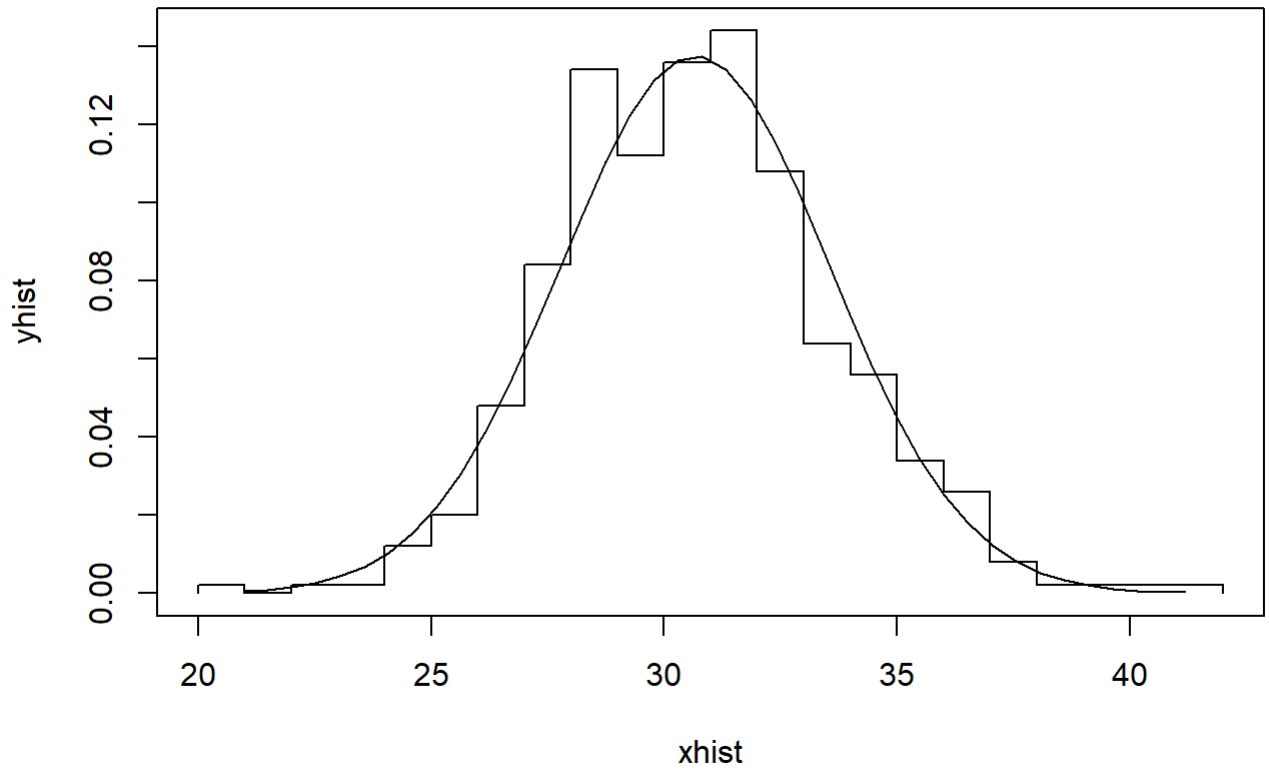

Histogram of sample_mean_med$x_bar
Frequency
0
2
4
6
8
26
28
30
32
34
36
sample_mean_med$x_bar


yhist
0.00
0.05
0.10
0.15
26
28
30
32
34
36
xhist

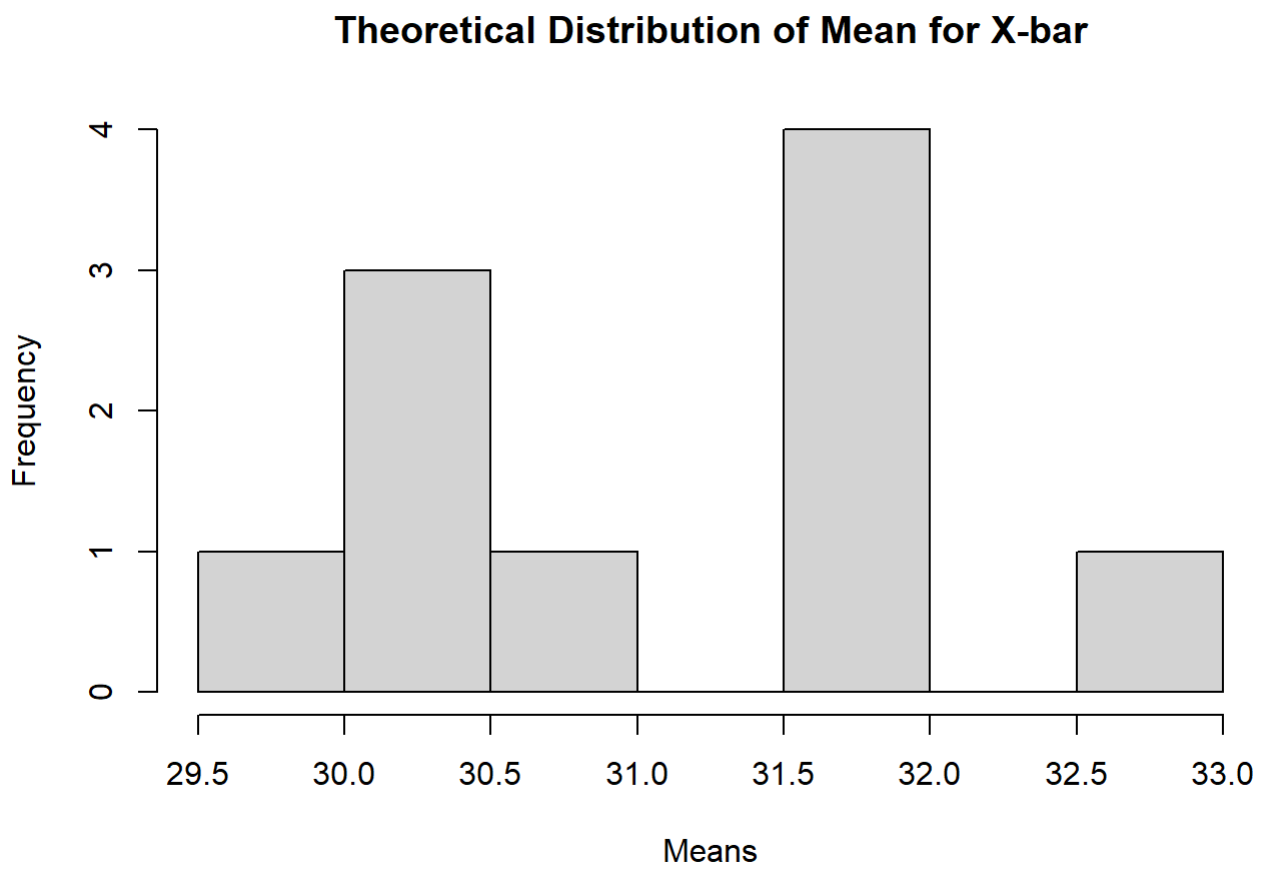

Theoretical Distribution of Mean for X-bar
Frequency
0
20
40
60
80
100
140
20
25
30
35
40
Means

```
## [1] 30.66419
```

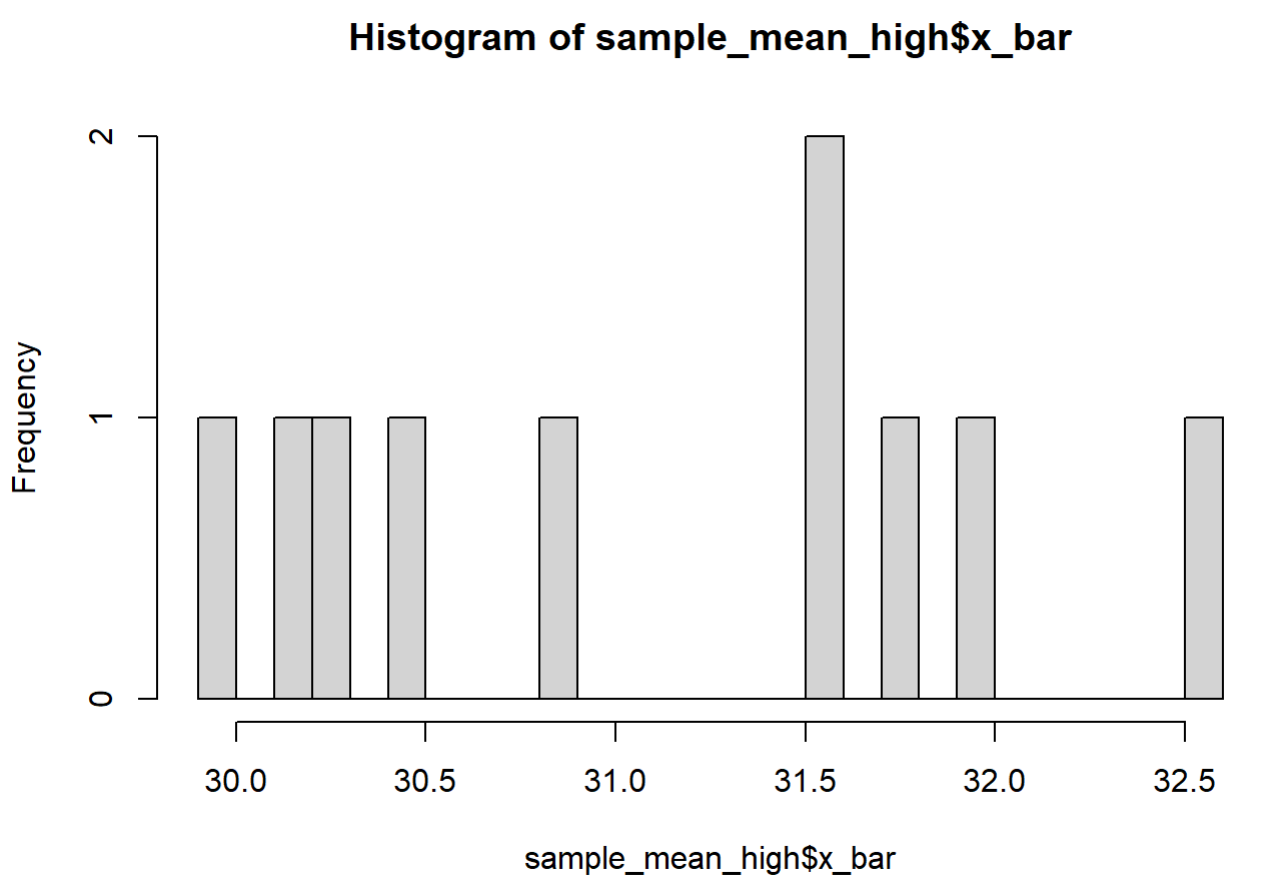

Histogram of sample_mean_med$x_bar
Frequency
0
10
20
30
40
50
60
70
20
25
30
35
40
sample_mean_med$x_bar

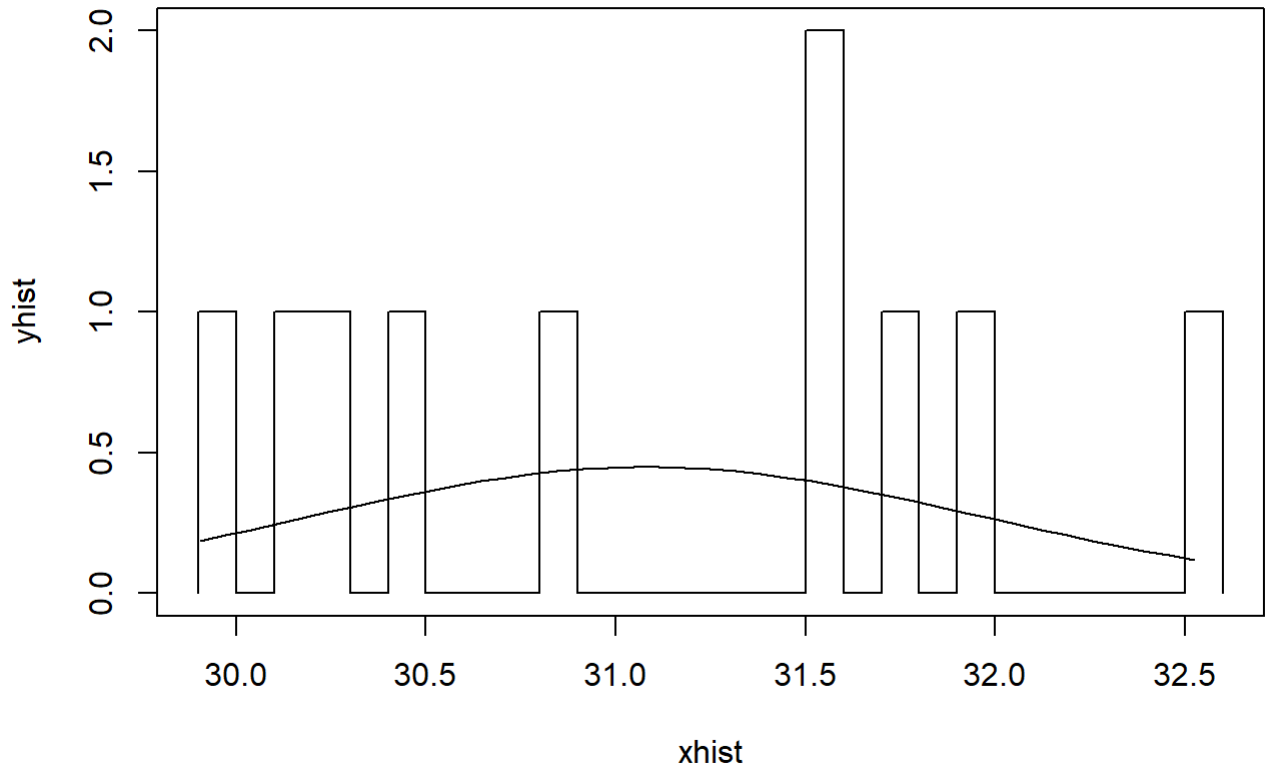

yhist
0.00
0.04
0.08
0.12
20
25
30
35
40
xhist

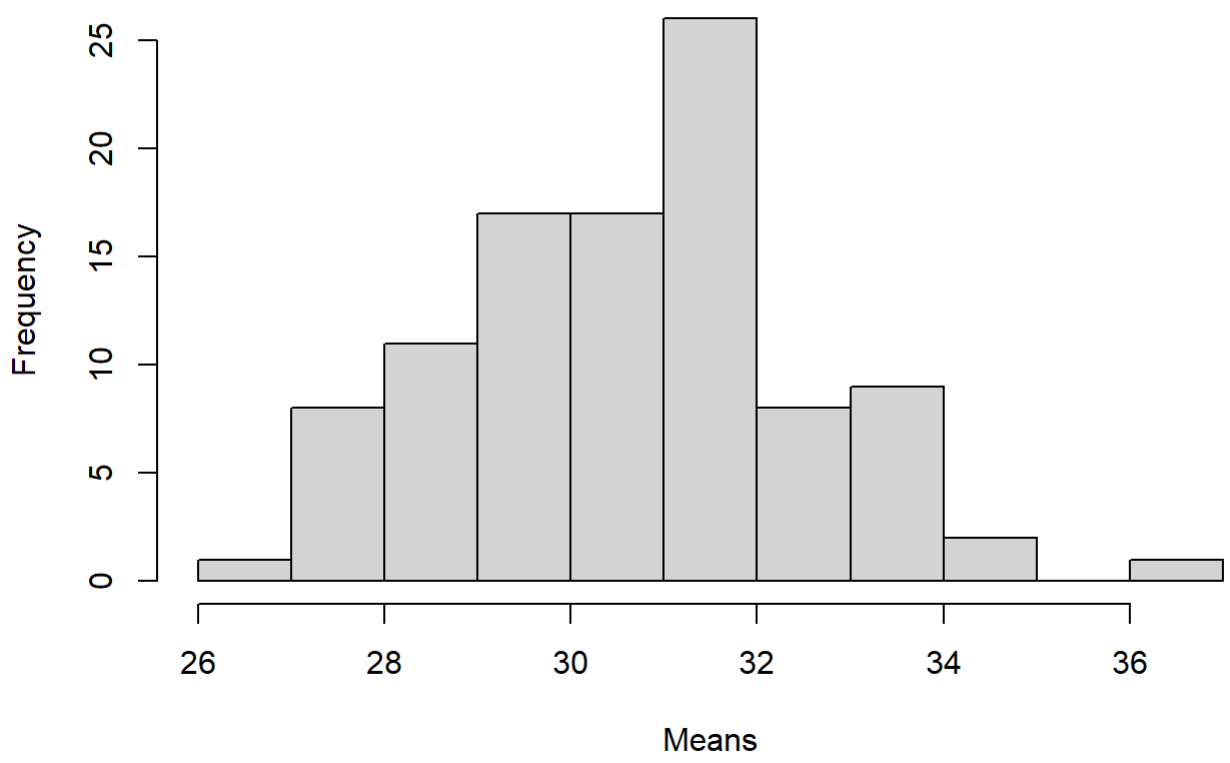
```
for (j in c(10,100,500)){
  sample_mean_high <- vector(mode = "numeric", length = 0)
  for (i in 1:j){
    sample_mean <- insurance %>%
                sample_n(size = 10) %>%
                summarise(x_bar = mean(bmi))
    sample_mean_high <- rbind(sample_mean_high, sample_mean)
  }
  hist(sample_mean_high$x_bar, main= 'Theoretical Distribution of Mean for X-bar',xlab= 'Means')
  print(mean(sample_mean_high$x_bar))

  h <- hist(sample_mean_high$x_bar, breaks = 20)
xhist <- c(min(h$breaks),h$breaks)
yhist <- c(0, h$density,0)
xfit<- seq(min(sample_mean_high$x_bar),max(sample_mean_high$x_bar),length=40)
yfit<- dnorm(xfit, mean = mean(sample_mean_high$x_bar), sd=sd(sample_mean_high$x_bar))
plot(xhist,yhist,type='s')
lines(xfit,yfit)
}
```



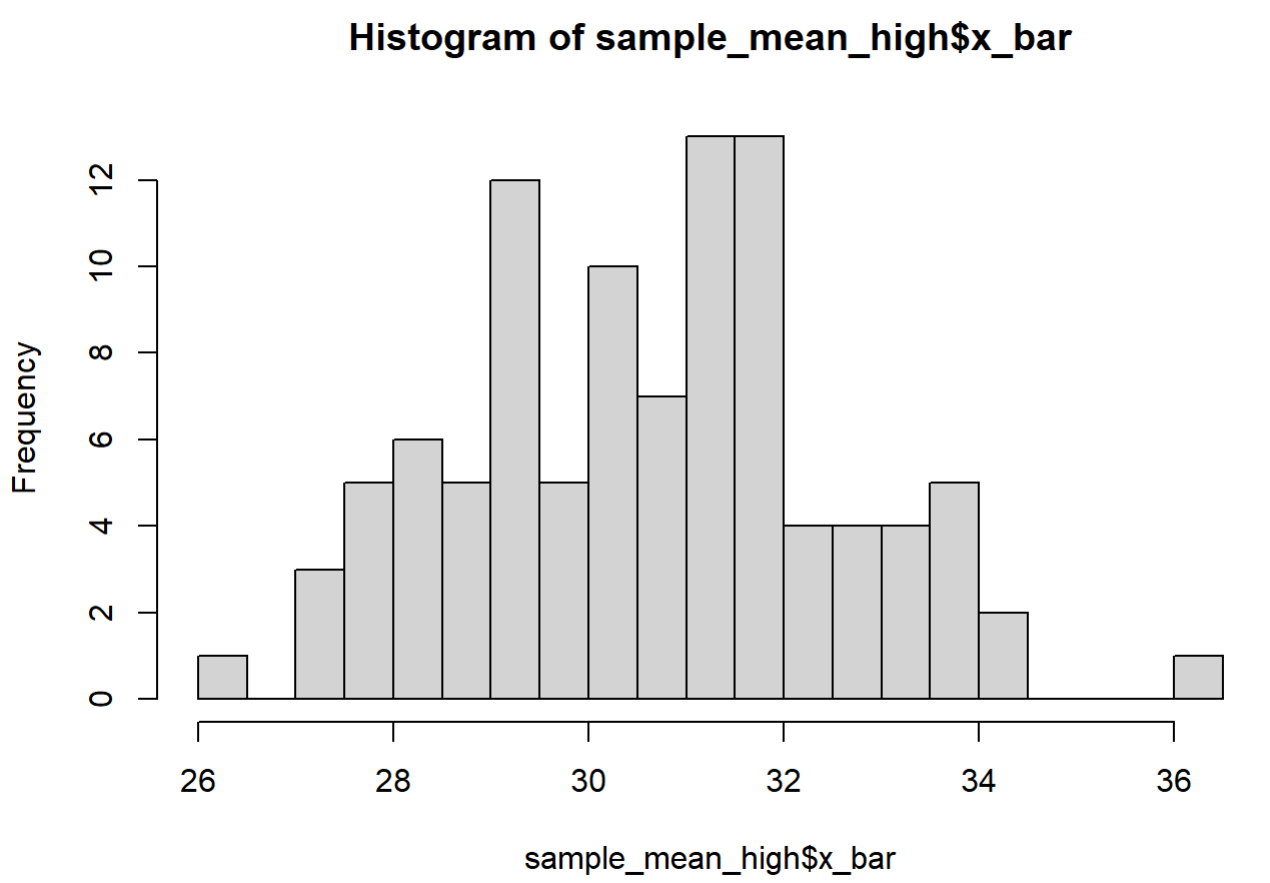

```
## [1] 31.07965
```

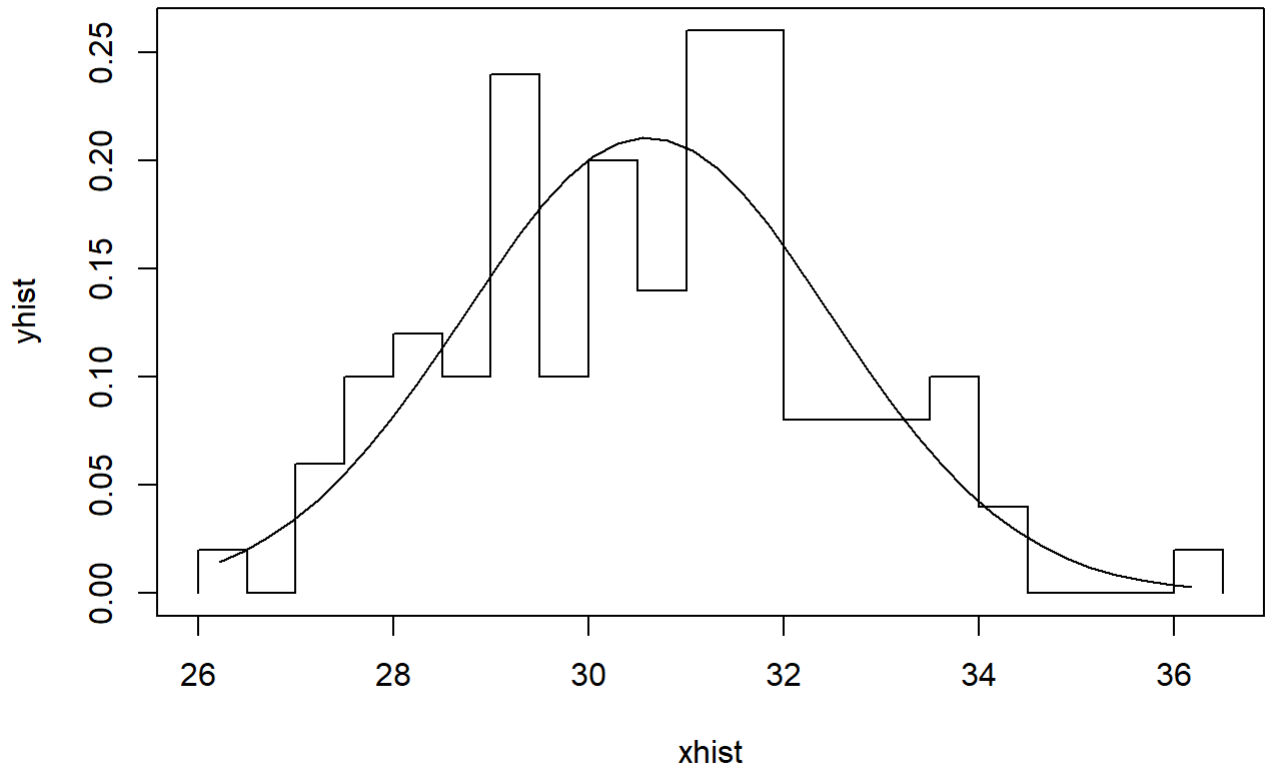

Histogram of sample_mean_high$x_bar
Frequency
0
1
2
30.0
30.5
31.0
31.5
32.0
32.5
sample_mean_high$x_bar

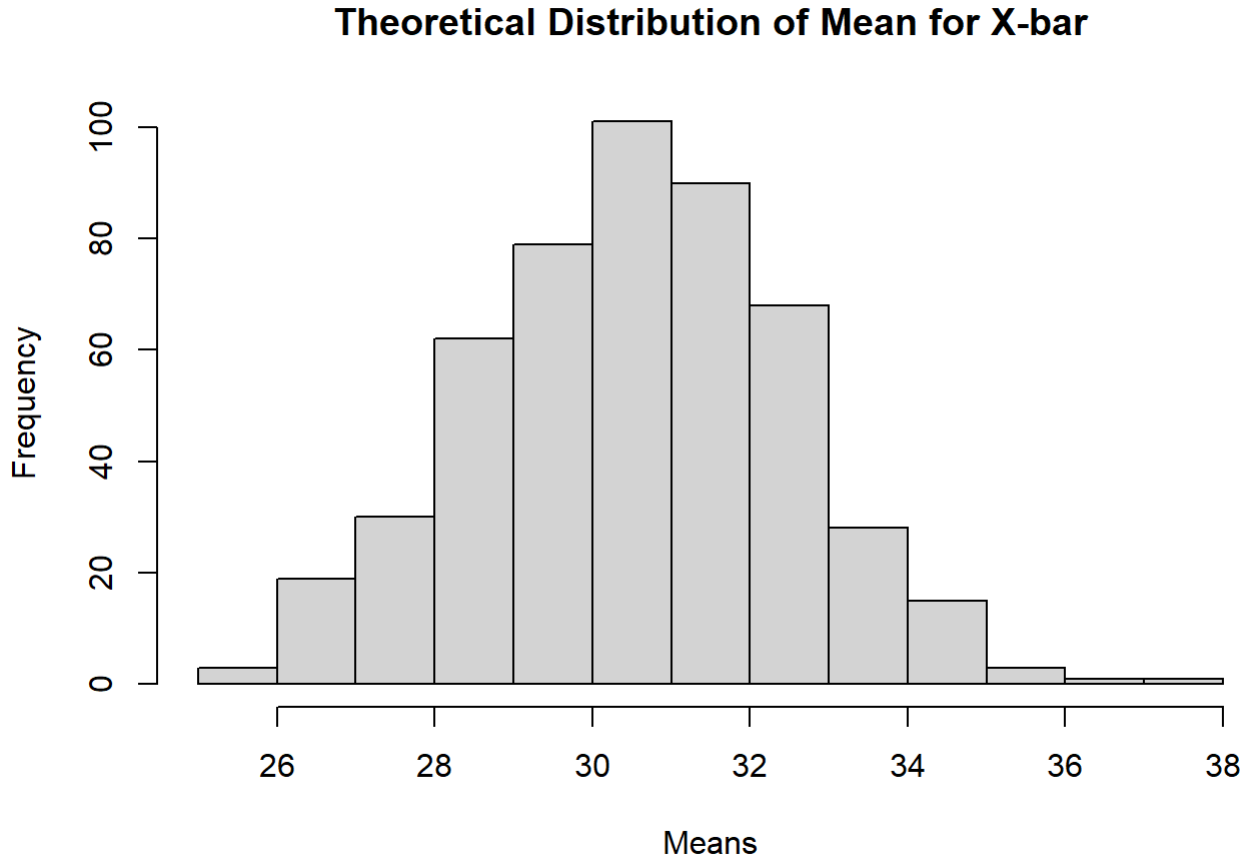

yhist
0.0
0.5
1.0
1.5
2.0
30.0
30.5
31.0
31.5
32.0
32.5
xhist


Theoretical Distribution of Mean for X-bar
Frequency
0
5
10
15
20
25
26
28
30
32
34
36
Means

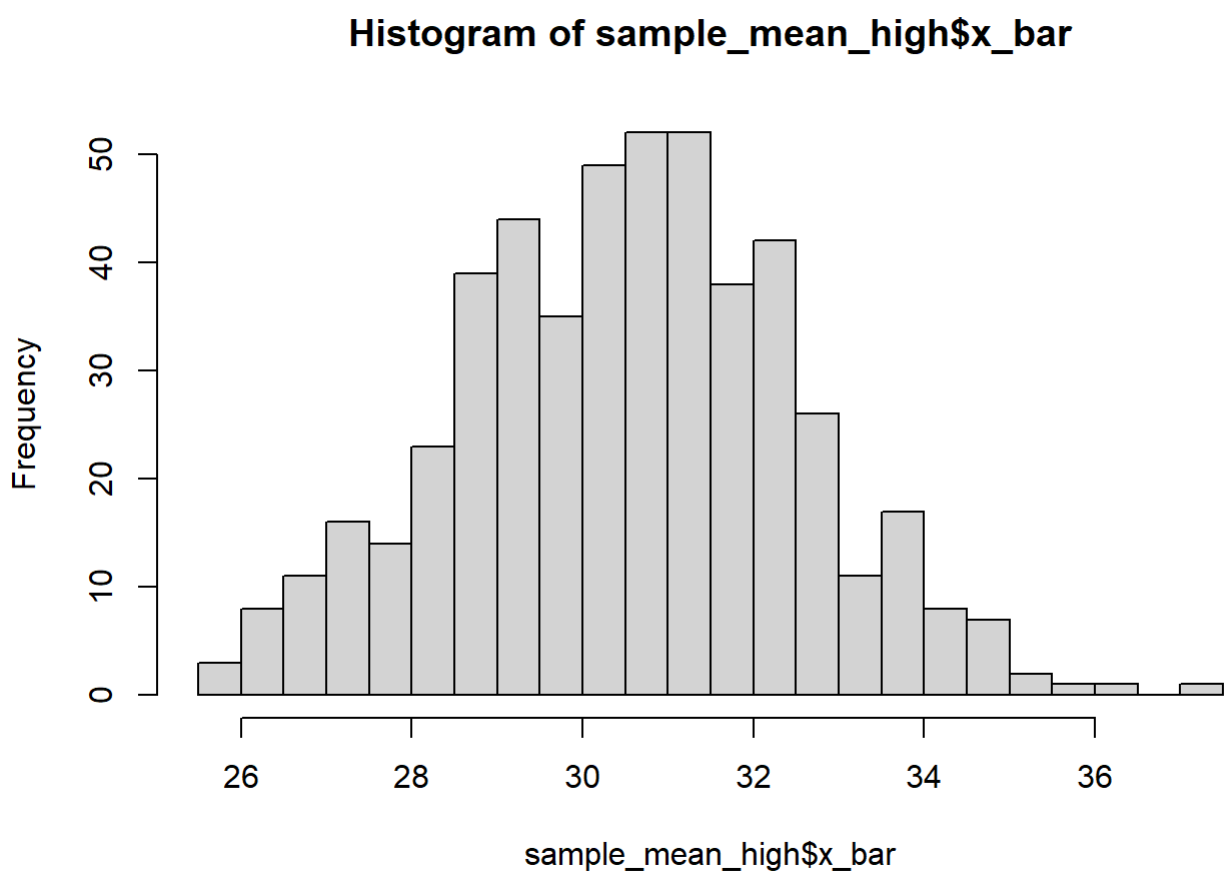
```
## [1] 30.60169
```

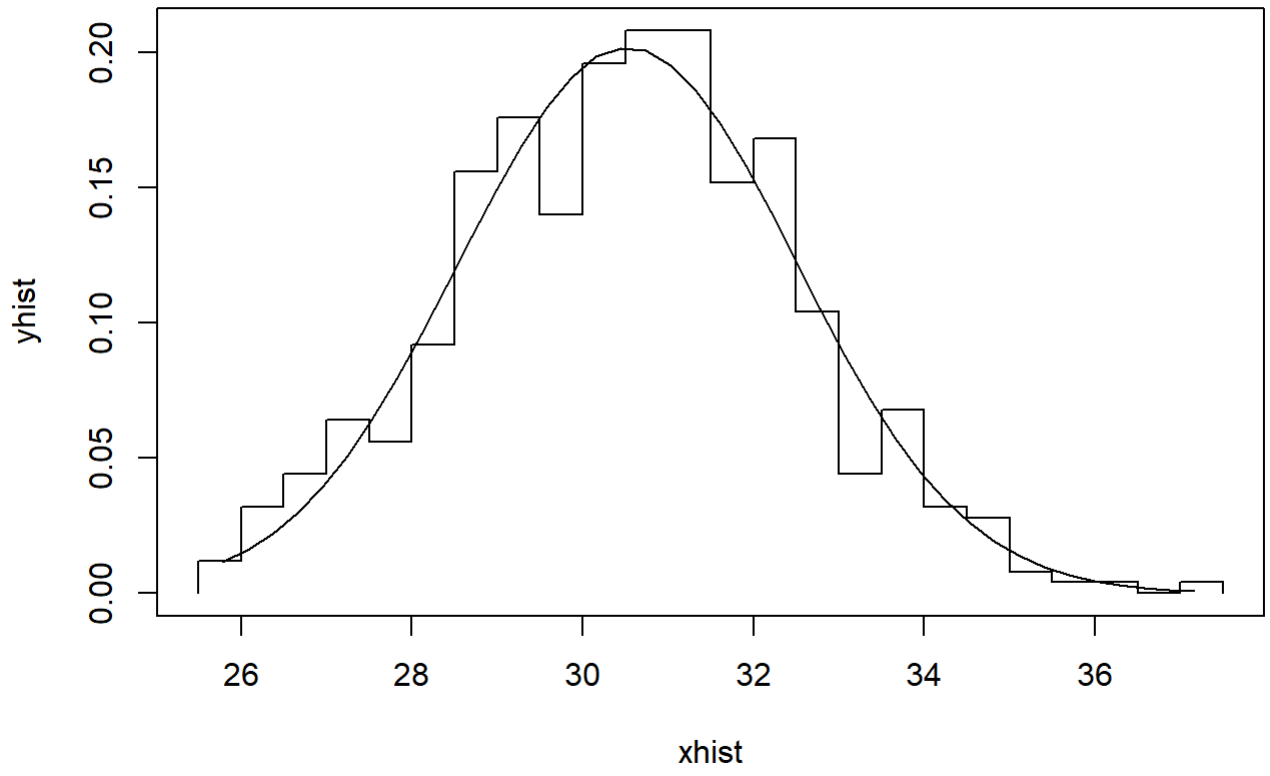

Histogram of sample_mean_high$x_bar
Frequency
0
2
4
6
8
10
12
26
28
30
32
34
36
sample_mean_high$x_bar


yhist
0.00
0.05
0.10
0.15
0.20
0.25
26
28
30
32
34
36
xhist


Theoretical Distribution of Mean for X-bar
Frequency
0
20
40
60
80
100
26
28
30
32
34
36
38
Means

```
## [1] 30.52537
```


Histogram of sample_mean_high$x_bar
Frequency
0
10
20
30
40
50
26
28
30
32
34
36
sample_mean_high$x_bar


yhist
0.00
0.05
0.10
0.15
0.20
26
28
30
32
34
36
xhist

From all observation it appear that the data of averages fit the distribution of X * well.

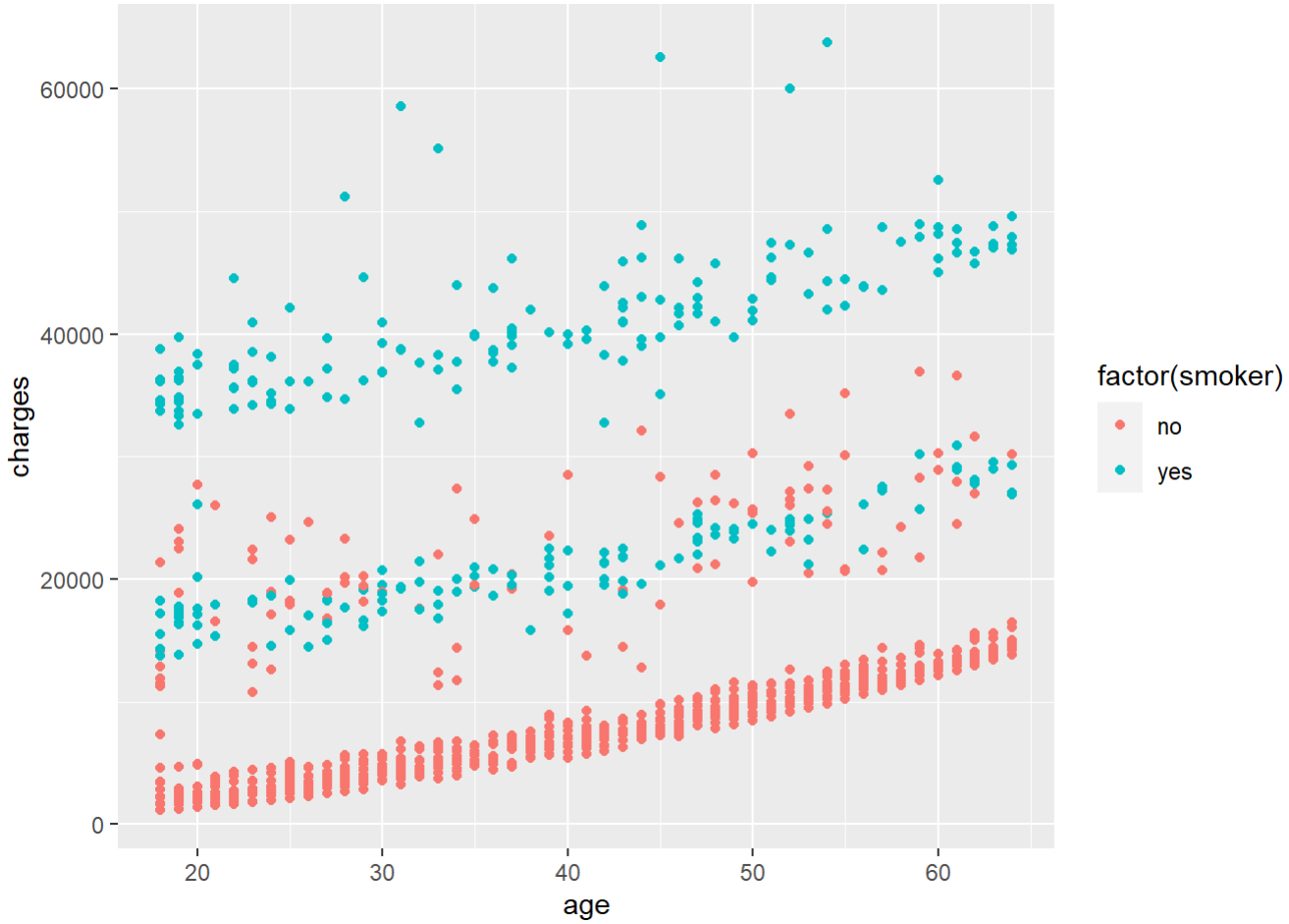
The shape and distribution became closer to the normal with greater sample size. And when the number of average means increases the the shape of the histogram fits a normal distribution as it should according to the Central Limit Theorem.

6. Explore a relationship between variables.

Here we are choosing age & charges to explore relation between them.

a. Drawing a scatter diagram relating these variables.

```
#creating a scatter plot

ggplot(insurance, aes(x=age, y=charges)) + geom_point(aes(color = factor (smoker)))
```



This scatter plot shows the relationship between a patients age and their medical charges, with an additional variable for whether or not the patient is a smoker. Looking at this graph, we can notice two distinct patterns, the first being that the medical costs for patients increase gradually as the age of the patient increases, and the second being that individuals who smoke have on average, far higher medical costs than non-smokers.

b. Determining the sample correlation coefficient & Estimating CI for the correlation coefficient.

For this we will use built-in function inspect_cor() from “inspectdf” package. from this single function we get correlation coefficient and confidant interval for the correlation coefficient.

```
# Determining the sample correlation coefficient
# Estimating CI for correlaton coefficient.

library("inspectdf")
inspect_cor(insurance)
```

```
## # A tibble: 6 × 7
##   col_1    col_2      corr  p_value   lower  upper pcnt_nna
##   <chr>    <chr>     <dbl>    <dbl>   <dbl>  <dbl>    <dbl>
## 1 charges  age      0.299  8.75e-28  0.249  0.347       100
## 2 charges  bmi      0.198  4.26e-13  0.146  0.249       100
## 3 bmi      age      0.109  6.54e- 5  0.0560 0.162       100
## 4 charges  children 0.0680 1.30e- 2  0.0145 0.121       100
## 5 children age      0.0425 1.21e- 1 -0.0111 0.0958      100
## 6 children bmi      0.0128 6.41e- 1 -0.0409 0.0663      100
```

d. Performing the test for independence. State the null and alternative hypothesis.

```
t.test(insurance$charges, insurance$age, alternative = "greater", var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  insurance$charges and insurance$age
## t = 39.965, df = 1337, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  12686.28      Inf
## sample estimates:
##   mean of x   mean of y
## 13270.42227    39.20703
```

Null hypothesis means the variables are independent, there in no relationship between them. And in other hand Alternative hypothesis means the variables are dependent, there is a relationship between them.

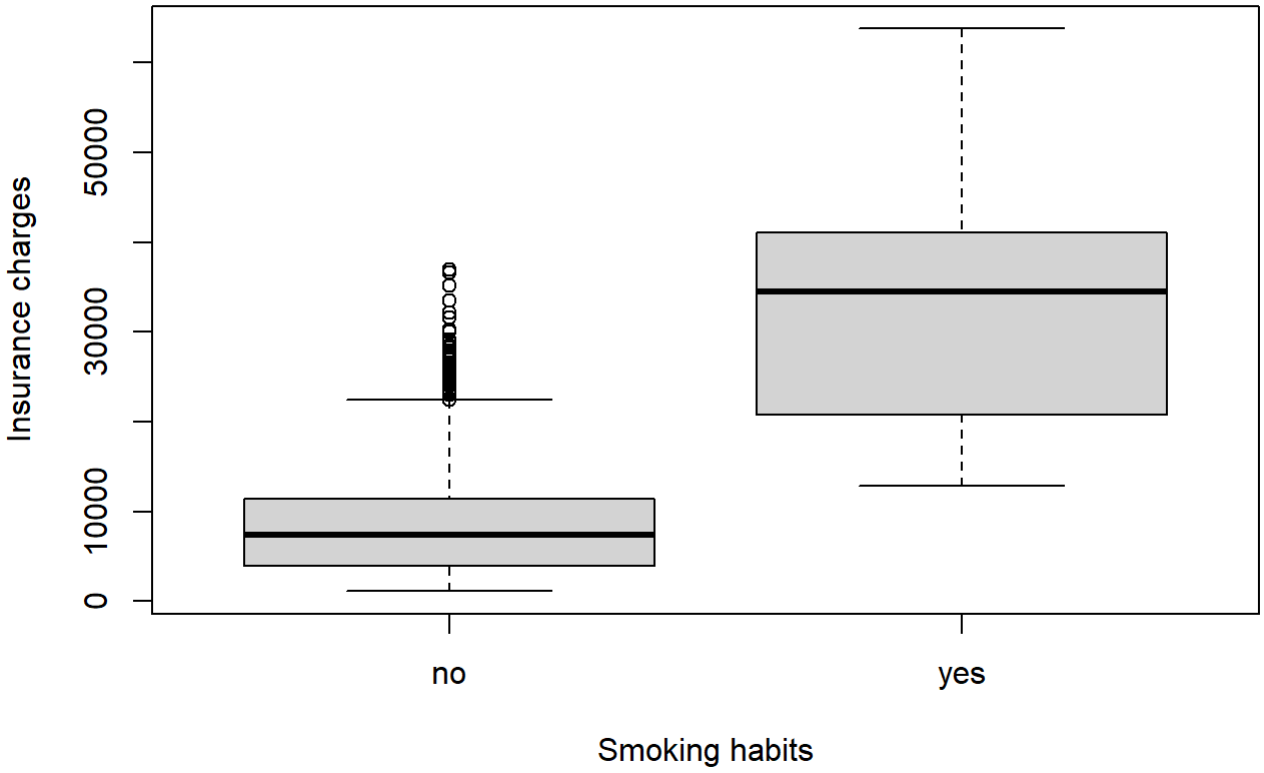
From the result we see that the p-value is less than the significance level of 5%. Like any other statistical test, if the p-value is less than the significance level, we can reject the null hypothesis.

So in our result, rejecting the null hypothesis means that there is a significant relationship between the age and the charges.

7. Hypothesis testing for categorical variables.

We are going to choose a categorical variable and a numerical variable. And going to do a hypothesis test

```
# boxplot for smoker and charges

boxplot(charges ~ smoker, data = insurance, xlab = "Smoking habits", ylab = "Insurance charges")
```

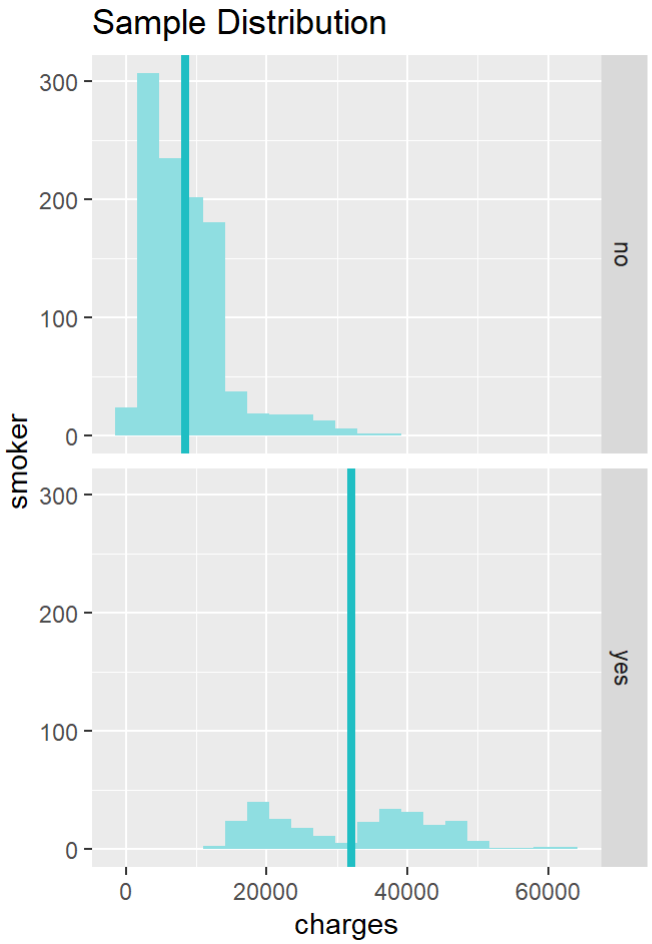


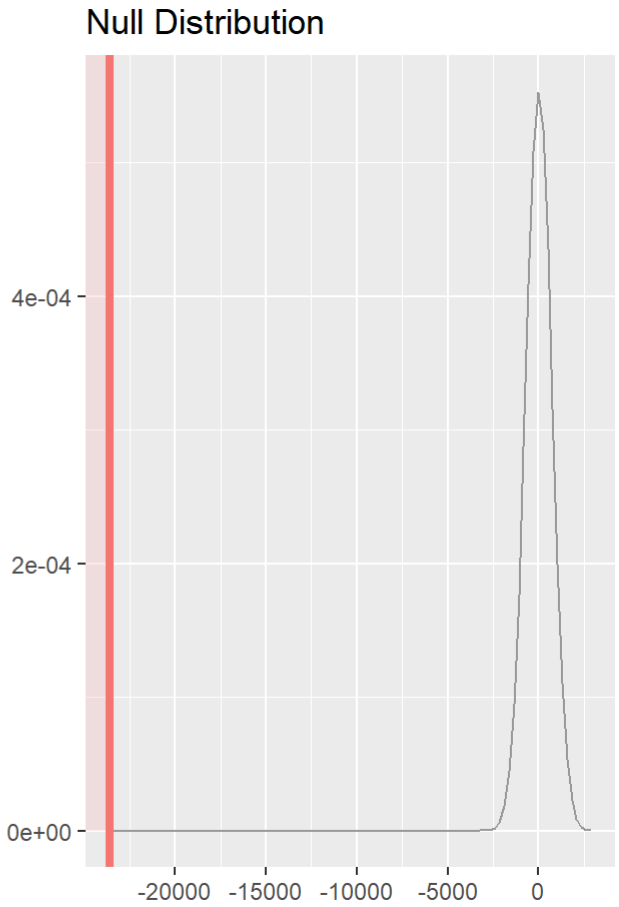
We will conduct the following hypothesis test:

$$H_0 : \mu_{non_smoker} = \mu_{smoker}$$
$$H_A : \mu_{non_smoker} < \mu_{smoker}$$

```
library("statsr")
inference(y = charges, x = smoker, data = insurance, statistic = "mean", type = "ht", null = 0,
          alternative = "less", method = "theoretical")
```

```
## Response variable: numerical
## Explanatory variable: categorical (2 levels)
## n_no = 1064, y_bar_no = 8434.2683, s_no = 5993.7818
## n_yes = 274, y_bar_yes = 32050.2318, s_yes = 11541.5472
## H0: mu_no =  mu_yes
## HA: mu_no < mu_yes
## t = -32.7519, df = 273
## p_value = < 0.0001
```





From the result we see that the p-value is less than the significance level of 5%. Like any other statistical test, if the p-value is less than the significance level, we can reject the null hypothesis.

So in our result, rejecting the null hypothesis means that there is a significant relationship between the smokers and the charges.