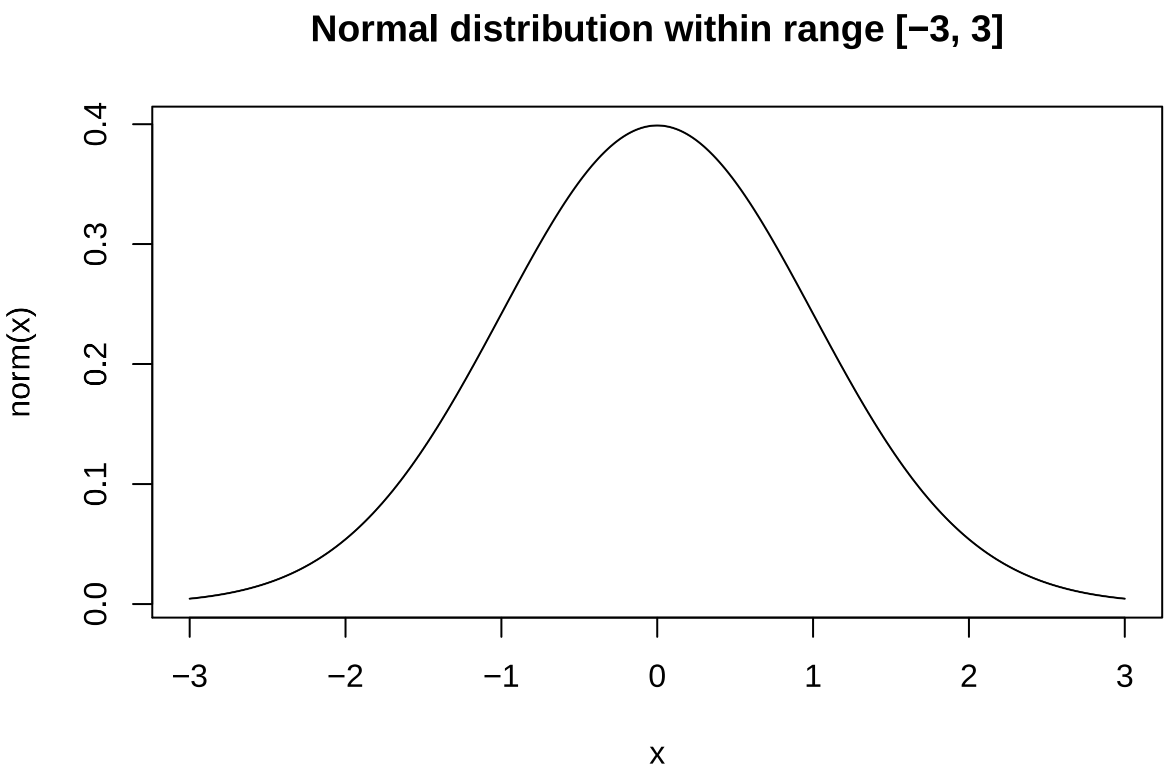
R Notebook

Statistical Inference Lab1

Aryan Soltani

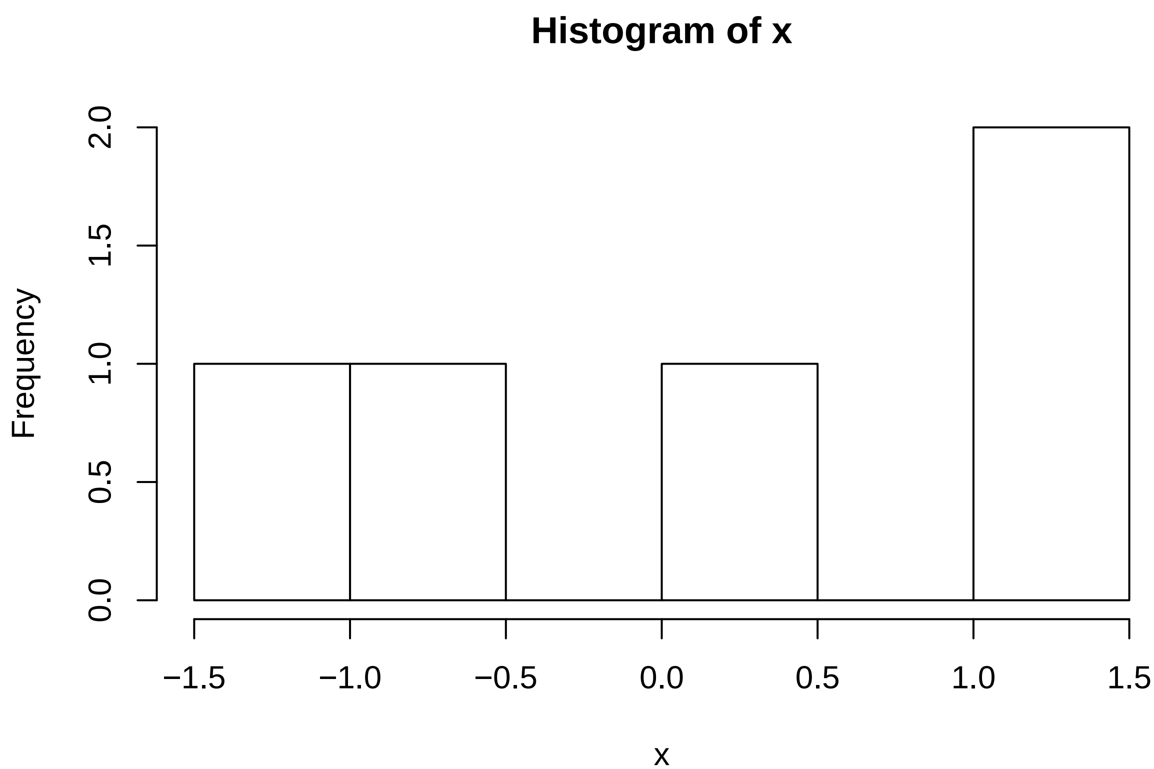
1.

```
norm_x <- seq(-3, 3, length = 1000)
norm_y <- dnorm(norm_x)
plot(norm_x, norm_y, type = "l", xlab = "x", ylab = "norm(x)", main = "Normal distribution within range
```

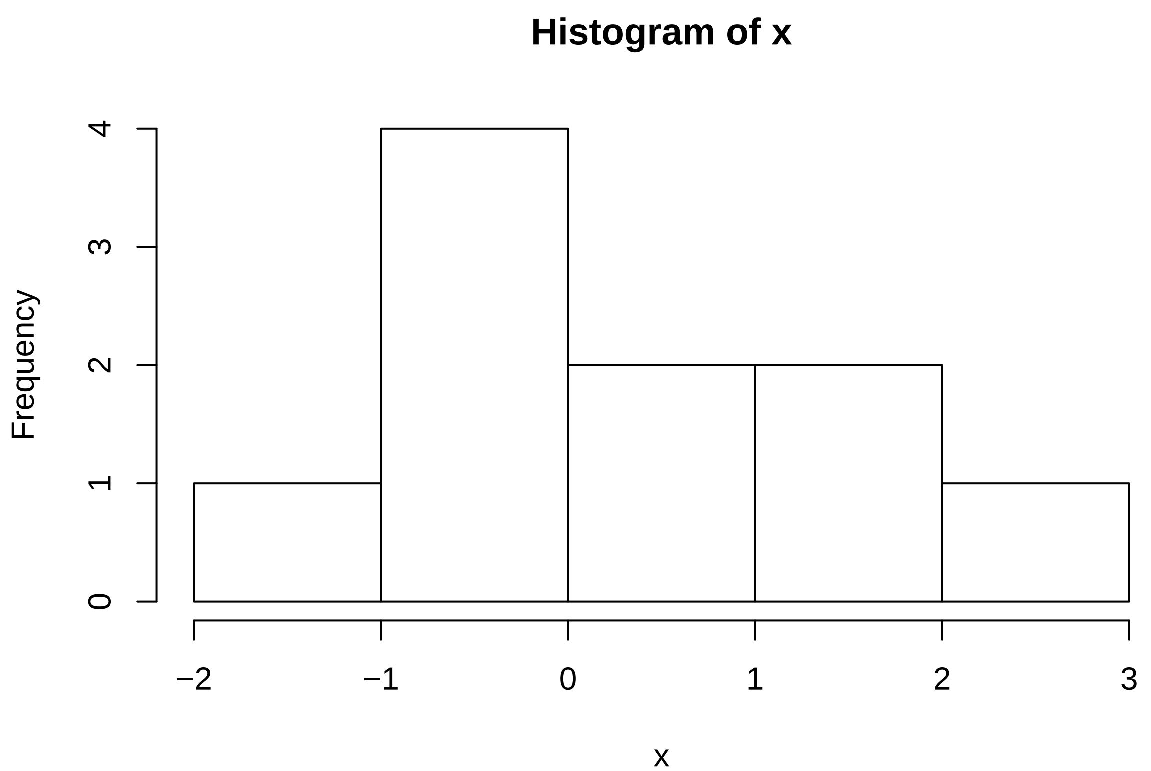


2.

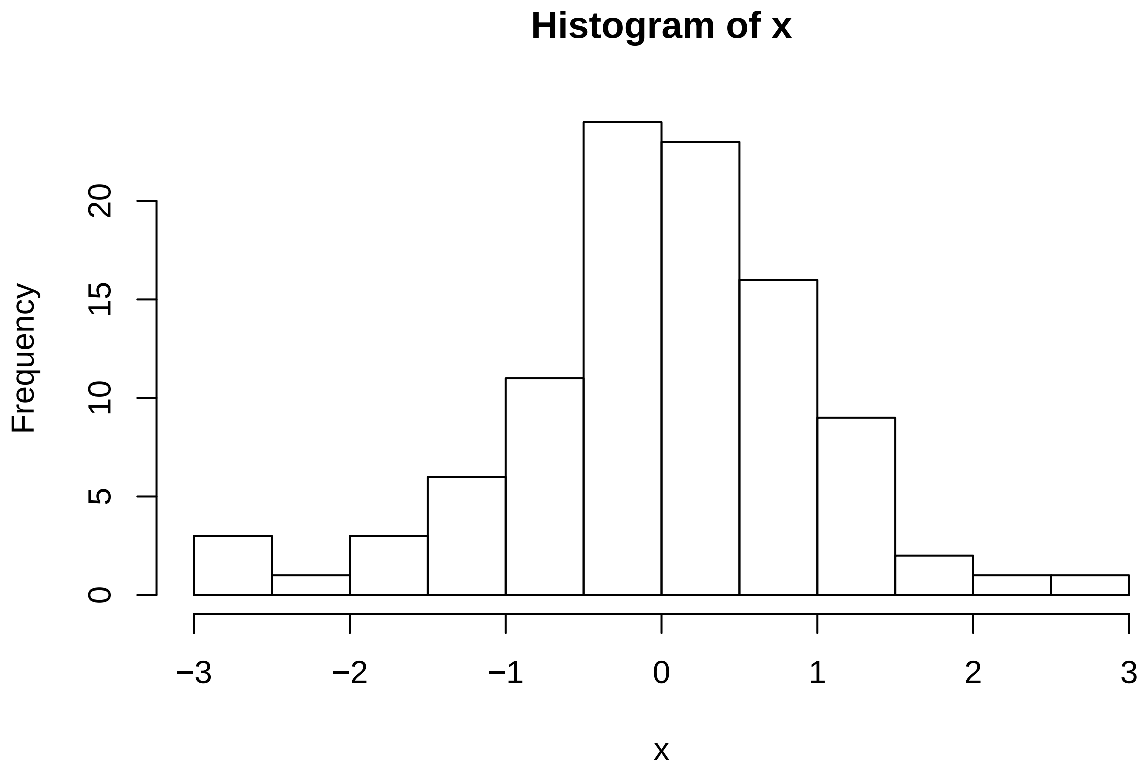
```
x <- rnorm(5)
hist(x)
```



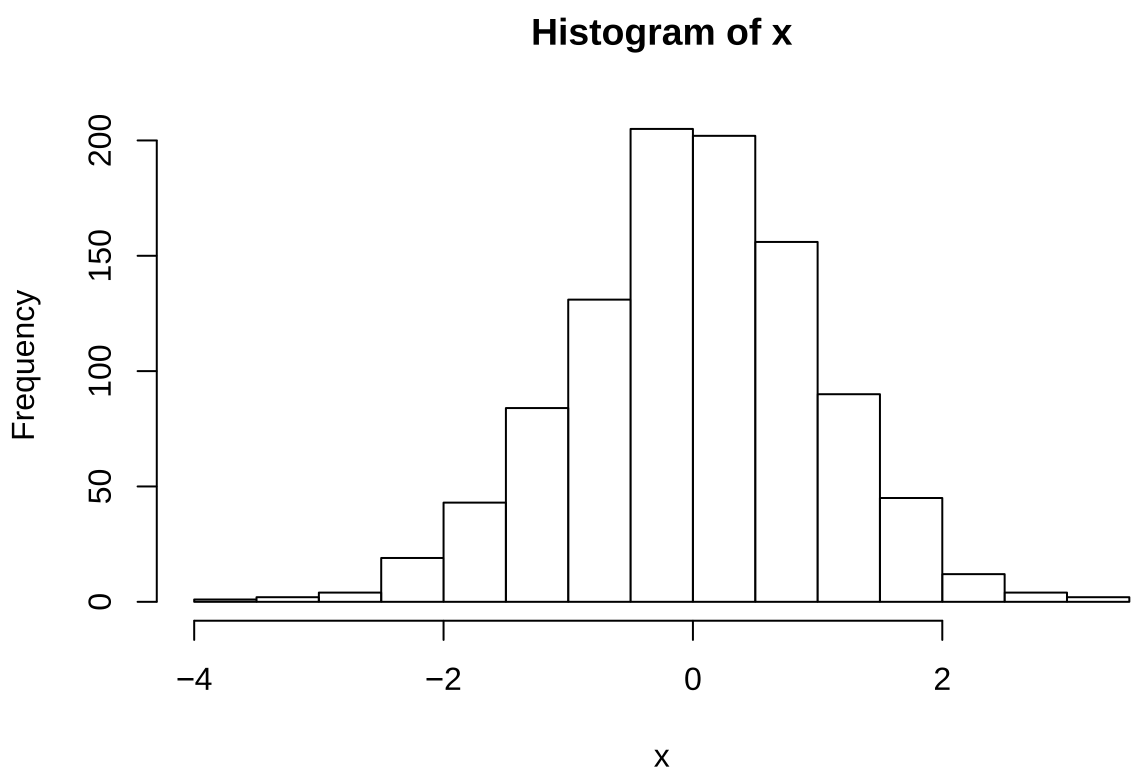

```r
x <- rnorm(10)
hist(x)
```




```r
x <- rnorm(100)
hist(x)
```

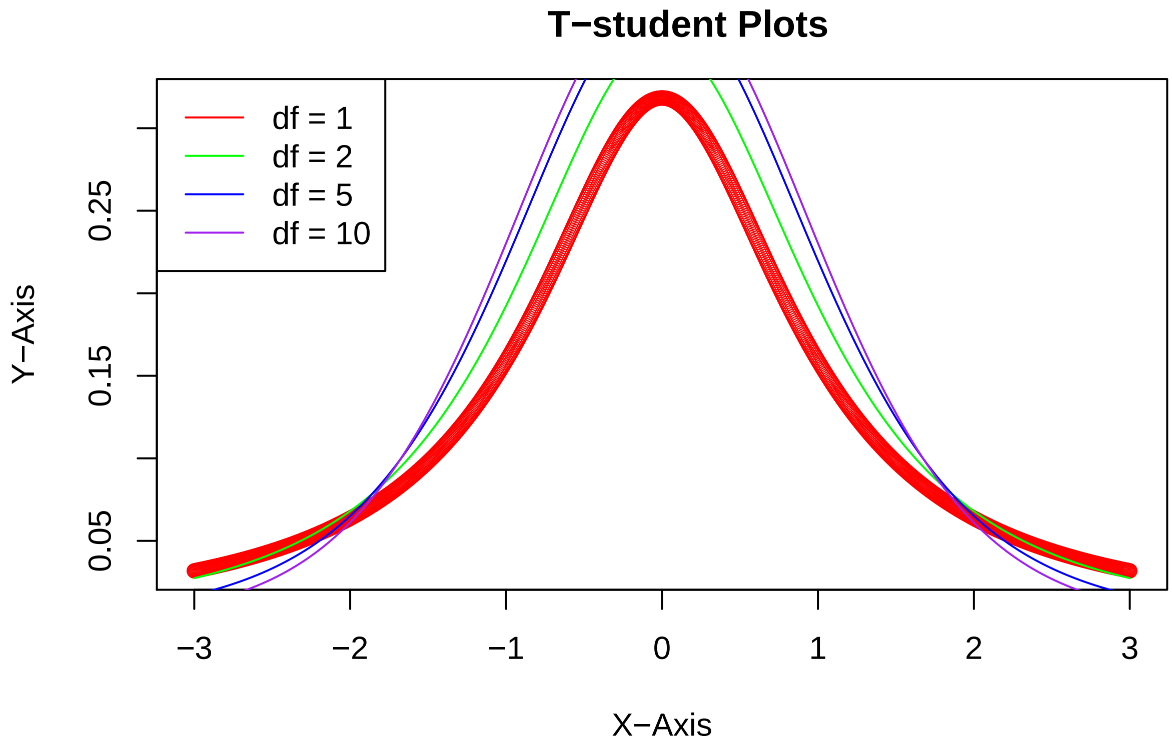



```
x <- rnorm(1000)
hist(x)
```

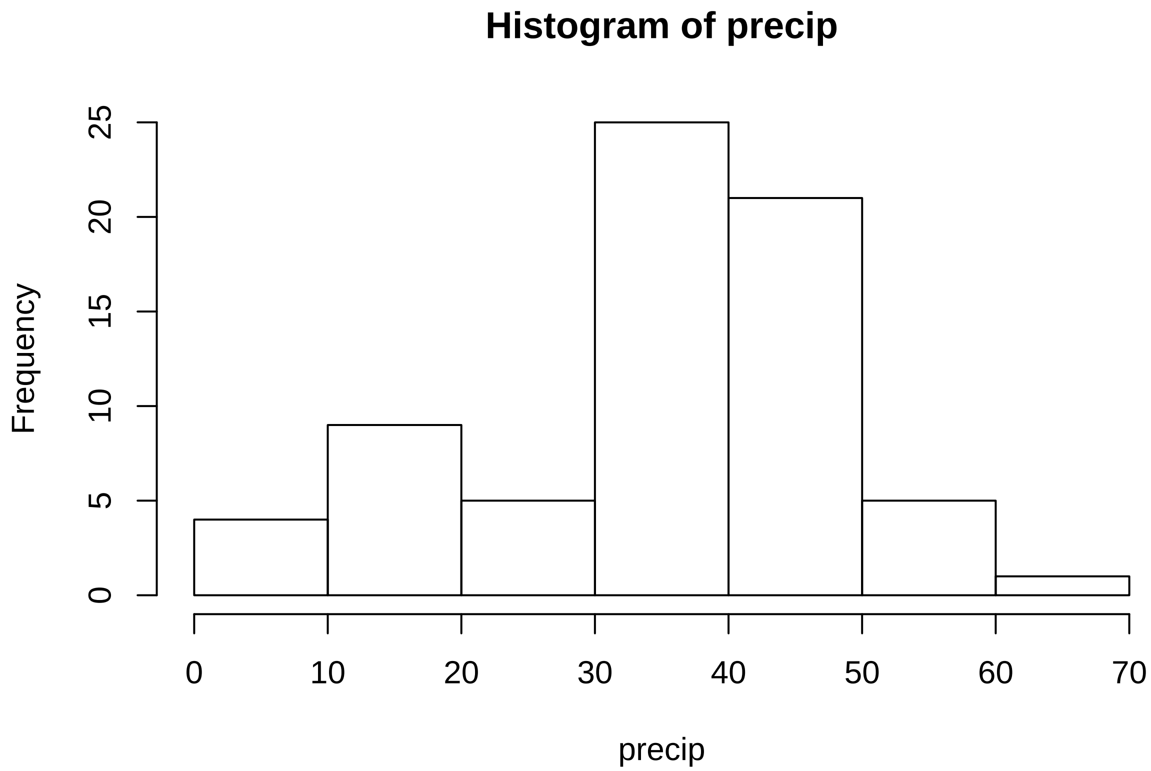


The one with 1000 number of samples has produced the most accurate answer, and the more sample we have the more accurate histogram we would get.

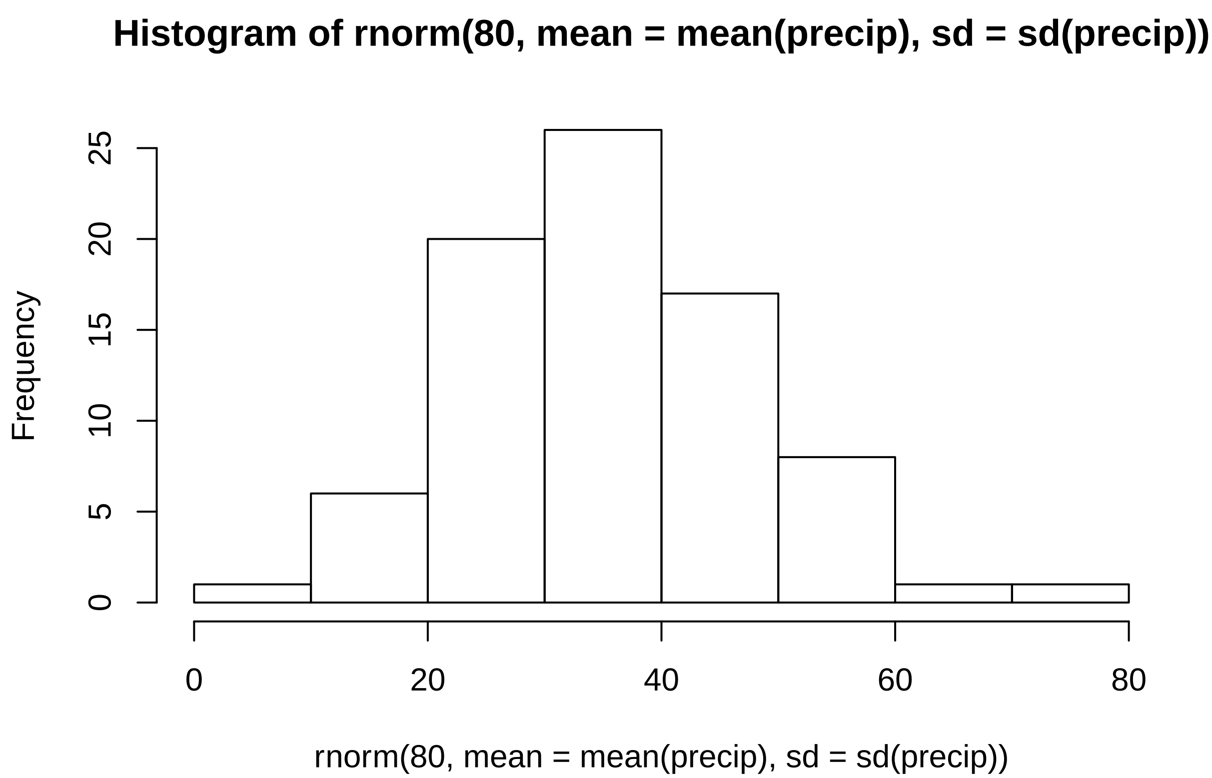
```
t_x <- seq(-3, 3, length = 1000)
colors <- c("red", "green", "blue", "purple")
df_plots <- c(1, 2, 5, 10)
plot(t_x, dt(t_x, df_plots[1]), col = colors[1], main = "
     T-student Plots", ylab = "Y-Axis", xlab = "X-Axis", lwd = 0.5)
for(i in 2:4)
{
  lines(t_x, dt(t_x, df_plots[i]), col = colors[i])
}
legend("topleft", c("df = 1", "df = 2", "df = 5", "df = 10"), col = colors, lty = c(1,1,1,1))
```




```
hist(precip)
```

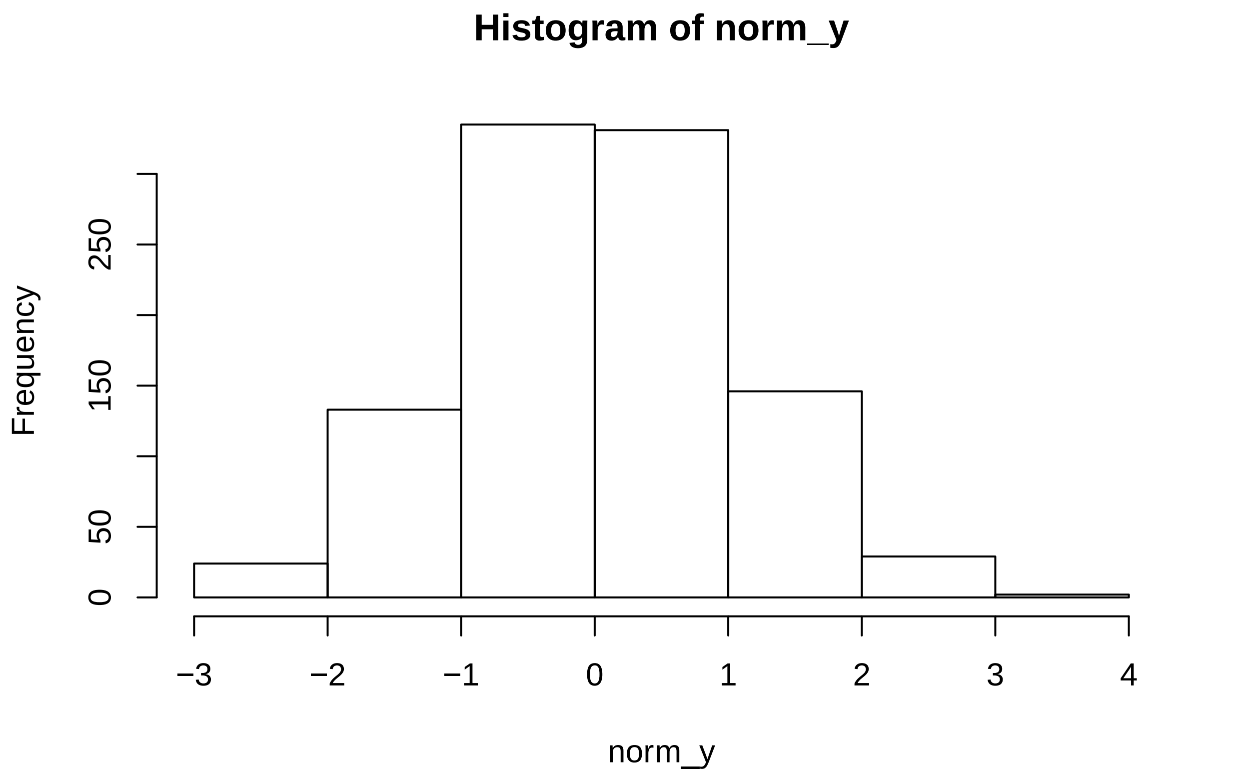



```
hist(rnorm(80, mean = mean(precip), sd = sd(precip)))
```

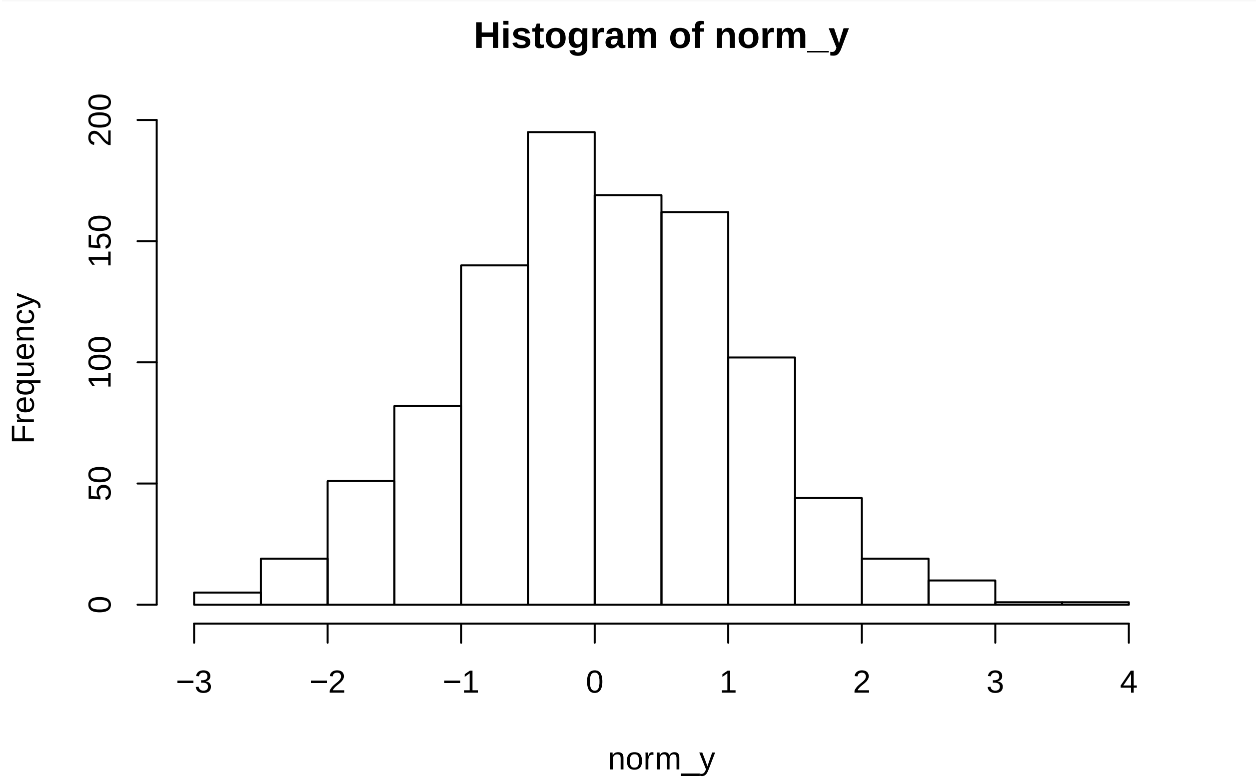


They are similar and the precipitation distribution is very close to normal distribution.

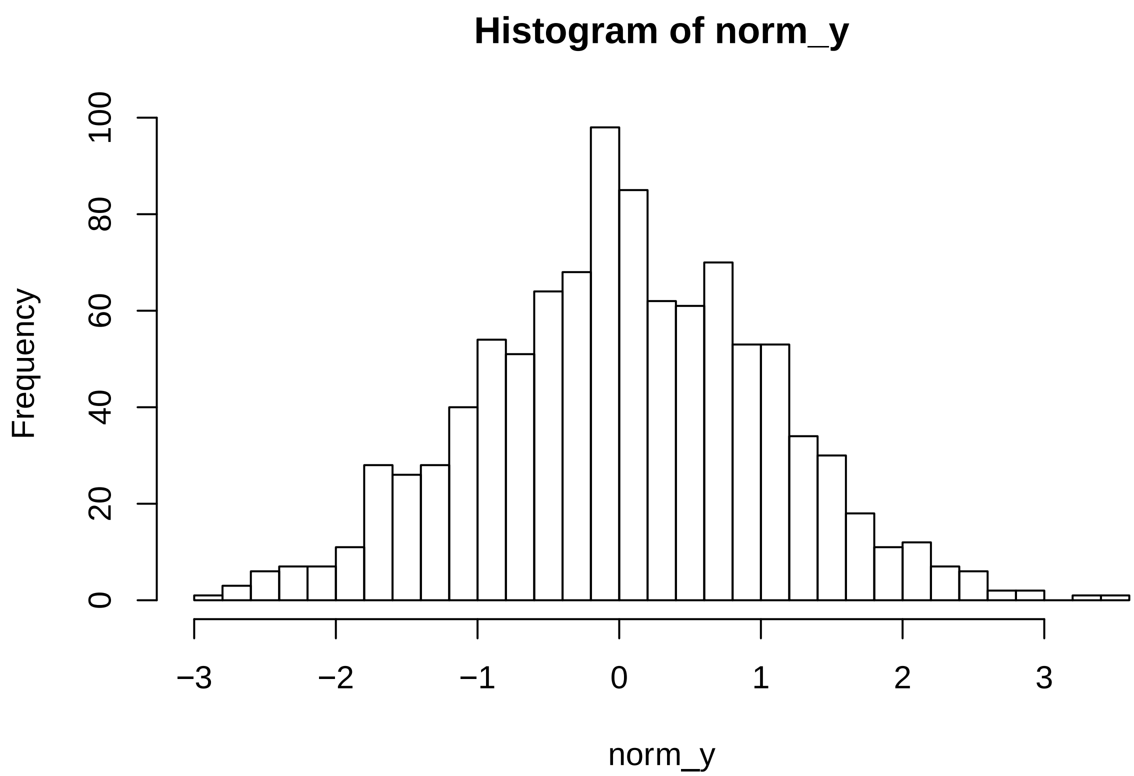
```
norm_y <- rnorm(1000)
hist(norm_y, breaks = 7)
```



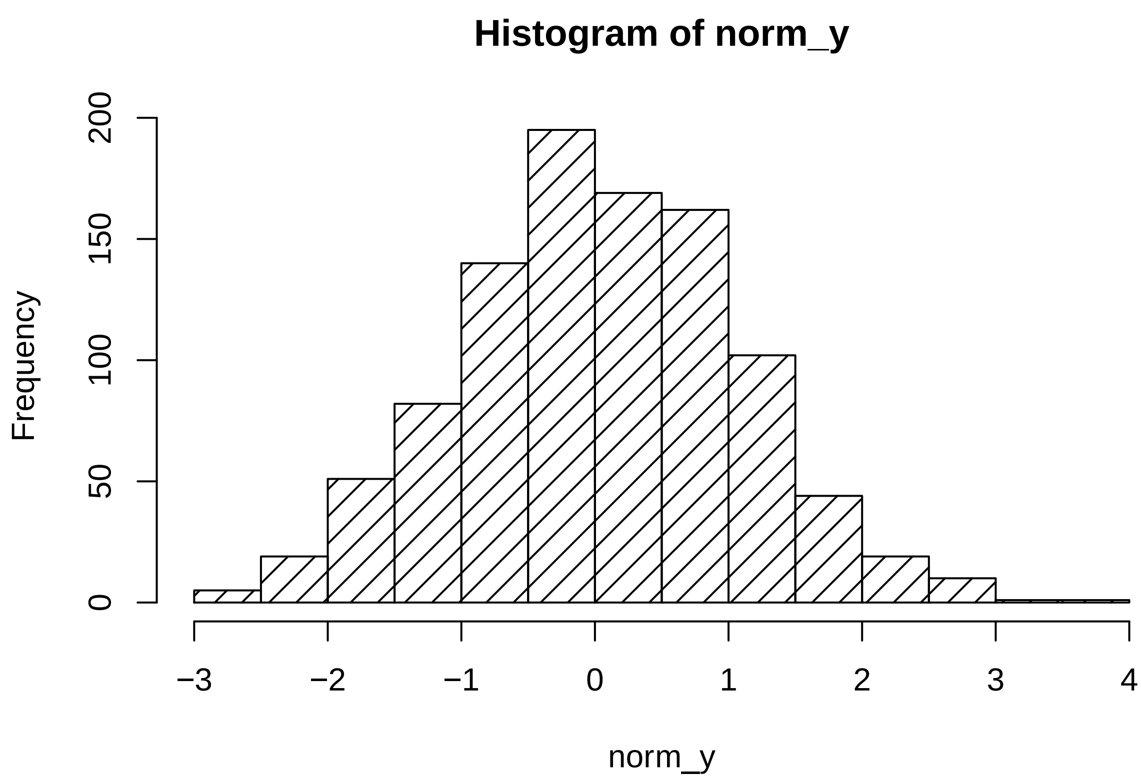

```
hist(norm_y, breaks = 13)
```




```
hist(norm_y, breaks = 30)
```

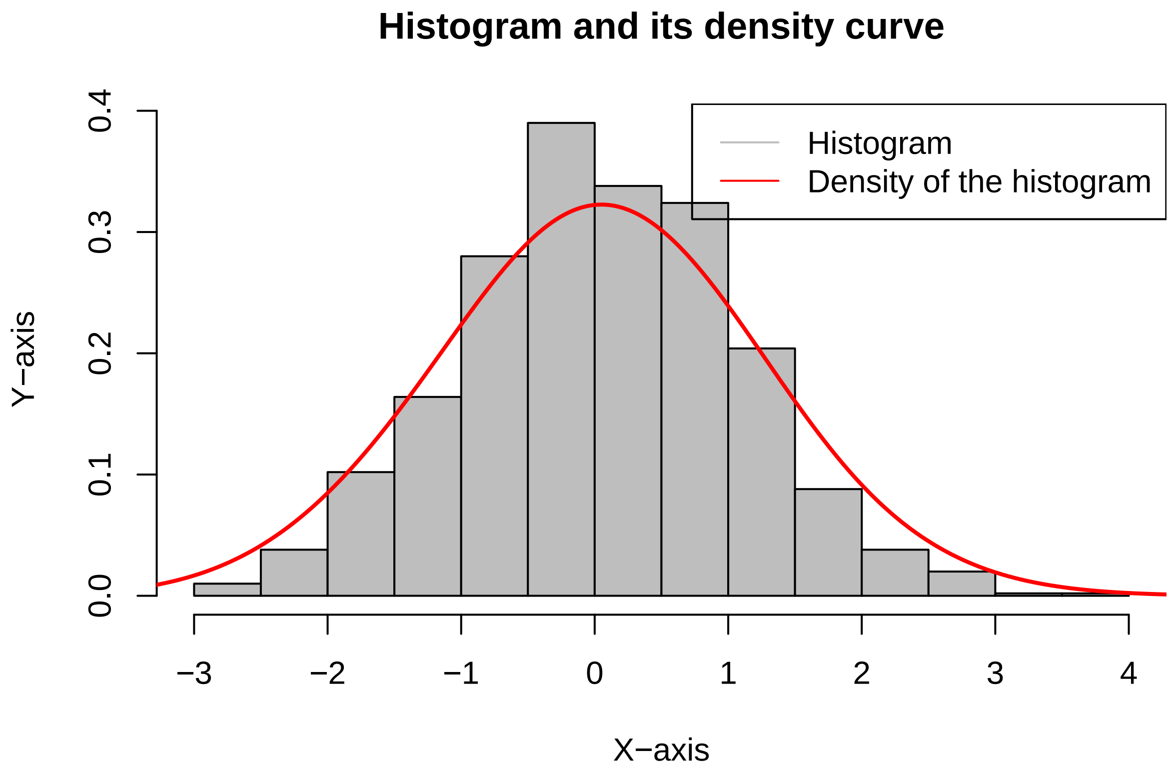



```r
hist(norm_y, breaks = 13, density = 10)
```

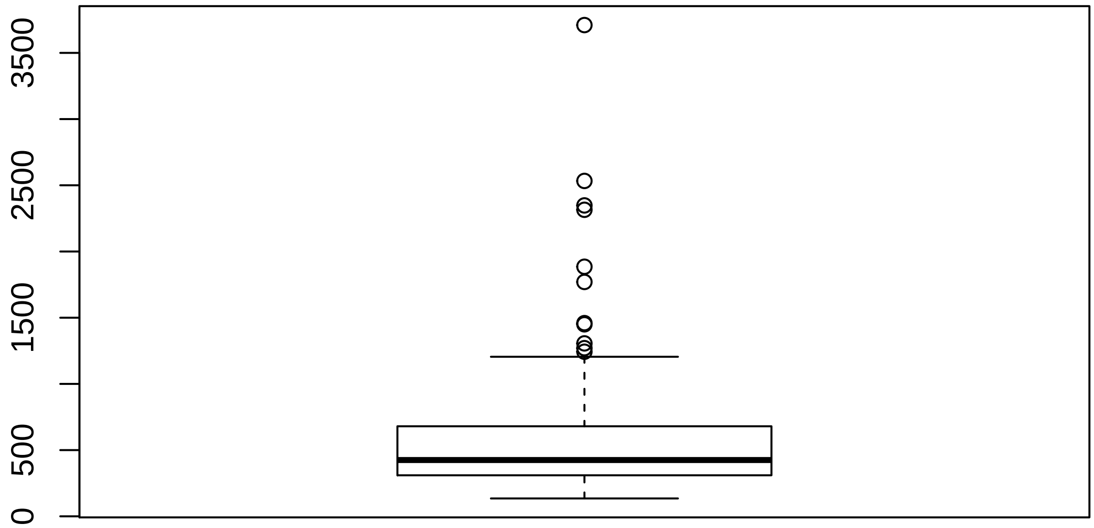


The one with the density visualized the data better than the others and between the histograms with different bins the one with the more bins has visualized better.

```r
hist(norm_y, breaks = 13, prob = TRUE, col = "grey", main = "Histogram and its density curve", xlab = "X
     ylab = "Y-axis")
lines(density(norm_y, adjust = 3), col = "red", lwd = 2)
legend("topright", c("Histogram", "Density of the histogram"), col = c("grey", "red")
       , lty = c(1, 1, 1, 1))
```




```r
whiskers = boxplot(rivers)
```



As we can see in the boxplot the rivers dataset is right-skewed.

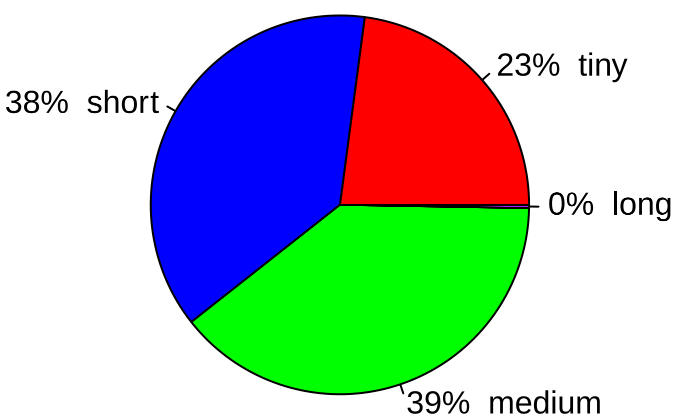
```r
whiskers
```

```
## $stats
##      [,1]
```

```
## [1,]  135
## [2,]  310
## [3,]  425
## [4,]  680
## [5,] 1205
##
## $n
## [1] 141
##
## $conf
##          [,1]
## [1,] 375.7678
## [2,] 474.2322
##
## $out
##  [1] 1459 1450 1243 2348 3710 2315 2533 1306 1270 1885 1770
##
## $group
##  [1] 1 1 1 1 1 1 1 1 1 1 1
##
## $names
## [1] "1"
```

The whiskers values is 135 and 1205 and theses can be extract as the stats output of return value of the boxplot and yes there are some outliers above the upper whisker and their exact values are [1459 1450 1243 2348 3710 2315 2533 1306 1270 1885 1770]

```
group_1 = rivers[rivers < 500]
group_2 = rivers[rivers < 1500]
group_3 = rivers[rivers < 3000]
group_4 = rivers[rivers >= 3000]
vec_count_group = c(length(group_1), length(group_2), length(group_3), length(group_4))
vec_count_group = (vec_count_group * 100) / sum(vec_count_group)
pie(vec_count_group, col = c("red", "blue", "green", "purple"),  label = paste0(
  round(vec_count_group) , "%  " , c("tiny", "short", "medium", "long")))
```

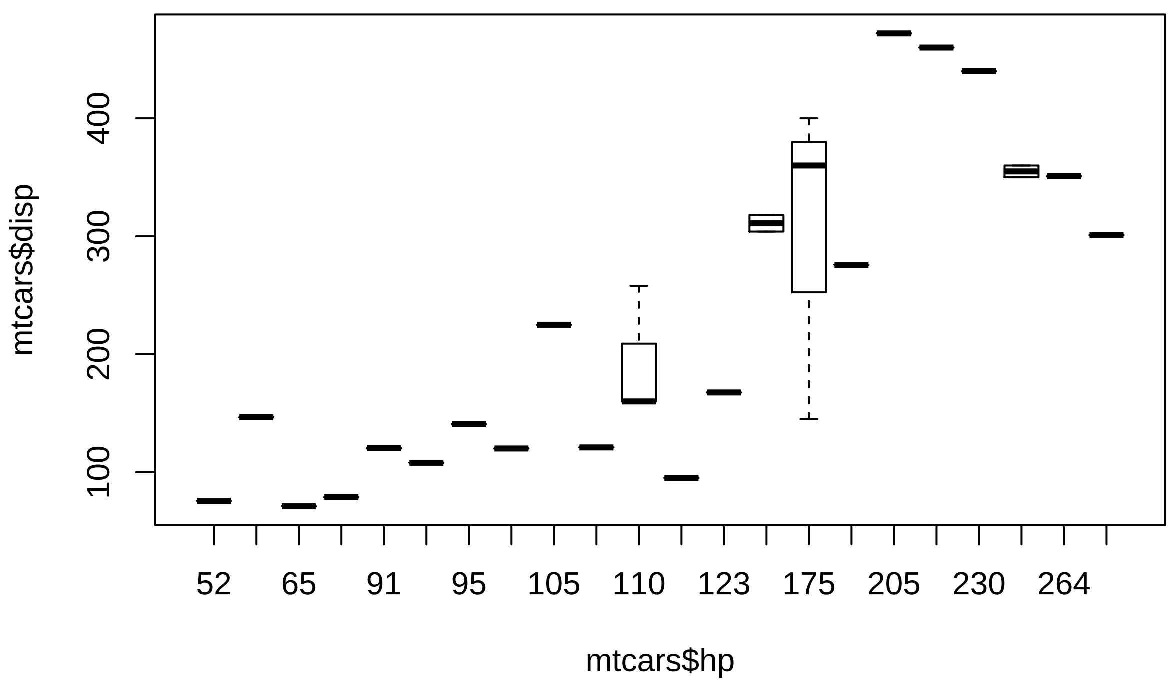



```
data()
mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
```

```
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

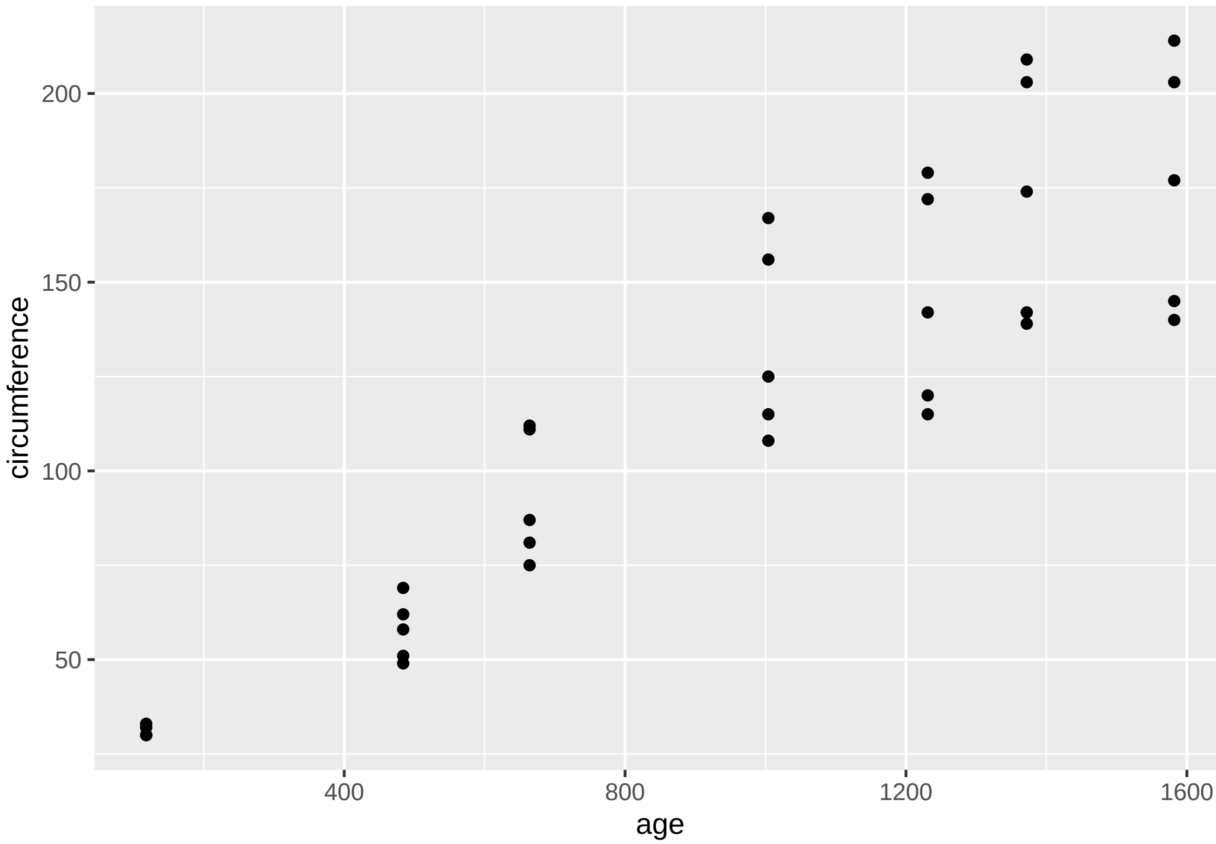
```
boxplot(mtcars$disp ~ mtcars$hp)
```



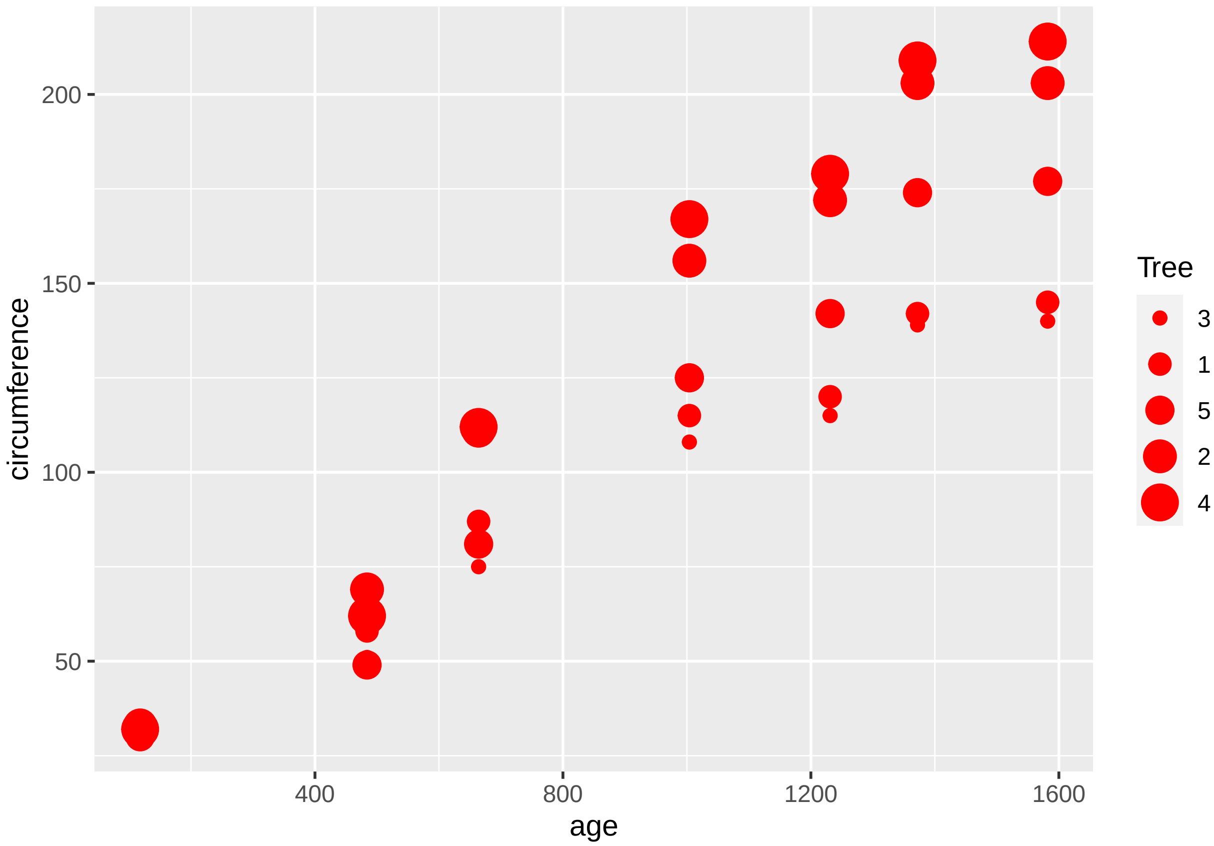
It is showing the box for each category within their disp value and the ~ symbol is used for showing the formula.

```
library(ggplot2)


ggplot(Orange, aes(x = age, y = circumference)) + geom_point()
```

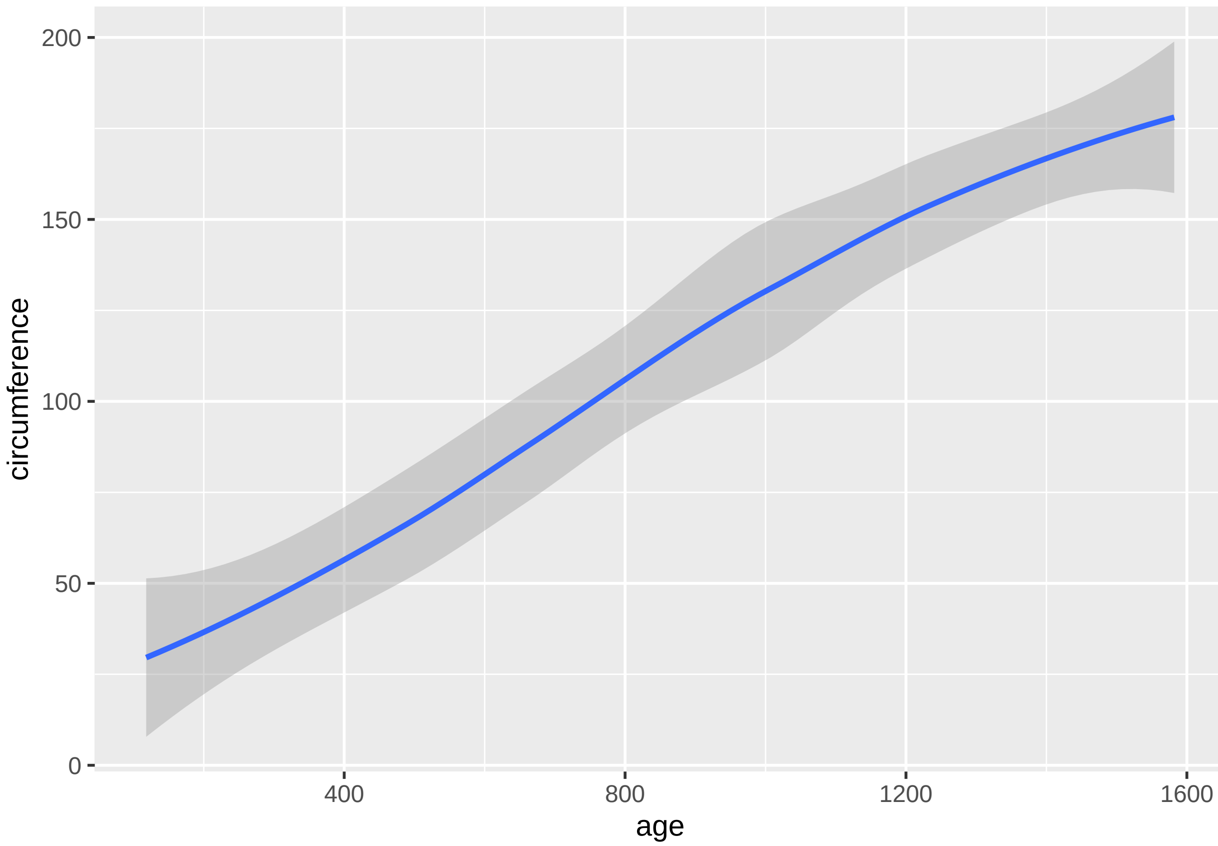



```
ggplot(Orange, aes(x = age, y = circumference)) + geom_point(aes(size = Tree), color = "red")
```




```
ggplot(Orange, aes(x = age, y = circumference)) + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



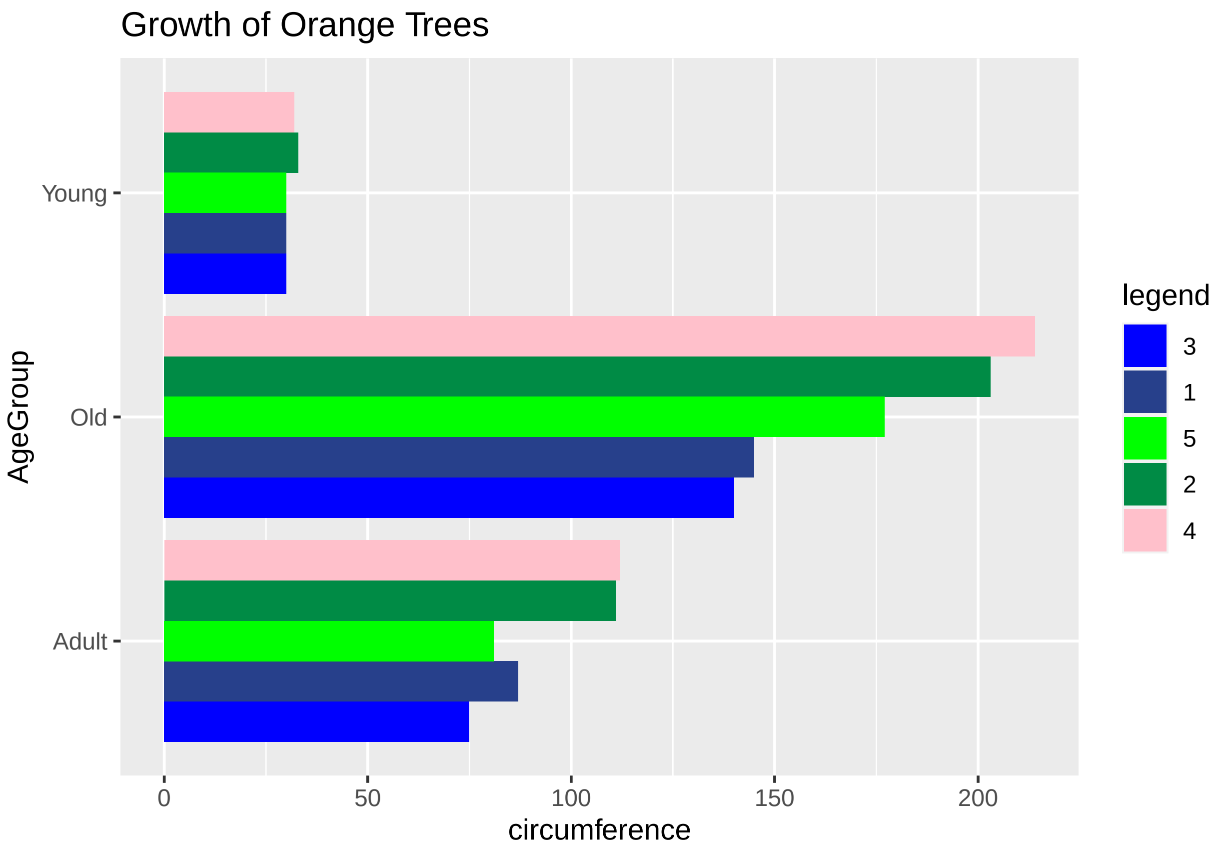
It is using the loess method.

```
Orange$AgeGroup = Orange$age

colour <- c("blue", "royalblue4", "green", "springgreen4", "pink")

Orange$AgeGroup[Orange$age <= 250] = "Young"
Orange$AgeGroup[Orange$age > 250 & Orange$age <= 900] = "Adult"
Orange$AgeGroup[Orange$age > 900] = "Old"

ggplot(Orange, aes(fill = Tree, x = AgeGroup, y = circumference)) + geom_bar(stat = "identity", positio
```


Growth of Orange Trees
Young
Old
Adult
AgeGroup
0
50
100
150
200
circumference
legend
3
1
5
2
4