Data Visualization

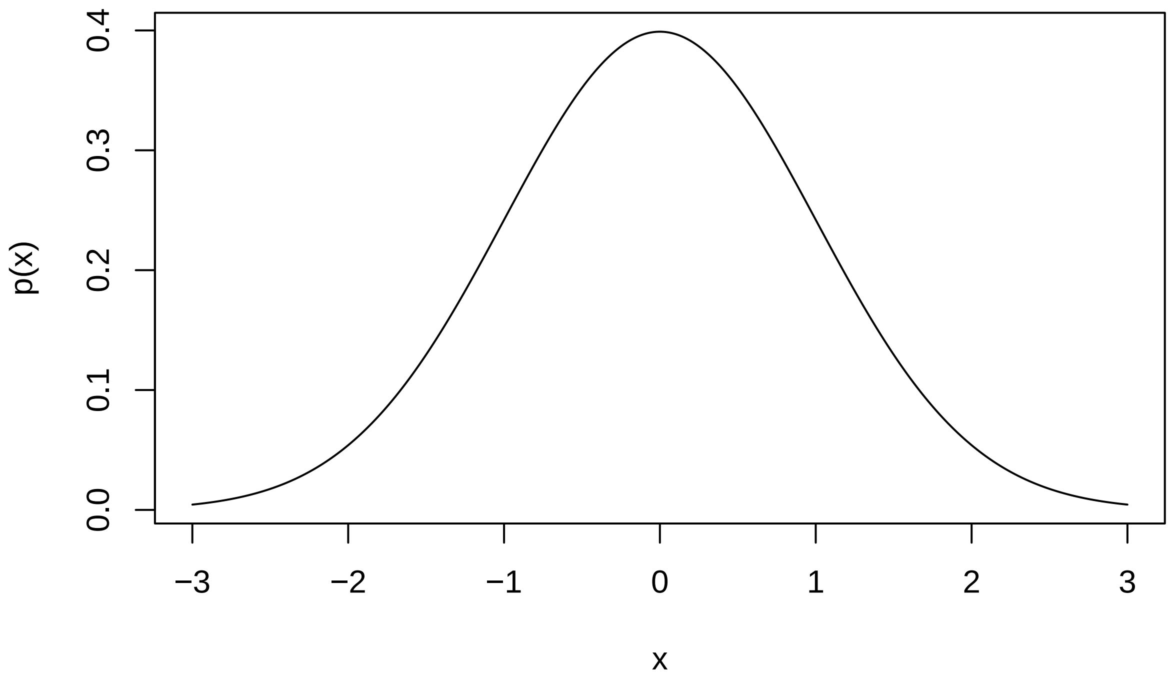
Moein Karami

3/9/2022

a) Simple Plots

1)

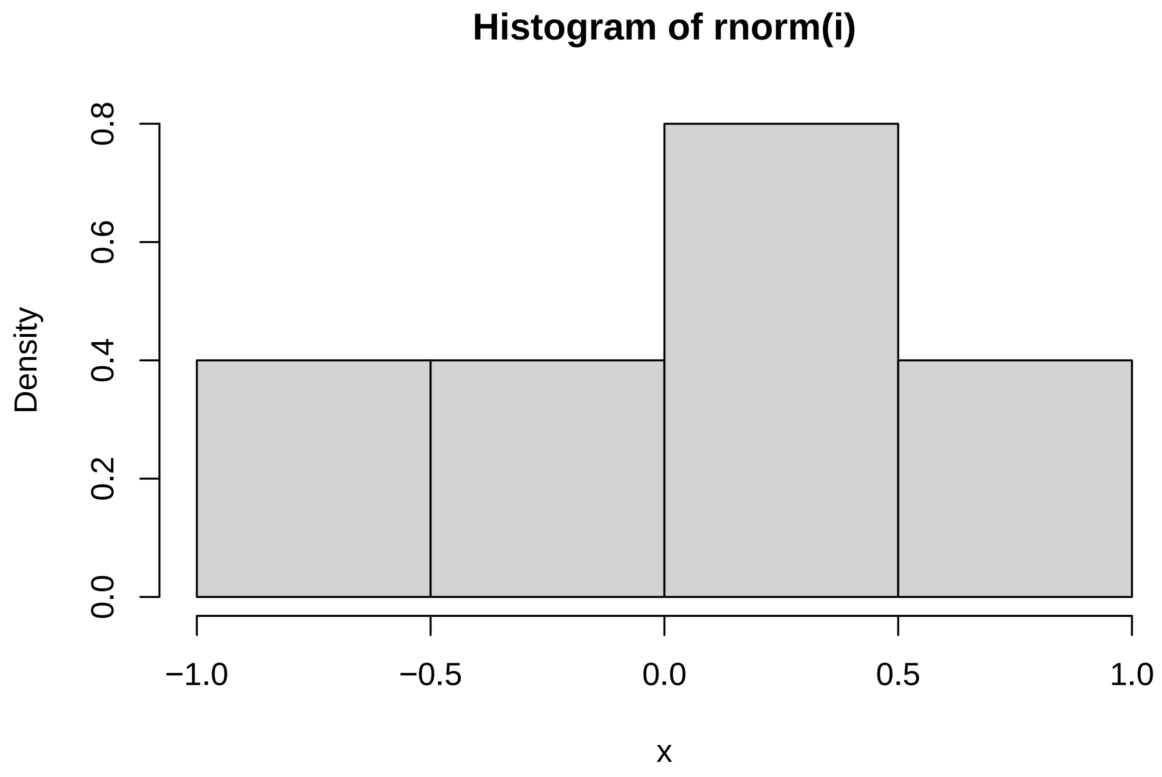
```r
x <- seq(-3, 3, by = 0.001)
plot(x, dnorm(x, mean = 0, sd = 1), type = 'l', ylab = 'p(x)')
```

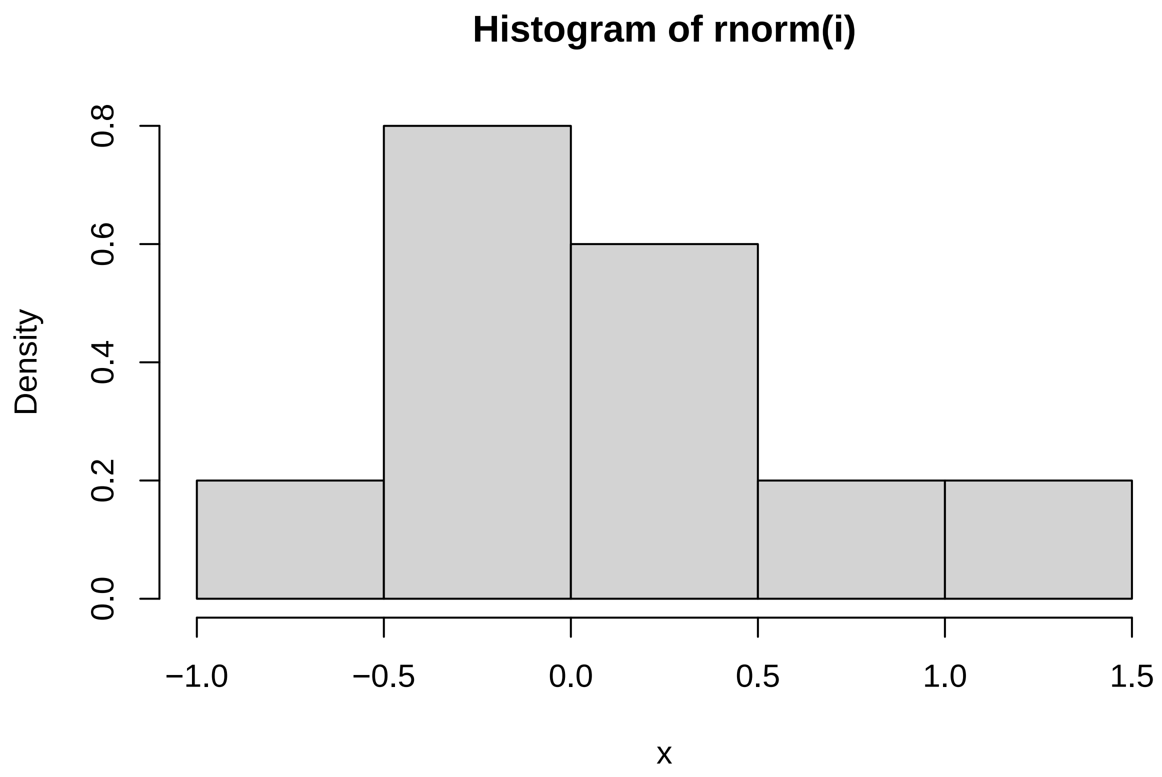


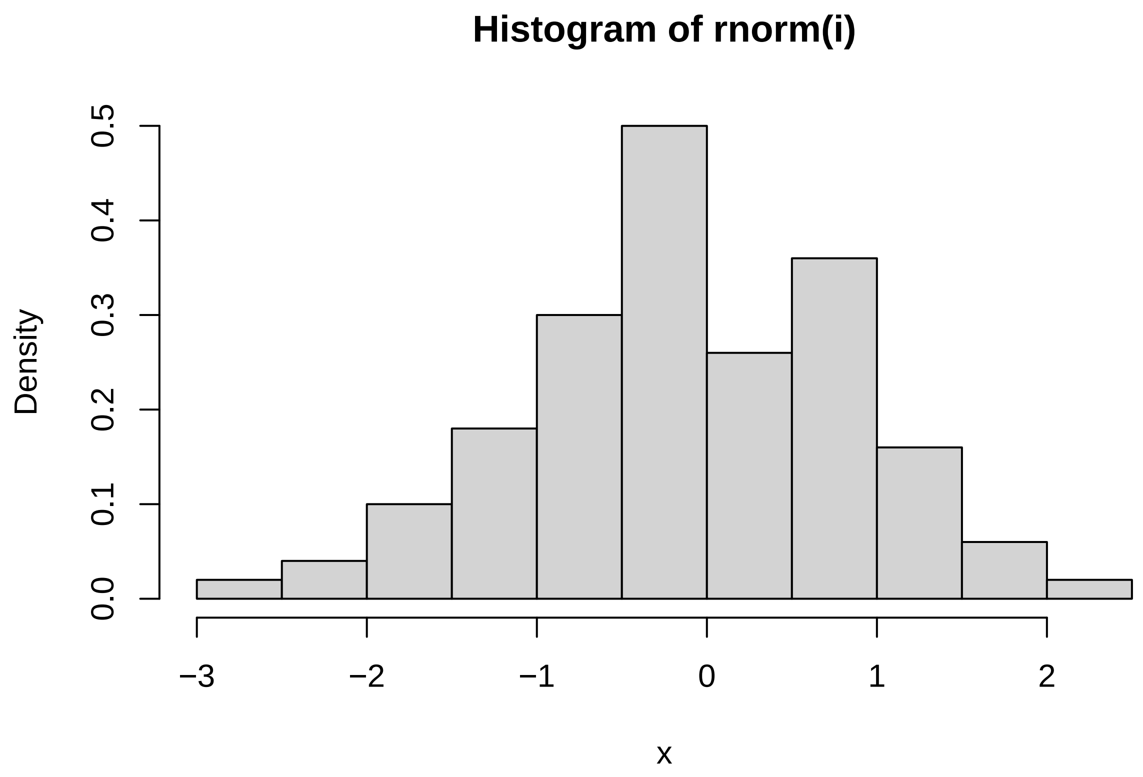
2)

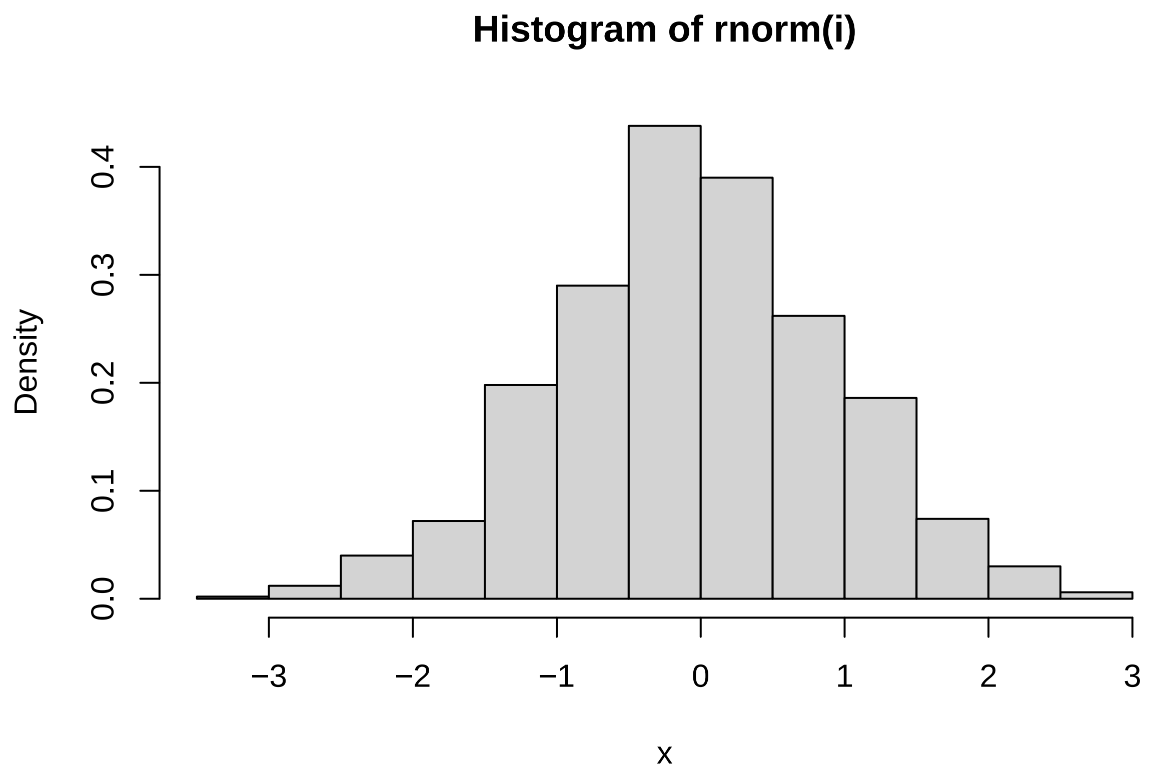
Samples with larger size are closer to real distribution.

```r
for (i in c(5, 10, 100, 1000))
{
  hist(rnorm(i), xlab = 'x', freq = FALSE)
}
```


Histogram of rnorm(i)
Density
0.0
0.2
0.4
0.6
0.8
−1.0
−0.5
0.0
0.5
1.0
x

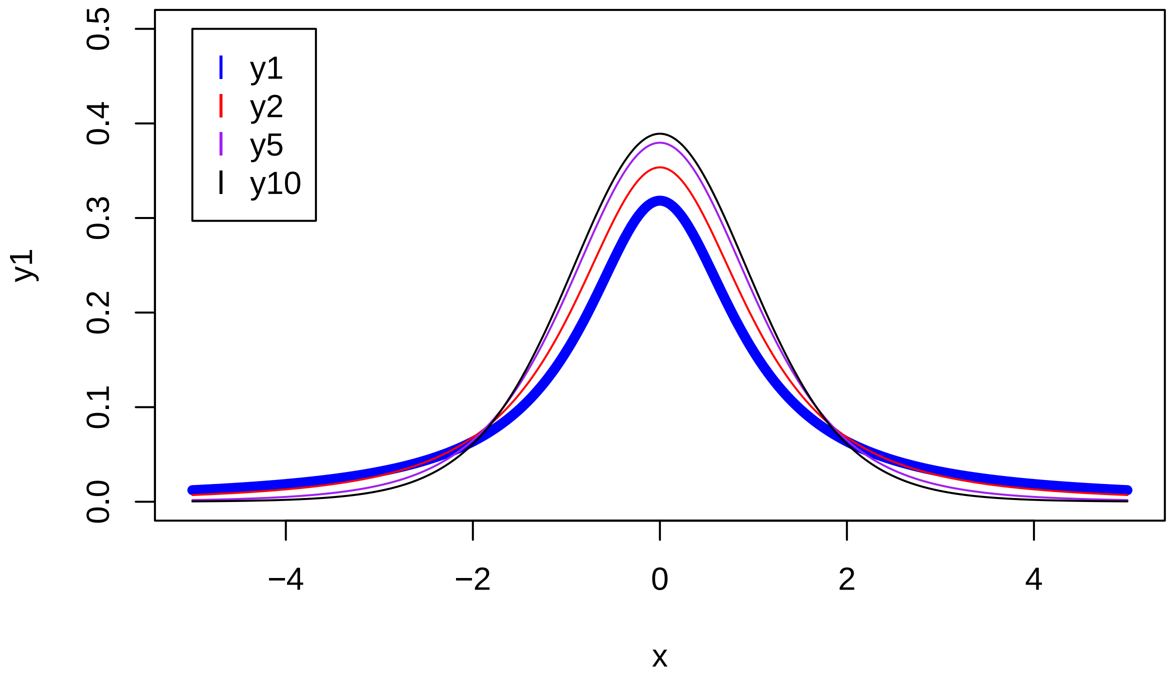

Histogram of rnorm(i)
Density
0.0
0.2
0.4
0.6
0.8
−1.0
−0.5
0.0
0.5
1.0
1.5
x


Histogram of rnorm(i)
Density
0.0
0.1
0.2
0.3
0.4
0.5
−3
−2
−1
0
1
2
x



3)

```
x = seq(-5, 5, by = 0.01)
y1= dt(x, df = 1)
y2 = dt(x, df = 2)
y5 = dt(x, df = 5)
y10 = dt(x, df = 10)

plot(x, y1, type = 'l', col = 'blue', lwd = 5, ylim = c(0, 0.5))
lines(x, y2, col = 'red')
lines(x, y5, col = 'purple')
lines(x, y10)
legend(x = -5, y = 0.5, legend = c('y1', 'y2', 'y5', 'y10'), col = c('blue', 'red', 'purple', 'black'),
```

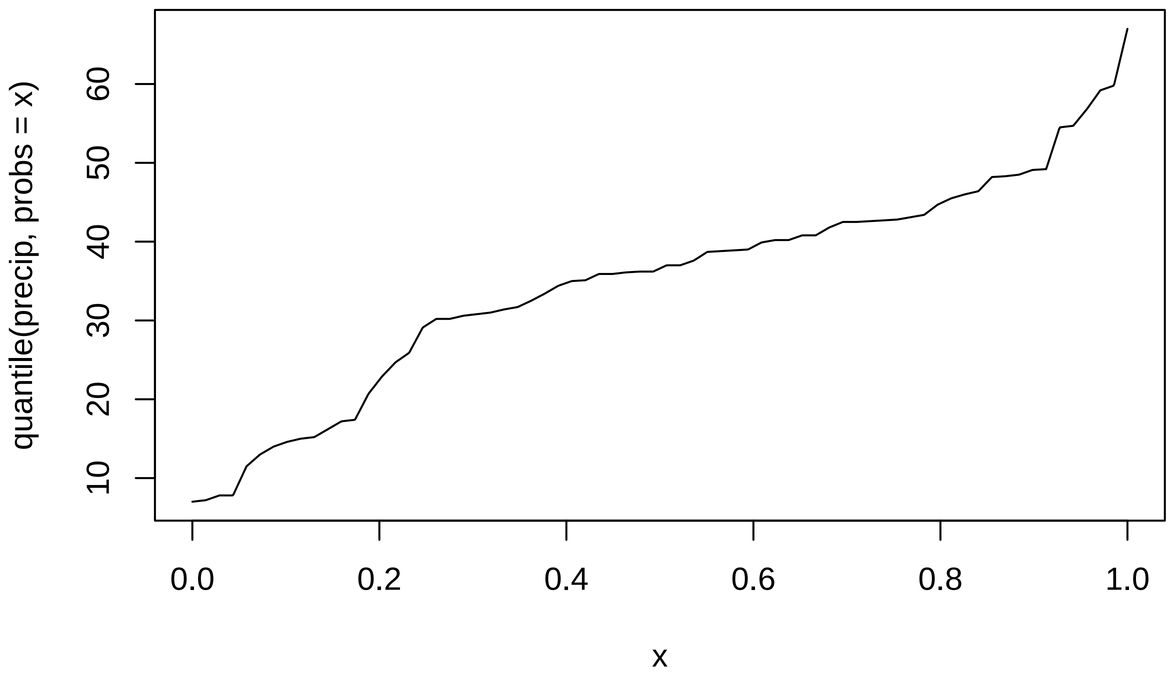


b) QQ-Plots, Histograms, Box-Plots and Pie-Charts

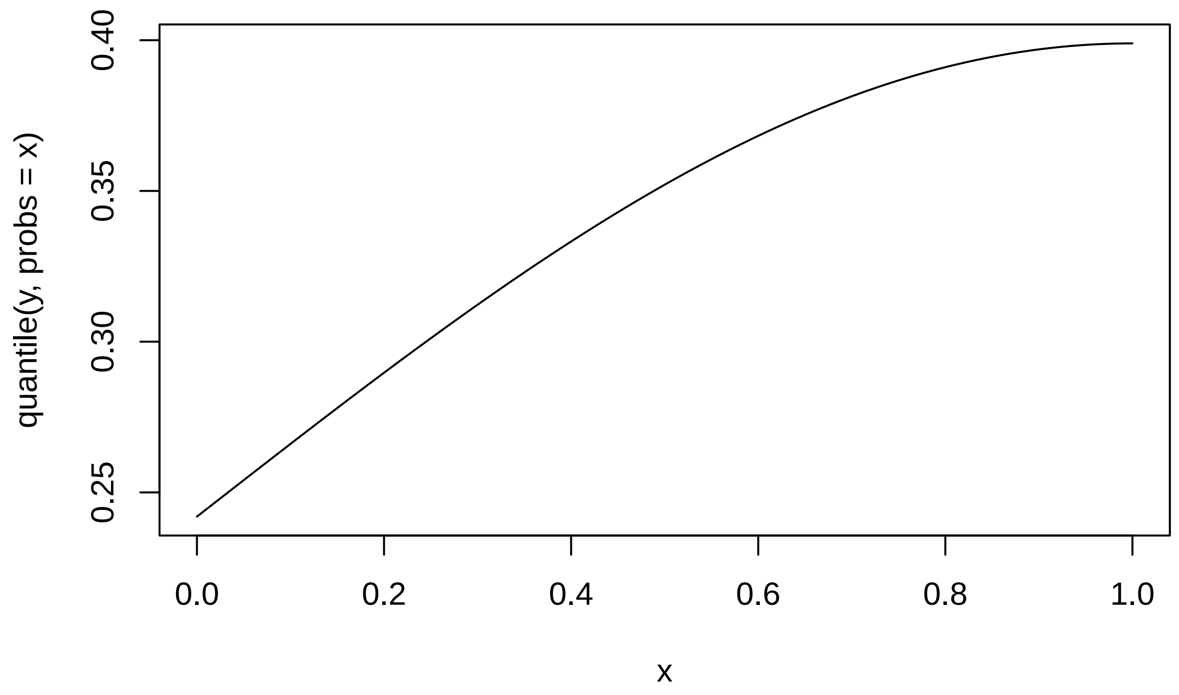
4)

As you can see these two plots are not very similar and precip data-set is not quite symmetric and has a heavy tail, so it does not have a normal distubition.

```
x = seq(0, 1, by = 0.001)
y = dnorm(x, mean = 0, sd = 1)

plot(x, quantile(precip, probs = x), type = 'l')
```



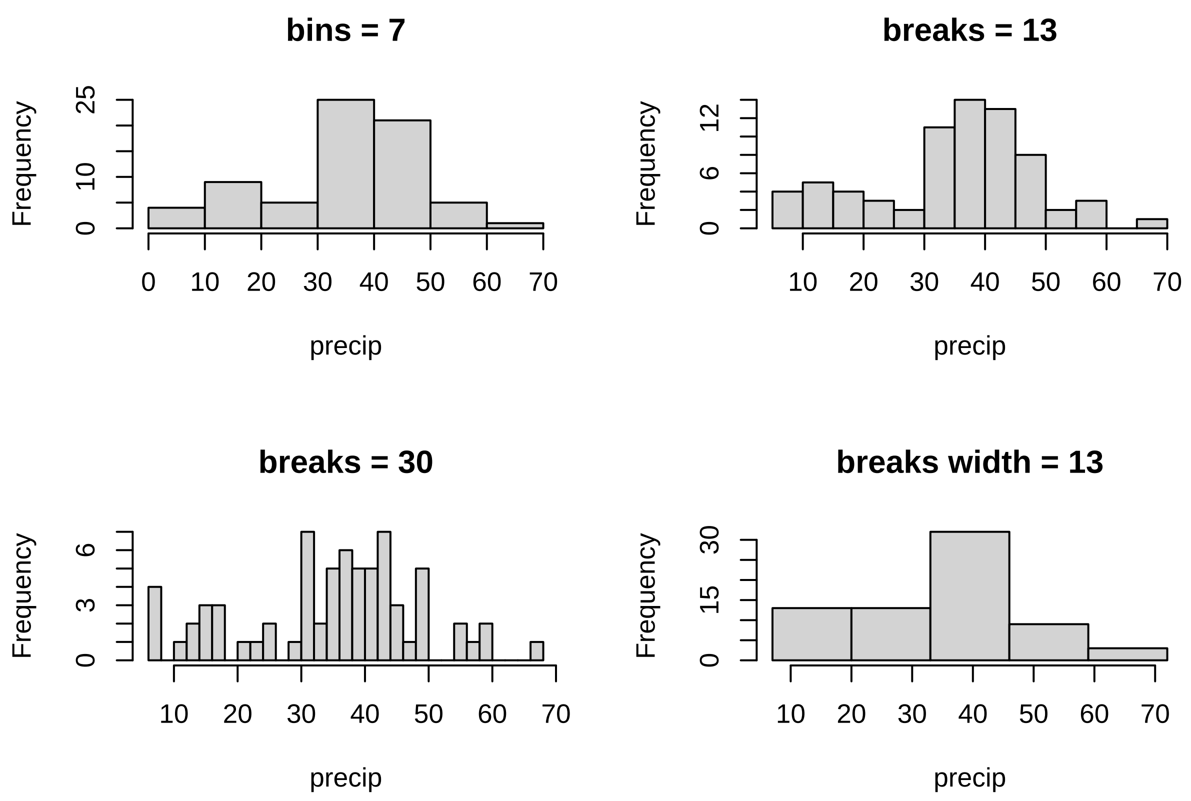
```
plot(x, quantile(y, probs = x), type = 'l')
```



5)

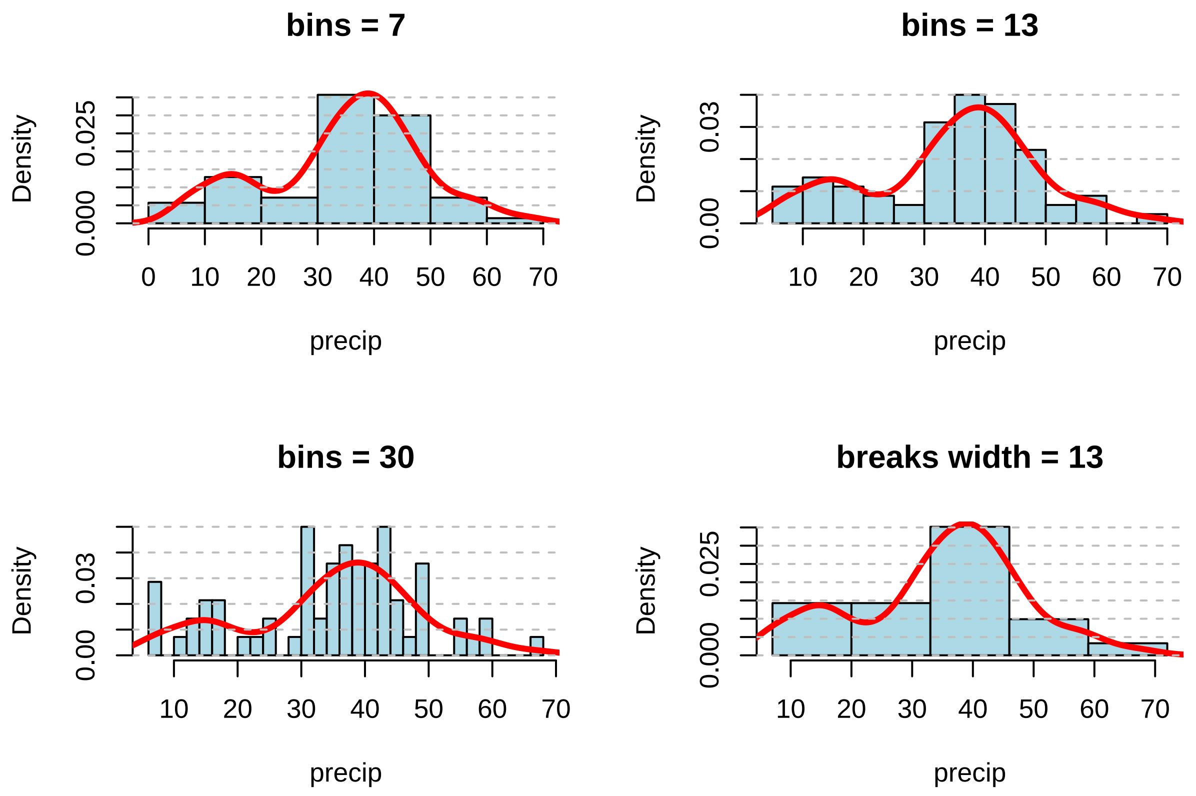
For this specific vector, the histogram with 13 bins visualizes the data more appropriately. In the first and last histogram due to low number of bins we can't recognize any distribution and in the third histogram due to large number of bins, we actually see the original data and each bin is representative of very limited range of data.

```
par(mfrow = c(2, 2))
hist(precip, breaks = 7, main = 'bins = 7')
hist(precip, breaks = 13, main = 'breaks = 13')
hist(precip, breaks = 30, main = 'breaks = 30')
hist(precip, breaks = seq(min(precip), max(precip) + 13, by = 13), main = 'breaks width = 13')
```



6)

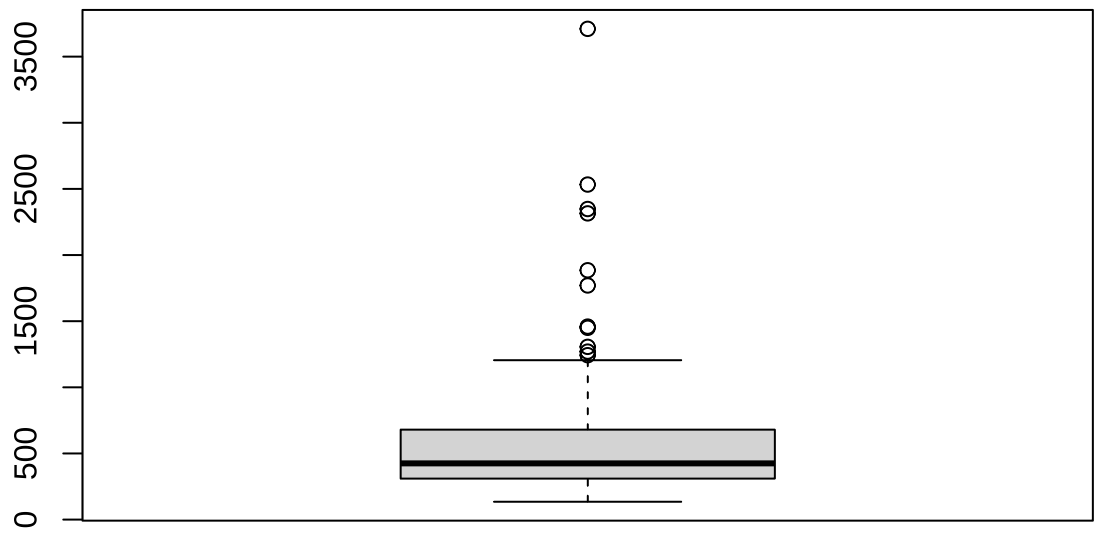
```r
par(mfrow = c(2, 2))
for (i in c(7, 13, 30))
{
    hist(precip, breaks = i, main = sprintf('bins = %d', i), prob = T, col = 'light blue')
    lines(density(precip), col = 'red', lwd = 3 )
    grid(nx = NA, ny = NULL, lty = 2, col = "gray", lwd = 1)
}

hist(precip, breaks = seq(min(precip), max(precip) + 13, by = 13), main = 'breaks width = 13', prob = T
lines(density(precip), col = 'red', lwd = 3 )
grid(nx = NA, ny = NULL, lty = 2, col = "gray", lwd = 1)
```



7)

This distribution is very right skewed.

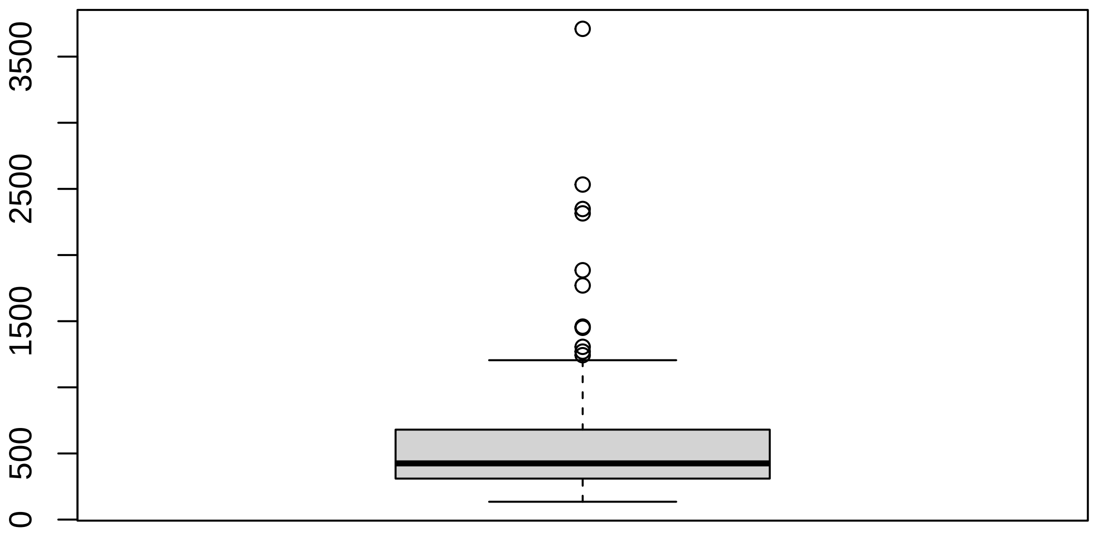
```
boxplot(rivers)
```



8)

You can see whiskers and outliners below.

```
box <- boxplot(rivers)
```



```
box
```

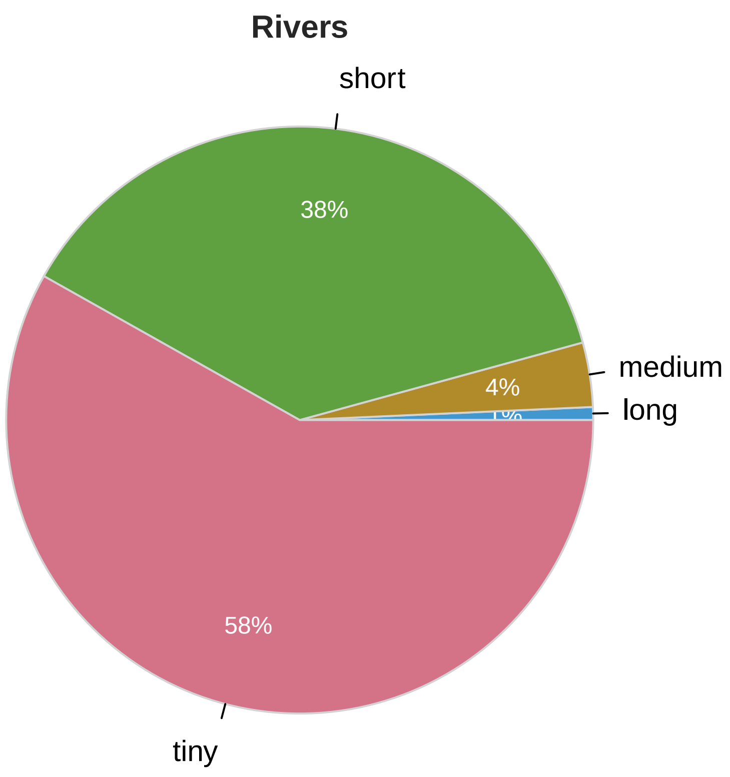
```
## $stats
##      [,1]
## [1,]  135
## [2,]  310
## [3,]  425
## [4,]  680
## [5,] 1205
##
## $n
## [1] 141
##
## $conf
##          [,1]
## [1,] 375.7678
## [2,] 474.2322
##
## $out
##  [1] 1459 1450 1243 2348 3710 2315 2533 1306 1270 1885 1770
##
## $group
##  [1] 1 1 1 1 1 1 1 1 1 1 1
##
## $names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

9)

```
rivers_cat = ifelse(rivers < 500, 'tiny', ifelse(rivers < 1500, 'short',
                                            ifelse(rivers < 3000, 'medium',
                                                   'long')))

# install.packages('lessR')
library(lessR)
```

```
##
## lessR 4.1.6  feedback: gerbing@pdx.edu  web: lessRstats.com/new
## ---------------------------------------------------------------
## > d <- Read("")   Read text, Excel, SPSS, SAS, or R data file
##   d is default data frame, data= in analysis routines optional
##
## Learn about reading, writing, and manipulating data, graphics,
## testing means and proportions, regression, factor analysis,
## customization, and descriptive statistics from pivot tables.
##   Enter:  browseVignettes("lessR")
##
## View changes in this or recent versions of lessR.
##   Enter: help(package=lessR)  Click: Package NEWS
##   Enter: interact()  for access to interactive graphics
##   New function: reshape_long() to move data from wide to long
```

```
PieChart(rivers_cat, hole = 0, main = 'Rivers')
```



```
## >>> Suggestions
## PieChart(rivers_cat, hole=0)  # traditional pie chart
## PieChart(rivers_cat, values="%")  # display %'s on the chart
## PieChart(rivers_cat)  # bar chart
## Plot(rivers_cat)  # bubble plot
## Plot(rivers_cat, values="count")  # lollipop plot
##
## --- rivers_cat ---
##
##                  long  medium  short   tiny     Total
## Frequencies:        1       5     53     82       141
## Proportions:   0.007   0.035  0.376  0.582     1.000
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 130.177, df = 3, p-value = 0.000
```

10)

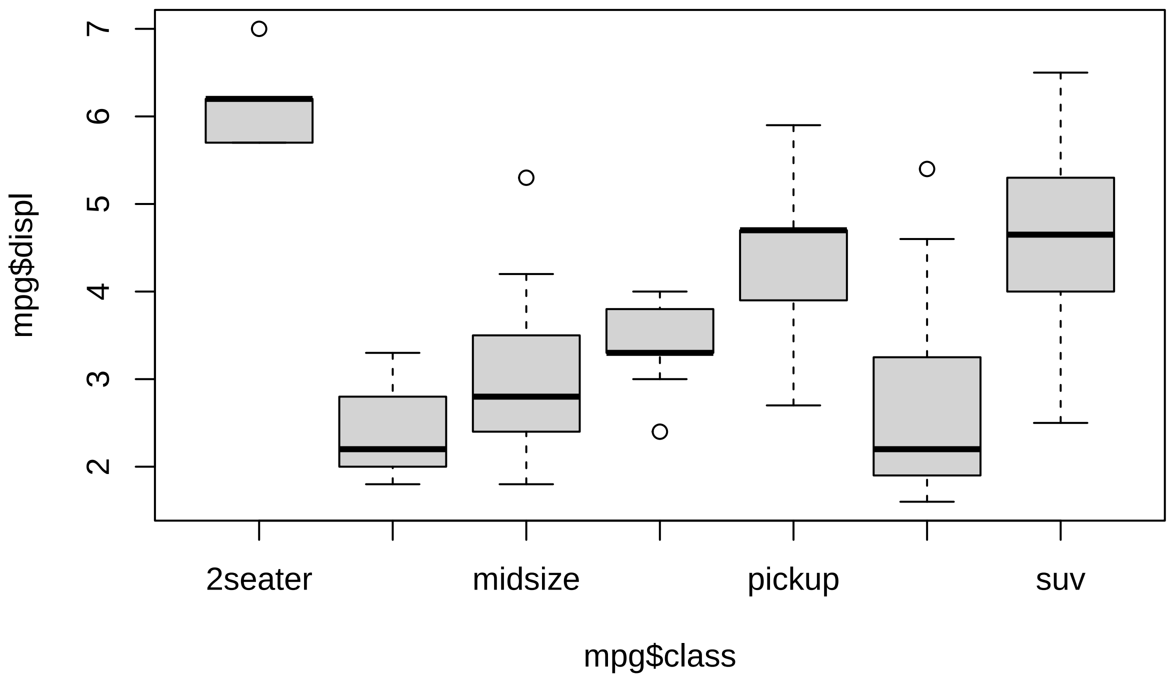
This command results boxplots of "displ vs class".

In general, the tilde ('~') separates the left side of a formula with the right side of the formula.

For example, in a linear function, it would separate the dependent variable from the independent variables and can be interpreted as saying, "as a function of."

In this command we considered mpg$displ as a function of mpg$class and then draw box plot for each class based on ditribution of displ.

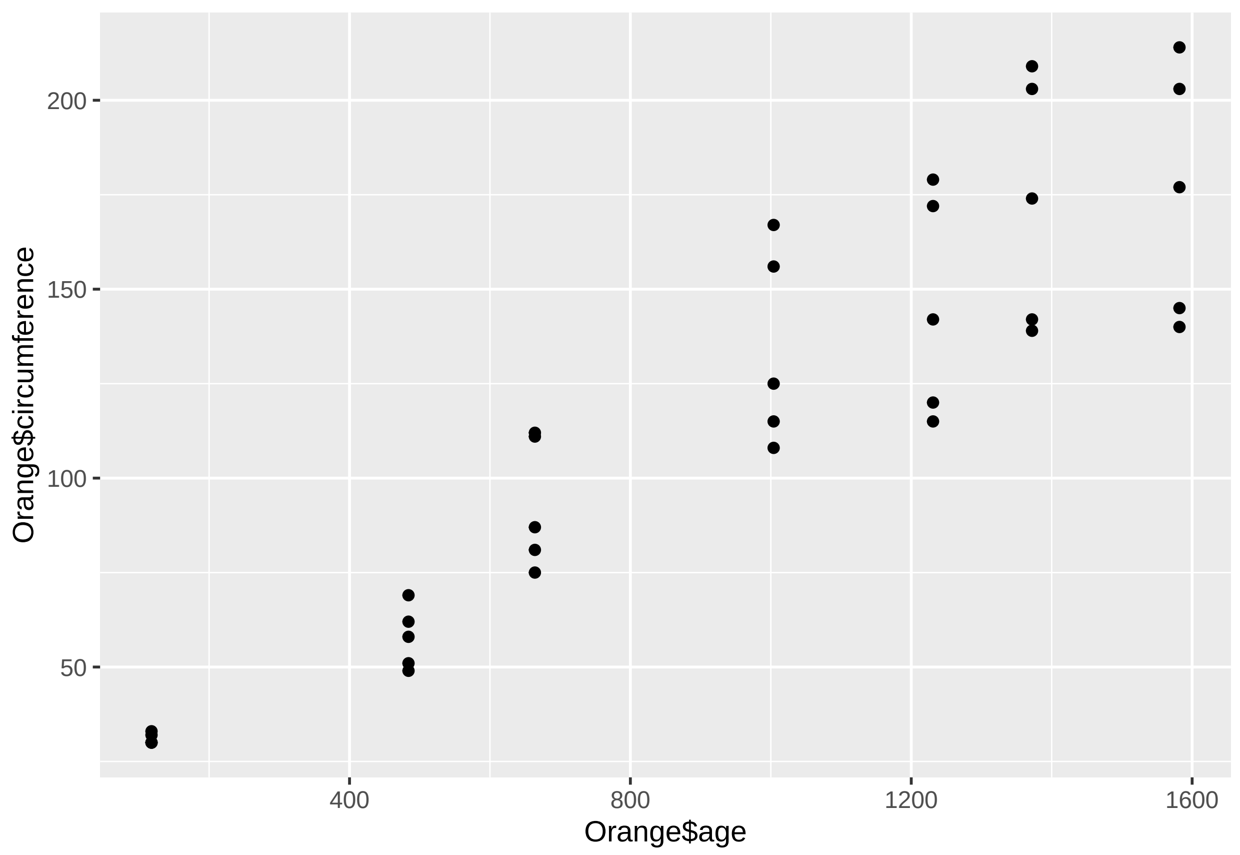
```
library(ggplot2)
boxplot(mpg$displ ~ mpg$class)
```



c) ggplot2

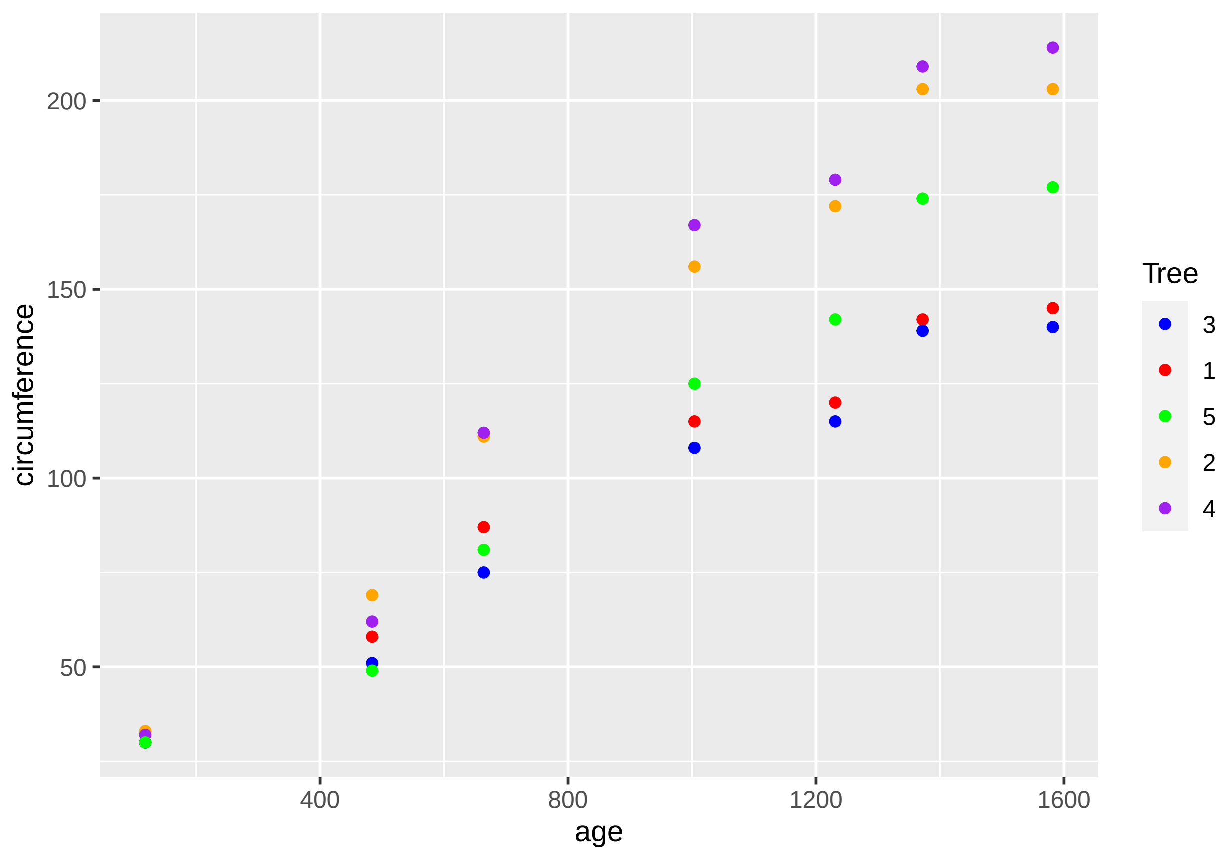
11)

```
library(ggplot2)
ggplot(Orange, aes(x = Orange$age, y = Orange$circumference)) +
  geom_point()
```



12)

```
ggplot(Orange, aes(x = age, y = circumference)) +
  geom_point(aes(color = Tree)) + scale_color_manual(values=c("blue", "red", "green", "orange", "purple
```

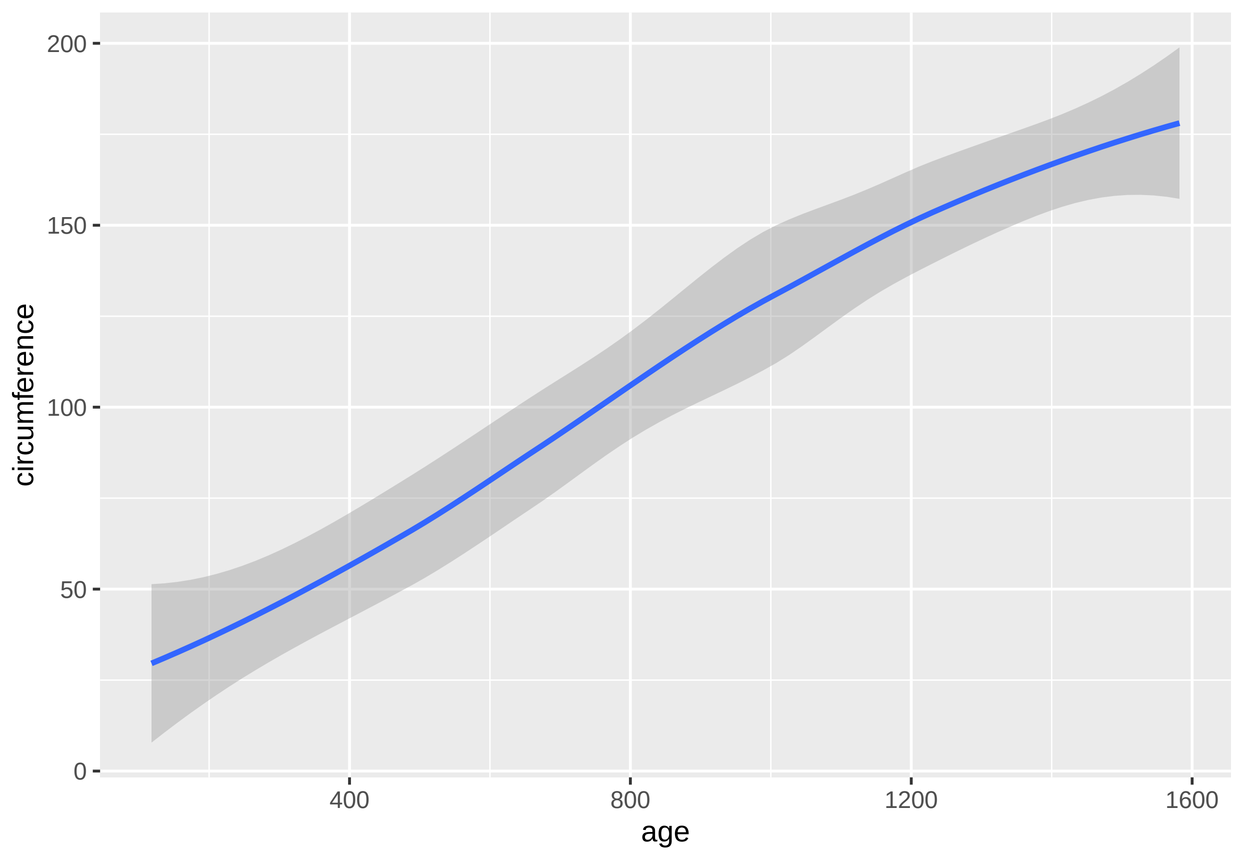


13)

method = "loess": This is the default value for small number of observations. It computes a smooth local regression.

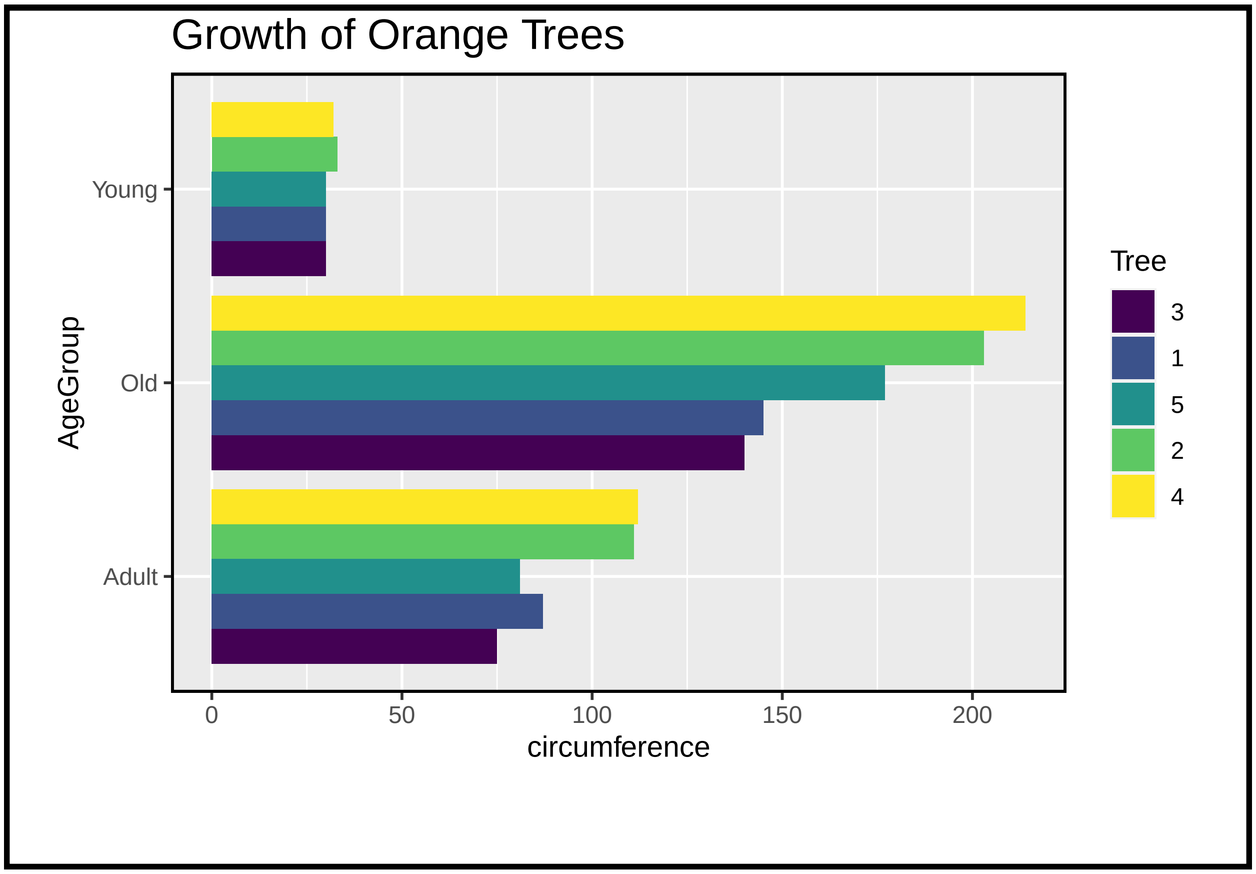
```
ggplot(Orange, aes(age, circumference)) + geom_smooth()
```

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



14)

```
Orange$AgeGroup <- ifelse(Orange$age <= 250, 'Young',
                          ifelse(Orange$age <= 950, 'Adult', 'Old'))

ggplot(Orange, aes(x = AgeGroup, y = circumference, fill = Tree), main = 'Growth of Orange Trees') +
  geom_bar(stat = 'identity', position = 'dodge') +
  coord_flip() +
  theme(panel.border = element_rect(color = 'black', fill = NA, size = 1), plot.margin = unit(c(5, 20, 
  labs(title = "Growth of Orange Trees")
```



d) ggmap

15)

```
data(state)
centers_df <- data.frame(state.center)
centers_df
```

```
##             x       y
## 1   -86.7509 32.5901
## 2  -127.2500 49.2500
## 3  -111.6250 34.2192
## 4   -92.2992 34.7336
## 5  -119.7730 36.5341
## 6  -105.5130 38.6777
## 7   -72.3573 41.5928
## 8   -74.9841 38.6777
## 9   -81.6850 27.8744
## 10  -83.3736 32.3329
## 11 -126.2500 31.7500
## 12 -113.9300 43.5648
## 13  -89.3776 40.0495
## 14  -86.0808 40.0495
## 15  -93.3714 41.9358
## 16  -98.1156 38.4204
## 17  -84.7674 37.3915
## 18  -92.2724 30.6181
## 19  -68.9801 45.6226
## 20  -76.6459 39.2778
## 21  -71.5800 42.3645
## 22  -84.6870 43.1361
## 23  -94.6043 46.3943
## 24  -89.8065 32.6758
## 25  -92.5137 38.3347
## 26 -109.3200 46.8230
## 27  -99.5898 41.3356
## 28 -116.8510 39.1063
## 29  -71.3924 43.3934
## 30  -74.2336 39.9637
## 31 -105.9420 34.4764
## 32  -75.1449 43.1361
## 33  -78.4686 35.4195
## 34 -100.0990 47.2517
## 35  -82.5963 40.2210
## 36  -97.1239 35.5053
## 37 -120.0680 43.9078
## 38  -77.4500 40.9069
## 39  -71.1244 41.5928
## 40  -80.5056 33.6190
## 41  -99.7238 44.3365
## 42  -86.4560 35.6767
## 43  -98.7857 31.3897
## 44 -111.3300 39.1063
## 45  -72.5450 44.2508
## 46  -78.2005 37.5630
## 47 -119.7460 47.4231
## 48  -80.6665 38.4204
## 49  -89.9941 44.5937
## 50 -107.2560 43.0504
```

16)

We should pay money for this one.

```
library(ggplot2)
library(ggmap)
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
library(qmap)
```

```
## Loading required package: fitdistrplus
```

```
## Loading required package: MASS
```

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

```
ggmap::register_google(key = "AIzaSyAIPS2fbBhllG0dZphsaPmPFluVwMnLoVg")
#usa = qmap('usa')
#ggmap(usa)
```