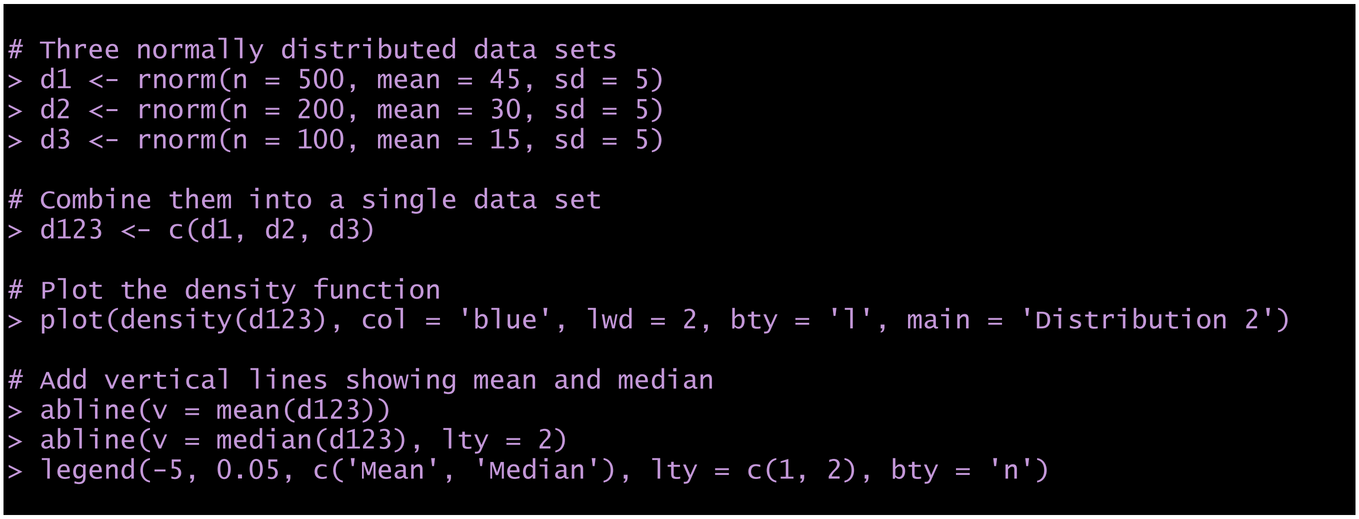
BUSINESS ANALYTICS USING STATISTICAL MODELING – ASSIGNMENT 2

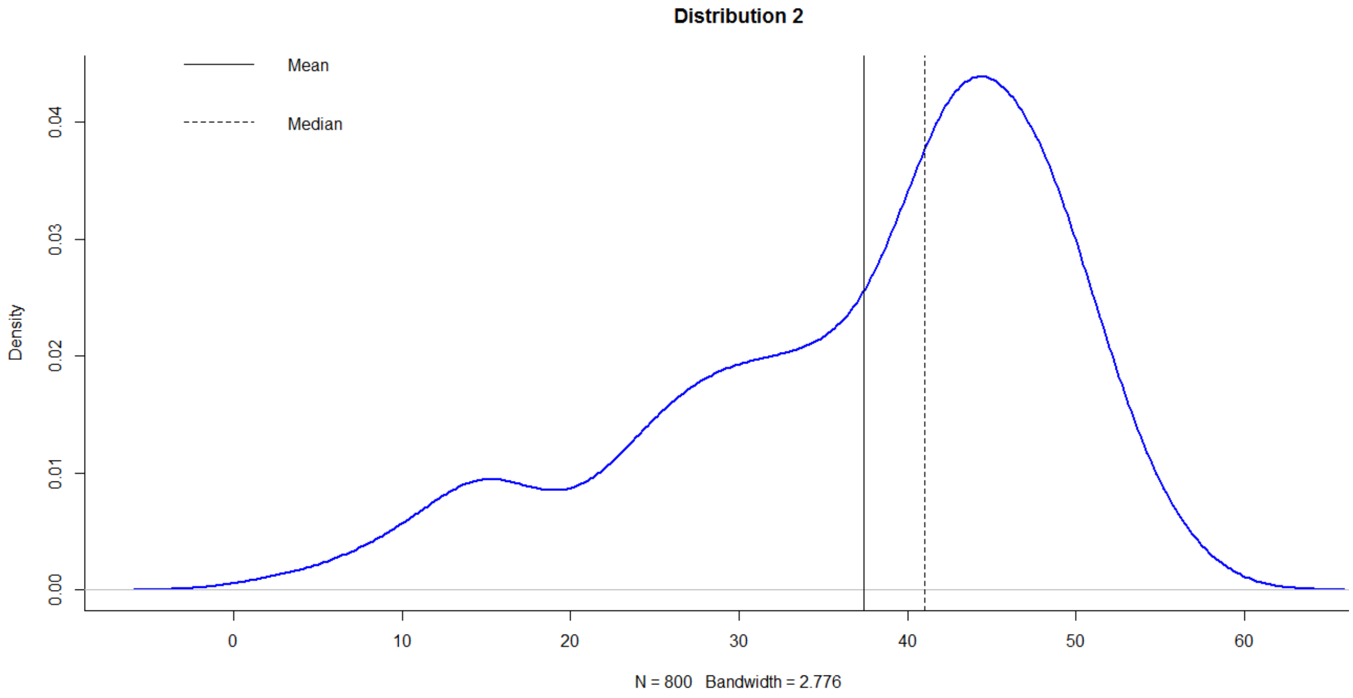
2017 / 03 / 01

1.

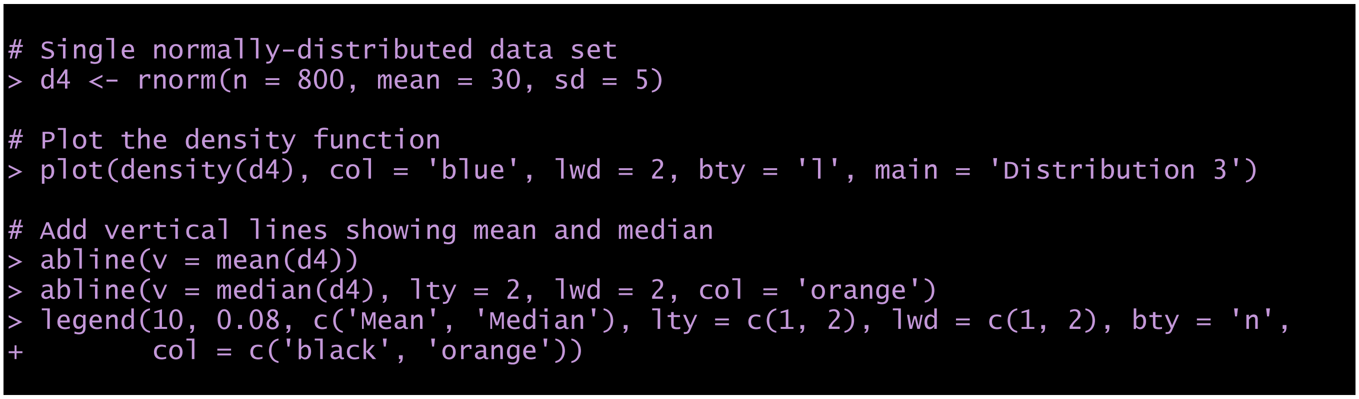
a. Create and visualize “Distribution 2”: a combined dataset (n=800) that is negatively skewed (tail stretches to the left). Change the mean and standard deviation of a, b, and c to achieve this new distribution. Compute the mean and median, and draw lines showing the mean (thick line) and median (thin line).



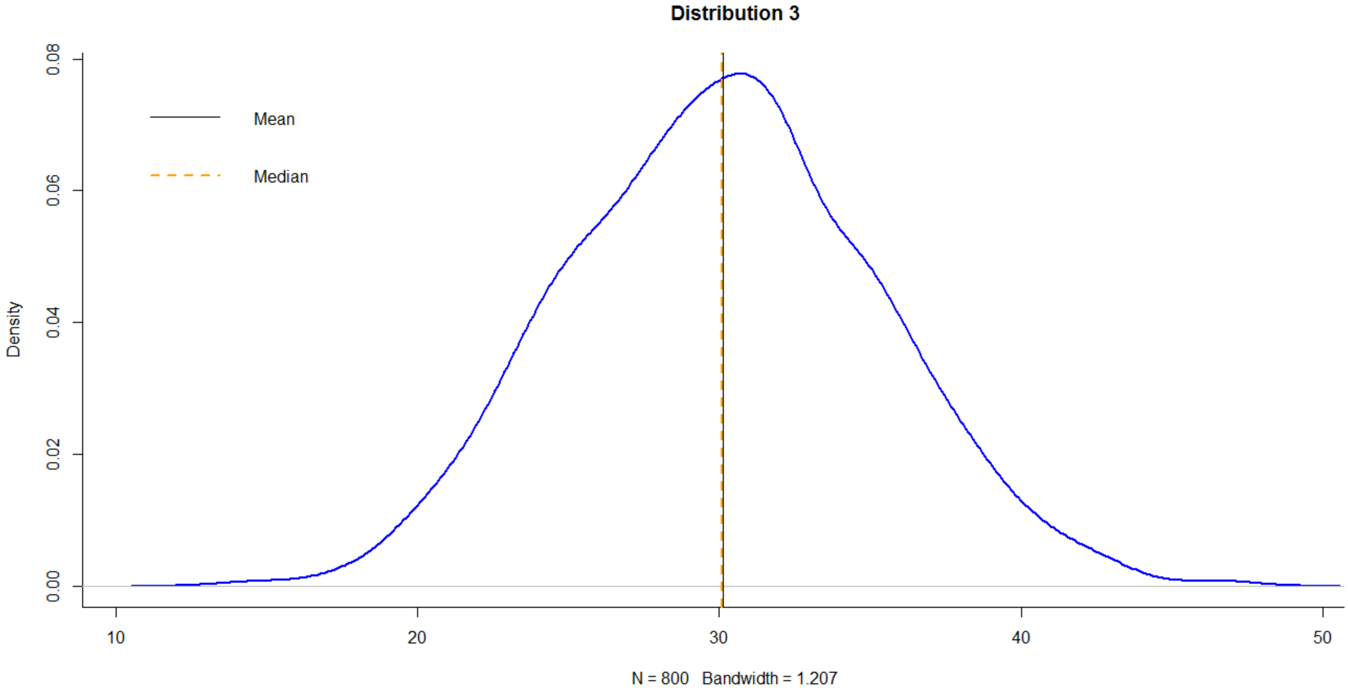
```
# Three normally distributed data sets
> d1 <- rnorm(n = 500, mean = 45, sd = 5)
> d2 <- rnorm(n = 200, mean = 30, sd = 5)
> d3 <- rnorm(n = 100, mean = 15, sd = 5)

# Combine them into a single data set
> d123 <- c(d1, d2, d3)

# Plot the density function
> plot(density(d123), col = 'blue', lwd = 2, bty = 'l', main = 'Distribution 2')

# Add vertical lines showing mean and median
> abline(v = mean(d123))
> abline(v = median(d123), lty = 2)
> legend(-5, 0.05, c('Mean', 'Median'), lty = c(1, 2), bty = 'n')
```



b. Create “Distribution 3”: a single dataset that is normally distributed (bell-shaped, symmetric) -- you do not need to combine datasets, just use the rnorm function to create a single large dataset (n=800). Show your code, compute the mean and median, and draw lines showing the mean (thick line) and median (thin line).



```
# Single normally-distributed data set
> d4 <- rnorm(n = 800, mean = 30, sd = 5)

# Plot the density function
> plot(density(d4), col = 'blue', lwd = 2, bty = 'l', main = 'Distribution 3')

# Add vertical lines showing mean and median
> abline(v = mean(d4))
> abline(v = median(d4), lty = 2, lwd = 2, col = 'orange')
> legend(10, 0.08, c('Mean', 'Median'), lty = c(1, 2), lwd = c(1, 2), bty = 'n',
+        col = c('black', 'orange'))
```



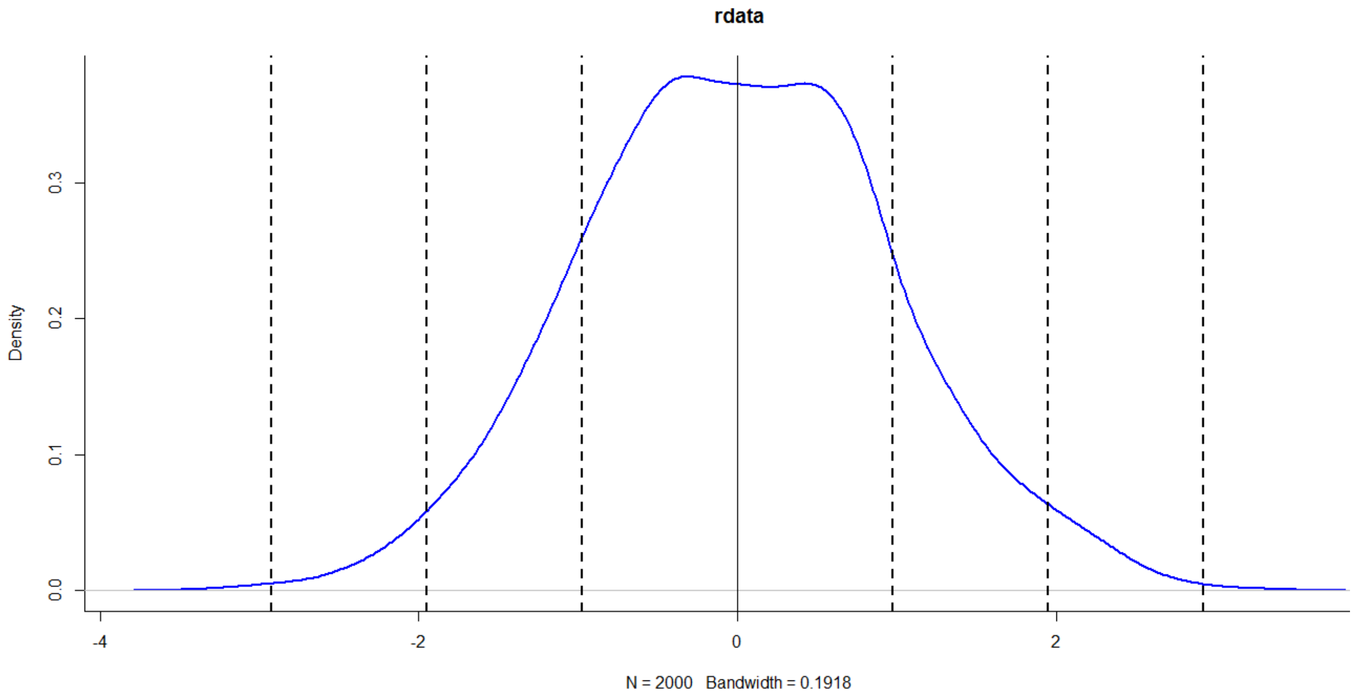
c. In general, which measure of central tendency (mean or median) do you think will be more sensitive (will change more) to outliers being added to your data?

- Mean will be more sensitive (will change more) to outliers being added to the data.

2.

a. Create a random dataset (call it ‘rdata’) that is normally distributed with: n=2000, mean=0, sd=1. Draw a density plot and put a solid vertical line on the mean, and dashed vertical lines at the 1st, 2nd, and 3rd standard deviations on both sides of the mean. You should have a total of 7 vertical lines.

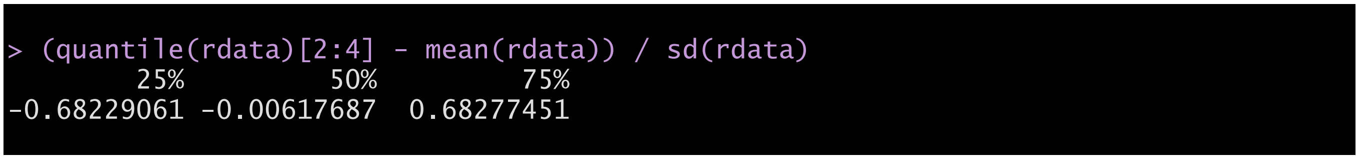
```
# Random normally-distributed data set
> rdata <- rnorm(n = 2000, mean = 0, sd = 1)

# Plot the density function
> plot(density(rdata), col = 'blue', lwd = 2, bty = 'l', main = 'rdata')

# Add vertical lines showing mean and the 1st, 2nd, and 3rd standard deviations on
# both sides of the mean
> abline(v = mean(rdata))
> for (i in -3:3){
+   if(i == 0) next
+   abline(v = i * sd(rdata), lty = 2, lwd = 2)
+ }
```



b. Using the quantile function, which data points correspond to the 1st, 2nd, and 3rd quartiles (i.e., 25th, 50th, 75th percentiles).

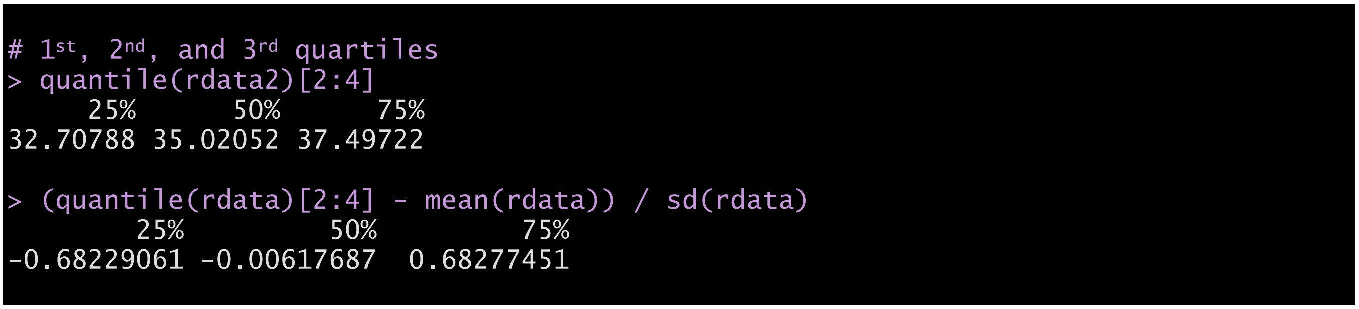
```
# 1st, 2nd, and 3rd quartiles
> quantile(rdata)[2:4]
         25%          50%          75% 
-0.667372510 -0.008453127  0.662977418 
```

How many standard deviations away from the mean (use positive or negative) are those points corresponding to the 1st, 2nd, and 3rd quartiles?



```
> (quantile(rdata)[2:4] - mean(rdata)) / sd(rdata)
        25%         50%         75% 
-0.68229061 -0.00617687  0.68277451 
```

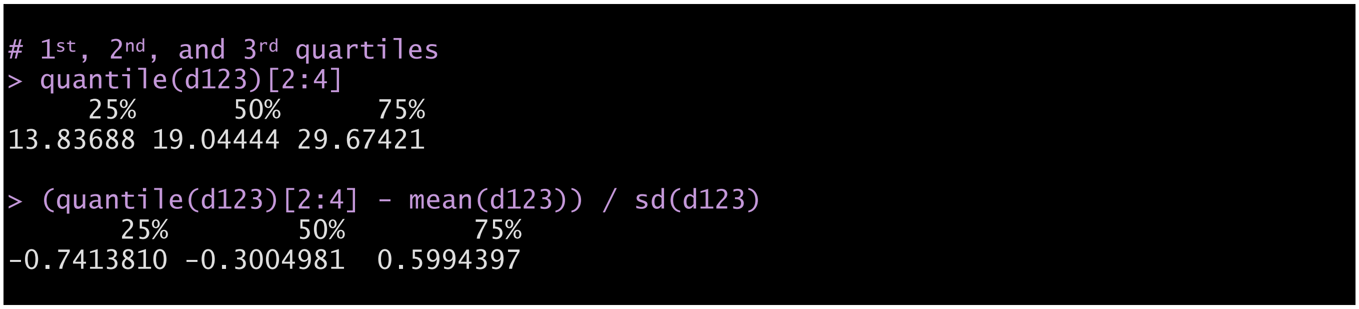
c. Now create a new random dataset that is normally distributed with: n=2000, mean=35, sd=3.5. In this distribution, how many standard deviations away from the mean (use positive or negative) are those points corresponding to the 1st and 3rd quartiles? Compare your answer to (b)



```
# 1st, 2nd, and 3rd quartiles
> quantile(rdata2)[2:4]
     25%      50%      75% 
32.70788 35.02052 37.49722 

> (quantile(rdata)[2:4] - mean(rdata)) / sd(rdata)
        25%         50%         75% 
-0.68229061 -0.00617687  0.68277451 
```

- ➢ The 1st, 2nd, and 3rd quartiles for both rdata and rdata2 have the same distances (in standard deviation unit) from their corresponding means, because they both are normally distributed.

d. Finally, recall the dataset d123 shown in question 1. In that distribution, how many standard deviations away from the mean (use positive or negative) are those data points corresponding to the 1st and 3rd quartiles? Compare your answer to (b)



```
# 1st, 2nd, and 3rd quartiles
> quantile(d123)[2:4]
     25%      50%      75% 
13.83688 19.04444 29.67421 

> (quantile(d123)[2:4] - mean(d123)) / sd(d123)
       25%        50%        75% 
-0.7413810 -0.3004981  0.5994397 
```

- ➢ The 1st, 2nd, and 3rd quartiles for d123 does not have the same distances (in standard deviation unit) from their corresponding means as the 1st, 2nd, and 3rd quartiles for rdata, because d123 is not normally distributed like rdata.

3.

a. From the StackOverflow question, which formula does *Rob Hyndman’s* answer (1st answer) suggest to use for bin widths/number?

➢ The “Freedman – Diaconis’ rule”

- The bin-width:

  $h = 2 * IQR * n^{-1/3}$

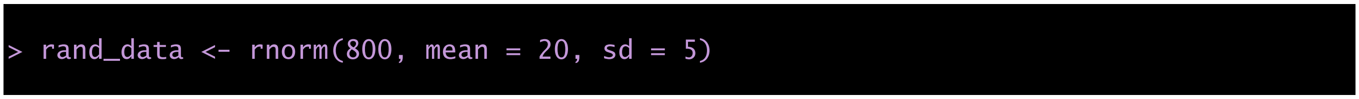
- The number of bins:

  k = (max – min) / h

Also, what does the Wikipedia article say is the benefit of that formula?

➢ It uses the IQR (Inter-Quartile Range), which is less sensitive than the standard deviation to outliers in the data.

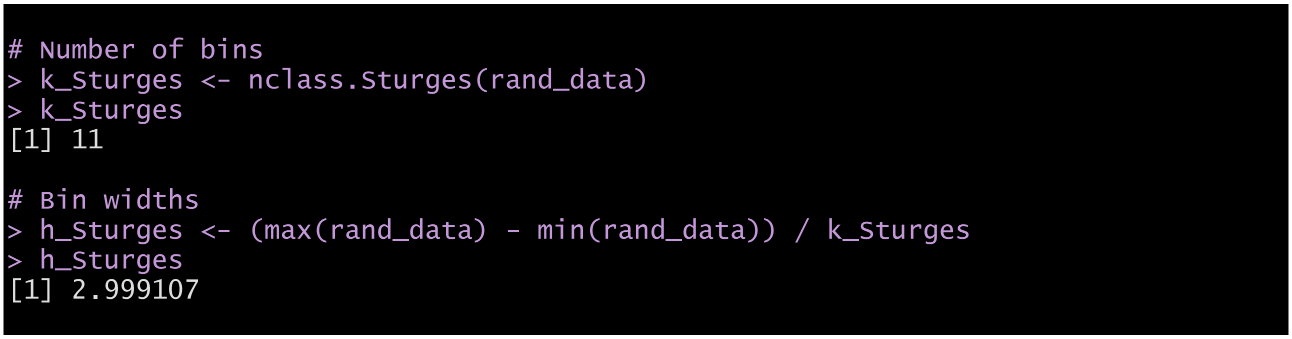
b. Given a random normal distribution:



```
> rand_data <- rnorm(800, mean = 20, sd = 5)
```

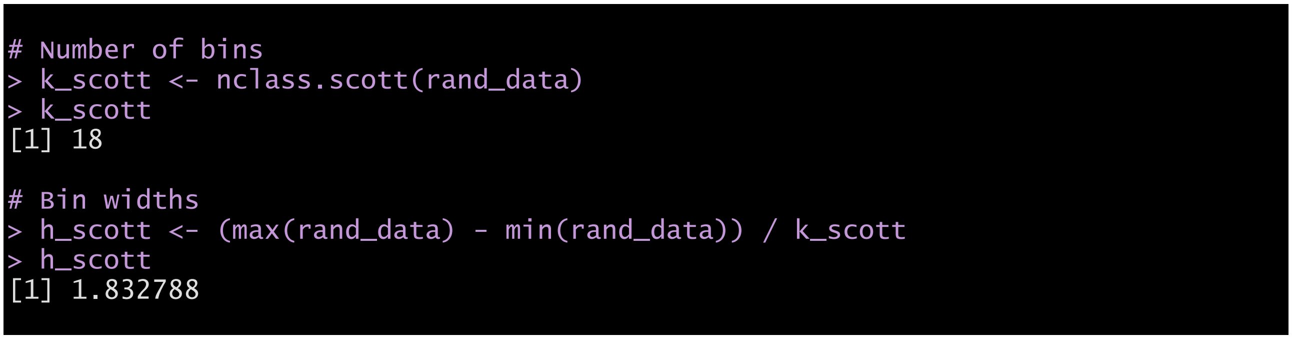
Compute the bin widths (h) and number of bins (k) according to each of the following formula:

i. Sturges’ formula



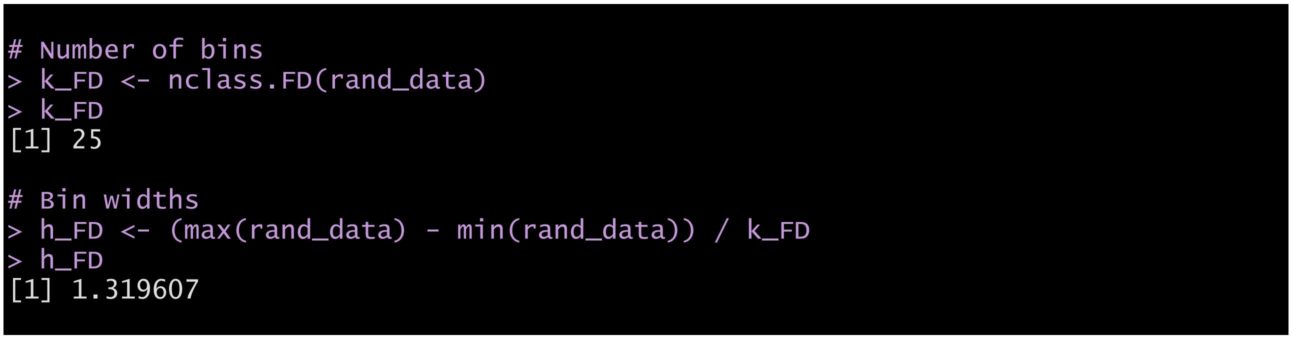
```
# Number of bins
> k_Sturges <- nclass.Sturges(rand_data)
> k_Sturges
[1] 11

# Bin widths
> h_Sturges <- (max(rand_data) - min(rand_data)) / k_Sturges
> h_Sturges
[1] 2.999107
```

ii. Scott’s normal reference rule (uses standard deviation)



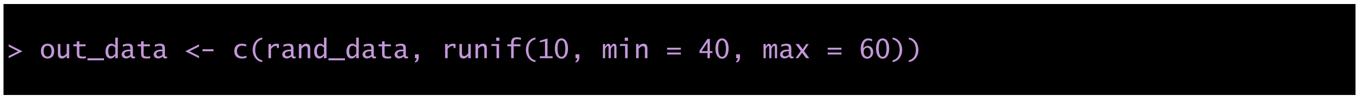
```
# Number of bins
> k_scott <- nclass.scott(rand_data)
> k_scott
[1] 18

# Bin widths
> h_scott <- (max(rand_data) - min(rand_data)) / k_scott
> h_scott
[1] 1.832788
```

iii. Freedman – Diaconis’ choice (uses IQR)



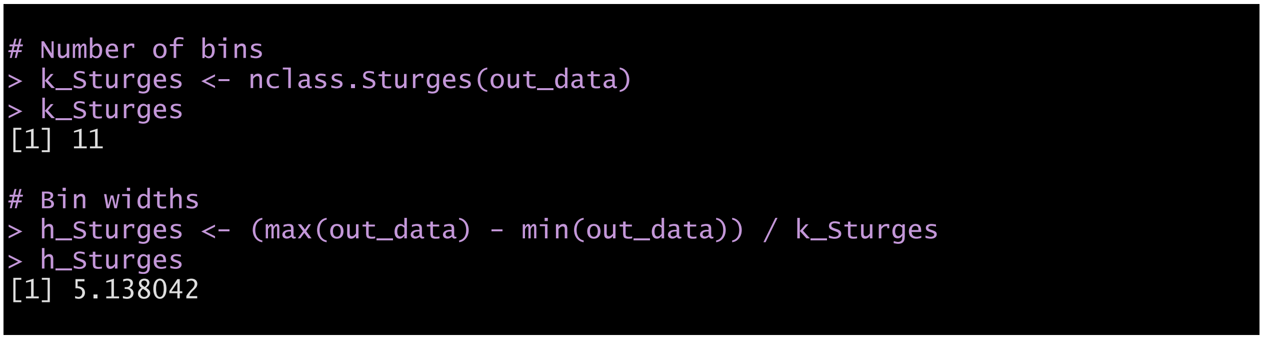
```
# Number of bins
> k_FD <- nclass.FD(rand_data)
> k_FD
[1] 25

# Bin widths
> h_FD <- (max(rand_data) - min(rand_data)) / k_FD
> h_FD
[1] 1.319607
```

c. Repeat part (b) but extend the rand_data dataset with some outliers (use a new dataset out_data):



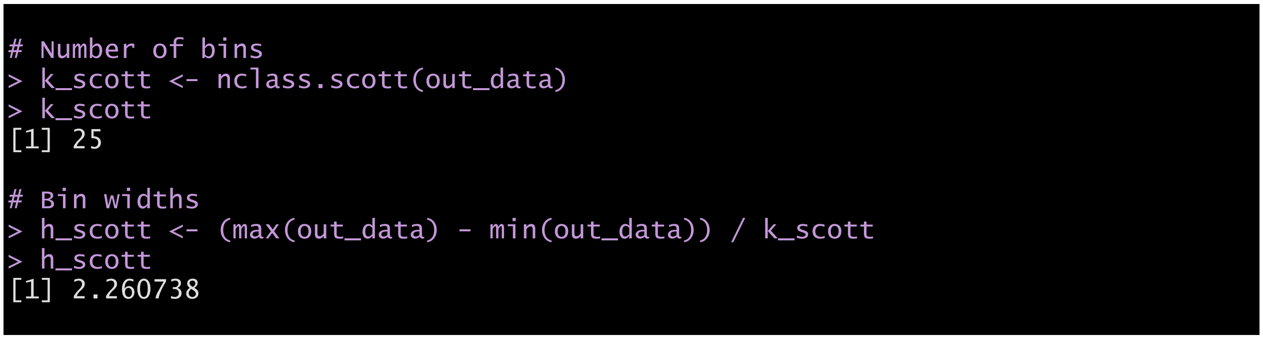
```
> out_data <- c(rand_data, runif(10, min = 40, max = 60))
```

i. Sturges’ formula



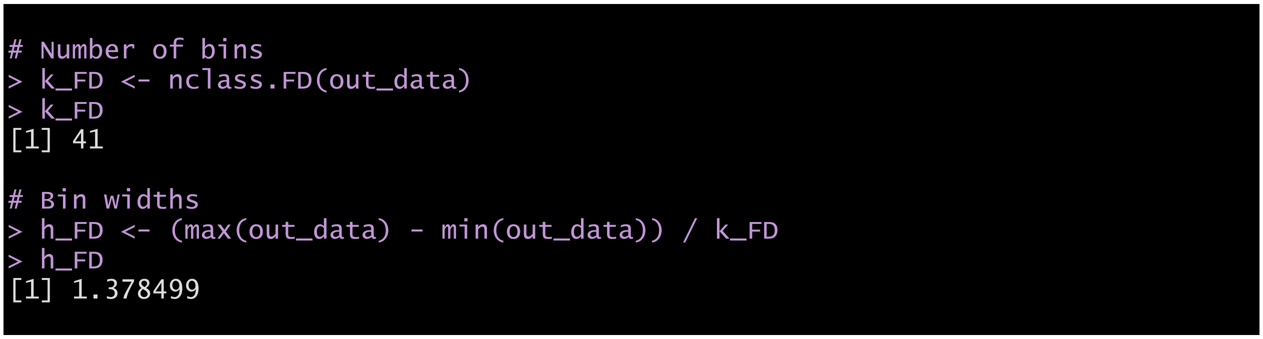
```
# Number of bins
> k_Sturges <- nclass.Sturges(out_data)
> k_Sturges
[1] 11

# Bin widths
> h_Sturges <- (max(out_data) - min(out_data)) / k_Sturges
> h_Sturges
[1] 5.138042
```

ii. Scott’s normal reference rule (uses standard deviation)



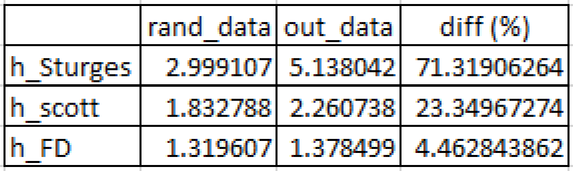
```
# Number of bins
> k_scott <- nclass.scott(out_data)
> k_scott
[1] 25

# Bin widths
> h_scott <- (max(out_data) - min(out_data)) / k_scott
> h_scott
[1] 2.260738
```

iii. Freedman – Diaconis’ choice (uses IQR)



```
# Number of bins
> k_FD <- nclass.FD(out_data)
> k_FD
[1] 41

# Bin widths
> h_FD <- (max(out_data) - min(out_data)) / k_FD
> h_FD
[1] 1.378499
```

d. From your answers above, in which of the three methods does the bin width (h) change the least when outliers are added (i.e., which is least sensitive to outliers), and (briefly) WHY do you think that is?



	rand_data	out_data	diff (%)
h_Sturges	2.999107	5.138042	71.31906264
h_scott	1.832788	2.260738	23.34967274
h_FD	1.319607	1.378499	4.462843862

- In Sturges’ formula, since it is affected so much by the range of the data, and in this case, the range of the data increased dramatically, so the bin width will also increase dramatically.

- In Scott’s normal reference rule, if the standard deviation is increased by 1, the bin width is increased by 3.5, all else equal.

  The addition of 10 outliers makes the standard deviation increased dramatically, so the bin width also increased dramatically.

- In Freedman – Diaconis’ choice, considering it uses the IQR (Inter-Quartile Range), the addition of 10 outliers to 800 samples will not make the 1st quartile and 3rd quartile shifted too much, so that the IQR will not change dramatically, thus the bin width will just change a little bit also.