BACS - HW2

109006241

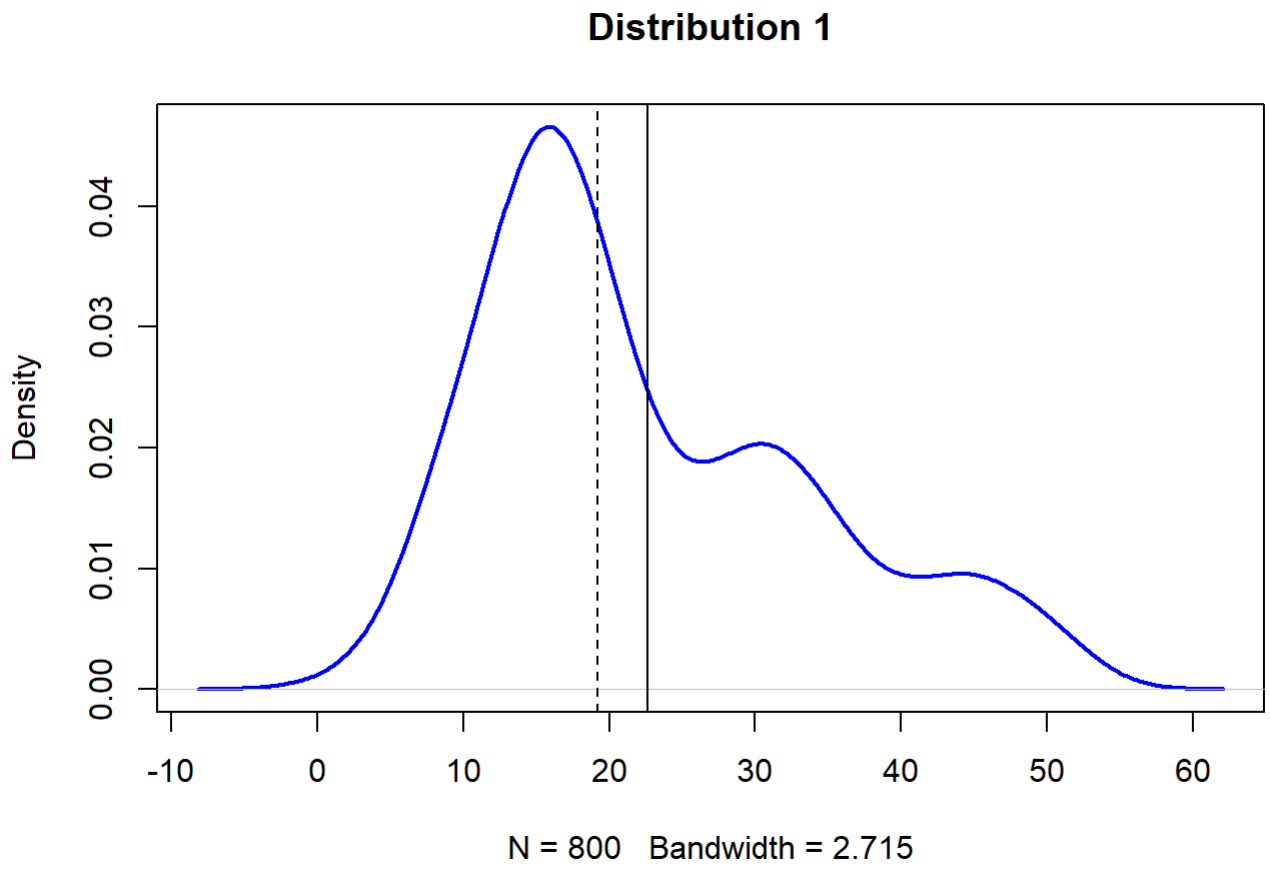
Question 1

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

# Combining them into a composite dataset
d123 <- c(d1, d2, d3)

# Let's plot the density function of d123
plot(density(d123), col="blue", lwd=2,
     main = "Distribution 1")

# Add vertical lines showing mean and median
abline(v=mean(d123))
abline(v=median(d123), lty="dashed")
```



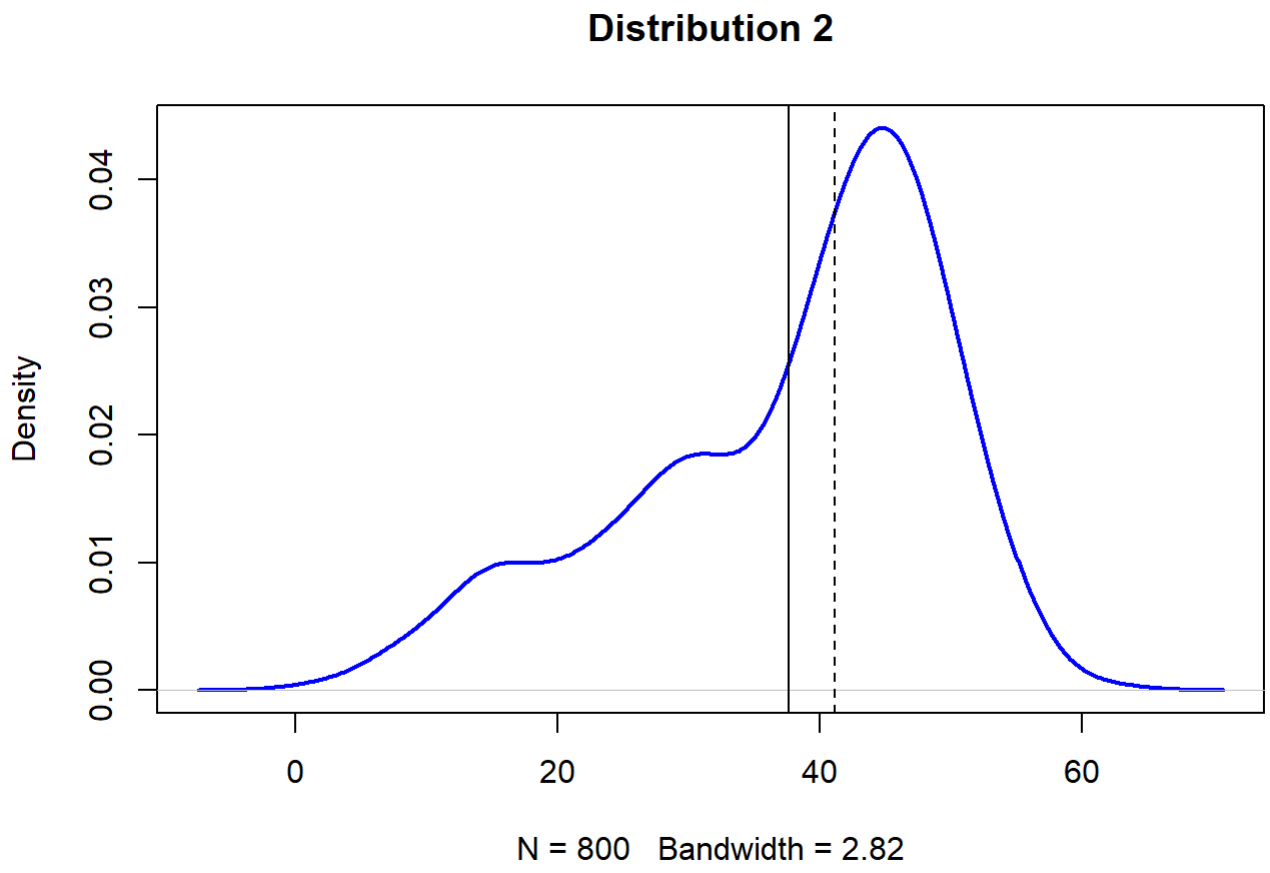
a. Create and visualize a new “Distribution 2”: a combined dataset (n=800) that is negatively skewed (tail stretches to the left). Change the mean and standard deviation of d1, d2, and d3 to achieve this new distribution. Compute the mean and median, and draw lines showing the mean (thick line) and median (thin line).

```
d1 = rnorm(n=100, mean=15, sd=5)
d2 = rnorm(n=200, mean=30, sd=5)
d3 = rnorm(n=500, mean=45, sd=5)

d123 = c(d1, d2, d3)

plot(density(d123), col="blue", lwd=2,
     main = "Distribution 2")

abline(v=mean(d123))
abline(v=median(d123), lty="dashed")
```

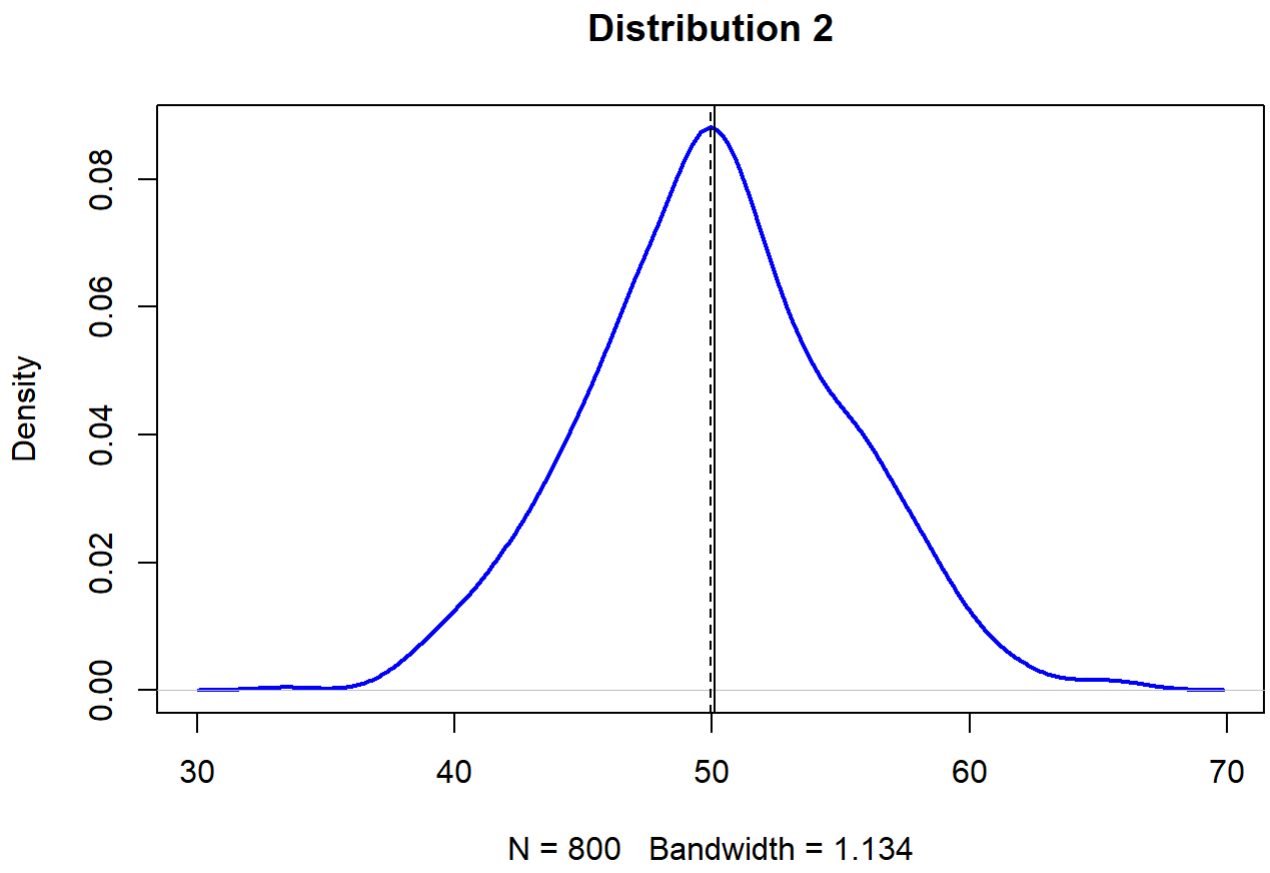


b. Create a “Distribution 3”: a single dataset that is normally distributed (bell-shaped, symmetric) – you do not need to combine datasets, just use the rnorm() function to create a single large dataset (n=800). Show your code, compute the mean and median, and draw lines showing the mean (thick line)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

```
d = rnorm(n=800, mean=50, sd=5)

plot(density(d), col="blue", lwd=2,
     main = "Distribution 2")

abline(v=mean(d))
abline(v=median(d), lty="dashed")
```



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?

> The mean value will be more sensitive towards outliers on our data.

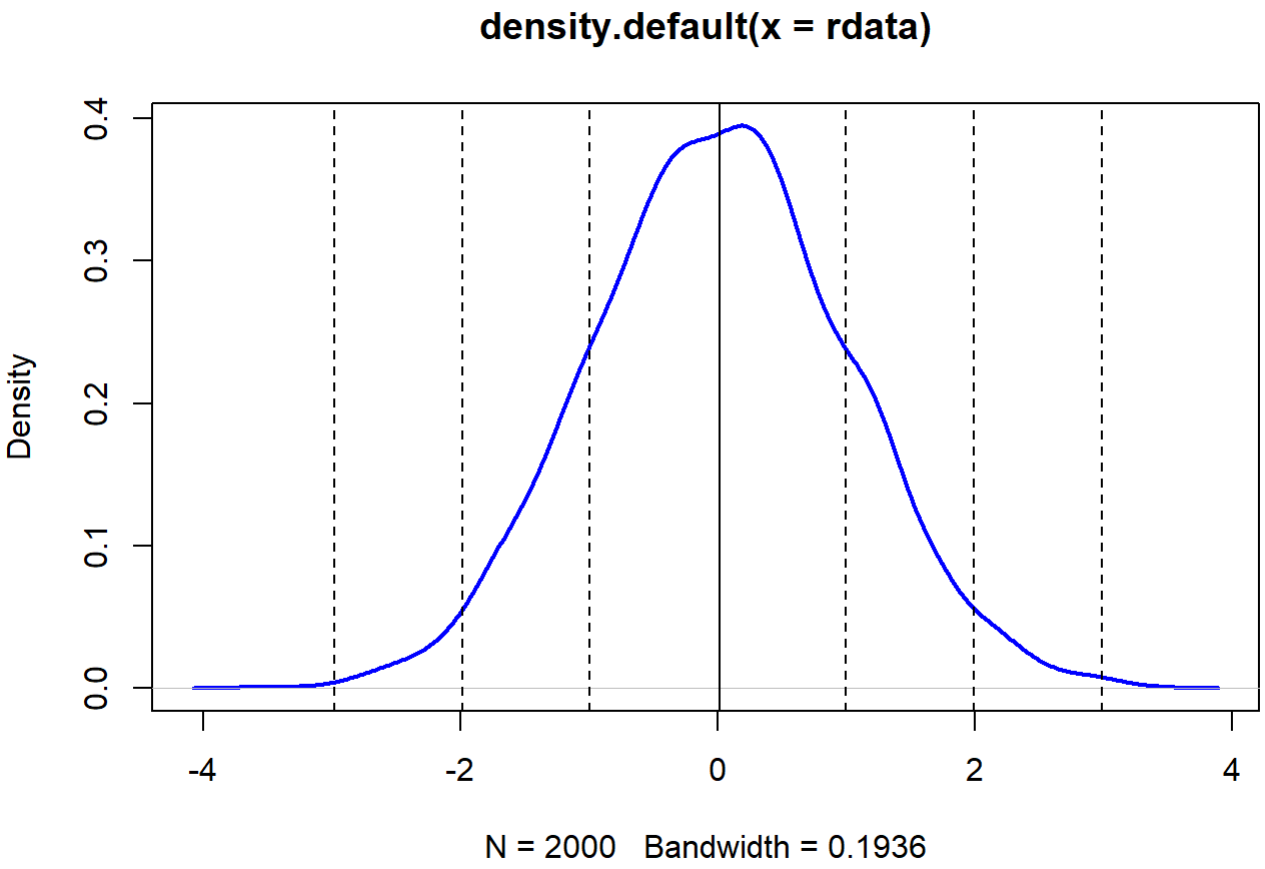
Question 2

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 to the left and right of the mean. You should have a total of 7 vertical lines (one solid, six dashed).

```
rdata = rnorm(n=2000, mean=0, sd=1)

plot(density(rdata), col="blue", lwd=2)

abline(v=mean(rdata))
sd = sd(rdata)
abline(v=sd, lty="dashed")
abline(v=-sd, lty="dashed")
abline(v=2*sd, lty="dashed")
abline(v=-2*sd, lty="dashed")
abline(v=3*sd, lty="dashed")
abline(v=-3*sd, lty="dashed")
```



b. Using the quantile() function, which data points correspond to the 1st, 2nd, and 3rd quartiles (i.e., 25th, 50th, 75th percentiles) of rdata? How many standard deviations away from the mean (divide by standard-deviation; keep positive or negative sign) are those points corresponding to the 1st, 2nd, and 3rd quartiles?

```
quantiles = quantile(rdata, c(0.25, 0.5, 0.75))
cat(" 1st quartile:", quantiles[1], "\n",
     "2nd quartile:", quantiles[2], "\n",
     "3rd quartile:", quantiles[3])
```

```
##  1st quartile: -0.6550485
##  2nd quartile: 0.01378465
##  3rd quartile: 0.6634514
```

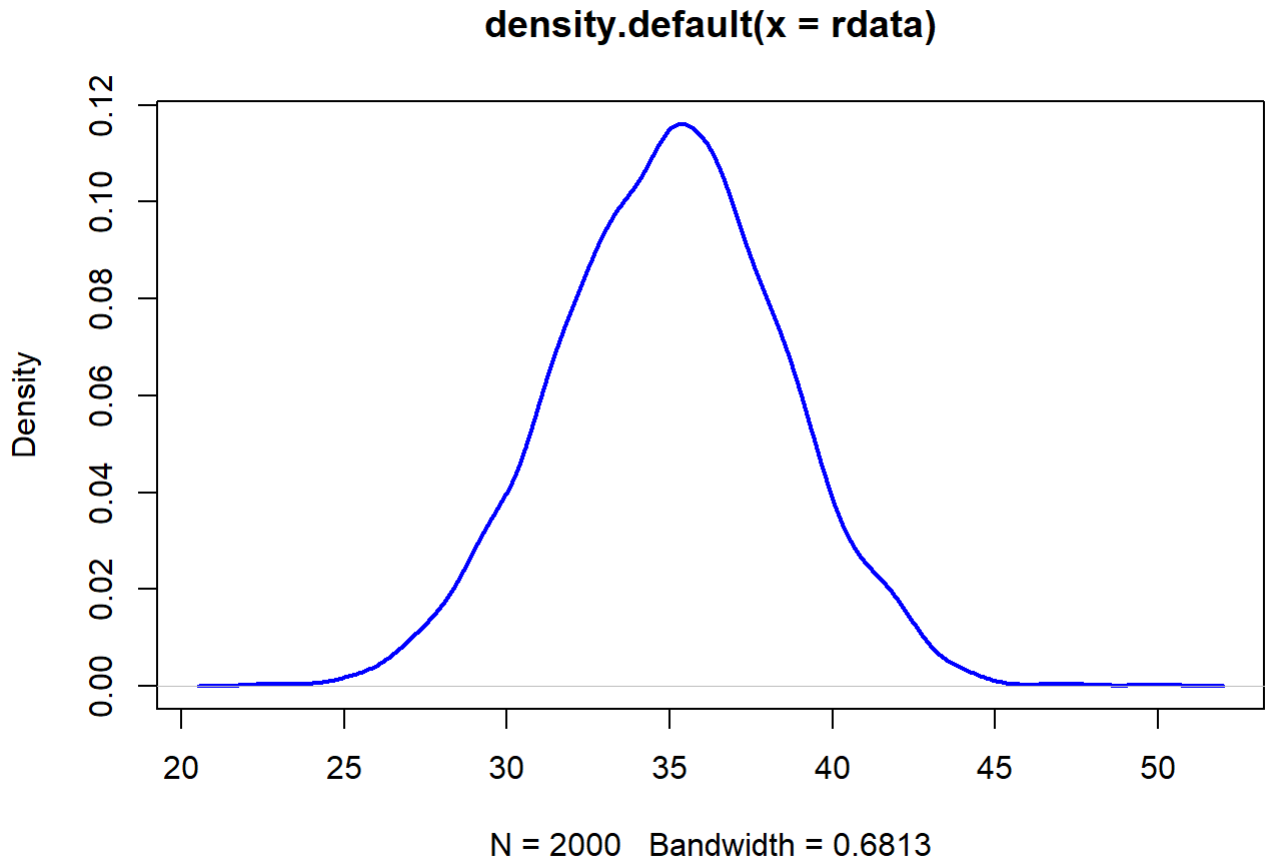
```
quantiles_dist = (quantiles - mean(rdata)) / sd(rdata)
cat(" Distance from the mean corresponding to the 1st quartile:", quantiles_dist[1], "standard d
eviations\n",
     "Distance from the mean corresponding to the 2nd quartile:", quantiles_dist[2], "standard d
eviations\n",
     "Distance from the mean corresponding to the 3rd quartile:", quantiles_dist[3], "standard d
eviations")
```

```
##  Distance from the mean corresponding to the 1st quartile: -0.6713654 standard deviations
##  Distance from the mean corresponding to the 2nd quartile: 0.001043788 standard deviations
##  Distance from the mean corresponding to the 3rd quartile: 0.6541841 standard deviations
```

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)

```
rdata = rnorm(n=2000, mean=35, sd=3.5)

plot(density(rdata), col="blue", lwd=2)
```



```
quantiles = quantile(rdata, c(0.25, 0.75))
cat(" 1st quartile:", quantiles[1], "\n",
     "3rd quartile:", quantiles[2])
```

```
##  1st quartile: 32.6713
##  3rd quartile: 37.32635
```

```
quantiles_dist = (quantiles - mean(rdata)) / sd(rdata)
cat(" Distance from the mean corresponding to the 1st quartile:", quantiles_dist[1], "standard d
eviations\n",
     "Distance from the mean corresponding to the 3rd quartile:", quantiles_dist[2], "standard d
eviations")
```

```
##  Distance from the mean corresponding to the 1st quartile: -0.6759593 standard deviations
##  Distance from the mean corresponding to the 3rd quartile: 0.6687327 standard deviations
```

> The distances have a small difference, compared to the distances in (b)

d. Finally, recall the dataset d123 shown in the description of 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)

```
quantiles = quantile(d123, c(0.25, 0.75))
cat(" 1st quartile:", quantiles[1], "\n",
     "3rd quartile:", quantiles[2])
```

```
##  1st quartile: 29.9476
##  3rd quartile: 46.29831
```

```
quantiles_dist = (quantiles - mean(d123)) / sd(d123)
cat(" Distance from the mean corresponding to the 1st quartile:", quantiles_dist[1], "standard d
eviations\n",
     "Distance from the mean corresponding to the 3rd quartile:", quantiles_dist[2], "standard d
eviations")
```

```
##  Distance from the mean corresponding to the 1st quartile: -0.6433949 standard deviations
##  Distance from the mean corresponding to the 3rd quartile: 0.727269 standard deviations
```

> The distance corresponding to the 1st quartile has a small difference, while the distance corresponding to the 3rd quartile is slightly farther away from the mean, when compared to the distances in (b)

Question 3

a. From the question on the forum, which formula does Rob Hyndman's answer (1st answer) suggest to use for bin widths/number? Also, what does the Wikipedia article say is the benefit of that formula?

> He suggests to use the Freedman-Diaconsis rule. According to Wikipedia, the benefit of this formula is that this formula is less sensitive towards outliers in the data, because it uses IQR to compute the optimal bin width value, instead of standard deviation.

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
ii. Scott's normal reference rule (uses standard deviation)
iii. Freedman-Diaconis' choice (uses IQR)

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

n = length(rand_data)

# Sturges' formula
k = ceiling(log2(n)) + 1
h = (max(rand_data) - min(rand_data)) / k
cat("Bin Width:", h, ", Number of Bins:", k)
```

```
## Bin Width: 2.799136 , Number of Bins: 11
```

```
# Scott's normal reference rule
sd = sd(rand_data)
h = 3.49*sd / n^(1/3)
k = ceiling((max(rand_data) - min(rand_data)) / h)
cat("Bin Width:", h, ", Number of Bins:", k)
```

```
## Bin Width: 1.874852 , Number of Bins: 17
```

```
# Freedman-Diaconis' choice
h = 2*IQR(rand_data) / n^(1/3)
k = ceiling((max(rand_data) - min(rand_data)) / h)
cat("Bin Width:", h, ", Number of Bins:", k)
```

```
## Bin Width: 1.455798 , Number of Bins: 22
```

c. Repeat part (b) but let's extend rand_data dataset with some outliers (creating a new dataset out_data):

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

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?

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

n = length(out_data)

# Sturges' formula
k = ceiling(log2(n)) + 1
h = (max(out_data) - min(out_data)) / k
cat("Bin Width:", h, ", Number of Bins:", k)
```

```
## Bin Width: 4.996658 , Number of Bins: 11
```

```
# Scott's normal reference rule
sd = sd(out_data)
h = 3.49*sd / n^(1/3)
k = ceiling((max(out_data) - min(out_data)) / h)
cat("Bin Width:", h, ", Number of Bins:", k)
```

```
## Bin Width: 2.242084 , Number of Bins: 25
```

```
# Freedman-Diaconis' choice
h = 2*IQR(out_data) / n^(1/3)
k = ceiling((max(out_data) - min(out_data)) / h)
cat("Bin Width:", h, ", Number of Bins:", k)
```

```
## Bin Width: 1.470287 , Number of Bins: 38
```

Using the Freedman-Diaconis' formula, the bin width (h) changes the least when outliers are added. I think this is because the formula uses IQR (interquartile range), which is not that sensitive towards outliers compared to standard deviation and range, while computing the bin width value.