HW_Week2_108020033

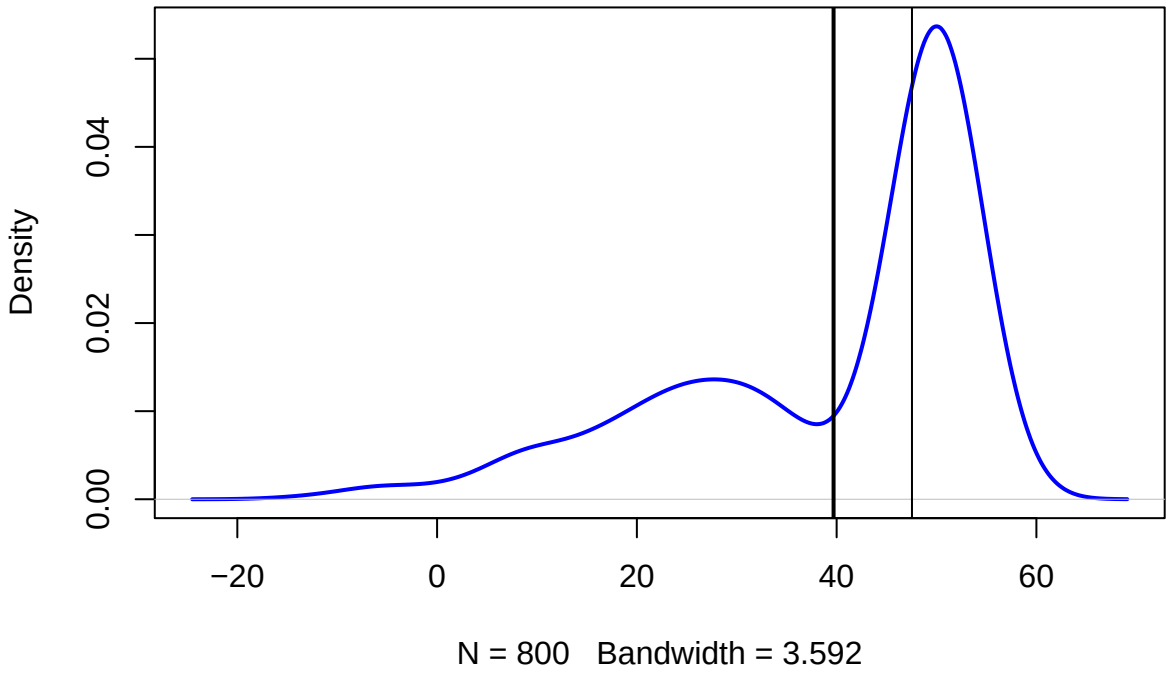
Che-Wei, Chang

2023-03-05

Question 1

```
library(moments)
# Three normally distributed data sets
d1 <- rnorm(n = 500, mean = 50, sd = 3)
d2 <- rnorm(n = 200, mean = 28, sd = 7)
d3 <- rnorm(n = 100, mean = 12, sd = 10)

# 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 2")

# Check skewness
skew <- skewness(d123)

# Add vertical lines showing mean and median
abline(v = mean(d123), lwd = 2)
abline(v = median(d123), lwd = 1)
```

(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).




```
# show the mean and median of the dataset d123
mean(d123)
```

```
## [1] 39.69197
```

```
median(d123)
```

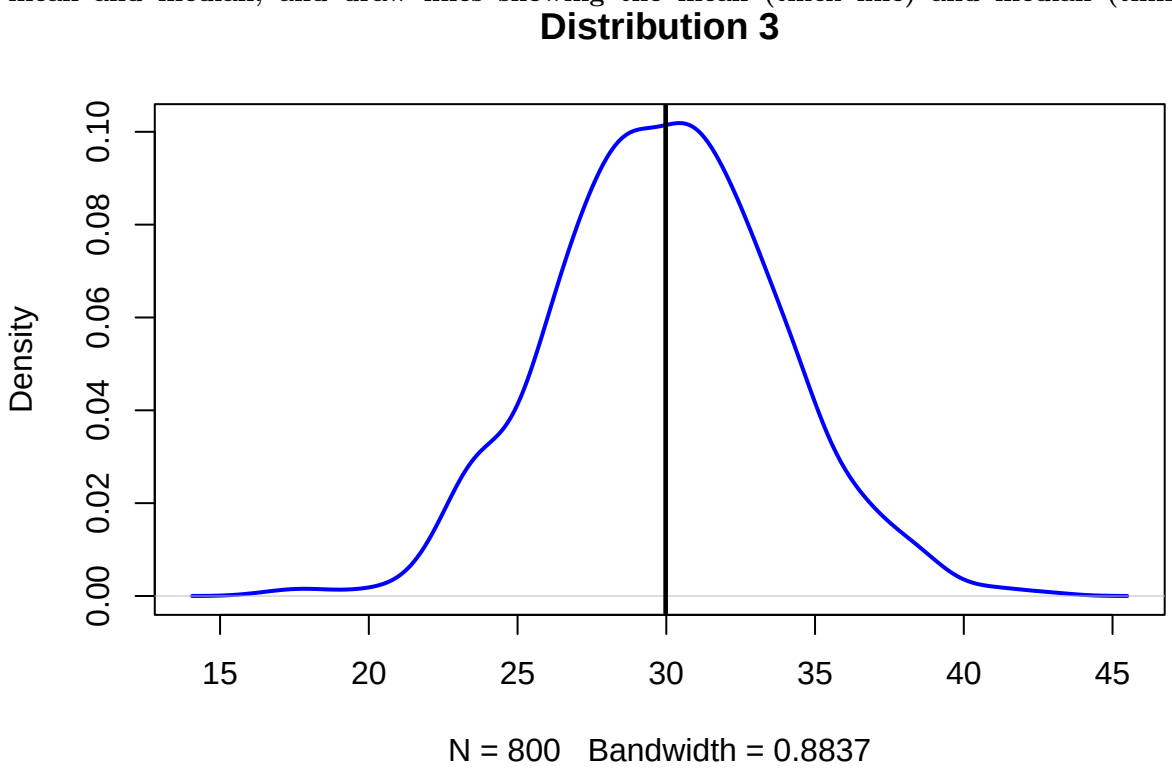
```
## [1] 47.54221
```

We create the data set d123 composed of d1, d2, d3 and use skewness to check the data set is negatively skewed. The values above are the mean and median, respectively.

```
# Create the data set
data <- rnorm(n = 800, mean = 30, sd = 4)

# Let's plot the density function of data
plot(density(data), col = "blue", lwd = 2,
     main = "Distribution 3")

# Add vertical lines showing mean and median
abline(v = mean(data), lwd = 2)
abline(v = median(data), lwd = 1)
```

(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).




```
# show the mean and median of the data
mean(data)
```

```
## [1] 29.96766
```

```
median(data)
```

```
## [1] 30.02152
```

We use rnorm() to create the data set and set mean = 30, sd = 4, and then, we draw a plot and show the median and mean on it. The values above are the mean and median, respectively.

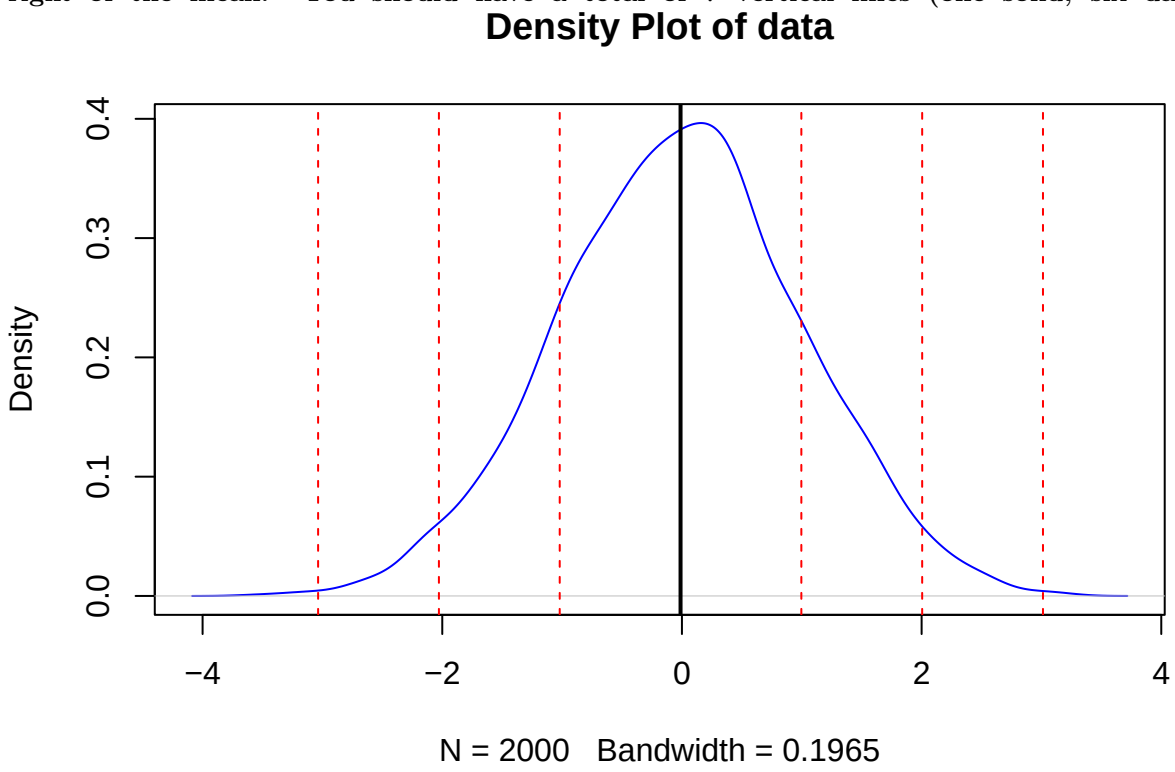
(c) In general, which measure of central tendency (mean or median) do you think will be more sensitive (will change more) to outliers being added to your data? When the outliers are significantly distant from the other data, the mean value will be greatly affected, whereas the median value will only be slightly influenced. In my opinion, the mean value is more sensitive to outliers.

Question 2

```
# Create data set
rdata <- rnorm(n = 2000, mean = 0, sd = 1)
```

```r
plot(density(rdata), col = "blue", main = "Density Plot of data")
abline(v = mean(rdata), col = "black", lwd = 2)
abline(v = c(mean(rdata) - sd(rdata), mean(rdata) - 2 * sd(rdata), mean(rdata) - 3 * sd(rdata),
             mean(rdata) + sd(rdata), mean(rdata) + 2 * sd(rdata), mean(rdata) + 3 * sd(rdata)), col =
```

(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).



We use rnorm() to create the data set and draw a density plot and show the line on mean, 1st, 2nd, and 3rd standard deviation on it.

```r
# Create variables to store mean and standard deviation
Part_b_mean <- mean(data)
Part_b_sd <- sd(data)

# Create variables to store 1st, 2nd, and 3rd quantiles
first_quantile <- unname(quantile(data, 0.25))
second_quantile <- unname(quantile(data, 0.5))
third_quantile <- unname(quantile(data, 0.75))

# Calculate the result
(first_quantile - Part_b_mean) / Part_b_sd
```

(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?

```
## [1] -0.660503
```

```
(second_quantile - Part_b_mean) / Part_b_sd
```

```
## [1] 0.01423561
```

```
(third_quantile - Part_b_mean) / Part_b_sd
```

```
## [1] 0.6633489
```

We declare some variables to store mean, standard deviation, 1st, 2nd, and 3rd quantiles. Then, we let these quantiles minus mean and divide standard deviation to get the result.The values above are 1st, 2nd, and 3rd quantiles minus mean and divide standard deviation, respectively.

```
# Generate the new data set
new_data_set <- rnorm(n = 2000, mean = 35, sd = 3.5)

# Create variables to store mean and standard deviation
Part_c_mean <- mean(new_data_set)
Part_c_sd <- sd(new_data_set)

# Create variables to store 1st and 3rd quantiles
first_quantile <- unname(quantile(new_data_set, 0.25))
third_quantile <- unname(quantile(new_data_set, 0.75))

# Calculate the result
(first_quantile - Part_c_mean) / Part_c_sd
```

(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)

```
## [1] -0.6520339
```

```
(third_quantile - Part_c_mean) / Part_c_sd
```

```
## [1] 0.659023
```

We create the new data set and declare some variables to store the information. Then, we let 1st and 3rd quantiles minus mean and divide standard deviation to get the result. The values above are 1st and 3rd quantiles minus mean and divide standard deviation, respectively. Compare the answer to (b), we can see that it doesn't change greatly, it is because the process that we let these quantiles minus mean and divide the standard deviation call normalization in Statistics and they all distributed in normal distribution. Therefore, the result in part (c) is approximately same as the result in part (b).

```r
# Create the data set in the description of question 1
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)

# Create variables to store mean and standard deviation
Part_d_mean <- mean(d123)
Part_d_sd <- sd(d123)

# Create variables to store 1st and 3rd quantiles
first_quantile <- unname(quantile(d123, 0.25))
third_quantile <- unname(quantile(d123, 0.75))

# Calculate the result
(first_quantile - Part_d_mean) / Part_d_sd
```

(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)

```
## [1] -0.7445963
```

```r
(third_quantile - Part_d_mean) / Part_d_sd
```

```
## [1] 0.6690215
```

We use the new data set at part a of Question1 and declare some variables to store the information.We let 1st and 3rd quantiles minus mean and divide standard deviation to get the result. The values above are 1st and 3rd quantiles minus mean and divide standard deviation, respectively. Compare the answer to part (b), the result is a little different. Since the the distribution in the description of question 1 is positively skewed, it doesn't distribute with normal distribution.Therefore, the result in part (d) is different to the result in part (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? Rob Hyndman suggests that Freedman–Diaconis' choice is robust and works well in practice. Because it replaces 3.49 of Scott's normal reference rule with 2*IQR, it is less sensitive than the standard deviation to outliers in data. Therefore, the bin width won't change greatly.

```r
# Create data set rand_data
rand_data <- rnorm(800, mean = 20, sd = 5)
```

```r
# Sturges' formula
k1 <- ceiling(log2(800)) + 1
h1 <- (max(rand_data) - min(rand_data)) / k1

# Scott's normal reference rule
h2 <- 3.49 * sd(rand_data) / (800 ^ (1 / 3))
k2 <- ceiling((max(rand_data) - min(rand_data)) / h2)

# Freedman-Diaconis'choice
q1 <- unname(quantile(rand_data, 0.25))
q3 <- unname(quantile(rand_data, 0.75))
IQR <- q3 - q1
h3 <- 2 * IQR / (800 ^ (1 / 3))
k3 <- ceiling((max(rand_data) - min(rand_data)) / h3)

# Put the result into the vector
result_of_h_part_b <- c(h1, h2, h3)
result_of_k_part_b <- c(k1, k2, k3)
result_of_h_part_b
```

(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)

```
## [1] 2.728588 1.887920 1.470055
```

```r
result_of_k_part_b
```

```
## [1] 11 16 21
```

We create the data set rand_data and calculate the number of bins and bin width in three different methods.The values above are the bin width calculated using Sturges' formula, Scott's normal reference rule, and Freedman-Diaconis' choice and number of bins calculated using Sturges' formula, Scott's normal reference rule, and Freedman-Diaconis' choice, respectively.

```r
# Create data set out_data
out_data <- c(rand_data, runif(10, min = 40, max = 60))

# Sturges' formula
k1 <- ceiling(log2(810)) + 1
h1 <- (max(out_data) - min(out_data)) / k1

# Scott's normal reference rule
h2 <- 3.49 * sd(out_data) / (810 ^(1 / 3))
k2 <- ceiling(max(out_data) - min(out_data) / h2)

# Freedman-Diaconis'choice
q1 <- unname(quantile(out_data, 0.25))
```

```
q3 <- unname(quantile(out_data, 0.75))
IQR <- q3 - q1
h3 <- 2 * IQR / (810 ^ (1 / 3))
k3 <- ceiling((max(out_data) - min(out_data)) / h3)

# Put the result into the vector
result_of_h_part_c <- c(h1, h2, h3)
result_of_k_part_c <- c(k1, k2, k3)
result_of_h_part_c
```

(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?

```
## [1] 4.814696 2.258676 1.491205
```

```
result_of_k_part_c
```

```
## [1] 11 56 36
```

```
minus_result <- abs(result_of_h_part_b - result_of_h_part_c)
```

We add some outliers into the data and calculate the result.The values above are the bin width (h) calculated using Sturges' formula, Scott's normal reference rule, and Freedman-Diaconis' choice and number of bins (k) calculated using Sturges' formula, Scott's normal reference rule, and Freedman-Diaconis' choice, respectively. I use minus_result to store the h in part_b minus the h in part_c to check which method get the least value. I find that Freedman-Diaconis'choice change the least. Because the formula of bin width in Freedman-Diaconis'choice is $\frac{2*IQR}{\sqrt[3]{n}}$, when we add outliers in the data, it just changes slightly. Whereas the formula of bin width in Sturges' formula and Scott's normal reference rule is $\frac{max(x)-min(x)}{k}$ and $\frac{3.49\hat{\sigma}}{\sqrt[3]{n}}$, so when we add outliers into the data, $\hat{\sigma}$ and max(x) - min(x) will greatly affected. Therefore, this is the idea that I think bin width in Freedman-Diaconis'choice will change the least.