HW2

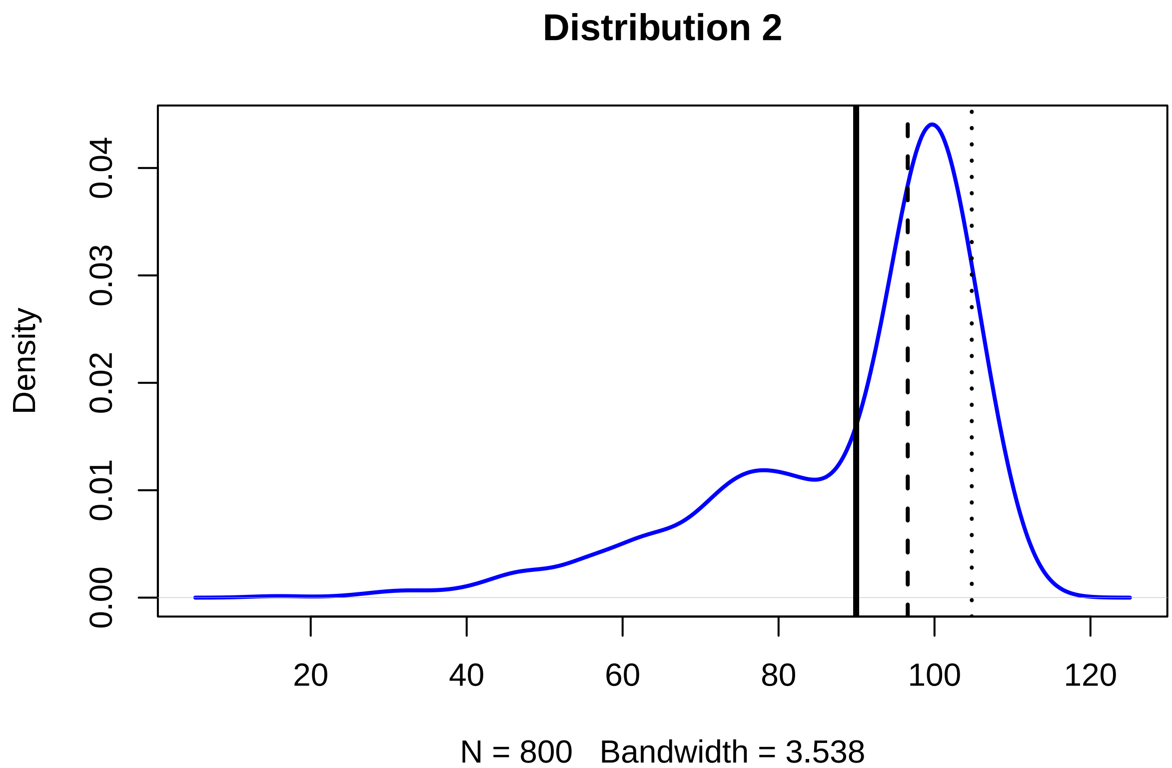
108048110

2/24/2022

Question 1)

(a) Create and visualize a new **"Distribution 2"**: a combined dataset (n=800) that is negatively skewed. Change the mean and standard deviation of d1, d2, and d3 to achieve this new distribution. mean=**solid**, median=**dashed**, and I write a mode function myself, which is indicated through **dotted** line in the diagram.

```
Mode <- function(x){
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]}
d1 <- rnorm(n=500, mean=100, sd=5)
d2 <- rnorm(n=200, mean=80, sd=10)
d3 <- rnorm(n=100, mean=60, sd=15)
d123 <- c(d1, d2, d3)
plot(density(d123), col="blue", lwd=2, main="Distribution 2")
abline(v=mean(d123), lwd=3)
abline(v=median(d123), lty="dashed", lwd=2)
abline(v=Mode(d123), lty="dotted", lwd=2)
```

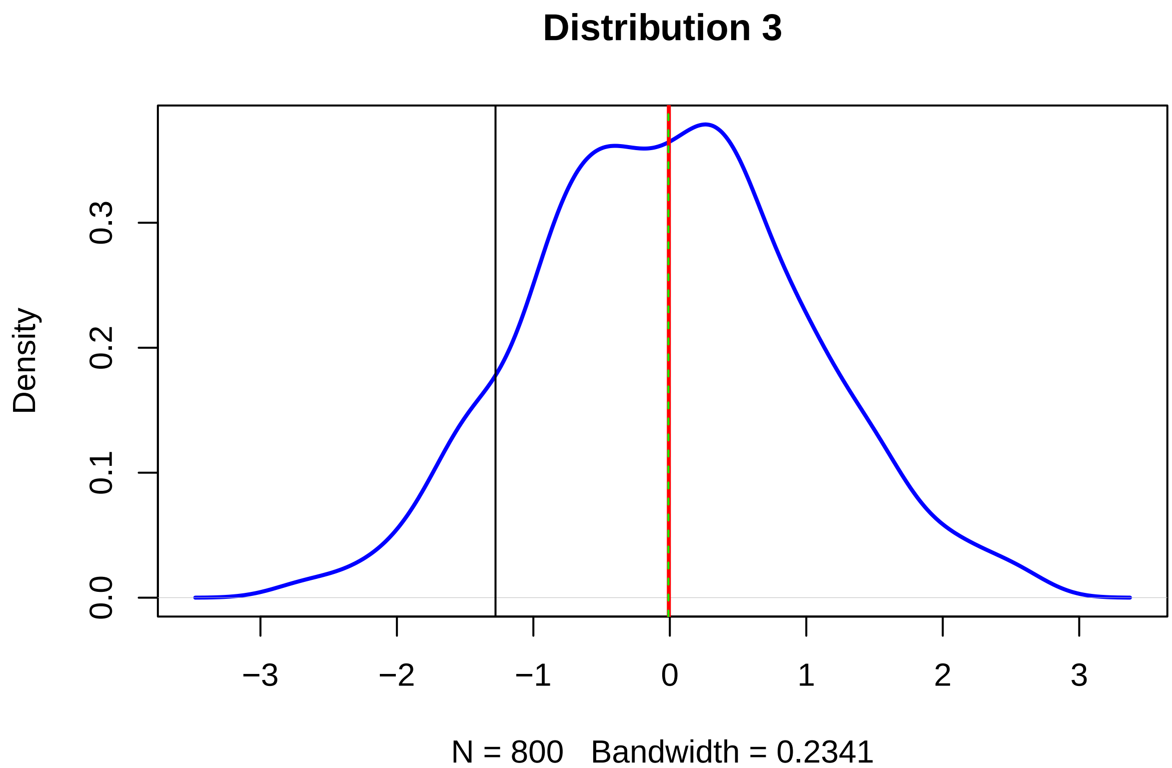



```
paste("Mean: ", mean(d123), "; Median: ", median(d123), "; Mode: ", Mode(d123))
```

```
## [1] "Mean:  89.9514582667222 ; Median:  96.5650488285323 ; Mode:  104.777193184465"
```

(b) Create a **"Distribution 3"**, a single dataset that is normally distributed (bell-shaped, symmetric), using 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 (red line) ,median (green line) and mode (black line).

```
datasets <- rnorm(n=800)
plot(density(datasets), main="Distribution 3", lwd=2, col="blue")
abline(v=mean(datasets), lwd=2, lty="solid", col="red")
abline(v=median(datasets), lwd=1, lty="dashed", col="green")
abline(v = Mode(datasets), lwd=1, col="black")
```




```
paste("Mean: ", mean(datasets), "; Median: ", median(datasets), "; Mode: ", Mode(datasets))
```

```
## [1] "Mean:  -0.00726347938283763 ; Median:  -0.01060807676938 ; Mode:  -1.27748449904778"
```

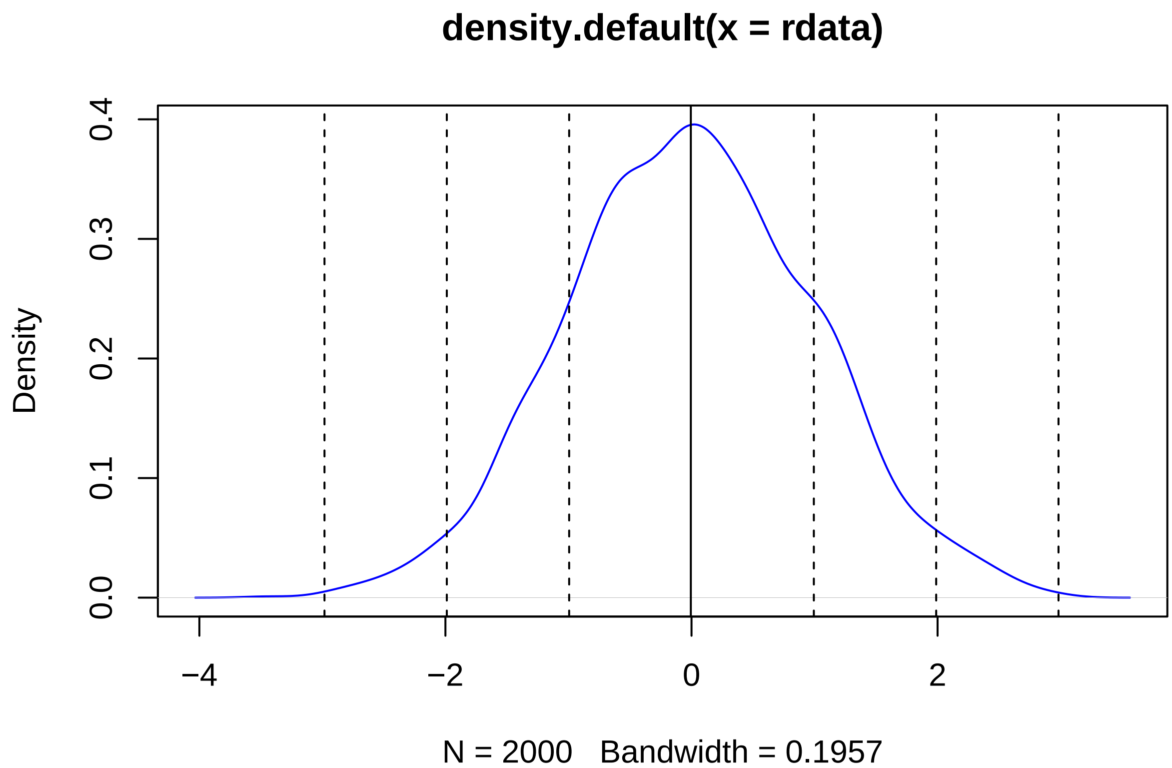
(c) In general, which measure of central tendency (mean or median) do you think will be more sensitive (will change more) to outliers being added to your data?

Ans. Mean is more sensitive to outliers being added to a dataset. As the example professor gave in class, mean is the average of the entire dataset, while median is the middle position among the dataset; thus value of outliers affect mean more than median.

Question 2)

(a) Create a random dataset (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")
abline(v=mean(rdata), lty="solid")
abline(v=c(sd(rdata), -sd(rdata)), lty="dashed")
abline(v=c(sd(rdata)*2, -sd(rdata)*2), lty="dashed")
abline(v=c(sd(rdata)*3, -sd(rdata)*3), lty="dashed")
```



(b) Using the `quantile()` function, which data points correspond to the 1st, 2nd, and 3rd quartile? How many standard deviations away from the mean are those points corresponding to the 1st, 2nd, and 3rd quartiles?

```
summary(rdata)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -3.446335 -0.677576 -0.011892 -0.005499  0.669648  2.976799
```

```
Q1 <- quantile(rdata, 1/4)
Q2 <- quantile(rdata, 2/4) # Q2 is the same as the ordinary median.
Q3 <- quantile(rdata, 3/4)
Q4 <- quantile(rdata, 4/4)
iqr <- IQR(rdata)
```

```
unname(Q1/sd(rdata))
```

```
## [1] -0.681431
```

```
unname(Q2/sd(rdata))
```

```
## [1] -0.01195984
```

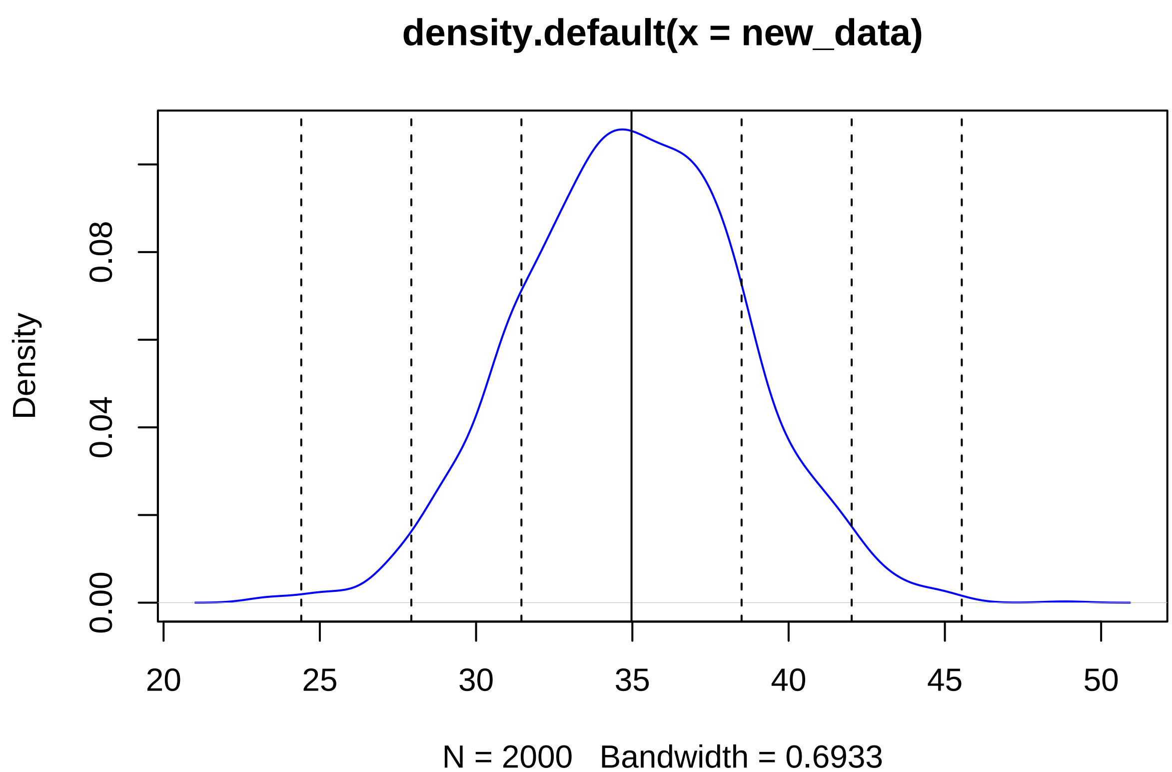
```
unname(Q3/sd(rdata))
```

```
## [1] 0.6734584
```

NOTE. As we can see from the above, the values of Q1, Q2 and Q3 are very close to the corresponding the 1st, 2nd, and 3rd quartiles.

(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 are those points corresponding to the 1st and 3rd quartiles? Compare your answer to **(b)**.

```
new_data <- rnorm(n=2000, mean=35, sd=3.5)
plot(density(new_data), col="blue")
new_mean <- mean(new_data)
abline(v=mean(new_data), lty="solid")
abline(v=c(new_mean-sd(new_data), new_mean+sd(new_data)), lty="dashed")
abline(v=c(new_mean-sd(new_data)*2, new_mean+sd(new_data)*2), lty="dashed")
abline(v=c(new_mean-sd(new_data)*3, new_mean+sd(new_data)*3), lty="dashed")
```




```
summary(new_data)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   23.09   32.56   34.98   34.97   37.37   48.84
```

```
unname((quantile(new_data, 1/4)-mean(new_data))/sd(new_data))
```

```
## [1] -0.6860774
```

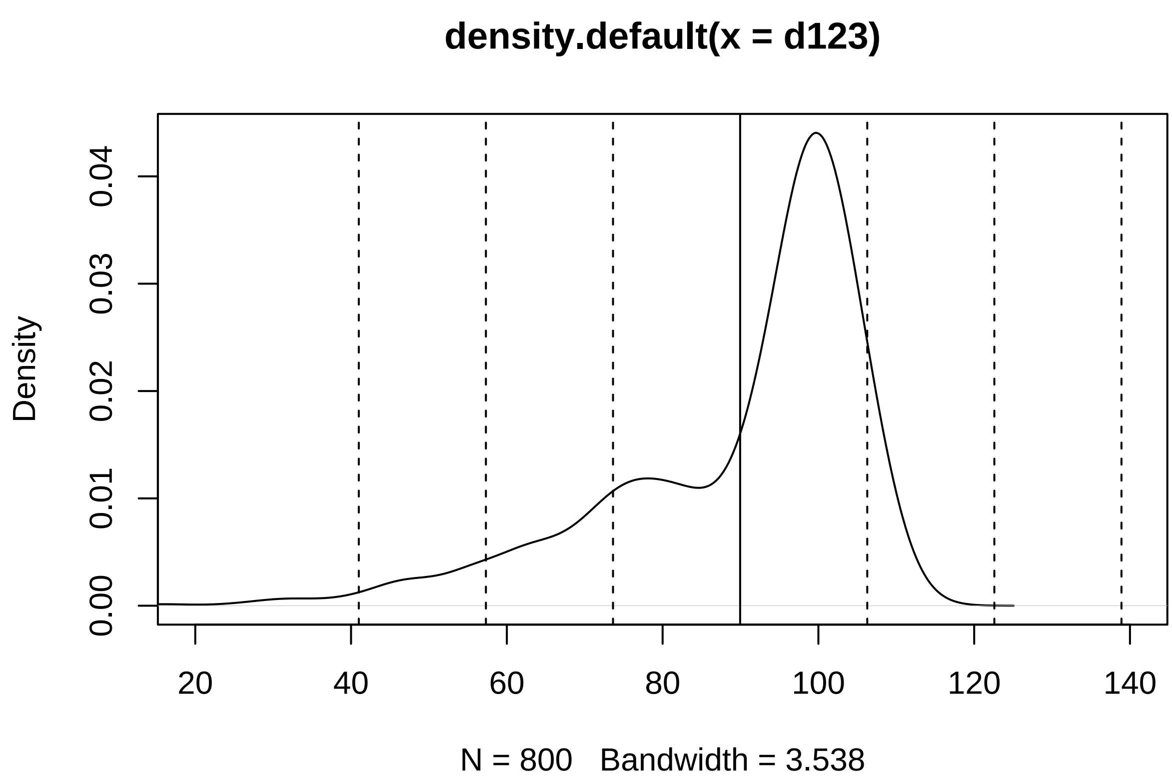
```
unname((quantile(new_data, 3/4)-mean(new_data))/sd(new_data))
```

```
## [1] 0.6816965
```

Ans. The 1st, 2nd, and 3rd quartiles of the new_data are very different from the rdata. Since **new_data** and **rdata** are both normally distributed, the distance between their quartiles and the mean divided by the corresponding standard deviation is almost the same.

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

```
plot(density(d123), xlim=c(20, 140))
abline(v=mean(d123), lty="solid")
new_mean <- mean(d123)
abline(v=c(new_mean-sd(d123), new_mean+sd(d123)), lty="dashed")
abline(v=c(new_mean-sd(d123)*2, new_mean+sd(d123)*2), lty="dashed")
abline(v=c(new_mean-sd(d123)*3, new_mean+sd(d123)*3), lty="dashed")
```



```
summary(d123)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.81   81.05   96.57   89.95  101.10  114.45
```

```
unname((quantile(d123, 1/4)-mean(d123))/sd(d123))
```

```
## [1] -0.5455278
```

```
unname((quantile(d123, 3/4)-mean(d123))/sd(d123))
```

```
## [1] 0.6833087
```

Ans. As we can observe from the plot, **d123** is a left skewed distribution; hence, the distance between the quartiles and the mean divided by the corresponding standard deviation is no way near **(b)**'s results.

Question 3)

We mentioned in class that there might be some objective ways of determining the bin size of histograms. Note that, for any dataset d, we can calculate number of bins (k) from the bin width (h): `k = ceiling((max(d) - min(d))/h)` and bin width from number of bins: `h = (max(d) - min(d)) / k`

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

Ans. He suggest to use **Freedman-Diaconis rule** for bin widths and numbers. In addition, the Wikipedia article says that the FD method is less sensitive than the standard deviation to outliers in 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
ii. Scott's normal reference rule (uses standard deviation)
iii. Freedman-Diaconis' choice (uses IQR)

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

# sd for samples
sample_sd <- function(x){
  variance <- sum((x-mean(x))^2)/(length(x)-1)
  sqrt(variance)
}

# Freedman-Diaconis' choice
FD <- function(x){
  h <- 2*IQR(x)/(length(x)^(1/3))
  k <- ceiling((max(x)-min(x))/h)
  paste("Freedman-Diaconis => ", "h: ", h, "; k: ", k)
}

# Scott's normal reference rule
```

```r
SCOTT <- function(x){
  h <- 3.49*sample_sd(x)/(length(x)^(1/3))
  k <- ceiling((max(x)-min(x))/h)
  paste("Scott => ", "h: ", h, "; k: ", k)
}

# Sturges' formula (this is default)
STRUGES <- function(x){
  k <- ceiling(log(length(x), base=2))+1
  h <- ceiling((max(x)-min(x))/k)
  paste("Struges => ", "h: ", h, "; k: ", k)
}
```

```r
# (i)
STRUGES(rand_data)
```

```
## [1] "Struges =>  h:  3 ; k:  11"
```

```r
# (ii)
SCOTT(rand_data)
```

```
## [1] "Scott =>  h:  1.8877324395713 ; k:  17"
```

```r
# (iii)
FD(rand_data)
```

```
## [1] "Freedman-Diaconis =>  h:  1.46828240283988 ; k:  21"
```

(c) Repeat part **(b)** but extend the `rand_data` dataset with some outliers: `out_data <- c(rand_data, runif(10, min=40, max=60))`

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

```r
STRUGES(out_data)
```

```
## [1] "Struges =>  h:  6 ; k:  11"
```

```r
# (ii)
SCOTT(out_data)
```

```
## [1] "Scott =>  h:  2.32653994679577 ; k:  24"
```

```r
# (iii)
FD(out_data)
```

```
## [1] "Freedman-Diaconis =>  h:  1.49904116605154 ; k:  38"
```

From your answers above, in which of the three methods does the bin width (h) change the **least** when outliers are added, and **WHY** do you think that is?

Ans. As we can see from the answer above, among the three methods, "Freedman-Diaconis's choice's" bin width (h) changes the least. Since, "Struges' formula" implicitly basing bin sizes in the range of the data; "Scott's choice" is used for a normal distribution based on the estimate of the standard error; and "Freedman-Diaconis choice" is calculated based on the `inter-quartile range` (computes interquartile range of the given data.)

In my opinion, it is the way **Freedman-Diaconis choice** computes the bin width, h, that lowers the affect from the outliers in the datasets.