BACS HW2 - 109006234

Credit: 109006278

March 5th 2023

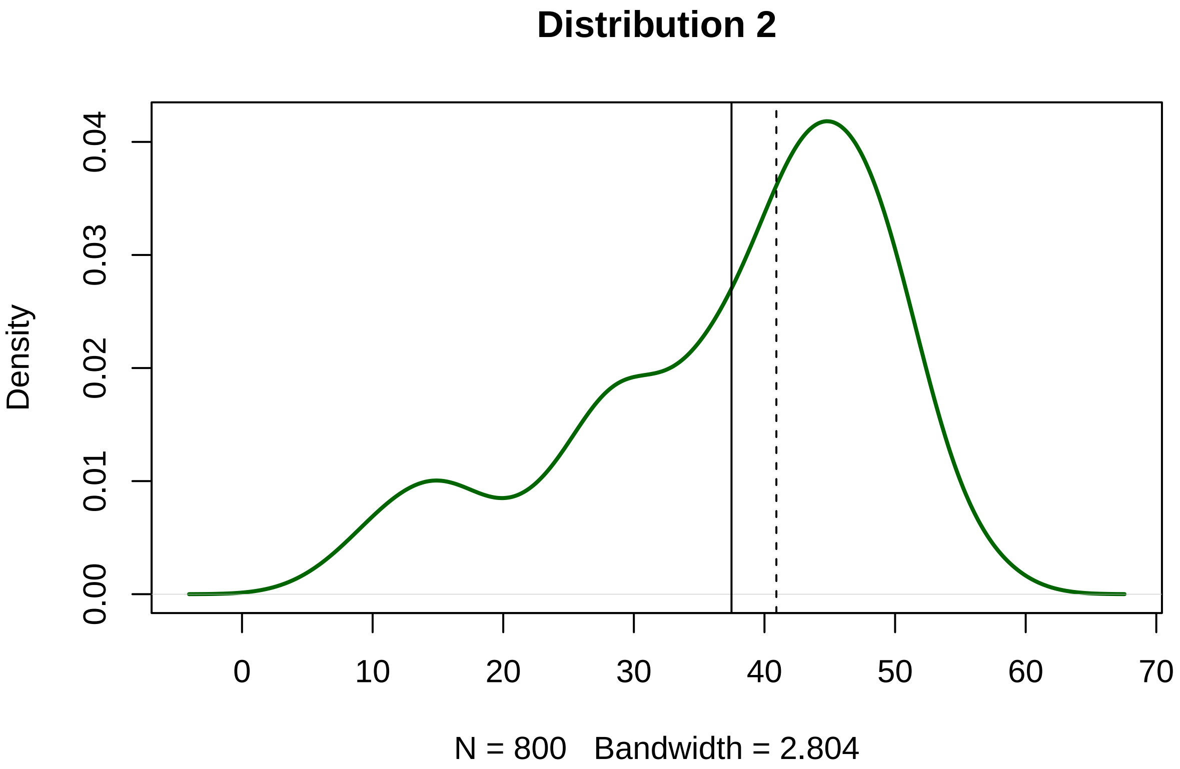
Q1(a) Visualizing a graph with a negatively skewed tail

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

d123 <- c(d1, d2, d3)

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

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



Computing Mean and Median

```
mean(d123)
```

```
## [1] 37.47523
```

```
median(d123)
```

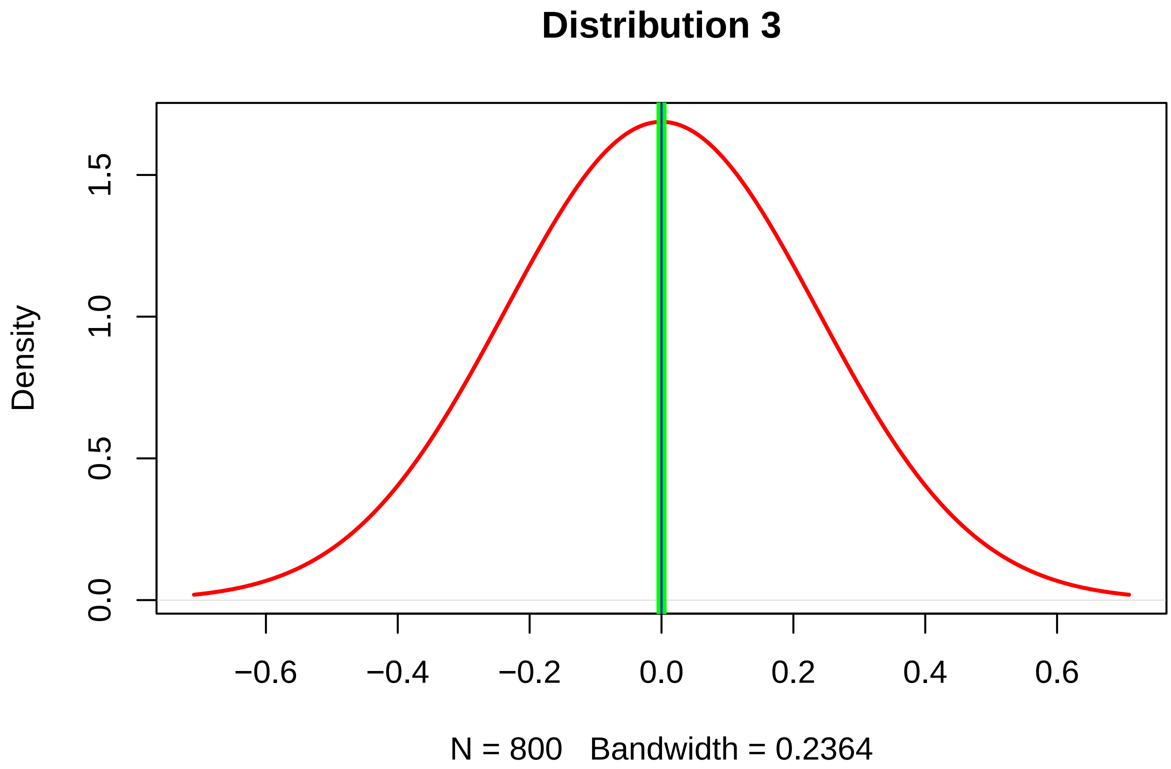
```
## [1] 40.91157
```

Q1(b) Visualizing a bell-shaped graph

```
dist <- rnorm(n = 800, mean = 0,  sd = 0)

plot(density(dist), col = "red", lwd = 2, main = "Distribution 3")

abline(v = mean(dist), lwd = 5, col = "green")
abline(v = median(dist), col = "blue")
```



Computing Mean and Median

```
mean(dist)
```

```
## [1] 0
```

```
median(dist)
```

```
## [1] 0
```

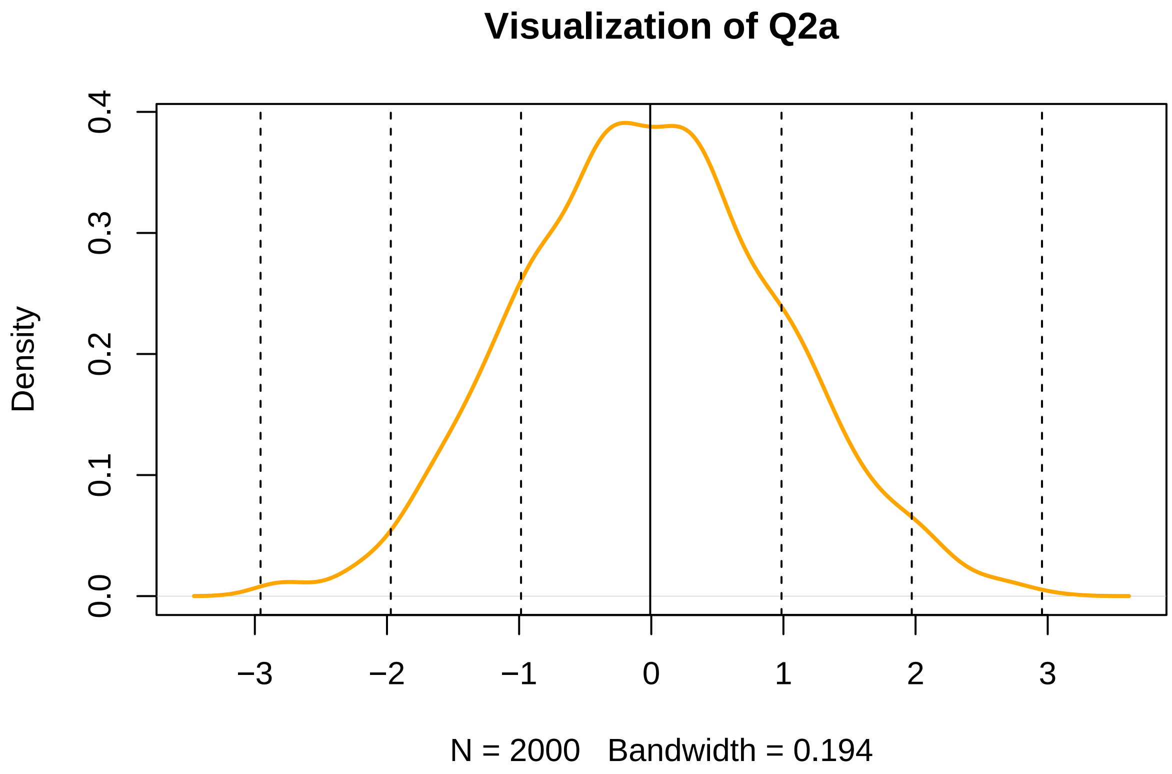
Q1(c) Which of the central measurements is more likely to be affected by outliers in the data? Is it mean or median?
In measures of central tendency, mean is more sensitive that median because the result of mean is highly affected by how many outliers the data has. In addition to that, mean will also be affected by skewed distribution like the ones visualized in Q1(a).

Q2(a) A normally distributed graph with: n=2000, mean=0, sd=1 with its mean and six standard deviation lines

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

plot(density(rdata), col = "orange", lwd = 2, main = "Visualization of Q2a")

abline(v = mean(rdata))
for(i in 1:3) {
    abline(v = i*sd(rdata), lty = "dashed")
    abline(v = -i*sd(rdata), lty = "dashed")
}
```



Q2(b) Computing quantile distances from Q2(a) normally distributed dataset

```r
q <- quantile(rdata, probs = c(.25, .5, .75))
q
```

```
##         25%         50%         75% 
## -0.69926561 -0.02657002  0.65630824
```

```
result <- (q - mean(rdata)) / sd(rdata)
result
```

```
##         25%         50%         75%
## -0.70129634 -0.01877557  0.67407660
```

Q2(c) Computing quantile distances the given normally distributed dataset

```
rdata <- rnorm(n = 2000, mean = 35, sd = 3.5)
q1 <- quantile(rdata, probs = c(.25, .5, .75))
result <- (q1 - mean(rdata)) / sd(rdata)
result
```

```
##         25%         50%         75%
## -0.66839174 -0.02242875  0.66304857
```

We can see that in Q2(c) that we have a larger mean and standard deviation compared to Q2(b). We can observe that with a larger standard deviation, the quantile distance becomes wider. We can also see that the mean shifts to the right, hence the quantile distance will follow accordingly.

Q2(d) Computing quantile distances from Q1(a) normally distributed dataset

```
q2 <- quantile(d123, probs = c(.25, .5, .75))
result <- (q2-mean(d123)) / sd(d123)
result
```

```
##        25%        50%        75%
## -0.6547922  0.2896749  0.7567899
```

We can observe that mean and standard deviation of the Q1 dataset are bigger that the Q2b dataset. Hence, we can observe that as the standard deviation increase, the quantile distance is wider. However, as the Q2 sample size is significantly larger, its quantile distance can be more precise.

Q3(a) Analysing histograms' formulas and its benefits In his comment, the number of bins is pretty much the same as said in the question that is k = (max - min)/h, where k is the number of bins, max(min) is the maximum(minimum) value in the observation group and h is the size of each bin.

In this case, h is 2 x InterQuartileRange x n^-1/3. To calculate the width of the bin, the comment is using the Freedman-Diaconis' rule. Its benefits is that in case of occuring outliers, it won't so much affect the width of the bin.

Q3(b) Calculating histograms' numbers of bin and bin width using n=800, mean=20, sd = 5

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

(i) Sturges' Formula

```
bins_sturges <- ceiling(log2(n)) + 1
width_sturges <- (max(rand_data) - min(rand_data)) / bins_sturges
bins_sturges
```

```
## [1] 11
```

```
width_sturges
```

```
## [1] 2.566282
```

(ii) Scott's Formula

```
width_scott <- (3.49 * sd(rand_data)) / n^(1/3)
bins_scott <- ceiling(max(rand_data) - min(rand_data)) / width_scott
width_scott
```

```
## [1] 1.832079
```

```
bins_scott
```

```
## [1] 15.82901
```

(iii) Freedman-Diaconis' Formula

```
width_fd <- (2*IQR(rand_data)) / n^(1/3)
bins_fd <- ceiling(max(rand_data) - min(rand_data)) / width_fd
width_fd
```

```
## [1] 1.465058
```

```
bins_fd
```

```
## [1] 19.79444
```

Q3(c) Calculating histograms' numbers of bin and bin width using some random dataset with outliers.

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

(i) Sturges' Formula

```
bins_sturges <- ceiling(log2(n)) + 1
width_sturges <- (max(out_data) - min(out_data)) / bins_sturges
bins_sturges
```

```
## [1] 11
```

```
width_sturges
```

```
## [1] 4.89678
```

(ii) Scott's Formula

```
width_scott <- (3.49 * sd(out_data)) / n^(1/3)
bins_scott <- ceiling(max(out_data) - min(out_data)) / width_scott
width_scott
```

```
## [1] 2.237765
```

```
bins_scott
```

```
## [1] 24.13122
```

(iii) Freedman-Diaconis' Formula

```
width_fd <- (2*IQR(out_data)) / n^(1/3)
bins_fd <- ceiling(max(out_data) - min(out_data)) / width_fd
width_fd
```

```
## [1] 1.47155
```

```
bins_fd
```

```
## [1] 36.696
```

Freedman-Diaconis' formula can be said to be the most stable formula in case of existing outliers because it computes the number of bins and the width of the bins without needing to compute its mean as said in Q1(c).