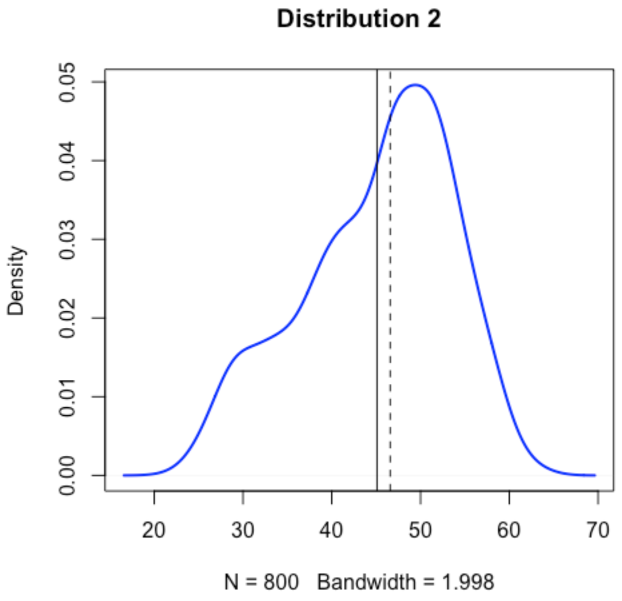
Question 1

a. Create a “Distribution 2”, which is negatively skewed, and compute the median and mean.

```
# Three normally distributed data sets
d1 <- rnorm(n=500, mean=50, sd=5)
d2 <- rnorm(n=200, mean=40, sd=4)
d3 <- rnorm(n=100, mean=30, sd=3)
# Let’s combine them into a single dataset
dist2 <- c(d1, d2, d3)
# Let’s plot the density function of abc
plot(density(dist2), col="blue", lwd=2,
    main = "Distribution 2")
# Add vertical lines showing mean and median
abline(v=mean(dist2))
abline(v=median(dist2), lty="dashed")
```

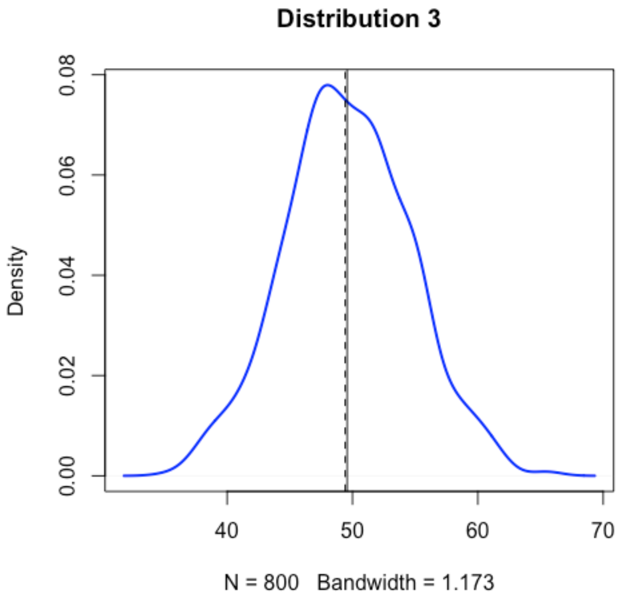


b. Create a “Distribution 3”: a single dataset that is normally distributed (n=800)

```
# One normally distributed datasets
d1 <- rnorm(n=800, mean=50, sd=5)

# Let’s plot the density function d1
plot(density(d1), col="blue", lwd=2,
     main = "Distribution 3")

# Add vertical lines showing mean and median
abline(v=mean(d1))
abline(v=median(d1), lty="dashed")
```

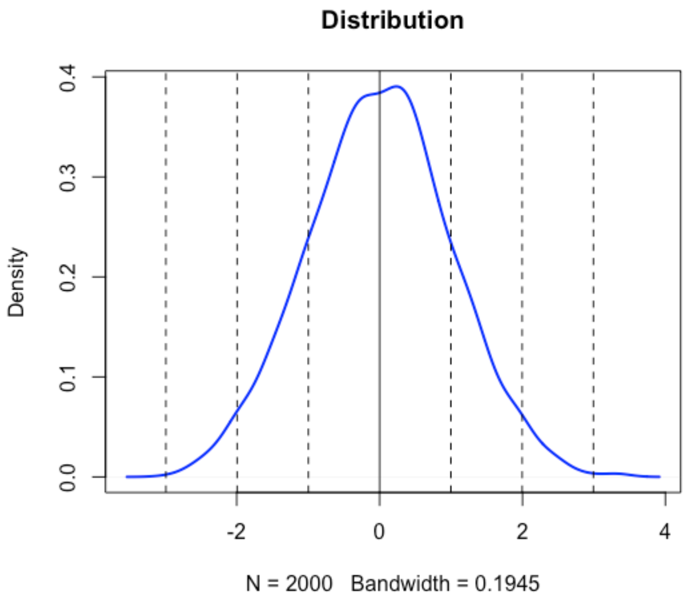


c. From Distribution 1-3, it can be observed that median is closer to the mean of the dataset which n is equal to 500. That is, mean is more sensitive to outliers (n=100 or n=200). So, median is more appropriate to be the measure of central tendency.

Question 2

a. Create a random dataset that is normally distributed with: n=2000, mean=0, sd=1. Draw density plot and mark mean and standard deviation.

```
# Generate data which is normally
distributed
rdata <- rnorm(n=2000, mean=0, sd=1)
# Plot density funtion of rdata
plot(density(rdata), col="blue", lwd=2,
    main = "Distribution")
# Calculate mean and standard deviation
m=mean(rdata)
d=sd(rdata)
# Add vertical lines showing mean and
median
abline(v=m)
abline(v=m+d, lty="dashed")
abline(v=m+2*d, lty="dashed")
abline(v=m+3*d, lty="dashed")
abline(v=m-d, lty="dashed")
abline(v=m-2*d, lty="dashed")
abline(v=m-3*d, lty="dashed")
```



b. Use quantile() function to find 1st, 2nd and 3trd quantiles, and calculate how many standard deviations away from the mean are those points?

```
# Calculate quantile
Q <- quantile(rdata,c(0.25,0.5,0.75))
Q
        25%         50%         75% 
-0.66673516  0.01100651  0.65769283 

# How many standard deviation away from mean
distance <- (Q - mean(rdata))/sd(rdata)
distance
        25%         50%         75% 
-0.66932282  0.01028876  0.65875932 
```

c. Generate new dataset with mean=35 and sd=3.5. Then, calculate how many standard deviations away from mean are 1st and 3rd quantiles.

```
# Generate new dataset
rdata2 <- rnorm(n=2000, mean=35, sd=3)

# Calculate quantile
Q2<- quantile(rdata2,c(0.25,0.75))
Q2
     25%      75% 
33.06096 36.91783 

# How many standard deviation away from mean
distance <- (Q2 - mean(rdata2))/sd(rdata2)
distance
       25%        75% 
-0.6631406  0.6554537 
```

Compare the result with the answer in (b), it can be observed that regardless of mean and standard deviation of the normal distribution, the first and the third quantile will locate at the -0.6631406 and 0.6554537 standard deviation.

d. Calculate how many standard deviations away from the mean are the first and third quantiles of d123 (Distribution 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)

# Let's combine them into a single dataset
d123 <- c(d1, d2, d3)

# Calculate quantile
Q123<- quantile(d123,c(0.25,0.75))
Q123
     25%      75% 
13.78673 30.88486 

# How many standard deviation away from mean
distance <- (Q123 - mean(d123))/sd(d123)
distance
       25%        75% 
-0.7362348  0.7050108 
```

Compare the result with the answer in (b), we can see that the quantiles of Distribution 1 locate no longer at -0.6631406 and 0.6554537 standard deviation. This is because we concatenate datasets from different normal distributions, which are with different mean values and standard deviations, d123 no more distributes normally.

Question III

a. Rob Hyndman's suggests using Freedman-Diaconis rule, which is as follows:

$$h = 2\frac{IQR(x)}{\sqrt[3]{n}}$$

This method uses interquartile, which is less sensitive to outliers in data.

b. Compute the bin widths (h) and number of bins (k) according to different formula.

I. Sturges’ formula

```
# Generate random data
rand_data <- rnorm(800, mean=20, sd = 5)
# Sturges’ formula
# k is the number of bins and h is bin widths
k <- ceiling(log2(800))+1
k
[1] 11
h <- (max(rand_data) - min(rand_data)) / k
h
[1] 3.180234
```

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

```
# Scott’s normal reference rule
# k is the number of bins and h is bin widths
h <- 3.49 * sd(rand_data) / 800^(1/3)
h
[1] 1.887741
k <- ceiling((max(rand_data) - min(rand_data))/h)
k
[1] 19
```

III. Freedman-Diaconis’ choice (uses IQR)

```
# Freedman-Diaconis’ choice
# k is the number of bins and h is bin widths
h <- 2 * IQR(rand_data) / 800^(1/3)
h
[1] 1.449804
k <- ceiling((max(rand_data) - min(rand_data))/h)
k
[1] 25
```

c. Repeat part (b) but extend the rand_data with some outliers.

I. Sturges’ formula

```
# Adding outliers to rand_data
out_data <- c(rand_data, runif(10, min=40, max=60))
# Sturges’ formula
# k is the number of bins and h is bin widths
k <- ceiling(log2(800))+1
k
[1] 11
h <- (max(out_data) - min(out_data)) / k
h
[1] 5.112678
```

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

```
# Scott’s normal reference rule
# k is the number of bins and h is bin widths
h <- 3.49 * sd(out_data) / 800^(1/3)
h
[1] 2.222704
k <- ceiling((max(out_data) - min(out_data))/h)
k
[1] 26
```

III. Freedman-Diaconis’ choice (uses IQR)

```
# Freedman-Diaconis’ choice
# k is the number of bins and h is bin widths
h <- 2 * IQR(out_data) / 800^(1/3)
h
[1] 1.47248
k <- ceiling((max(out_data) - min(out_data))/h)
k
[1] 39
```

d. The bin width calculated based on Freedman-Diaconis’ choice changes the least when outliers are added. I think the reason is that the method takes IQR into consideration instead of standard deviation of data. IQR will be sensitive when the number of outliers increases a lot, and however, there are only ten outliers in this case.