BASM 2017 - HW 2

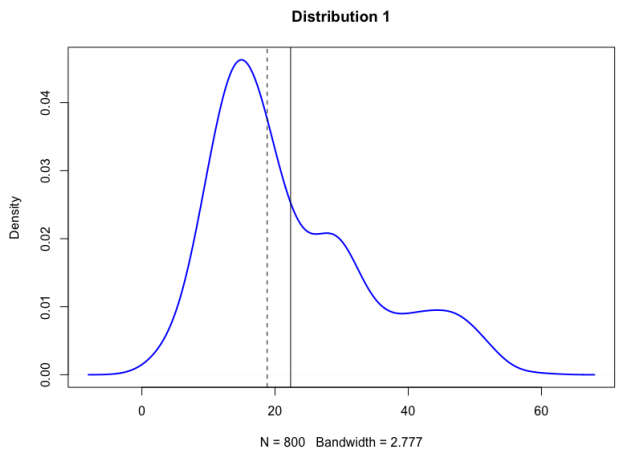
Question 1)

```
# 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)

# We can combine them into a single dataset
d123 <- c(d1, d2, d3)

# We can plot the density function of abc
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 "Distribution 2": a combined dataset (`n=800`) that is negatively skewed (tail stretches to the left). Change the mean and standard deviation of a, b, and c to achieve this new distribution. Compute the mean and median, and draw lines showing the mean (thick line) and median (thin line).

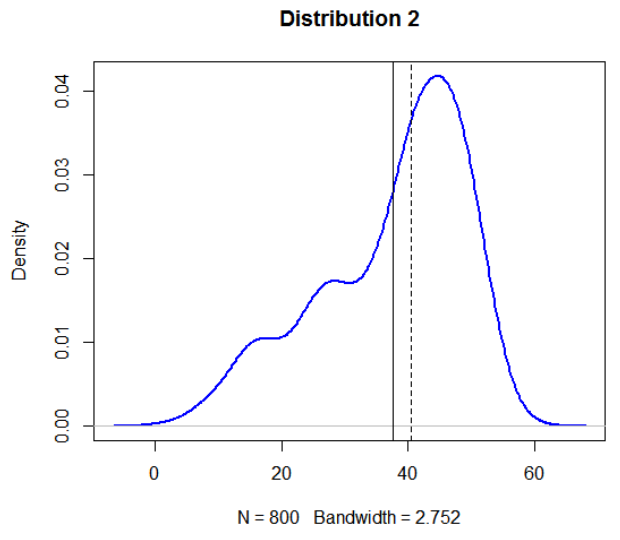
```
### a)

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

# We can combine them into a single dataset
d123 <- c(d1, d2, d3)

# We can plot the density function of abc
plot(density(d123), col="blue", lwd=2,
     main = "Distribution 2")

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



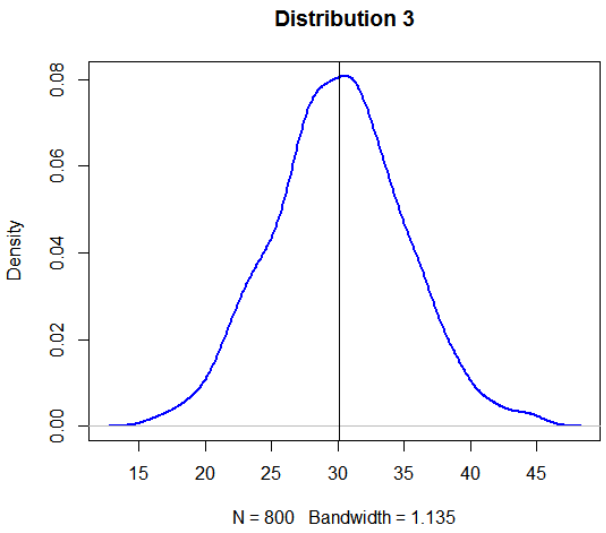
(b) Create "Distribution 3": a single dataset that is normally distributed (bell-shaped, symmetric) -- you do not need to combine datasets, just use the `rnorm` function to create a single large dataset (`n=800`). Show your code, compute the mean and median, and draw lines showing the mean (thick line)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

```
### b)

# A normally distributed data sets
d <- rnorm(n=800, mean=30, sd=5)

# We can plot the density function of abc
plot(density(d), col="blue", lwd=2,
     main = "Distribution 3")

# Add vertical lines showing mean and median
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?

As we saw from distribution 1 & 2. We know that mean will be more sensitive to outliers being added to dataset because mean is calculated with the “value” of the sampleset.

Question 2)

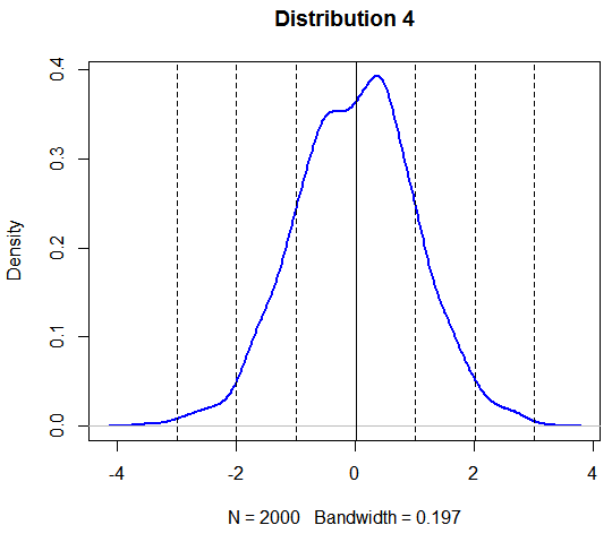
a) Create a random dataset (call it 'rdata') that is normally distributed with: n=2000, mean=0, sd=1. Draw a density plot and put a solid vertical line on the mean, and dashed vertical lines at the 1st, 2nd, and 3rd standard deviations on *both* sides of the mean. You should have a total of 7 vertical lines.

```
### a)

# A normally distributed data sets
rdata <- rnorm(n=2000, mean=0, sd=1)

# We can plot the density function of abc
plot(density(rdata), col="blue", lwd=2,
     main = "Distribution 4")

# Add vertical lines showing mean and median
abline(v=mean(rdata))
r_std = sd(rdata)
r_mean = mean(rdata)
vline = c(r_mean + r_std, r_mean + 2 * r_std,
r_mean + 3 * r_std, r_mean - r_std, r_mean -
2 * r_std, r_mean - 3 * r_std)
abline(v=vline, lty="dashed")
```



b) Using the `quantile` function, which data points correspond to the 1st, 2nd, and 3rd *quartiles* (i.e., 25th, 50th, 75th percentiles). How many *standard deviations away from the mean* (use positive or negative) are those points corresponding to the 1st, 2nd, and 3rd quartiles?

```
### b)

r_quantile = unname(quantile(rdata))
r_diff = (r_quantile - r_mean) / r_std
```

```
> r_quantile
[1] -3.56363850 -0.66887146  0.0406517
8  0.67509483  3.20436990
> r_diff
[1] -3.56861265 -0.67611833  0.0328478
5  0.66679278  3.19408208
```

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)

```
### c)

rdata <- rnorm(n=2000, mean=35, sd=3.5)
r_std = sd(rdata)
r_mean = mean(rdata)
r_quantile = unname(quantile(rdata))
r_diff = (r_quantile - r_mean) / r_std
```

```
> r_quantile
[1] 21.52114 32.46487 34.90232 37.3205
9 47.90967
> r_diff
[1] -3.723261549 -0.676872467  0.00163
8848  0.674808715  3.622473425
```

The r_diff of 1st and 3rd is almost the same compare to answer (b)

d) Finally, recall the dataset d123 shown in 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)

```
### d)

r_std = sd(d123)
r_mean = mean(d123)
r_quantile = unname(quantile(d123))
r_diff = (r_quantile - r_mean) / r_std
```

```
> r_quantile
[1]  1.560812 29.292197 40.485931 46.5
16642 59.894036
> r_diff
[1] -3.0872410 -0.7053451  0.2561037
0.7740917  1.9230988
```

According to distribution 1, we know that d123 is a right skewed distribution. It means that for the left hand side of mean contains more than 50% of sample points. So that means the 1st quantile will closer to mean than 3rd quantile.

Question 3) We mentioned in class that there might be some *objective* ways of determining the bin size of histograms. Take a quick look at the Wikipedia article on Histograms ("Number of bins and width") to see the different ways to calculate bin width (h) and number of bins (k).

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
```

Now, read the following discussion on StackOverflow about choosing the number of bins.

a) From the StackOverflow question, 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 suggested to use The Freedman-Diaconis rule to calculate bin widths/number.

h = 2*IQR*n^-1/3, k = (max - min) / h

The benefit 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)

```
### b)

# A normally distributed data sets
rand_data <- rnorm(800, mean=20, sd = 5)

k_sturges = nclass.Sturges(rand_data)
k_scott = nclass.scott(rand_data)
k_FD = nclass.FD(rand_data)
```

```
> nclass.Sturges(rand_data)
[1] 11
> nclass.scott(rand_data)
[1] 17
> nclass.FD(rand_data)
[1] 22
```

c) Repeat part (b) but extend the rand_data dataset with some outliers (use a new dataset `out_data`):

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

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

k_sturges = nclass.Sturges(out_data)
k_scott = nclass.scott(out_data)
k_FD = nclass.FD(out_data)
```

```
> k_sturges
[1] 11
> k_scott
[1] 24
> k_FD
[1] 37
```

d) 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?

Sturges is change the least. Because Sturges only care about the total number of data, not the value observed.