BACS HW (Week 2)

108020024

due on 03/05 (Sun)

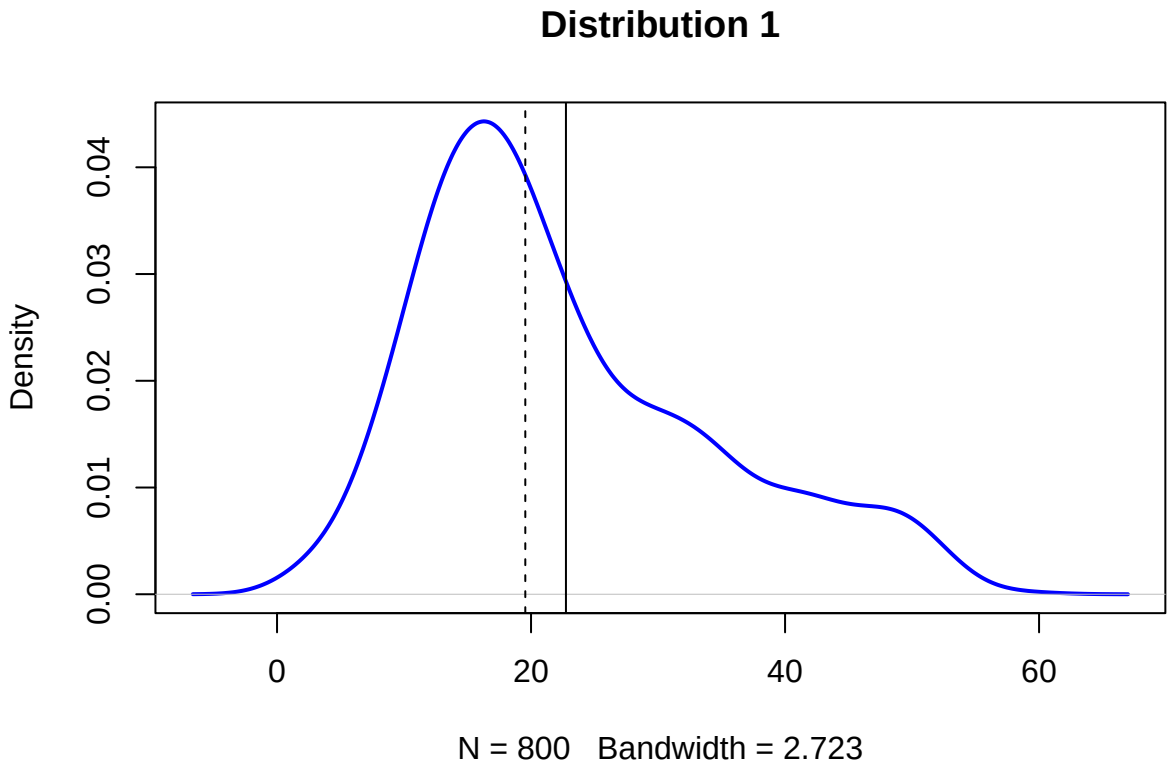
Question 1) Let's have a look at how the mean and median behave.

```r
# 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)

# 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 1")

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



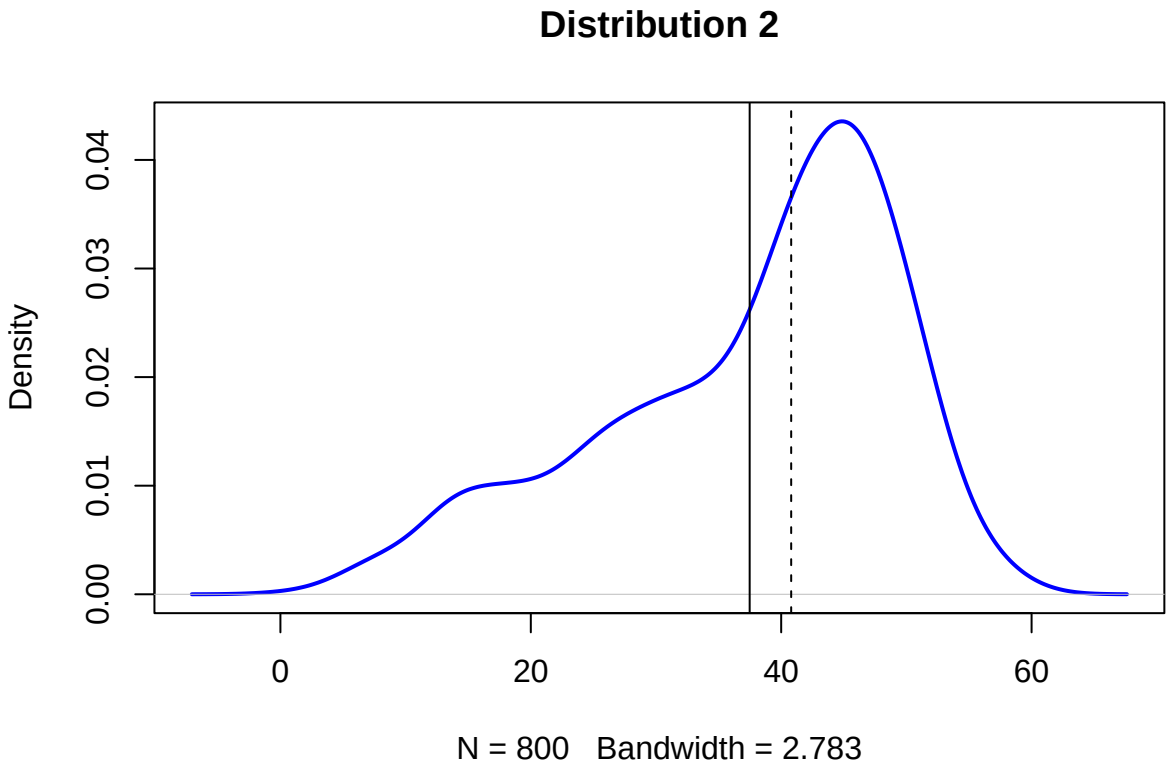
(a) Create and visualize a new "Distribution 2": a combined dataset (n=800) that is negatively skewed (tail stretches to the left). Change the mean and standard deviation of d1, d2, and d3 to achieve this new distribution. Compute the mean and median, and draw lines showing the mean (thick line) and median (thin line).

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

# Combining them into a composite dataset
Distribution_2 <- c(d1, d2, d3)

# Let's plot the density function of Distribution_2
plot(density(Distribution_2), col="blue", lwd=2,
     main = "Distribution 2")

# Add vertical lines showing mean and median
abline(v=mean(Distribution_2))
abline(v=median(Distribution_2), lty="dashed")
```



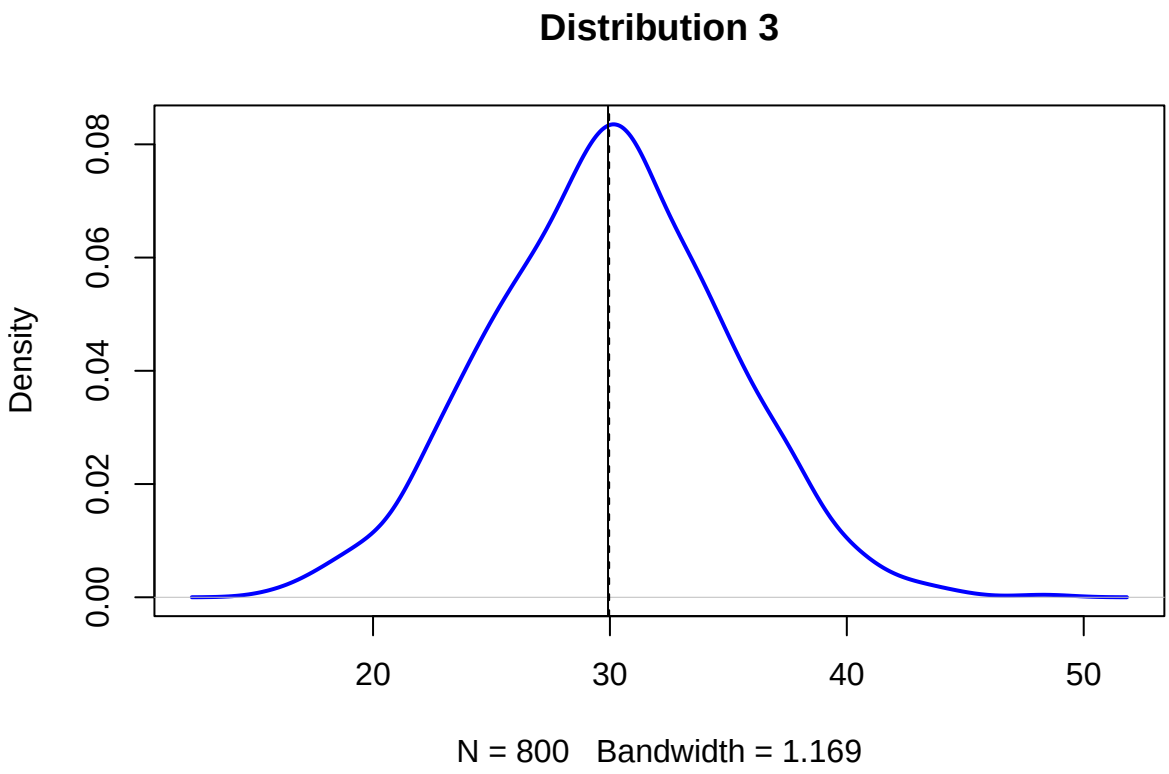
(b) Create a "Distribution 3": a single dataset that is normally distributed (bell-shaped, symmetric) – you do not need to combine datasets, just use the rnorm() function to create a single large dataset (n=800). Show your code, compute the mean and median, and draw lines showing the mean (thick line)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

```
# Three normally distributed data sets
d1 <- rnorm(n=800, mean=30, sd=5)

# Combining them into a composite dataset
Distribution_3 <- c(d1)

# Let's plot the density function of Distribution_2
plot(density(Distribution_3), col="blue", lwd=2,
     main = "Distribution 3")

# Add vertical lines showing mean and median
abline(v=mean(Distribution_3))
abline(v=median(Distribution_3), 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?

```
d1 <- rnorm(n=800, mean=30, sd=5)
#calculate 1.5 IQR
1.5*(quantile(d1,0.75) - quantile(d1,0.25))
```

```
##      75%
## 9.854064
```

After calculate the 1.5 IQR, a distribution with mean $= 3000 > 10$ with standard deviation $= 5$ is a reasonable outlier added to the original distribution.

```
#added outlier to d1
outlier <- rnorm(n=10, mean=3000, sd=5)
out <- c(d1,outlier)

#calculate the mean and median when outliers not being added to data

mean(d1)
```

```
## [1] 30.18976
```

```
median(d1)
```

```
## [1] 30.25365
```

```
#calculate the mean and median when outliers being added to data

mean(out)
```

```
## [1] 66.8539
```

```
median(out)
```

```
## [1] 30.30623
```

Calculate the difference.

```
#calculate the difference
abs(mean(d1) - mean(out))
```

```
## [1] 36.66414
```

```
abs(median(d1) - median(out))
```

```
## [1] 0.0525724
```

The mean has a big shift, however median has a slightly change.

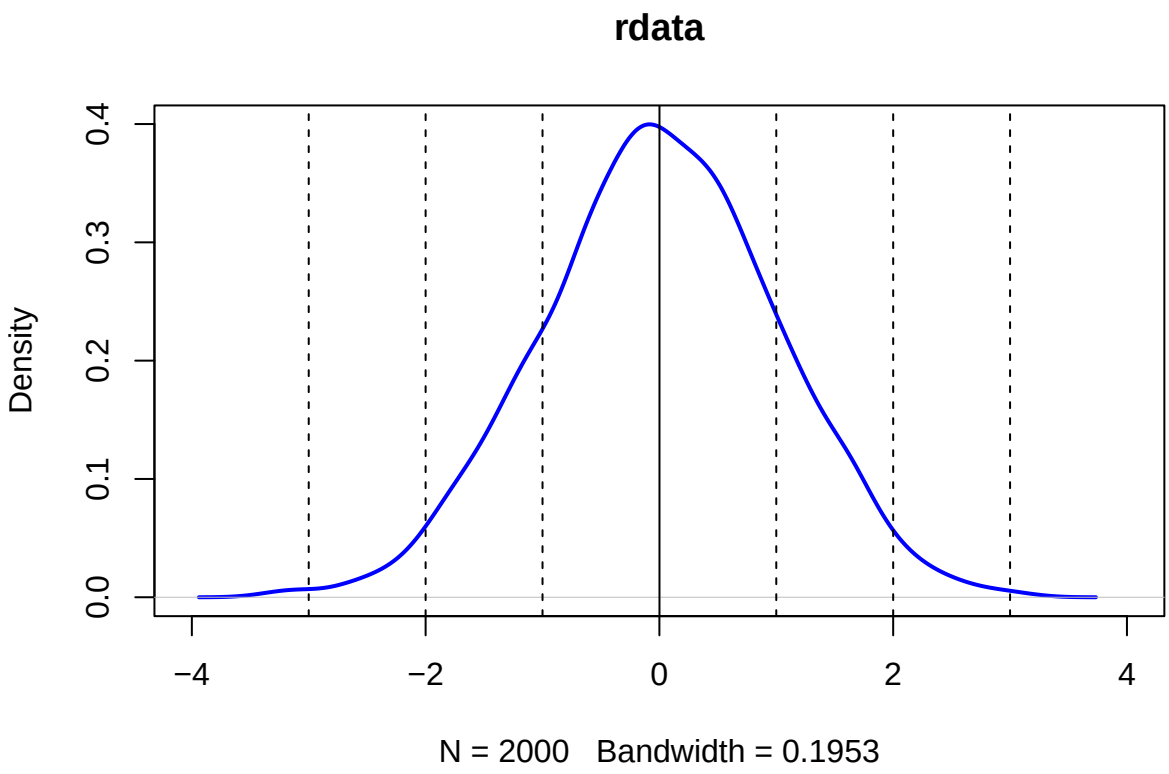
The conclusion is that mean will be more sensitive to outliers being added to your data.

Question 2) Let's try to get some more insight about what standard deviations are.

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).

```
rdata <- rnorm(n=2000, mean=0, sd=1)
# Let's plot the density function of Distribution_2
plot(density(rdata), col="blue", lwd=2,
     main = "rdata", xlim = c(-4,4))

# Add vertical lines showing mean and median
abline(v=mean(rdata))
abline(v=1, lty="dashed")
abline(v=2, lty="dashed")
abline(v=3, lty="dashed")
abline(v=-1, lty="dashed")
abline(v=-2, lty="dashed")
abline(v=-3, lty="dashed")
```



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?

```
quantile(rdata, c(0.25,0.5,0.75))
```

```
##         25%         50%         75%
## -0.66132375  0.00107297  0.66835049
```

```
rdataQ1 = unname(quantile(rdata, c(0.25,0.5,0.75)))[1]
rdataQ2 = unname(quantile(rdata, c(0.25,0.5,0.75)))[2]
rdataQ3 = unname(quantile(rdata, c(0.25,0.5,0.75)))[3]
```

Since the mean is 0, standard deviation is 1, the the 1st, 2nd, and 3rd quantile are the three values above, and the value shows how far from mean.

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)

```
rdata2 <- rnorm(n=2000, mean=35, sd=3.5)
#1st quantile
rdata2Q1 = unname(quantile(rdata2, c(0.25,0.5,0.75)))[1]
#2nd quantile
rdata2Q2 = unname(quantile(rdata2, c(0.25,0.5,0.75)))[2]
#3rd quantile
rdata2Q3 = unname(quantile(rdata2, c(0.25,0.5,0.75)))[3]


c(rdata2Q1,rdata2Q2,rdata2Q3)
```

```
## [1] 32.72203 34.98442 37.24983
```

The the 1st, 2nd, and 3rd quantile are the three values above.

```
(rdata2Q1-35)/3.5
```

```
## [1] -0.6508489
```

```
(rdata2Q3-35)/3.5
```

```
## [1] 0.6428092
```

It is nearly 0.66 standard deviation away from mean.

Compare with the answer in (b), there doesn't have any change, since our goal is the know the distance of Q1 and Q3 to the center of the distribution, the calculation about "how many standard deviations away from the mean" is doing normalization. Since both (b) and (c) are from a normal distribution, the distance for Q1 and Q3 to the center should be approximately 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 (use positive or negative) are those data points corresponding to the 1st and 3rd quartiles? Compare your answer to (b)

```
#1st quantile
d123Q1 <- unname(quantile(d123, c(0.25,0.5,0.75)))[1]
#2nd quantile
d123Q2 <- unname(quantile(d123, c(0.25,0.5,0.75)))[2]
#3rd quantile
d123Q3 <- unname(quantile(d123, c(0.25,0.5,0.75)))[3]

c(d123Q1,d123Q2,d123Q3)
```

```
## [1] 14.21483 19.54874 29.65073
```

The the 1st, 2nd, and 3rd quantile are the three values above.

```
(d123Q1-mean(d123))/sd(d123)
```

```
## [1] -0.7324957
```

```
(d123Q3-mean(d123))/sd(d123)
```

```
## [1] 0.5926287
```

It is nearly 0.74 standard deviation away from mean for Q1, and 0.63 standard deviation away from mean for Q3.

Compare with answer in (b), the result change, it is because in d123, the distribution has a positive skew (skew to the right), so Q1 is farther from the mean compare with the standard deviations away from the mean for Q3.

```
#calculate the change compare with (b)

abs((rdata2Q1-35)/3.5
 - (d123Q1-mean(d123))/sd(d123))
```

```
## [1] 0.08164678
```

```
abs((rdata2Q3-35)/3.5
 - (d123Q3-mean(d123))/sd(d123))
```

```
## [1] 0.05018054
```

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 this discussion on the Q&A forum called "Cross Validated" about choosing the number of bins

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?

The Rob Hyndman suggest to use Freedman–Diaconis rule, the bin width formula is

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

The Wikipedia article said the benefit of Freedman-Diaconis rule is that it is less sensitive than the standard deviation to outliers in data.

Freedman-Diaconis rule is designed to roughly minimize the integral of the squared difference between the histogram and the density of the theoretical probability distribution.

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

$$k = [\log_2 n] + 1$$

```
k1 = log2(800) +1
h1 = (max(rand_data)-min(rand_data))/k1

data.frame(
  k = k1,
  h = h1
)
```

```
##          k        h
## 1 10.64386 2.795954
```

ii. Scott's normal reference rule (uses standard deviation)

$$h = \frac{3.49\hat{\sigma}}{\sqrt[3]{n}}$$

where $\hat{\sigma}$ is the sample standard deviation.

```
h2 = 3.49*sd(rand_data)/(800^(1/3))
k2 = ceiling((max(rand_data) - min(rand_data))/h2)

data.frame(
  k = k2,
  h = h2
)
```

```
##    k       h
## 1 16 1.91884
```

iii. Freedman-Diaconis' choice (uses IQR)

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

```
h3 = 2*IQR(rand_data)/(800)^(1/3)
k3 = ceiling((max(rand_data) - min(rand_data))/h3)


data.frame(
  k = k3,
  h = h3
)
```

```
##    k        h
## 1 20 1.521825
```

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?

```
ko1 = log2(810) +1
ho1 = (max(out_data)-min(out_data))/ko1

data.frame(
  k = ko1,
  h = ho1
)
```

i. Sturges' formula

```
##          k        h
## 1 10.66178 4.583932
```

```
ho2 = 3.49*sd(out_data)/(810^(1/3))
ko2 = ceiling((max(out_data) - min(out_data))/ho2)

data.frame(
  k = ko2,
  h = ho2
)
```

ii. Scott's normal reference rule (uses standard deviation)

```
##    k        h
## 1 23 2.204443
```

```
ho3 = 2*IQR(out_data)/(810)^(1/3)
ko3 = ceiling((max(out_data) - min(out_data))/ho3)


data.frame(
  k = ko3,
  h = ho3
)
```

iii. Freedman-Diaconis' choice (uses IQR)

```
##    k        h
## 1 33 1.516817
```

See which method does the bin width (h) change the least when outliers are added.

```
abs(h1-ho1)
```

```
## [1] 1.787979
```

```
abs(h2-ho2)
```

```
## [1] 0.2856033
```

```
abs(h3-ho3)
```

```
## [1] 0.005007931
```

Freedman-Diaconis' choice change the least when outliers are added.The reason is because for the Sturges' formula, the range $(max(d) - min(d)$ increase when outliers are added, so h increase;for the Scott's normal reference rule, standard deviation $\hat{\sigma}$ increase when outliers are added, so h increase; the Freedman-Diaconis' choice used IQR, which are less sensitive to outliers compared with range and standard deviation, so that is the reason why for Freedman-Diaconis' choice, bin width (h) change the least when outliers are added.