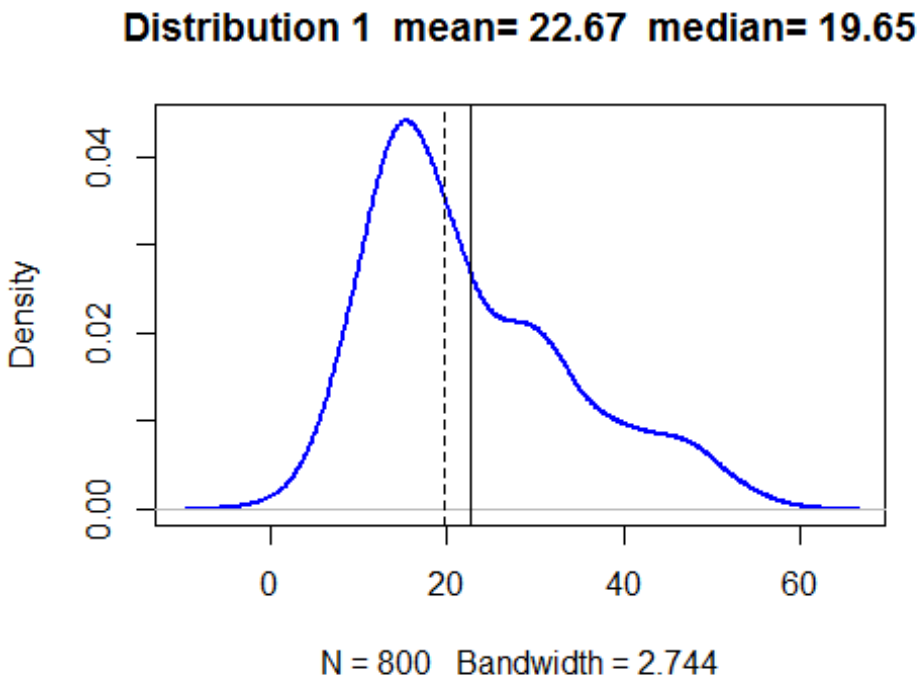
HW2

Black text are answers.

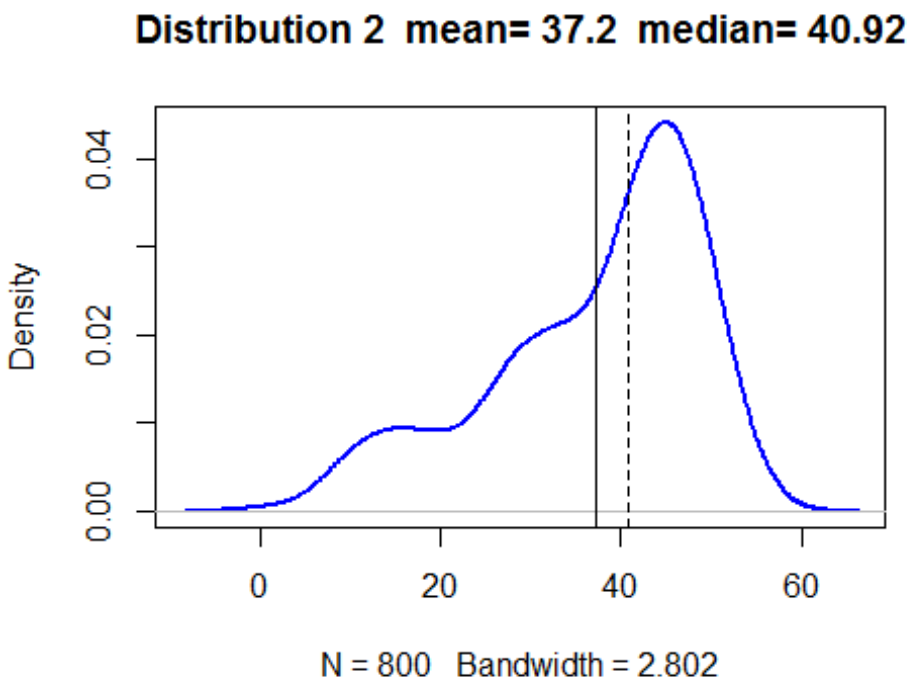
Question 1

```
#sample code
# 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 =paste("Distribution 1  mean=",round(mean(d123),2)," median=",
round(median(d123),2)))

# 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).

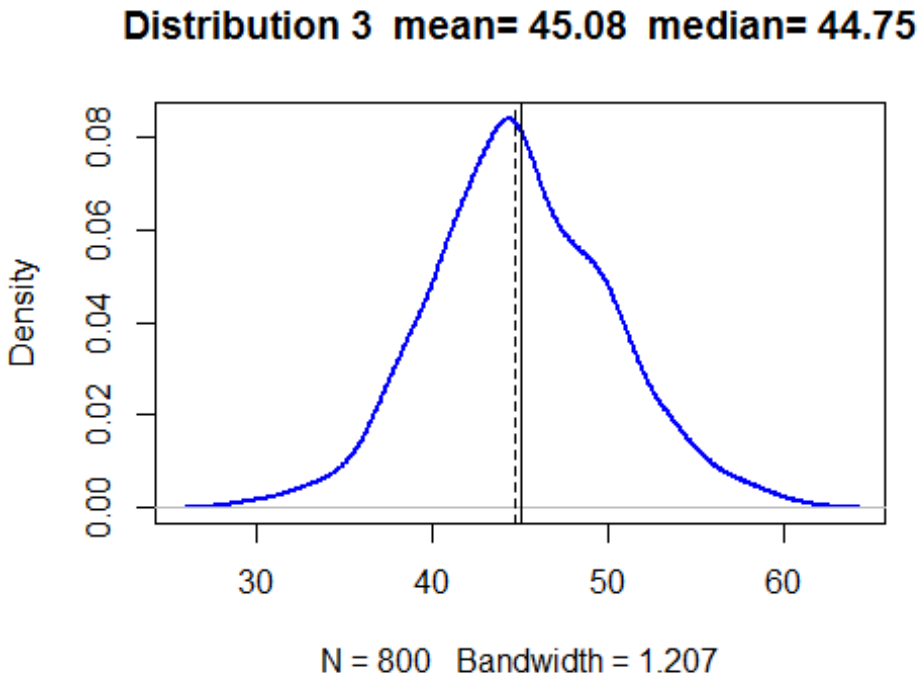
```
#(a)
# Three normally distributed data sets, a, b, and c
a <- rnorm(n=500, mean=45, sd=5)
b <- rnorm(n=200, mean=30, sd=5)
c <- rnorm(n=100, mean=15, sd=5)

# We can combine them into a single dataset, abc
abc <- c(a, b, c)

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

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



Note that in the above figure, the median is on the right side of the mean and more close to the peak.

(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)
single <- rnorm(n=800, mean=45, sd=5)

plot(density(single), col="blue", lwd=2,
     main = paste("Distribution 3  mean=",round(mean(single),2)," media
n=",round(median(single),2)))

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



In this dataset, it seems that the median and the mean are going to overlap because it's a normal distribution,whose data distribute symmetrically.

(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?

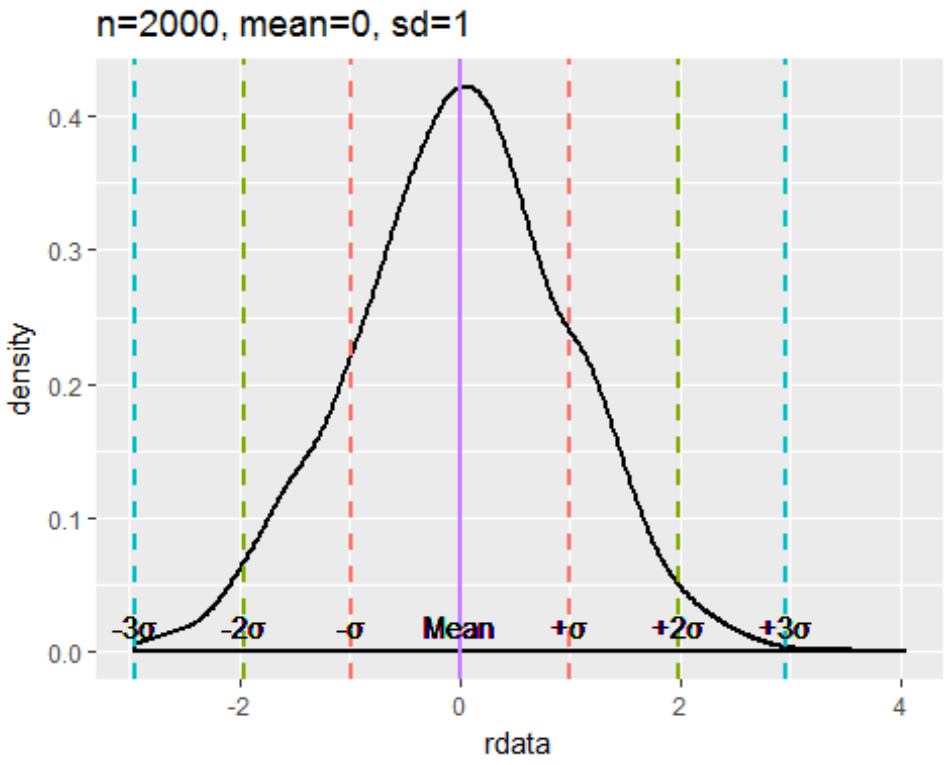
From the above figures, we can find out that the dashed lines, which represent "Median" for each distribution, are always closer to the central peak than the the thick lines (mean) if there exist outliers. In terms of the mean, it tends to show up at the side where tail stretches to. Therefore, I think the mean of a dataset is more

sensitive to outliers. Intuitively, that measure of central tendency calculate all the value in the dataset, including outliers, and will be affected by them.

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.

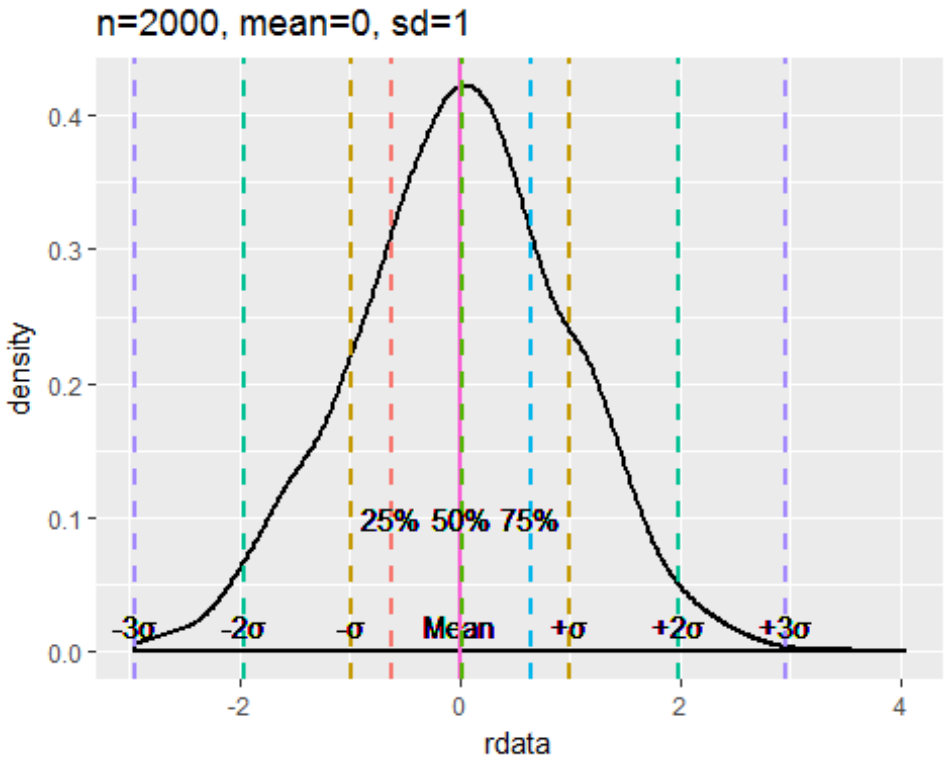
```
#In the following part, I will use ggplot to make figures more clearly
library(ggplot2)
#create a data frame for ggplot usage
rdata <- data.frame(norm = rnorm(n = 2000, mean = 0, sd = 1))

ggplot(rdata,aes(norm)) +
  geom_density(size = 1)+
  ggtitle("n=2000, mean=0, sd=1")+
  xlab("rdata")+

  #add line
  geom_vline(aes(xintercept = mean(rdata$norm),color="Mean"),size=1,sho
w.legend  =F,lty =1)+
  geom_vline(aes(xintercept = mean(rdata$norm)-sd(rdata$norm),color="1s
t standard deviations"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)-2*sd(rdata$norm),color="
2nd standard deviations"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)-3*sd(rdata$norm),color="
3rd standard deviations"),size=1,show.legend  =FALSE,lty =2 )+

  #add text to illustrate the line
  geom_text(aes(x=mean(rdata$norm)-sd(rdata$norm), label="-σ",y=0.02))+
  geom_text(aes(x=mean(rdata$norm)-2*sd(rdata$norm), label="-2σ",y=0.0
2))+
  geom_text(aes(x=mean(rdata$norm)-3*sd(rdata$norm), label="-3σ",y=0.0
2))+
  geom_vline(aes(xintercept = mean(rdata$norm)+sd(rdata$norm),color="1s
t standard deviations"),size=1,show.legend  =FALSE,lty =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)+2*sd(rdata$norm),color="
2nd standard deviations"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)+3*sd(rdata$norm),color="
3rd standard deviations"),size=1,show.legend  =F,lty =2 )+
  geom_text(aes(x=mean(rdata$norm)+sd(rdata$norm), label="+σ",y=0.02))+
  geom_text(aes(x=mean(rdata$norm)+2*sd(rdata$norm), label="+2σ",y=0.0
2))+
  geom_text(aes(x=mean(rdata$norm)+3*sd(rdata$norm), label="+3σ",y=0.0
2))+
geom_text(aes(x=mean(rdata$norm), label="Mean",y=0.02))
```



(b) Using the quantile function, which dat a 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?

```
ggplot(rdata,aes(norm)) +
  ggtitle("n=2000, mean=0, sd=1")+
  xlab("rdata")+
  geom_density(size = 1)+
  geom_vline(aes(xintercept = mean(rdata$norm),color="Mean"),size=1,sho
w.legend  =F,lty =1)+geom_vline(aes(xintercept = mean(rdata$norm)-1*sd
(rdata$norm),color="1st standard deviations"),size=1,show.legend  =F,lt
y =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)-2*sd(rdata$norm),color="
2nd standard deviations"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)-3*sd(rdata$norm),color="
3rd standard deviations"),size=1,show.legend  =FALSE,lty =2 )+
  geom_text(aes(x=mean(rdata$norm)-sd(rdata$norm), label="-σ",y=0.02))+
  geom_text(aes(x=mean(rdata$norm)-2*sd(rdata$norm), label="-2σ",y=0.0
2))+
  geom_text(aes(x=mean(rdata$norm)-3*sd(rdata$norm), label="-3σ",y=0.0
2))+
  geom_vline(aes(xintercept = mean(rdata$norm)+sd(rdata$norm),color="1s
t standard deviations"),size=1,show.legend  =FALSE,lty =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)+2*sd(rdata$norm),color="
2nd standard deviations"),size=1,show.legend  =F,lty =2)+
```

```
  geom_vline(aes(xintercept = mean(rdata$norm)+3*sd(rdata$norm),color="
3rd standard deviations"),size=1,show.legend  =F,lty =2 )+
  geom_text(aes(x=mean(rdata$norm)+sd(rdata$norm), label="+σ",y=0.02))+
  geom_text(aes(x=mean(rdata$norm)+2*sd(rdata$norm), label="+2σ",y=0.0
2))+
  geom_text(aes(x=mean(rdata$norm)+3*sd(rdata$norm), label="+3σ",y=0.0
2))+
geom_text(aes(x=mean(rdata$norm), label="Mean",y=0.02))+
  geom_vline(aes(xintercept = mean(rdata$norm),color="Mean"),size=1,sho
w.legend  =F,lty =1)+
  geom_vline(aes(xintercept = quantile(rdata$norm,0.25),color="1st quar
tiles"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = quantile(rdata$norm,0.5),color="2nd quart
iles"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = quantile(rdata$norm,0.75),color="3rd quar
tiles"),size=1,show.legend  =F,lty =2 )+
  geom_text(aes(x=quantile(rdata$norm,0.5), label="50%",y=0.1))+
  geom_text(aes(x=quantile(rdata$norm,0.25), label="25%",y=0.1))+
  geom_text(aes(x=quantile(rdata$norm,0.75), label="75%",y=0.1))
```



Here I put extra lines to show the 1st, 2nd, and 3rd quartiles. According to the figure above, we can find out that those quartiles are sit within one standard deviation away from the mean; that is, they show up between one neagative standard deviation and one positive standard deviation.

```
#compute how many standard deviations way from the mean
(quantile(rdata$norm,0.25)-mean(rdata$norm))/sd(rdata$norm)
```

```
##        25%
## -0.6365139
```

The 1st quartile is about -0.64 standard deviations away from the mean.

```
(quantile(rdata$norm,0.5)-mean(rdata$norm))/sd(rdata$norm)
```

```
##         50%
## 0.008909081
```

The 2nd quartile is about 0.01standard deviations away from the mean.

```
(quantile(rdata$norm,0.75)-mean(rdata$norm))/sd(rdata$norm)
```

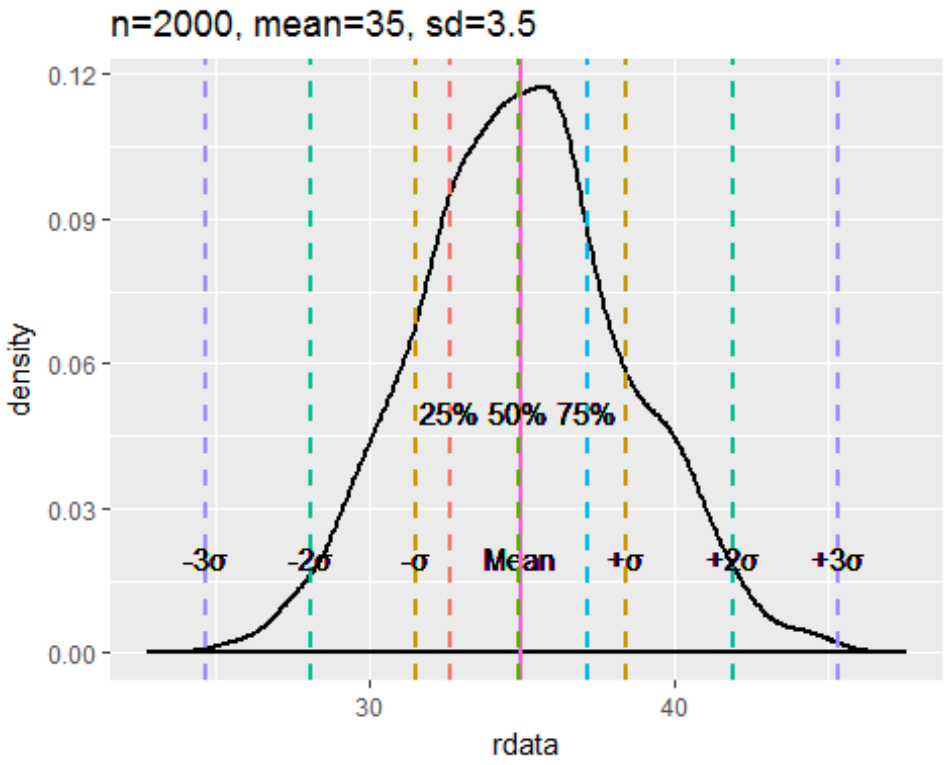
```
##       75%
## 0.6489403
```

The 3rd quartile is about 0.65 standard deviations away from the mean.

(c) Now create a new random dataset that is normally distributed with: n=2000, mean=35, sd=3.5. In this distribu tion, 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)

```
rdata <- data.frame(norm = rnorm(n = 2000, mean = 35, sd = 3.5))
ggplot(rdata,aes(norm)) +
  ggtitle("n=2000, mean=35, sd=3.5")+
  xlab("rdata")+
  geom_density(size = 1)+
  geom_vline(aes(xintercept = mean(rdata$norm),color="Mean"),size=1,sho
w.legend  =F,lty =1)+geom_vline(aes(xintercept = mean(rdata$norm)-1*sd
(rdata$norm),color="1st standard deviations"),size=1,show.legend  =F,lt
y =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)-2*sd(rdata$norm),color="
2nd standard deviations"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)-3*sd(rdata$norm),color="
3rd standard deviations"),size=1,show.legend  =FALSE,lty =2 )+
  geom_text(aes(x=mean(rdata$norm)-sd(rdata$norm), label="-σ",y=0.02))+
  geom_text(aes(x=mean(rdata$norm)-2*sd(rdata$norm), label="-2σ",y=0.0
2))+
  geom_text(aes(x=mean(rdata$norm)-3*sd(rdata$norm), label="-3σ",y=0.0
2))+
  geom_vline(aes(xintercept = mean(rdata$norm)+sd(rdata$norm),color="1s
t standard deviations"),size=1,show.legend  =FALSE,lty =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)+2*sd(rdata$norm),color="
2nd standard deviations"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)+3*sd(rdata$norm),color="
3rd standard deviations"),size=1,show.legend  =F,lty =2 )+
  geom_text(aes(x=mean(rdata$norm)+sd(rdata$norm), label="+σ",y=0.02))+
  geom_text(aes(x=mean(rdata$norm)+2*sd(rdata$norm), label="+2σ",y=0.0
2))+
```

```
  geom_text(aes(x=mean(rdata$norm)+3*sd(rdata$norm), label="+3σ",y=0.0
2))+
geom_text(aes(x=mean(rdata$norm), label="Mean",y=0.02))+
  geom_vline(aes(xintercept = mean(rdata$norm),color="Mean"),size=1,sho
w.legend  =F,lty =1)+
  geom_vline(aes(xintercept = quantile(rdata$norm,0.25),color="1st quar
tiles"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = quantile(rdata$norm,0.5),color="2nd quart
iles"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = quantile(rdata$norm,0.75),color="3rd quar
tiles"),size=1,show.legend  =F,lty =2 )+
  geom_text(aes(x=quantile(rdata$norm,0.5), label="50%",y=0.05))+
  geom_text(aes(x=quantile(rdata$norm,0.25), label="25%",y=0.05))+
  geom_text(aes(x=quantile(rdata$norm,0.75), label="75%",y=0.05))
```




```
(quantile(rdata$norm,0.25)-mean(rdata$norm))/sd(rdata$norm)
```

```
##      25%
## -0.66826
```

The 1st quartile is about -0.67 standard deviation away from the mean.

```
(quantile(rdata$norm,0.5)-mean(rdata$norm))/sd(rdata$norm)
```

```
##          50%
## -0.01573141
```

The 2nd quartile is about -0.02 standard deviation away from the mean.

```
(quantile(rdata$norm,0.75)-mean(rdata$norm))/sd(rdata$norm)
```

```
##       75%
## 0.6234469
```

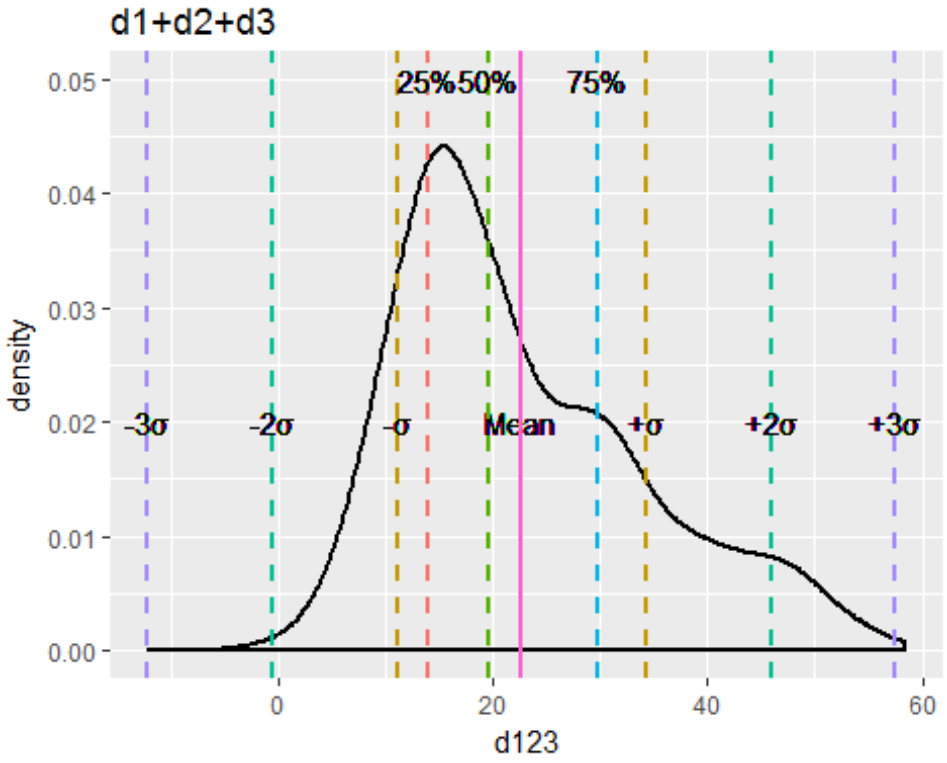
The 3rd quartile is about 0.62 standard deviation away from the mean.

In contrast with the answer to (b), their answer are quite similar. Because both of the "rdata" in each sub-question are typical normal distribution, the 1st quartile might be about -0.68 standard deviation away from the mean, the 2nd one approximately overlaps the mean, and the 3rd one is about 0.68 away from the mean (Because we use simulation, there might exist some empirical errors). In addition, from the figure above, which already marked out the quartiles, we can see that the distance are almost the same between each of them.

(d) Finally, recall the dataset d123 shown in qu estion 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)

```
rdata <- data.frame(norm = d123)

ggplot(rdata,aes(norm)) +
  ggtitle("d1+d2+d3")+
  xlab("d123")+
  geom_density(size = 1)+
  geom_vline(aes(xintercept = mean(rdata$norm),color="Mean"),size=1,sho
w.legend  =F,lty =1)+geom_vline(aes(xintercept = mean(rdata$norm)-1*sd
(rdata$norm),color="1st standard deviations"),size=1,show.legend  =F,lt
y =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)-2*sd(rdata$norm),color="
2nd standard deviations"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)-3*sd(rdata$norm),color="
3rd standard deviations"),size=1,show.legend  =FALSE,lty =2 )+
  geom_text(aes(x=mean(rdata$norm)-sd(rdata$norm), label="-σ",y=0.02))+
  geom_text(aes(x=mean(rdata$norm)-2*sd(rdata$norm), label="-2σ",y=0.0
2))+
  geom_text(aes(x=mean(rdata$norm)-3*sd(rdata$norm), label="-3σ",y=0.0
2))+
  geom_vline(aes(xintercept = mean(rdata$norm)+sd(rdata$norm),color="1s
t standard deviations"),size=1,show.legend  =FALSE,lty =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)+2*sd(rdata$norm),color="
2nd standard deviations"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = mean(rdata$norm)+3*sd(rdata$norm),color="
3rd standard deviations"),size=1,show.legend  =F,lty =2 )+
  geom_text(aes(x=mean(rdata$norm)+sd(rdata$norm), label="+σ",y=0.02))+
  geom_text(aes(x=mean(rdata$norm)+2*sd(rdata$norm), label="+2σ",y=0.0
2))+
  geom_text(aes(x=mean(rdata$norm)+3*sd(rdata$norm), label="+3σ",y=0.0
2))+
geom_text(aes(x=mean(rdata$norm), label="Mean",y=0.02))+
```

```
  geom_vline(aes(xintercept = mean(rdata$norm),color="Mean"),size=1,sho
w.legend  =F,lty =1)+
  geom_vline(aes(xintercept = quantile(rdata$norm,0.25),color="1st quar
tiles"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = quantile(rdata$norm,0.5),color="2nd quart
iles"),size=1,show.legend  =F,lty =2)+
  geom_vline(aes(xintercept = quantile(rdata$norm,0.75),color="3rd quar
tiles"),size=1,show.legend  =F,lty =2 )+
  geom_text(aes(x=quantile(rdata$norm,0.5), label="50%",y=0.05))+
  geom_text(aes(x=quantile(rdata$norm,0.25), label="25%",y=0.05))+
  geom_text(aes(x=quantile(rdata$norm,0.75), label="75%",y=0.05))
```




```
(quantile(rdata$norm,0.25)-mean(rdata$norm))/sd(rdata$norm)
```

```
##        25%
## -0.7447283
```

The 1st quartile is about -0.74 standard deviations away from the mean.

```
(quantile(rdata$norm,0.5)-mean(rdata$norm))/sd(rdata$norm)
```

```
##        50%
## -0.2601399
```

The 2nd quartile is about -0.26 standard deviations away from the mean.

```
(quantile(rdata$norm,0.75)-mean(rdata$norm))/sd(rdata$norm)
```

```
##       75%
## 0.6168632
```

The 3rd quartile is about 0.62 standard deviations away from the mean. In contrast with the answer to (b), since d123 is a composite distribution, the answers are not gonna be the same. Moreover, the distribution is asymmetric excessively. Therefore, the median (2nd quartile) is much differ from the mean.

Question 3

(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?

Rob Hyndman suggests to use Freedman–Diaconis rule ($h = 2 * IQR * n^{-1/3}$) to calculate The bin-width. Also, we can use the bin-with we get to compute bin-number $k = \left\lceil \frac{\max x - \min x}{h} \right\rceil$. The benefit of Freedman–Diaconis rule is that we use IQR here instead of standard deviation, which shows in Scott's rule, and that make the formula less sensitive 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)
rand_data <- rnorm(800, mean=20, sd = 5)

#Sturges' formula
k1 = ceiling(log2(length(rand_data)))+1
h1 = (max(rand_data) - min(rand_data)) / k1

#Scott's normal reference rule (uses standard deviation)
h2 = 3.5*sd(rand_data) / length(rand_data)^(1/3)
k2 = ceiling((max(rand_data) - min(rand_data))/h2)

#Freedman-Diaconis' choice (uses IQR)
h3 = 2 * IQR(rand_data) / length(rand_data)^(1/3)
k3 = ceiling((max(rand_data) - min(rand_data))/h3)
cat ("Sturges' formula   bin widths:",round(h1,4)," number of bins:",ro
und(k1,4))

## Sturges' formula   bin widths: 2.9608  number of bins: 11

cat ("Scott's normal reference rule   bin widths:",round(h2,4)," number
 of bins:",round(k2,4))
```

```
## Scott’s normal reference rule   bin widths: 1.9291  number of bins:
17

cat ("Freedman-Diaconis’ choice   bin widths:",round(h3,4)," number of
bins:",round(k3,4))

## Freedman-Diaconis’ choice   bin widths: 1.4192  number of bins: 23
```

Sturges’ formula {bin widths: 2.9608, number of bins: 11}

Scott’s normal reference rule {bin widths: 1.9291 , number of bins: 17}

Freedman-Diaconis’ choice {bin widths: 1.4192, number of bins: 23}

(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))
#Sturges’ formula
k1 = ceiling(log2(length(out_data)))+1
h1 = (max(out_data) - min(out_data)) / k1

#Scott’s normal reference rule (uses standard deviation)
h2 = 3.5*sd(out_data) / length(out_data)^(1/3)
k2 = ceiling((max(out_data) - min(out_data))/h2)

#Freedman-Diaconis’ choice (uses IQR)
h3 = 2 * IQR(out_data) / length(out_data)^(1/3)
k3 = ceiling((max(out_data) - min(out_data))/h3)
cat ("Sturges’ formula   bin widths:",round(h1,4)," number of bins:",ro
und(k1,4))

## Sturges’ formula   bin widths: 5.2104  number of bins: 11

cat ("Scott’s normal reference rule   bin widths:",round(h2,4)," number
 of bins:",round(k2,4))

## Scott’s normal reference rule   bin widths: 2.3638  number of bins:
25

cat ("Freedman-Diaconis’ choice   bin widths:",round(h3,4)," number of
bins:",round(k3,4))

## Freedman-Diaconis’ choice   bin widths: 1.4474  number of bins: 40
```

Sturges’ formula {bin widths: 5.2104 number of bins: 11}

Scott’s normal reference rule {bin widths: 2.3638 , number of bins: 25}

Freedman-Diaconis’ choice {bin widths: 1.4474, number of bins: 40}

(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?

By using Freedman-Diaconis' choice, the bin width change the least among the three methods when outliers are added. In addition, deriving from the formula of the three methods, we can have more insight on why Freedman-Diaconis' choice seems to be least sensitive to outliers. Firstly, we have already knew that Freedman-Diaconis' rule is least sensitive to outliers than Scott's normal reference rule since the latter one include standard deviation into the formula, which counts on all the value in the dataset. Moreover, the number of bins in Sturges' method changes minimally. However, when computing the bin width, the formula $h = \frac{\max x - \min x}{k}$ will greatly affect by the maximum or minimum. Not to mention, those maximum and minimum are likely to be outliers. Therefore, Freedman-Diaconis' choice might be the stablest approch among the three.