Assignment 2: Visualization and Summary Statistics

STATS 101

Please upload the `PDF` that you obtain by knitting the `Rmd` file that contains your `R` code and your text answering other questions. So this uploaded file will also show any output that `R` produces in addition to your code.

Part 1. A disguising plot

In this exercise, we will be comparing three types of plots for the same set of data. You will explore how different visualization techniques display the data, and describe your conclusions on which plot you think makes the most sense for this specific illustrating dataset.

We will beginning by generating some data from a Gaussian mixture model For those who are curious, you can read more about what the Gaussian mixture model is here (https://brilliant.org/wiki/gaussian-mixture-model/).

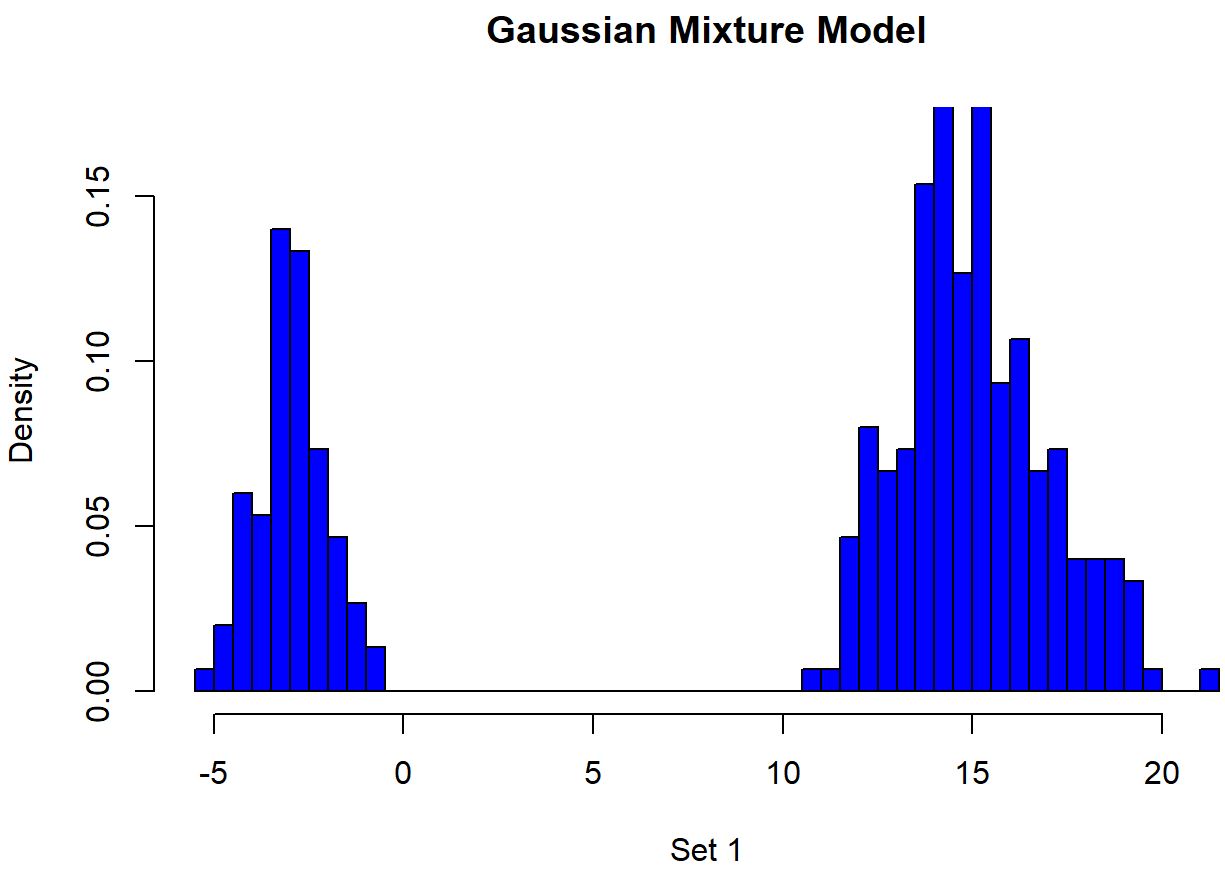
```
## set the seed for reproducibility
set.seed(123)
n = 300
num_in_cluster_1 = rbinom(1, size=n, prob=.3)
num_in_cluster_2 = n - num_in_cluster_1
data_cluster1 = rnorm(num_in_cluster_1, mean=-3, sd=1)
data_cluster2 = rnorm(num_in_cluster_2, mean=15, sd=2)
data = c(data_cluster1, data_cluster2)
head(data)
```

```
## [1] -3.230177 -1.441292 -2.929492 -2.870712 -1.284935 -2.539084
```

Exercise 1a.

Visualize `data` using a density histogram with 50 equally sized bins. You should give your plot appropriate title and axis labels. Briefly describe the data using a couple of sentences (i.e. what patterns do you observe?).

```
### YOUR CODE HERE
densityHistogram = hist(data, main = 'Gaussian Mixture Model',xlab = 'Set 1', labels = FALSE,
     col = 'blue', freq = FALSE, ylim = c(0,0.17), breaks = 50)
```



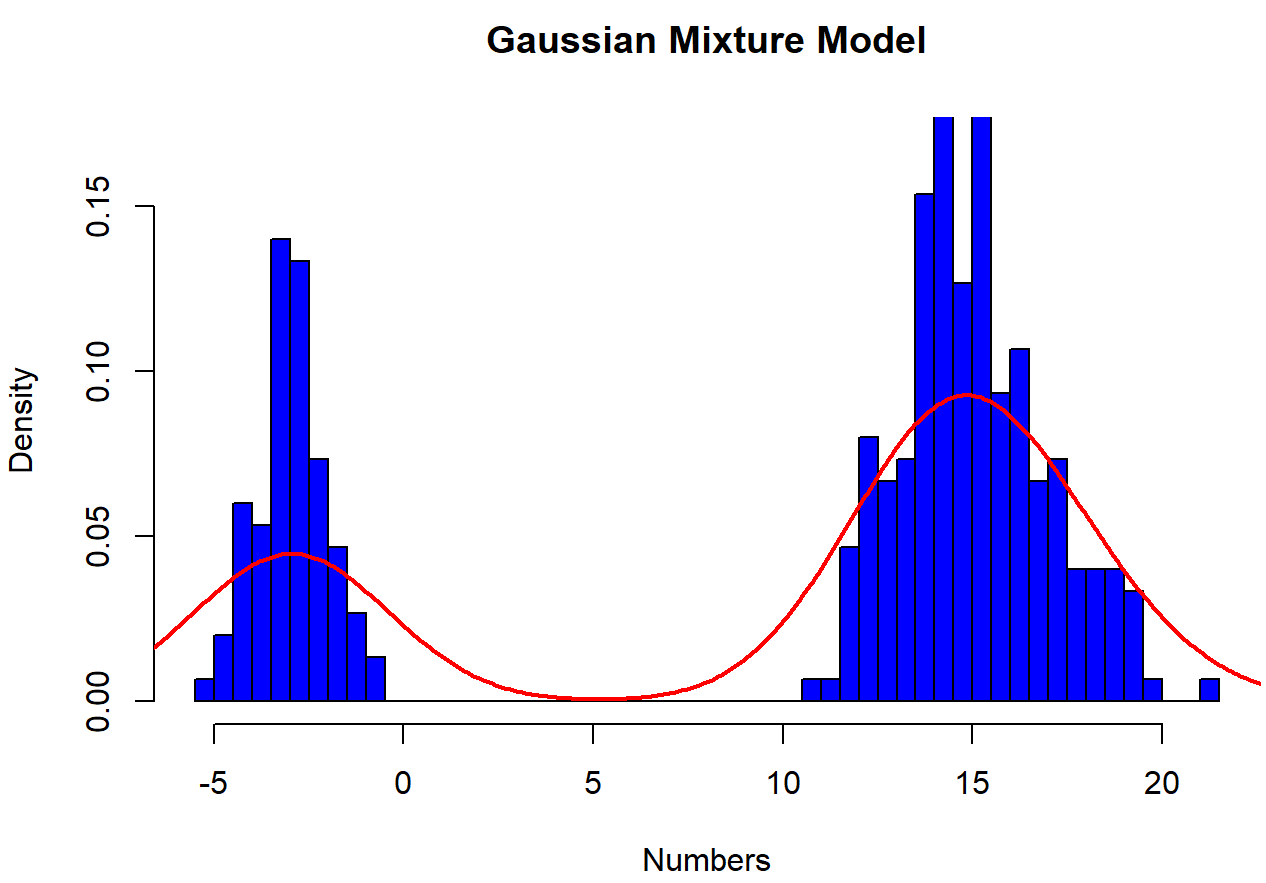

```
### END OF YOUR CODE
```

YOUR EXPLANATION HERE. #The data is bimodal and each section is rougly symetric.

Exercise 1b.

Now, add a density curve for `data` overlaying on top of your histogram using the default value for the bandwidth parameter either by not explicitly specifying the `bw` argument or passing in `bw = nrd0` using the `gaussian` kernel.

```
### YOUR CODE HERE
hist(data, main = 'Gaussian Mixture Model',xlab = 'Numbers', labels = FALSE,
     col = 'blue',
     freq = FALSE,
     ylim = c(0,0.17),
     breaks = 50)
lines(density(data, kernel = 'gaussian'), main = 'Gaussian Mixture Model', lwd = 2, col = 'red')
```



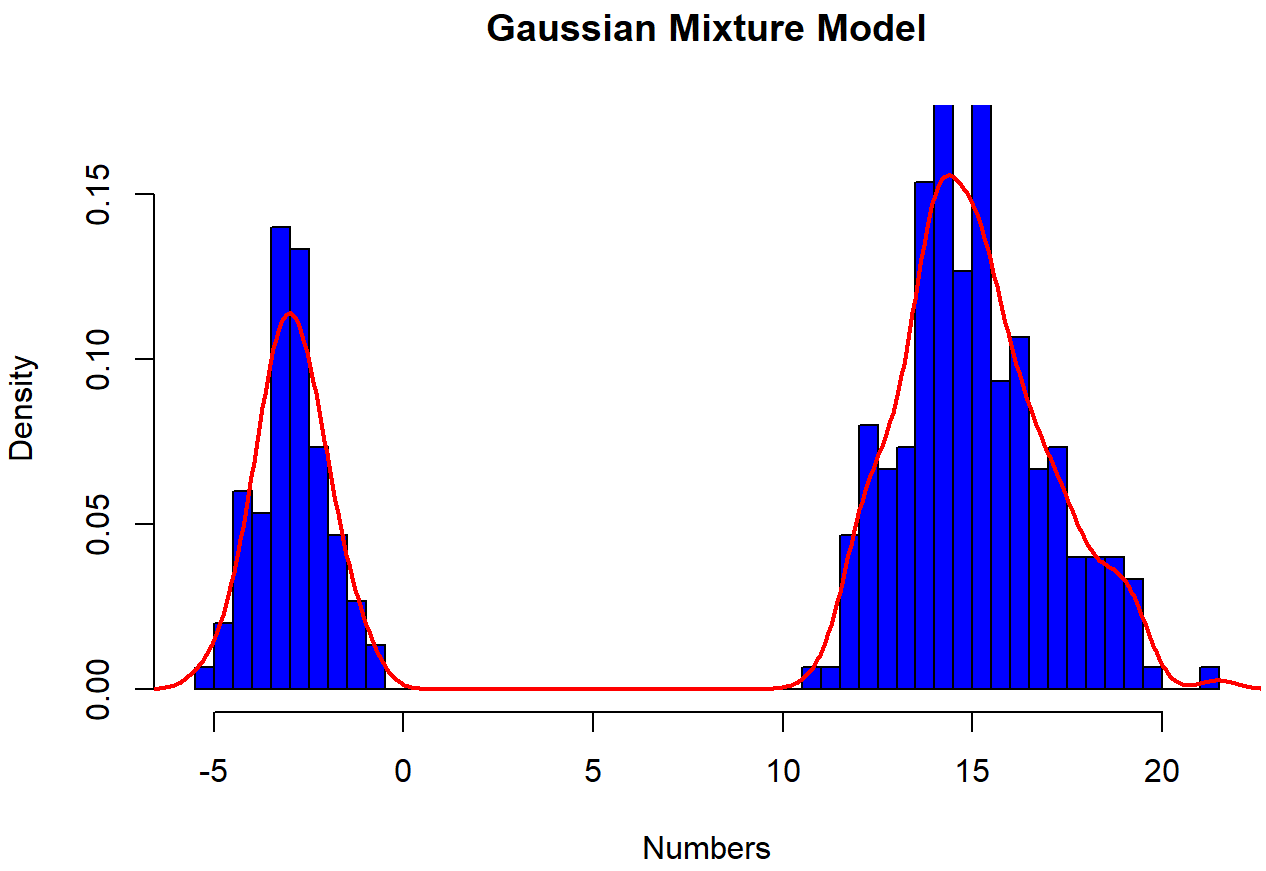

```
### END OF YOUR CODE
```

Exercise 1c.

Instead of using the default bandwidth as in (b), we will now input a specific value for the bandwidth parameter. Create a new density histogram plot using `data` with 50 bins and overlay it with a density curve with the `gaussian` kernel and bandwidth set to 0.5.

Describe what you observe from comparing the resulting plots from (b) and (c).

```
#R CODE HERE
densityHistogram = hist(data, main = 'Gaussian Mixture Model',xlab = 'Numbers', labels = FALSE,
     col = 'blue', freq = FALSE, ylim = c(0,0.17), breaks = 50)
lines(density(data, kernel = 'gaussian', bw=0.5), main = 'Gaussian Mixture Model', lwd = 2, col
= 'red')
```




```
### END OF YOUR CODE
```

YOUR EXPLANATION HERE. #Reducing the bandwidth helps capture more of the peaks and fits the data a bit better.

Optional reading: comparing density estimation fits

How do we quantify which of these density curves fit to the data better? We can evaluate how “fitting” a kernel density estimator (the density curve that you plotted) is using some goodness of fit test, for example, the two-sample Kolmogorov-Smirnov test. The null hypothesis of Kolmogorov-Smirnov test states that there is no difference between the two distributions; in our case, this means the fit of the KDE is adequate.

We will first generate some samples from the kernel density estimator (KDE) we fitted to our data.

```
## extract default bw
kde_gauss = density(data, kernel='gaussian')
default_bw = kde_gauss$bw
custom_bw = 0.5

gauss_kernel <- function(n, bw) {
    rnorm(n, sd=bw)
}

bootstrap_sample = sample(data, n, replace=T)
kde_sample_default_bw = bootstrap_sample + gauss_kernel(n, default_bw)
kde_sample_custom_bw = bootstrap_sample + gauss_kernel(n, custom_bw)
```

Let's print the resulting p-values from the Kolmogorov-Smirnov test comparing the default bandwidth KDE generated samples to our actual data:

```
ks.test(data, kde_sample_default_bw)
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  data and kde_sample_default_bw
## D = 0.12667, p-value = 0.01624
## alternative hypothesis: two-sided
```

Since the p-value is $0.034 < 0.05 = \alpha$, we reject the null hypothesis that the fit is adequate with 95% confidence.

Let's now print the resulting p-values from the Kolmogorov-Smirnov test comparing the generated samples from a KDE with custom bandwidth=0.4 to our actual data:

```
ks.test(data, kde_sample_custom_bw)
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  data and kde_sample_custom_bw
## D = 0.043333, p-value = 0.9408
## alternative hypothesis: two-sided
```

Since the p-value is $0.721 > 0.05 = \alpha$, we fail reject the null hypothesis and conclude that the fit is adequate with 95% confidence.

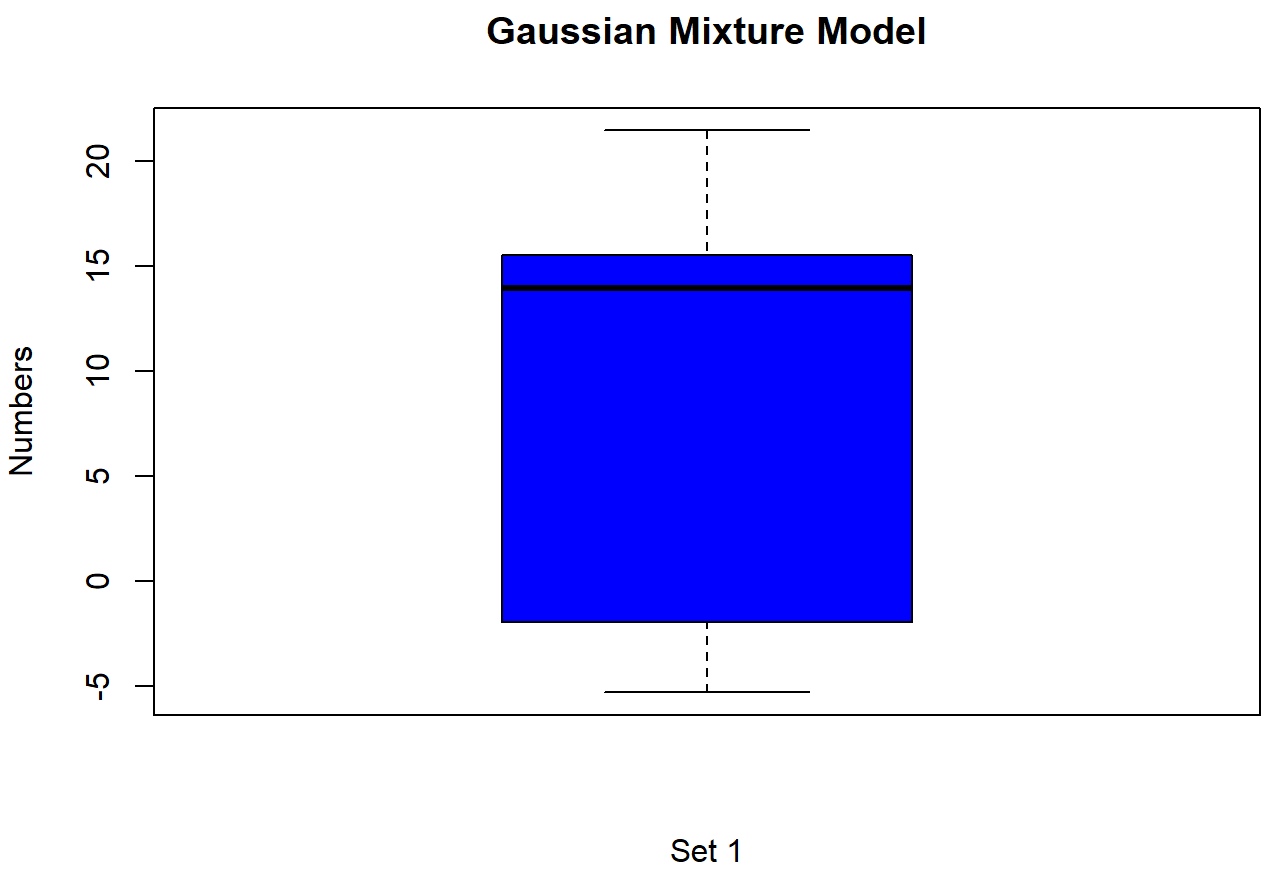
Thus, it seems like our customly chosen 0.5 bandwidth fits to the data better according to the Kolmogorov-Smirnov test than the default bandwidth of 2.4.

Exercise 1d.

In the last subpart of Exercise 1, we will visualize this dataset using a boxplot.

Visualize `data` below using a boxplot. You should give your plot appropriate title and axis labels. Briefly describe what you observe from the histogram from (a) and the boxplot from (d). Which plot do you think is more reasonable for this data and why?

```
### YOUR CODE HERE
boxplot(data, main = "Gaussian Mixture Model", xlab = "Set 1", ylab = "Numbers", col = 'blue')
```




```
### END OF YOUR CODE
```

YOUR EXPLANATION HERE. # The boxplot is unable to emhasize the bimodality of the distribution and as a result the histogram would be a better representation as we can more clearly see the distribution.

Part 2. Some more density curves

Exercise 2a. Prelude

In this warm up exercise, you will directly visualize the importance of choosing an suitable bandwidth for the density curve. First, we will start by generating 1000 samples from a Student t's distribution with 3 degrees of freedom. You will be using the function, `rt` . You should avoid naming your samples `x` due to naming conflicts with the starter code.

Hint. You can begin by looking up the helper page for `rt` .

Next, we will plot the density of actual Student t's distribution with 3 degrees of freedom. This is already done for you in the starter code.

Finally, you will plot the smoothed density curve from your generated samples using a suitable bandwidth parameter selected by you. You will overlay this density curve on top of the plot generated by the starter code displaying the true distribution. Your curve should be in another color for discernibility; set `lwd = 2` for your curve to make the line thicker.

Remark. You can read more about bandwidth selection for density curves here (https://aakinshin.net/posts/kde-bw/).

```
?rt
```

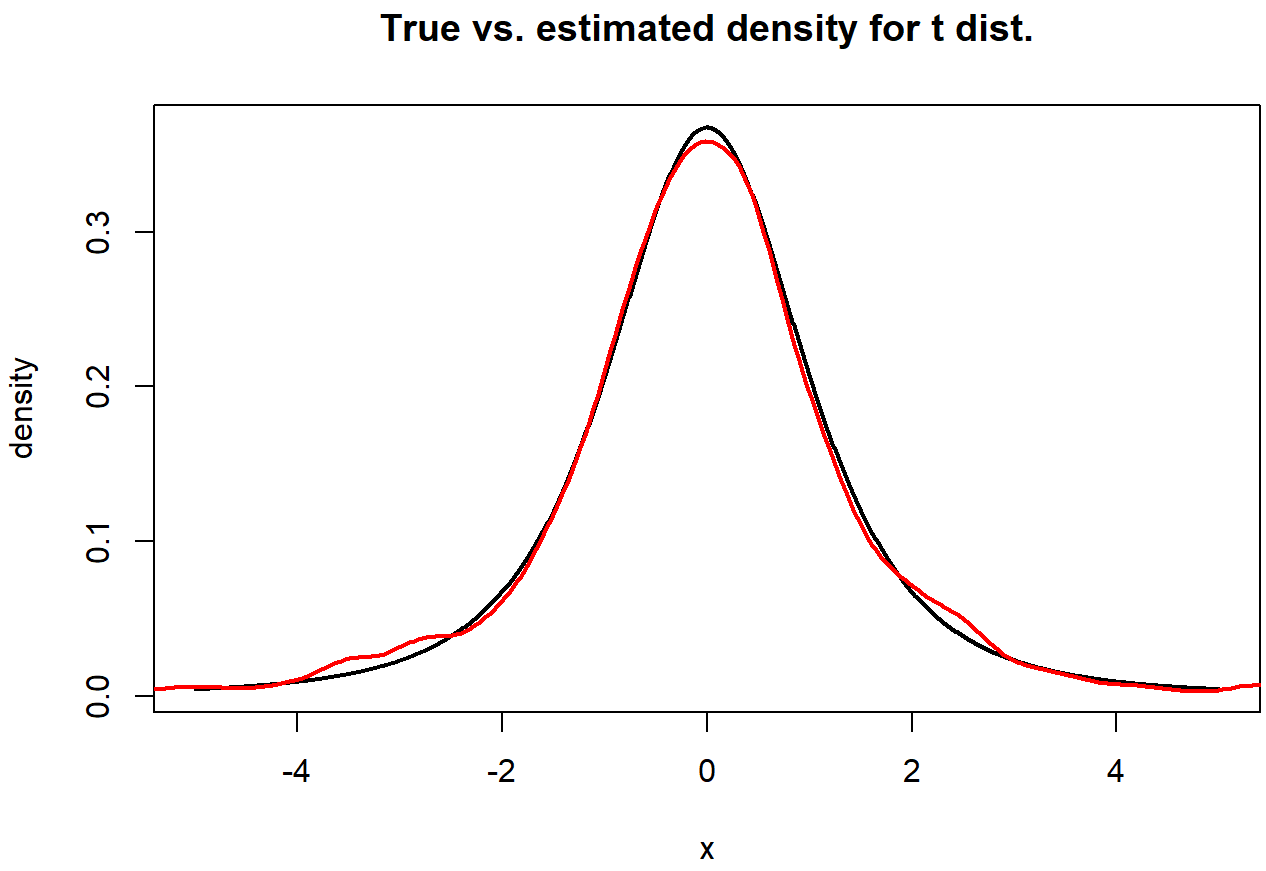
```
## starting httpd help server ... done
```

```
num_samples = 1000

### YOUR CODE HERE
samples = rt(num_samples, 3)
### END OF YOUR CODE

## plotting density of t dist
x = seq(-5, 5, length.out = num_samples)
y = dt(x, df=3)
graph = plot(x, y, type='l', lwd=2,
     main='True vs. estimated density for t dist.',
     xlab='x',
     ylab='density')
###

### YOUR CODE HERE
lines(density(samples, kernel = 'gaussian', bw = bw.nrd0(samples)), col = 'red', lwd = 2)
```




```
### END OF YOUR CODE
```

Exercise 2b.

Compute the mean, median, and standard deviation of your generated samples in (a) and include them below. Is the observed sample mean close to the observed sample median? What does that tell you? Does your intuition align with the density plot you generated?

```
### YOUR CODE HERE
print('Mean:')
```

```
## [1] "Mean:"
```

```
mean(samples)
```

```
## [1] -0.00901772
```

```
print('Median:')
```

```
## [1] "Median:"
```

```
median(samples)
```

```
## [1] -0.02134512
```

```
print('Standard Deviation:')
```

```
## [1] "Standard Deviation:"
```

```
sd(samples)
```

```
## [1] 1.626226
```

```
### END OF YOUR CODE
```

YOUR EXPLANATION HERE. The sample mean and median are very close together which indicates that the graph has no strong skew and is roughly symmetric. This intuition is consistent with the graph.

Exercise 2c. How many clusters of galaxies?

Now, once we are all warmed up, we are ready to proceed with into subsequent steps analyzing galaxy data. The galaxy data from Roeder (1990) Density Estimation With Confidence Sets Exemplified by Superclusters and Voids in the Galaxies (https://www.jstor.org/stable/2289993?seq=1#page_scan_tab_contents) is available in the `R` library `MASS`.

First, we will install the `MASS` library e.g. via `install.packages()` or simply saving this file and clicking on the `install` pop-up message.

The data are the measured velocities in km/second of 82 galaxies from the Corona Borealis region.

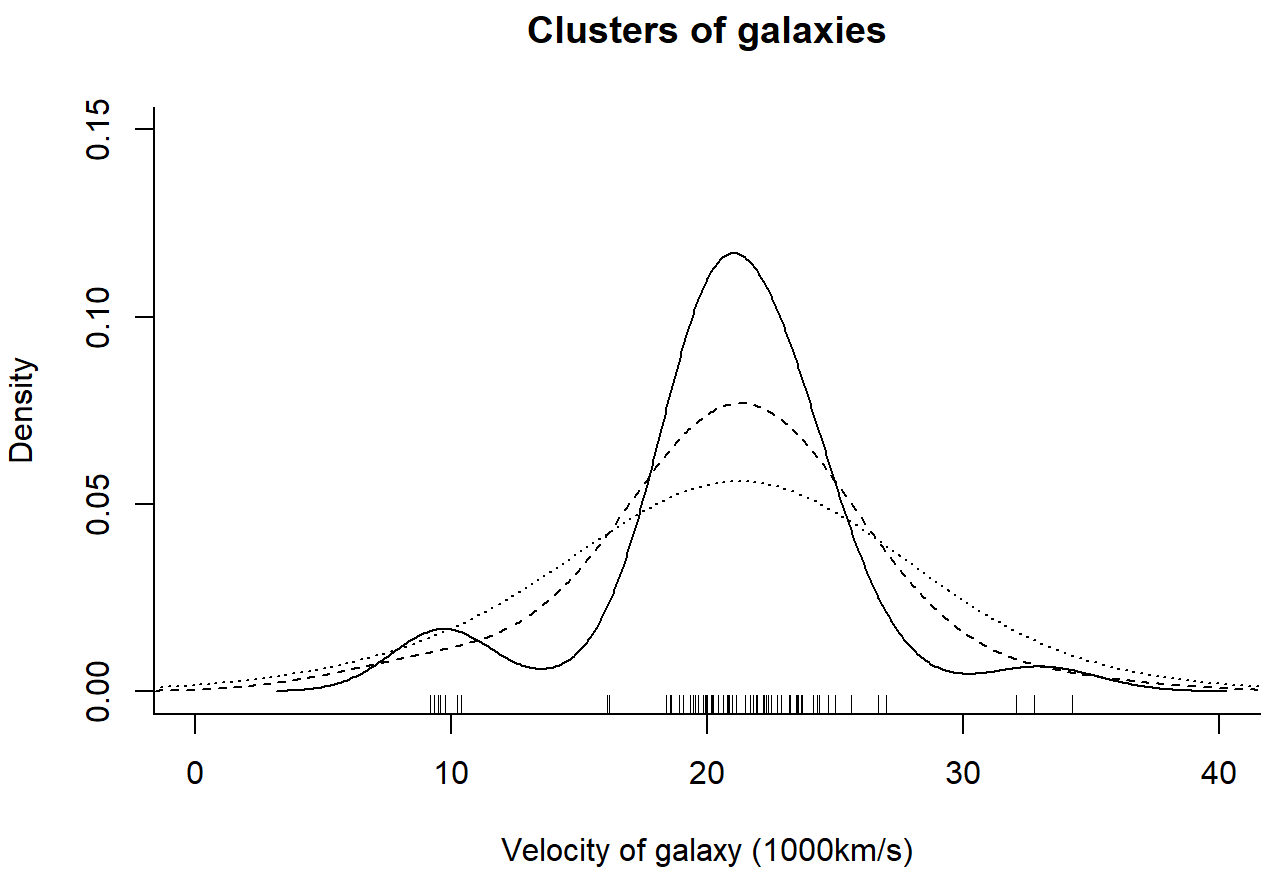
```
require(MASS)
```

```
## Loading required package: MASS
```

```
gal <- galaxies/1000
plot(x = c(0, 40), y = c(0, 0.15), type = "n", bty = "l",
     main='Clusters of galaxies',
     xlab = "Velocity of galaxy (1000km/s)", ylab = "Density")

## add a 'rug' (ticks along x axis)
rug(gal)

## lty controls y's appearance
lines(density(gal, bw = 6), lty = 3)
lines(density(gal, bw = 4), lty = 2)
lines(density(gal, bw = 2), lty = 1)
```



A theory of how the universe formed predicts the existence of clusters of galaxies. If galaxies travel at similar speeds, then the galaxies are clumped. This suggests that multimodal data corresponds to multiple clusters of galaxies. A unimodal distribution, by contrast, is what one would expect if there were no clusters and the data were just an artifact of how the galaxies were sampled.

Therefore, we are specifically interested in the number of modes (i.e. local maxima of the density) in this data.

The plot above includes 3 different density estimates for the data, using 3 different levels of smoothing as controlled by the bandwidth parameter.

Which bandwidth provides the clearest evidence that the galaxies are clustered? Which bandwidth would you choose to best represent the data and why?

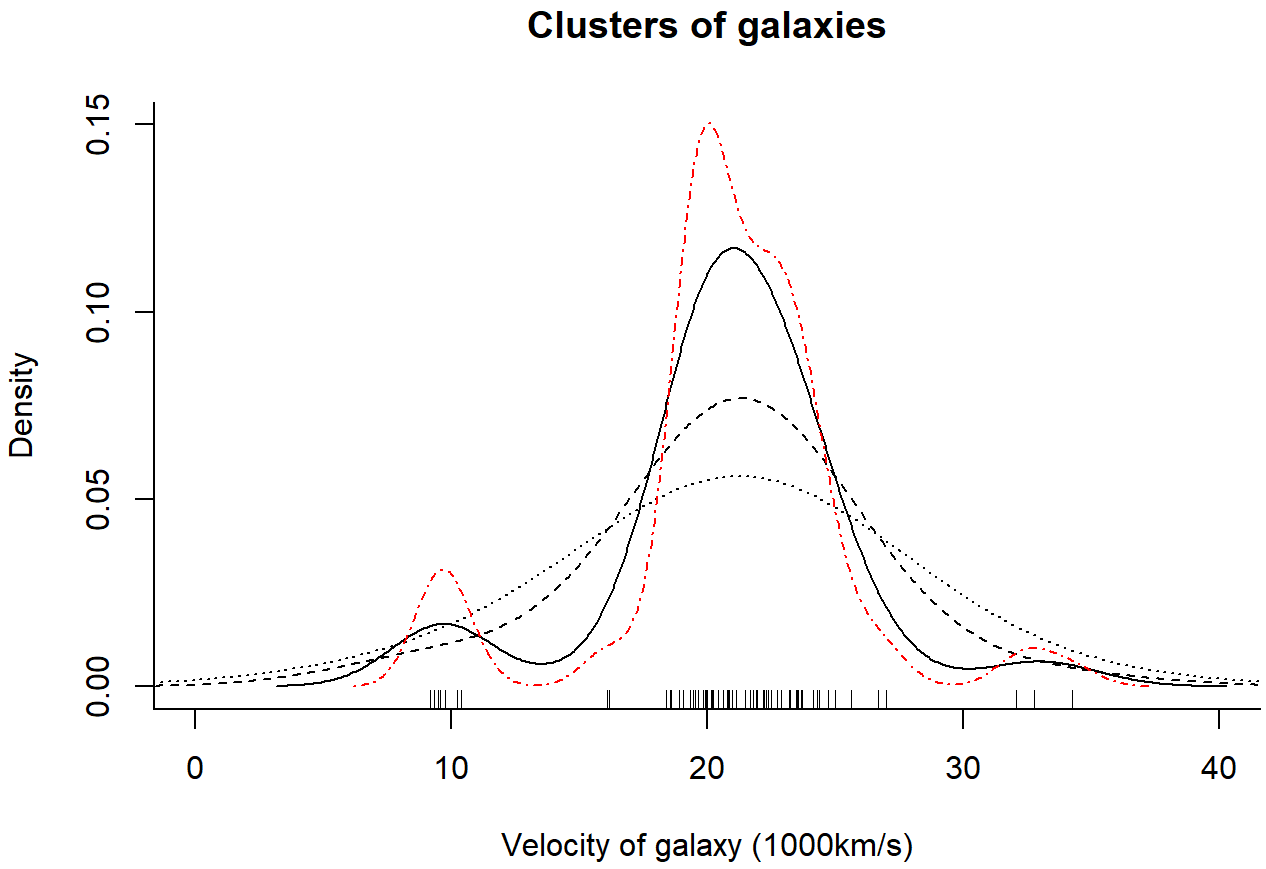
YOUR EXPLANATION HERE. #The line with bandwidth of 2 best indicates that the galaxies are clustered. #I would choose bandwidth 2 in this case to reflect the data as it more acurately captures the spread without having excessive variability.

Exercise 2d. How many clusters of galaxies, continued.

Adapt the code above and add 4th density curve that is less ‘smooth’ than any of present ones in the plot. Your density curve should be in red for discernibility.

```
plot(x = c(0, 40), y = c(0, 0.15), type = "n", bty = "l",
     main='Clusters of galaxies',
     xlab = "Velocity of galaxy (1000km/s)", ylab = "Density")
rug(gal)
lines(density(gal, bw = 6), lty = 3)
lines(density(gal, bw = 4), lty = 2)
lines(density(gal, bw = 2), lty = 1)

### YOUR CODE HERE
lines(density(gal,bw = 1), lty = 4, col = 'red')
```




```
### END OF YOUR CODE
```

Part 3. Groundhog Day

On Groundhog Day, February 2, a famous groundhog in Punxsutawney, PA is used to predict whether a winter will be long or not based on whether or not it sees its shadow.

Optional: an aside. According to Wikipedia, “[This tradition] derives from the Pennsylvania Dutch superstition that if a groundhog emerges from its burrow on this day and sees its shadow, it will retreat to its den and winter will go on for six more weeks; if it does not see its shadow, spring will arrive early. This is often due to the weather being cloudy or clear allowing for the groundhog to actually have a shadow or not.” You can read more about the tradition and view historical data here (http://www.stormfax.com/ghogday.html). A subset of the data on whether he saw his shadow or not is in this table (http://stats191.stanford.edu/data/groundhog.table).

Although the groundhog (named Phil) is on the East Coast, we are interested in whether this information says anything about whether or not we will experience a rainy winter out here in California. For this, we will be looking at the rainfall data, stored in a table here (http://stats191.stanford.edu/data/rainfall.csv).

Excerise 3a.

To answer the question, we will first visualize the data accordingly and identify any patterns of possible interest to us.

We will make a 2 side-by-side boxplots of the average monthly precipitation (total annual rainfall / 12) in Northern California for 1) years in which Phil sees its shadow, versus 2) years in which Phil does not see its shadow.

Hint. To compute the average precipitation, check out either the `rowSums` function or the `apply` function. To join the two dataframes together, check out the `left_join` function from the `dplyr` library. You should be very careful which dataset comes first in the `left_join` arguments and on which columns you perform the joining operation. To plot multiple boxplots side-by-side, check out the `~` operator.

```
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
shadow_url="http://web.stanford.edu/class/stats191/data/groundhog.table"
rain_url="http://web.stanford.edu/class/stats191/data/rainfall.csv"

shadow_data = read.table(shadow_url, header=TRUE, sep=',')
## see which columns are in the shadow dataset
names(shadow_data)
```

```
## [1] "year"    "mintemp" "shadow"
```

```
head(shadow_data)
```

```
##   year mintemp shadow
## 1 1990      24      N
## 2 1991      23      Y
## 3 1992      22      Y
## 4 1993      16      Y
## 5 1994      12      Y
## 6 1995      13      N
```

```
## convert `shadow` into a factor
shadow_data$shadow = as.factor(shadow_data$shadow)

### YOUR CODE HERE
#Make the rain data set
rain_data = read.csv(rain_url, header = TRUE, sep = ',')
names(rain_data)
```

```
##  [1] "WY"    "Oct"   "Nov"   "Dec"   "Jan"   "Feb"   "Mar"   "Apr"   "May"
## [10] "Jun"   "Jul"   "Aug"   "Sep"   "Total"
```

```
head(rain_data)
```

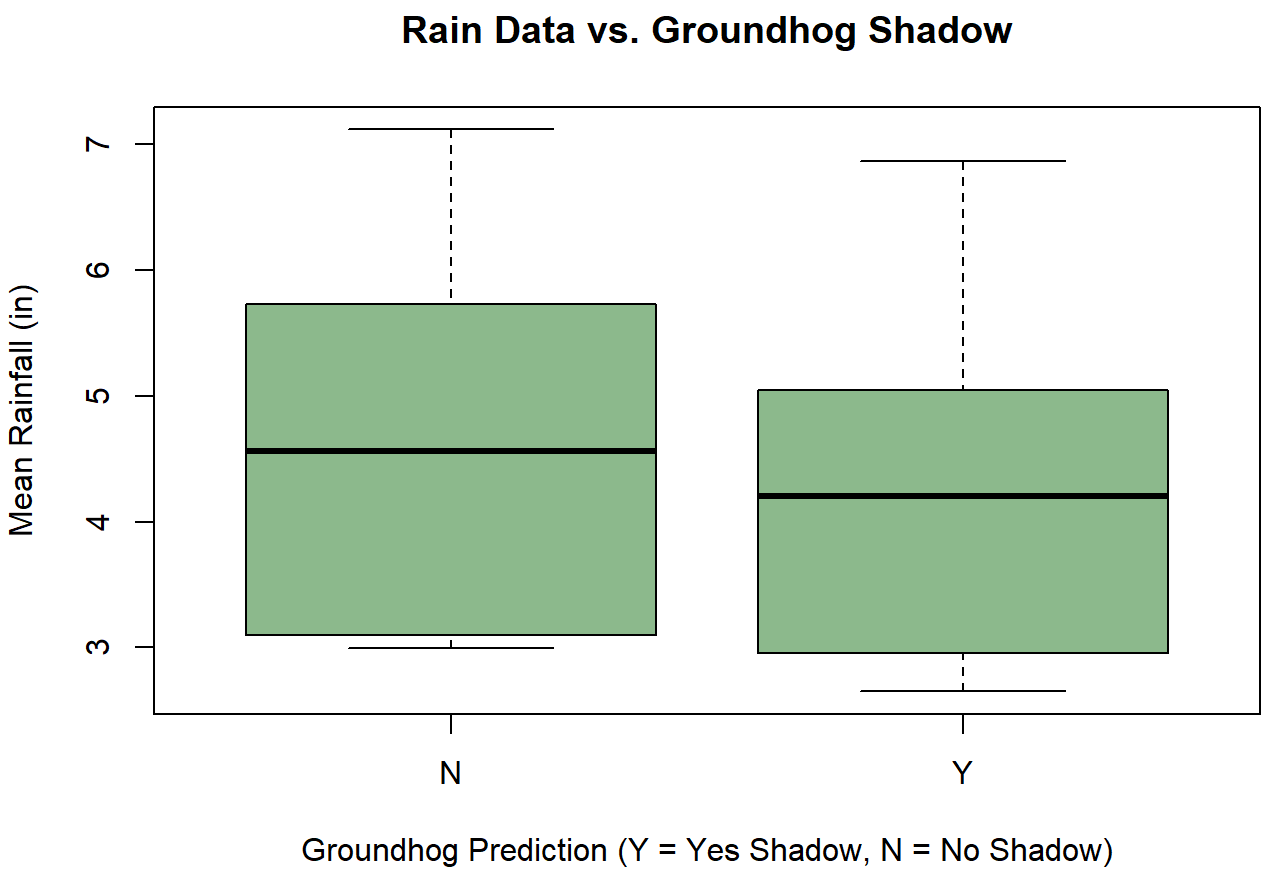
```
##     WY  Oct   Nov   Dec   Jan   Feb  Mar  Apr  May  Jun  Jul  Aug  Sep Total
## 1 1921 5.25 12.38 11.52 13.12  3.76 5.30 0.94 3.05 0.65 0.00 0.00 0.02 55.99
## 2 1922 1.39  3.16 11.22  3.21 14.42 8.37 1.58 2.22 0.98 0.14 0.08 0.01 46.78
## 3 1923 3.59  6.01 11.79  5.95  1.93 0.49 6.86 0.93 2.09 0.20 0.40 2.75 42.99
## 4 1924 2.15  0.46  2.77  3.55  3.94 2.67 0.89 0.05 0.08 0.00 0.14 0.40 17.10
## 5 1925 6.63  4.71  6.01  3.47 15.21 4.51 5.46 2.14 1.52 0.11 0.83 2.45 53.05
## 6 1926 1.90  3.53  2.88  5.63 11.55 0.61 6.45 1.49 0.00 0.00 0.24 0.08 34.36
```

```
#create average rainfall column
rain_data$avg = rowMeans(rain_data[,2:13])

#Combine the data sets
combined_data = dplyr::left_join(shadow_data, rain_data, by = join_by(year==WY))
combined_data
```

```
##    year mintemp shadow  Oct   Nov   Dec   Jan   Feb   Mar   Apr  May  Jun  Jul
## 1  1990      24      N 6.29  3.08  0.07  7.88  4.60  3.19  1.83 7.55 0.29 0.12
## 2  1991      23      Y 1.10  1.56  1.39  0.88  3.11 17.94  1.81 2.51 1.20 0.40
## 3  1992      22      Y 3.44  2.64  3.97  2.81 11.84  5.35  2.67 0.12 2.73 0.20
## 4  1993      16      Y 4.77  1.15 14.41 16.53 10.26  7.57  3.66 4.27 2.31 0.00
## 5  1994      12      Y 2.65  2.81  7.34  4.10  7.62  1.53  2.87 1.96 0.32 0.01
## 6  1995      13      N 0.77  8.62  7.57 27.14  1.94 22.85  8.53 4.62 3.05 0.30
## 7  1996      17      Y 0.01  0.57 13.90 12.44 14.07  5.88  5.86 6.97 0.48 0.14
## 8  1997      25      N 2.53  7.85 28.89 18.95  0.97  2.25  2.38 0.75 2.37 0.19
## 9  1998      28      Y 3.25  9.45  4.79 18.82 21.22  8.63  5.40 7.47 2.02 0.02
## 10 1999      22      N 1.51 12.69  4.69 10.03 15.15  5.45  3.24 0.93 0.54 0.10
## 11 2000      19      Y 2.72  6.78  1.53 14.31 19.08  3.66  3.66 2.54 0.82 0.21
## 12 2001      21      Y 4.71  1.86  2.62  5.05  9.45  3.76  3.71 0.11 0.87 0.03
## 13 2002      22      Y 1.82 10.93 14.31  5.41  3.99  5.76  2.08 1.72 0.17 0.02
## 14 2003      14      Y 0.01  6.95 23.84  5.21  3.02  6.27 10.52 2.17 0.12 0.16
## 15 2004      17      Y 0.14  5.37 15.85  5.69 14.48  2.23  1.67 1.15 0.23 0.04
## 16 2005      21      Y 6.95  2.65 10.94  8.33  4.44  9.25  3.53 8.29 2.64 0.00
## 17 2006      19      Y 1.47  6.53 25.82  9.80  8.00 14.50 12.14 1.53 0.35 0.01
## 18 2007       8      N 0.51  5.65  8.49  1.44 13.60  1.65  3.09 1.16 0.37 0.50
## 19 2008      17      Y 3.62  1.18  7.18 12.60  6.89  1.58  0.68 1.13 0.02 0.00
## 20 2009      19      Y 3.11  5.50  6.10  3.10 11.90  8.30  1.70 5.50 1.30 0.03
## 21 2010      17      Y 4.70  2.10  6.80 13.60  7.10  6.25  8.14 4.07 0.39 0.08
##     Aug  Sep Total      avg
## 1  0.54 0.53 35.97 2.997500
## 2  0.22 0.05 32.17 2.680833
## 3  0.16 0.08 36.01 3.000833
## 4  0.38 0.01 65.32 5.443333
## 5  0.00 0.62 31.83 2.652500
## 6  0.00 0.00 85.39 7.115833
## 7  0.01 0.98 61.31 5.109167
## 8  0.71 0.92 68.76 5.730000
## 9  0.03 1.30 82.40 6.866667
## 10 0.34 0.08 54.75 4.562500
## 11 0.07 1.32 56.70 4.725000
## 12 0.00 0.80 32.97 2.747500
## 13 0.03 0.10 46.34 3.861667
## 14 1.25 0.25 59.77 4.980833
## 15 0.05 0.39 47.29 3.940833
## 16 0.01 0.48 57.51 4.792500
## 17 0.00 0.00 80.15 6.679167
## 18 0.01 0.74 37.21 3.100833
## 19 0.10 0.01 34.99 2.915833
## 20 0.17 0.14 46.85 3.904167
## 21 0.08 0.28 53.59 4.465833
```

```
#Plot combined data set based on average vs. groundhog prediction.
boxplot(combined_data$avg ~ combined_data$shadow, main = "Rain Data vs. Groundhog Shadow", ylab
= 'Mean Rainfall (in)', xlab = 'Groundhog Prediction (Y = Yes Shadow, N = No Shadow)', col = 'da
rkseagreen')
```




```
### END OF YOUR CODE
```

Excerise 3b.

Describe your findings from your resulting plot in (a) in the context of the question, “whether this information says anything about whether or not we will experience a rainy winter out here in California”.

YOUR EXPLANATION HERE. #There is no significant difference in the average rainfall in a year regardless of whether or not the groundhog sees its shadow as the varition bars (spread) has an overlap

(Question 3 is adapted from Jonathan Taylor’s STATS 191 course.)