Stats 204 | Data Analysis

Jordan Berninger

10/3/2019

Homework 1

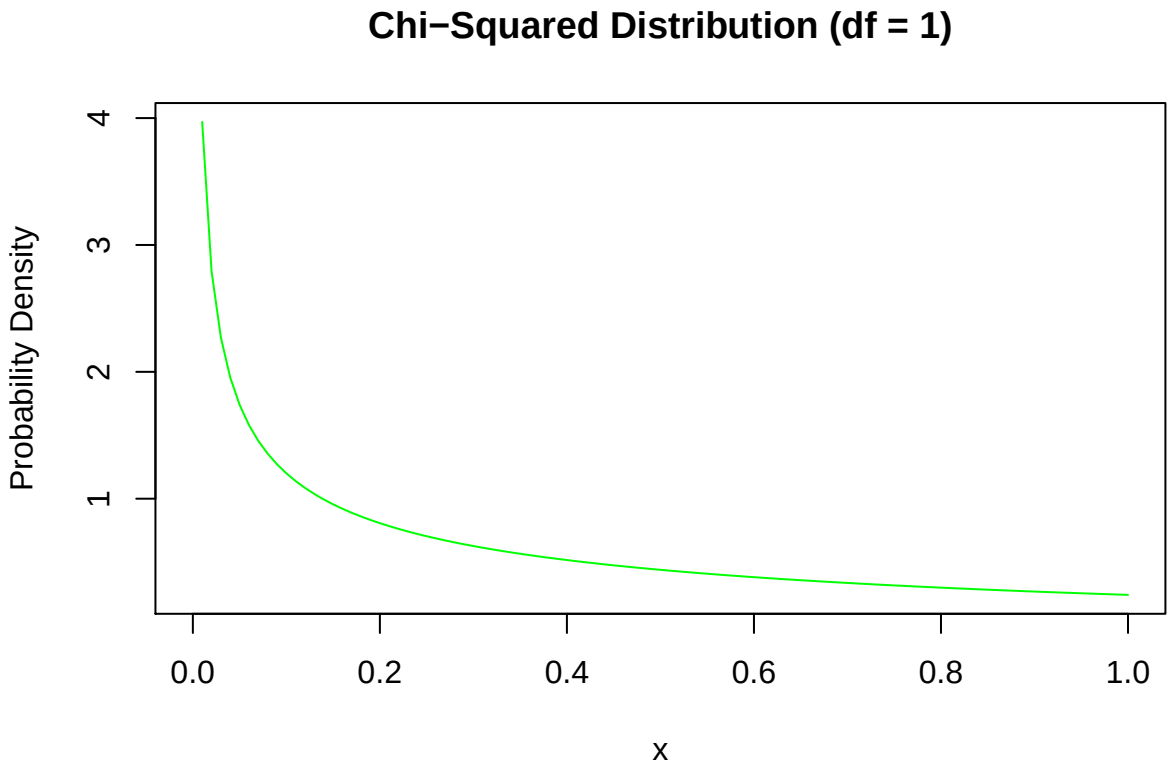
Chapter 1: 2, 3, 6, 7, 8, 10, 11, 12, 14

Chapter 2: 3, 4, 5, 6, 7, 8, 12, 13

Problem 1.2:

Use the *curve* function to display the graph of the $\chi^2(1)$ density. The chi-square density function is *dchisq*.

```
x <- rchisq(100, 1)
curve( dchisq(x, df=1), col='green', ylab = "Probability Density")
title("Chi-Squared Distribution (df = 1)")
```

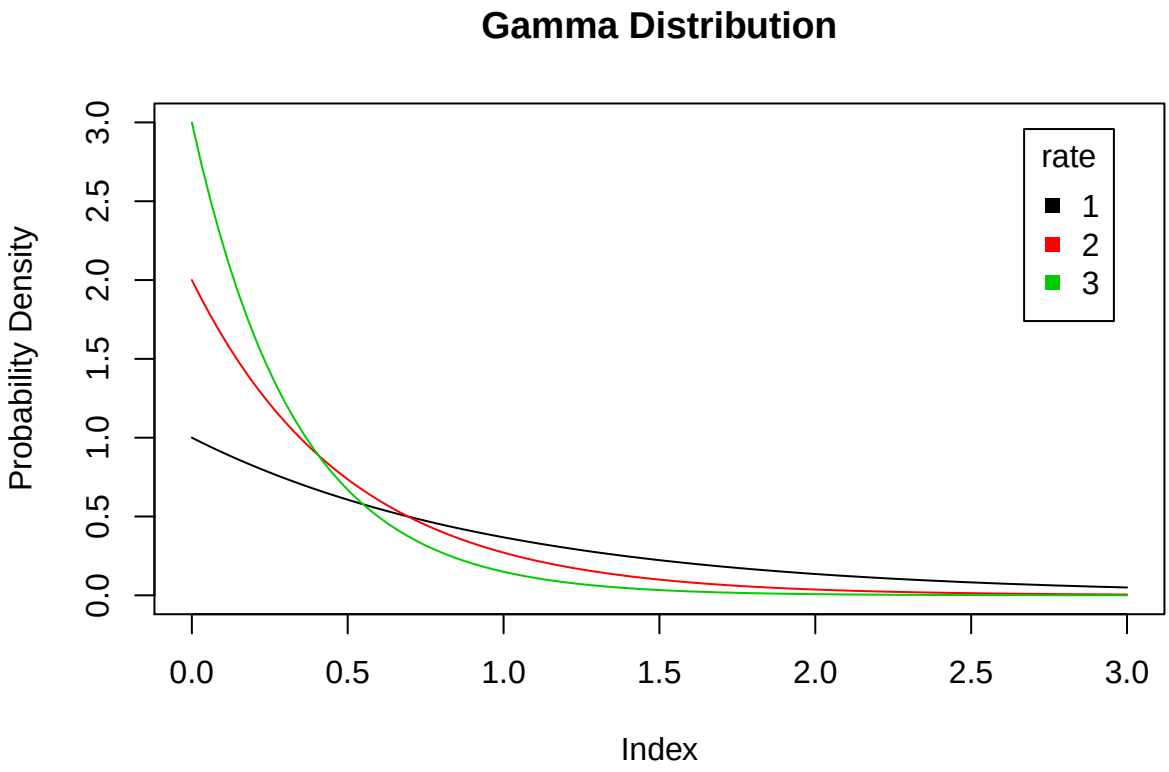


Problem 1.3:

Use the curve function to display the graph of the gamma density with shape parameter 1 and rate parameter 1. Then use the curve function with add=TRUE to display the graphs of the gamma density with shape parameter k and rate 1, 2, 3, all in the same graphics window. The gamma density function is dgamma. Consult the help file ?dgamma to see how to specify the parameters.

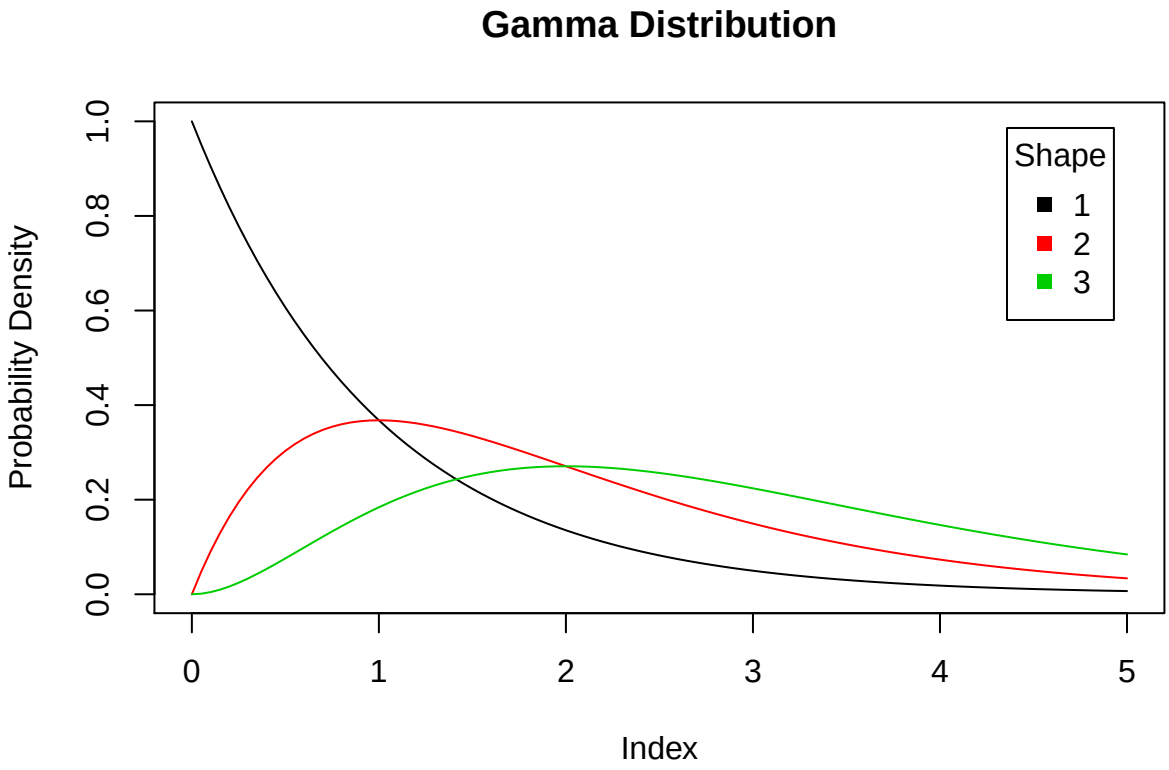
It is unclear from the wording of the problem if you want us to change the shape or the rate parameter, so I will do both independently.

```r
plot(x = c(), y = c(), xlim = c(0,3), ylim = c(0,3), ylab = "Probability Density")
curve( dgamma(x, shape = 1, rate = 1), col=1, add=TRUE)
curve( dgamma(x, shape = 1, rate = 2), col=2, add=TRUE)
curve( dgamma(x, shape = 1, rate = 3), col=3, add=TRUE)
#ylabel("Probability Density", add=TRUE)
title("Gamma Distribution")
legend("topright", c("1", "2", "3"), col = 1:3, pch = 15, inset = 0.05, title = 'rate')
```



Now, for fun, I will play around with the shape parameter. Note that the x and y axes have reset in this second plot.

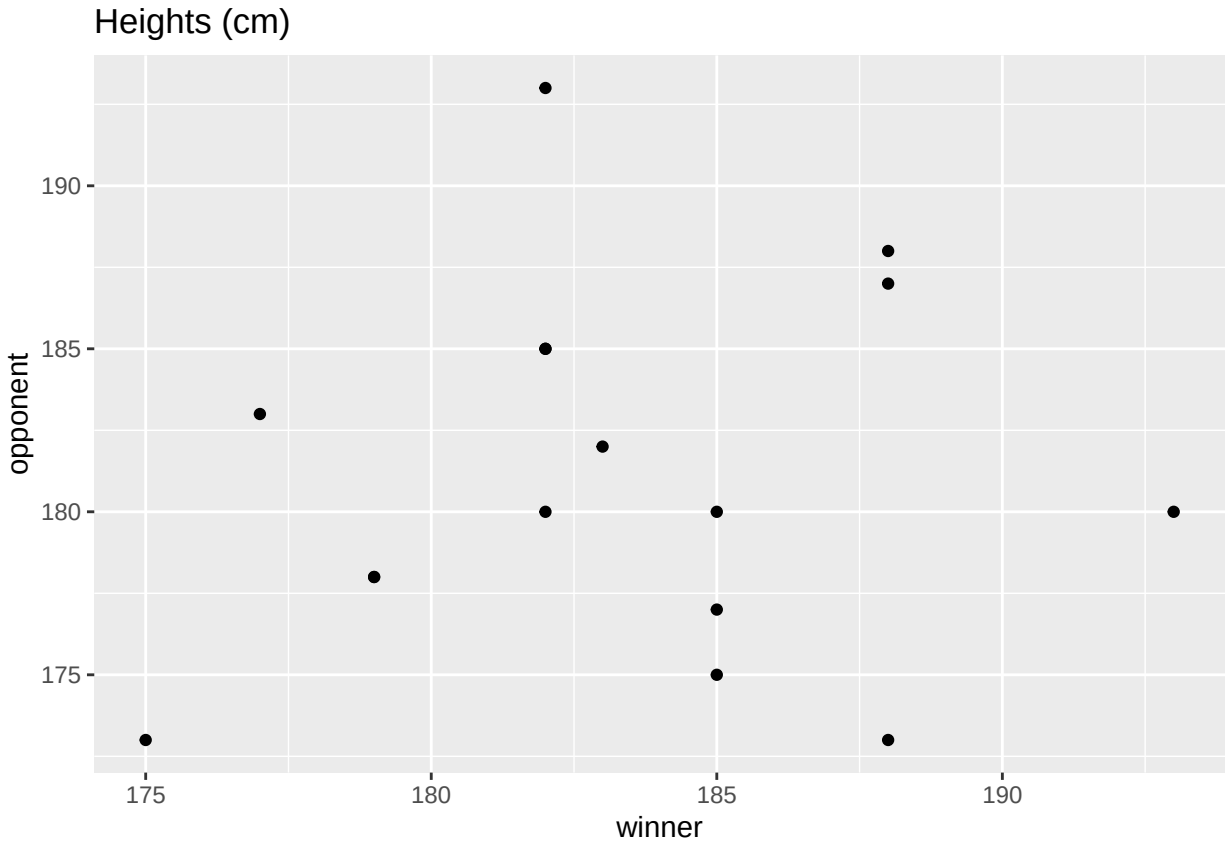
```r
plot(x = c(), y = c(), xlim = c(0,5), ylim = c(0,1), ylab = "Probability Density")
curve( dgamma(x, shape = 1, rate = 1), col=1, add=TRUE)
curve( dgamma(x, shape = 2, rate = 1), col=2, add=TRUE)
curve( dgamma(x, shape = 3, rate = 1), col=3, add=TRUE)
title("Gamma Distribution")
legend("topright", c("1", "2", "3"), col = 1:3, pch = 15, inset = 0.05, title = 'Shape')
```



Problem 1.6:

Refer to Example 1.2 where the heights of the United States Presidents are compared with their main opponent in the presidential election. Create a scatterplot of the loser's height vs the winner'sheight using the *plot* function. Compare the plot to the more detailed plotshown in the Wikipedia article "Heights of Presidents of the United States and presidential candidates" [54]

```
winner = c(185, 182, 182, 188, 188, 188, 185, 185, 177, 182, 182, 193, 183, 179, 179, 175)
opponent = c(175, 193, 185, 187, 188, 173, 180, 177, 183,  185, 180, 180, 182, 178, 178, 173)
df <- data.frame(winner, opponent)
ggplot(data = df, aes(x = winner, y = opponent)) + geom_point() + ggtitle("Heights (cm)")
```



Problem 1.7:

The *rpois* function generates random observations from a Poisson distribution. In Example 1.3, we compared the deaths due to horsekicks to a Poisson distribution with mean$\lambda = 0.61$, and in Example 1.4 we simulated random Poisson($\lambda = 0.61$) data. Use the *rpois* function to simulate very large ($n = 1000, n = 10000$) Poisson($\lambda = 0.61$) random samples. Find the frequency distribution, mean and variance for the sample. Compare the theoretical Poisson density with the sample proportions (see Example 1.4).

We will do this first for $n = 1000$ and later for $n = 10000$. First we return the frequency distribution of the sample:

```
y = rpois(1000, lambda=.61)
kicks = table(y)   #table of sample frequencies
kicks
```

```
## y
##   0   1   2   3   4
## 548 341  93  16   2
```

Now, we get the mean of the sample distribution

```
mean(kicks)
```

```
## [1] 200
```

We should not be surprised this is 200, since we asked for 1000 to be distributed across 0 to 4 years. We compute the variance of the sample distribution next

```r
var(kicks)
```

```
## [1] 56373.5
```

We should not be surprised this variance is large, considering the huge delta between the values of 0 and 4 deaths.

Finally, we compare the Theoretical and Sample distributions.

```r
Theoretical = dpois(1:5, lambda=.61)
Sample = kicks / 1000
cbind(Theoretical, Sample)
```

```
##    Theoretical Sample
## 0 0.3314440301  0.548
## 1 0.1010904292  0.341
## 2 0.0205550539  0.093
## 3 0.0031346457  0.016
## 4 0.0003824268  0.002
```

Problem 1.8:

Refer to Example 1.3. Using the *ppois* function, compute the cumulative distribution function (CDF) for the Poisson distribution with mean$\lambda = 0.61$, for the values 0 to 4. Compare these probabilities with the empirical CDF. The empirical CDF is the cumulative sum ofthe sample proportions p, which is easily computed using the *cumsum* function. Combine the values of 0:4, the CDF, and the empirical CDF in a matrix to display these results in a single table.

```r
Theoretical = ppois(c(0:4), lambda = .61)
Empirical = cumsum(kicks/sum(kicks))
cbind(Theoretical, Empirical)
```

```
##   Theoretical Empirical
## 0   0.5433509     0.548
## 1   0.8747949     0.889
## 2   0.9758853     0.982
## 3   0.9964404     0.998
## 4   0.9995750     1.000
```

Problem 1.10

1.10 (Euclidean norm function). Write a function norm that will compute the Euclidean norm of a numeric vector. The Euclidean norm of a vector.

```r
norm <- function(x){
  sqrt(apply(x^2, 1, sum))
}
norm(matrix(c(0,0,0,1), nrow = 1))
```

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

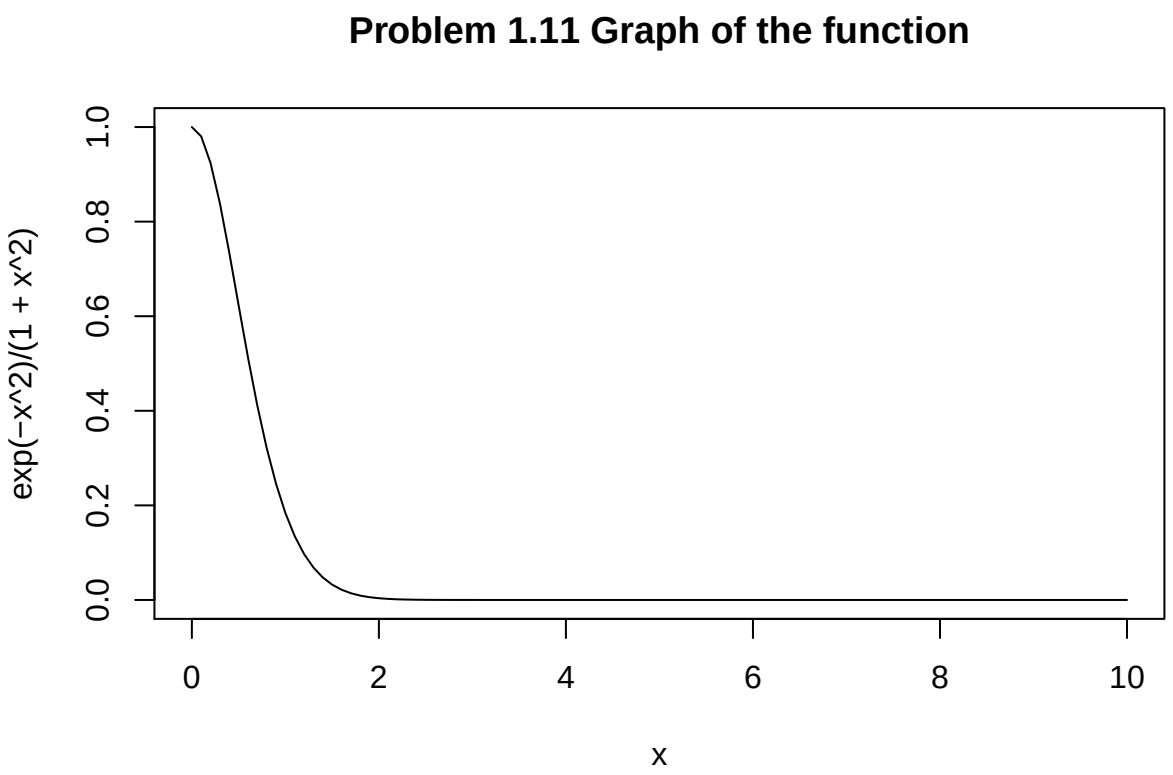
```
norm(matrix(c(2,5,2,4), nrow = 1))
```

```
## [1] 7
```

Problem 1.11

Use the *curve* function to display the graph of the function $f(x) = e - x^2/(1+x^2)$ on the interval $0 \leq x \leq 10$. Then use the *integrate* function to compute the value of the integral

```
curve(expr = exp(-x^2)/(1+x^2), from = 0, to = 10)
title("Problem 1.11 Graph of the function")
```




```
integrate(f = function(x){exp(-x^2)/(1+x^2)}, lower = 0, upper = Inf)
```

```
## 0.6716467 with absolute error < 8.3e-05
```

Problem 1.12

Construct a matrix with 10 rows and 2 columns, containing random standard normal data. This is a random sample of 10 observations from a standard bivariate normal distribution. Use the apply function and your norm function from Exercise 1.10 to compute the Euclidean norms for each of these 10 observations.

I built the the ability to iterate on rows into my *norm* function so I don't have to use *apply* here.

```
x = matrix(rnorm(20), 10, 2)
norm <- function(x){
  sqrt(apply(x^2, 1, sum))
}
norm(x)
```

```
##  [1] 0.9424404 2.9233325 1.0201432 0.6444420 1.0531181 0.7071925 1.3431601
##  [8] 1.7845766 0.7311829 0.6059533
```

Problem 1.14

The following data describe the tearing factor of paper manufactured under different pressures during pressing. The data is given in Hand et al. [21, Page 4]. Four sheets of paper were selected and tested from each of the five batches manufactured

```
pressure <- c(rep(35.0, 4), rep(49.5, 4), rep(70.0, 4), rep(99.0, 4), rep(140.0, 4))
tear <- c(112,119,17,113, 108,99,112,118, 120,106,102,109, 110,101,99,104, 100,102,96,101)
data.frame(pressure, tear)
```

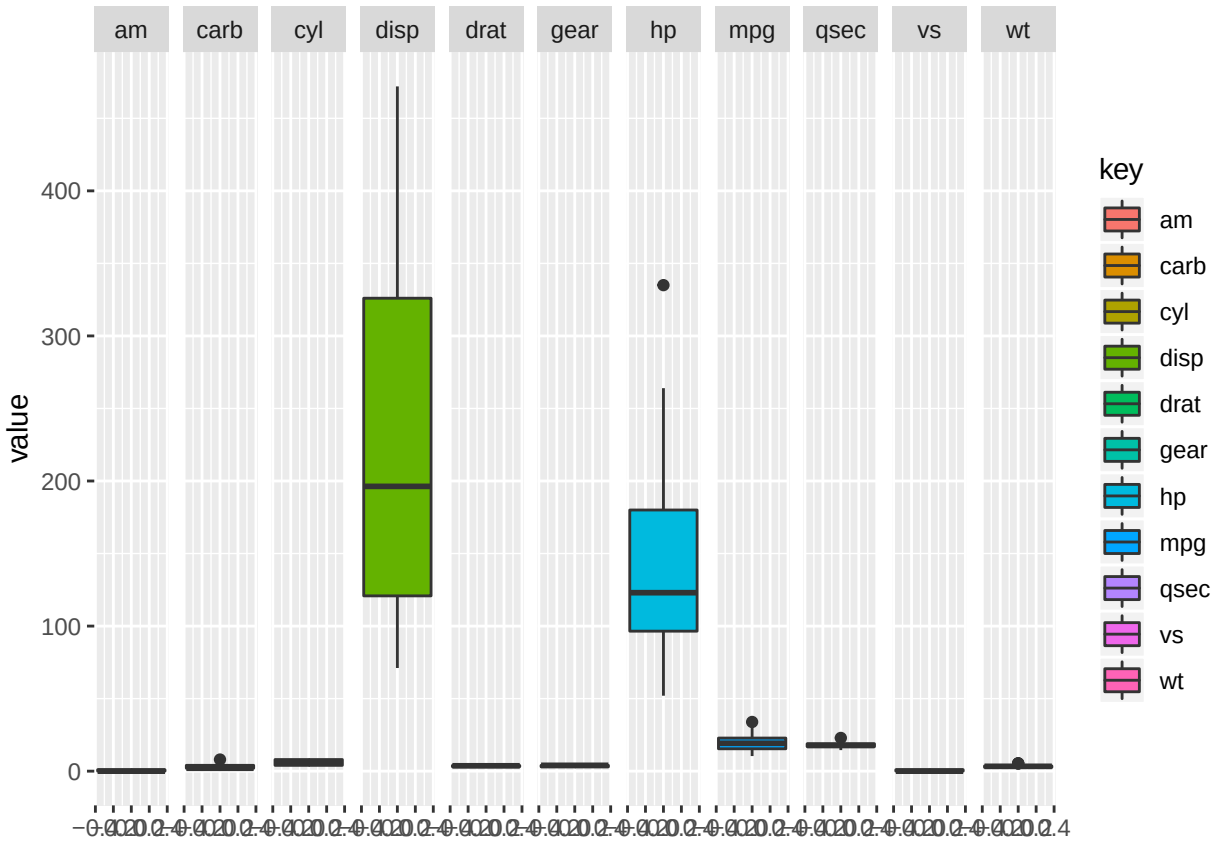
```
##    pressure tear
## 1      35.0  112
## 2      35.0  119
## 3      35.0   17
## 4      35.0  113
## 5      49.5  108
## 6      49.5   99
## 7      49.5  112
## 8      49.5  118
## 9      70.0  120
## 10     70.0  106
## 11     70.0  102
## 12     70.0  109
## 13     99.0  110
## 14     99.0  101
## 15     99.0   99
## 16     99.0  104
## 17    140.0  100
## 18    140.0  102
## 19    140.0   96
## 20    140.0  101
```

Problem 2.3 (pdf p 75)

Display the mtcars data included with R and read the documentation using ?mtcars. Display parallel boxplots of the quantitative variables. Display a pairs plot of the quantitative variables. Does the pairs plot reveal any possible relations between the variables?

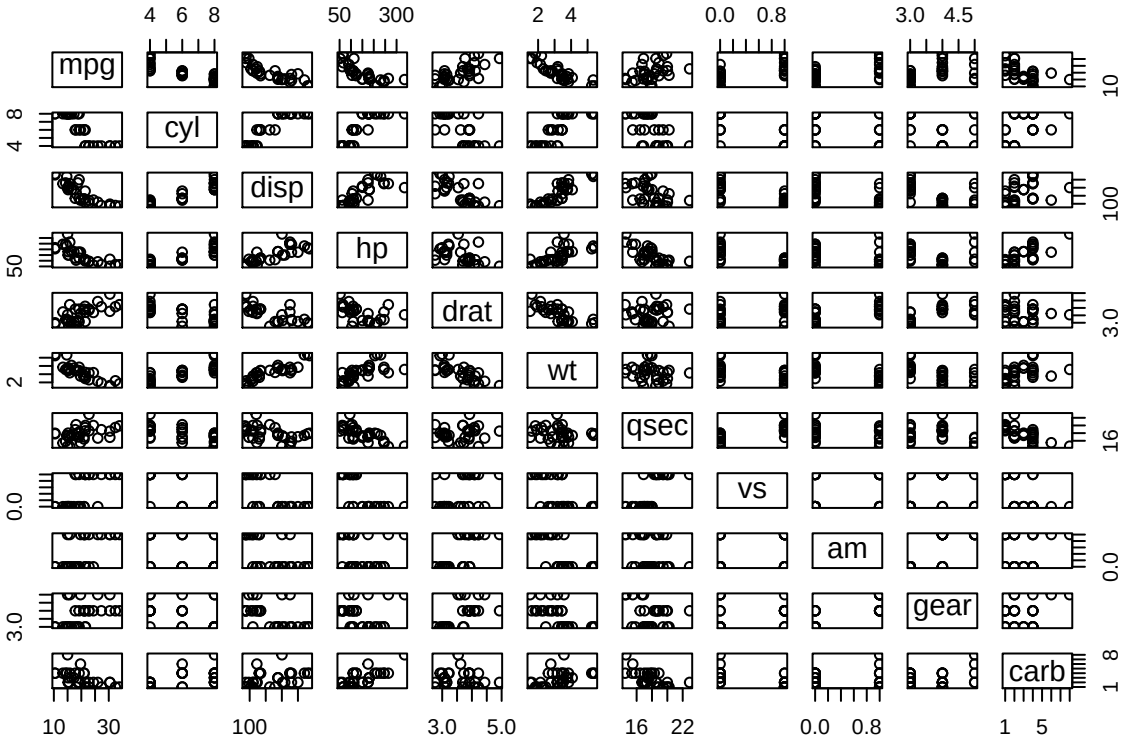
The following boxplot shows that the quantitative variables are using different scales, which it to be expected.

```
data(mtcars)
ggplot(data = gather(mtcars), aes(y = value, fill = key)) + geom_boxplot() + facet_grid(~key)
```



The pairwise scatterplots indicates that several variables are correlated with *mpg*, specifically, *disp*, *hp*, and *wt* seems to have negative correlations with *mpg*, while *drat* and *qsec* appear to have positive correlations with *mpg*. Furthermore, the scatterplots show that *vs* and *am* only have 2 values, while *gear* has 3 values.

```
pairs(mtcars)
```



Problem 2.4

Refer to Example 2.7. Create a new variable r equal to the ratio of brain size over body size. Using the full mammals data set, order the mammals data by the ratio r. Which mammals have the largest ratios of brain size to body size? Which mammals have the smallest ratios? (Hint: use head and tail on the ordered data.)

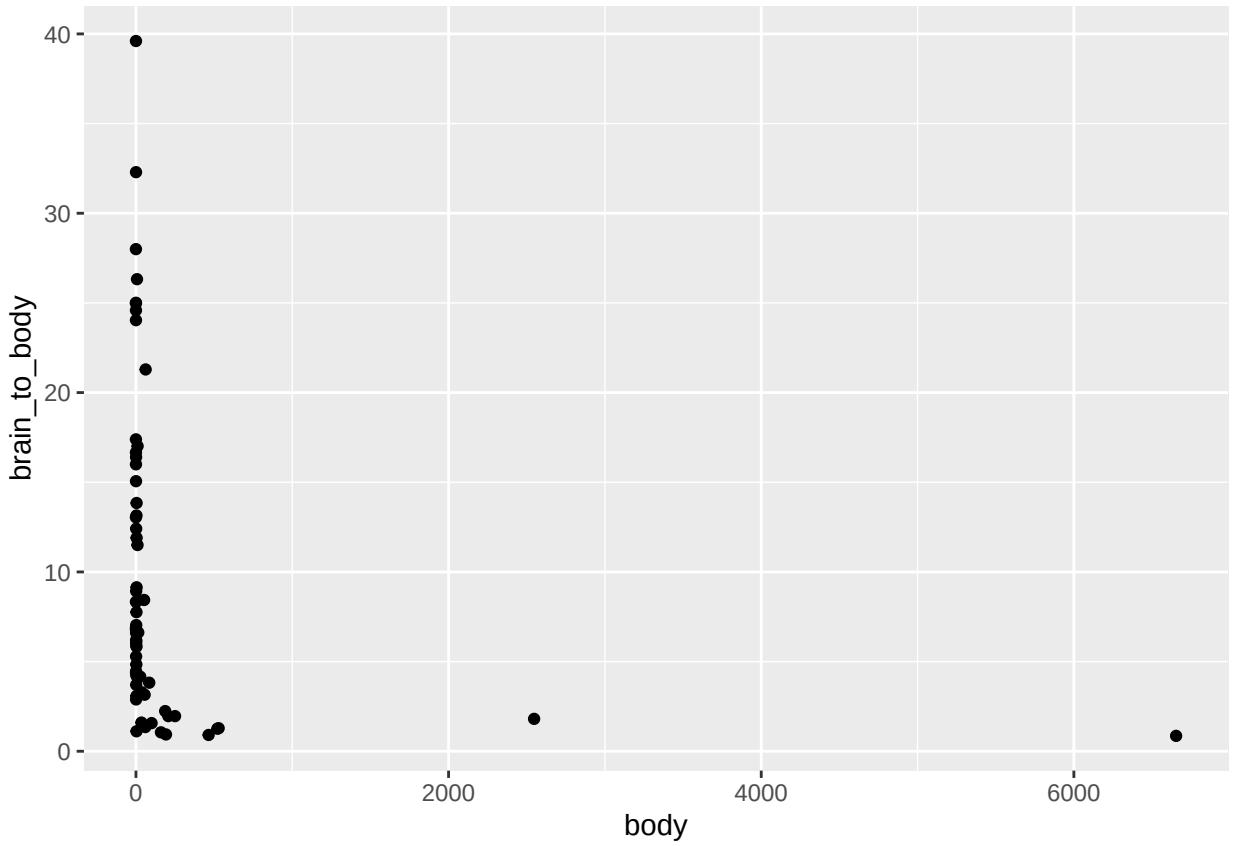
```
data(mammals)
mammals %>% mutate(brain_to_body = brain/body,
                   names = row.names(mammals)) %>%
  arrange(desc(brain_to_body)) %>%
  filter(brain_to_body == max(brain_to_body) | brain_to_body == min(brain_to_body))
```

```
##       body brain brain_to_body            names
## 1    0.101     4     39.603960  Ground squirrel
## 2 6654.000  5712      0.858431 African elephant
```

Problem 2.5

Refer to Exercise 2.5. Construct a scatterplot of the ratio $r = brain/body$ vs body size for the full mammals data set.

```
m1 <- mammals %>% mutate(brain_to_body = brain/body,
                         names = row.names(mammals))
ggplot(data = m1, aes(x = body, y = brain_to_body)) + geom_point()
```

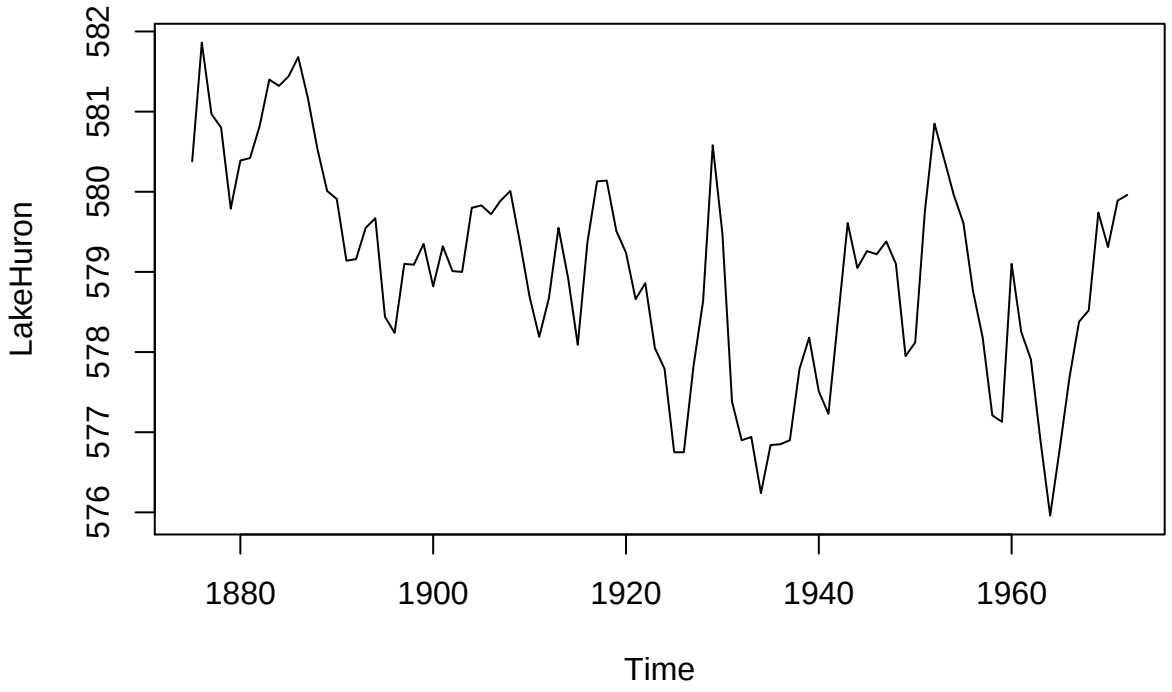


Problem 2.6

The LakeHuron data contains annual measurements of the level, in feet, of Lake Huron from 1875 through 1972. Display a time plot of the data. Does the average level of the lake appear to be stable or changing with respect to time? Refer to Example 2.4 for a possible method of transforming this series so that the mean is stable, and plot the resulting series. Does the transformation help to stabilize the mean?

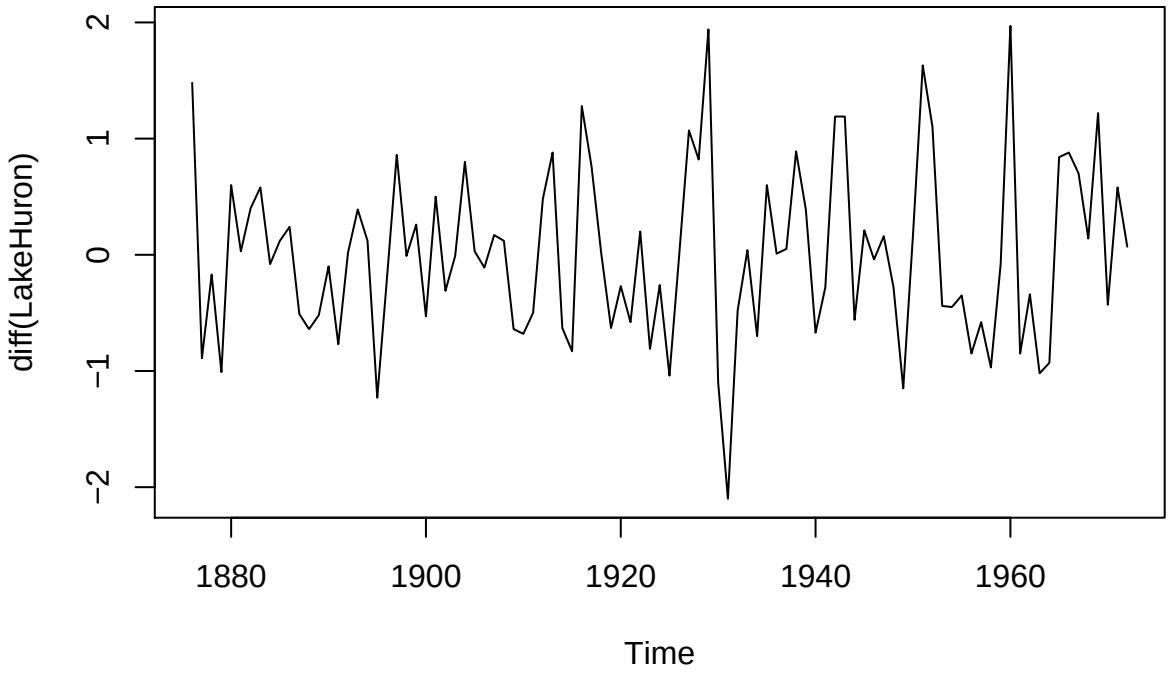
First we show the original data, which appears to have a mean and variance which is not constant with respect to time.

```
data(LakeHuron)
plot.ts(LakeHuron)
```



Now we show the differenced series, we appears to be more 'stable' over time, meaning its mean and variance appear to be consistent across different time windows:

```
plot.ts(diff(LakeHuron))
```



Problem 2.7

Refer to Example 2.6, where we computed sample means for each row of the *randu* data frame. Repeat the analysis, but instead of *randu*, create a matrix of random numbers using $runif$.

```r
set.seed(4)
mat <- matrix(runif(20), ncol = 5)
apply(mat, 1, mean)
```

```
## [1] 0.6839048 0.3761150 0.6301265 0.5372545
```

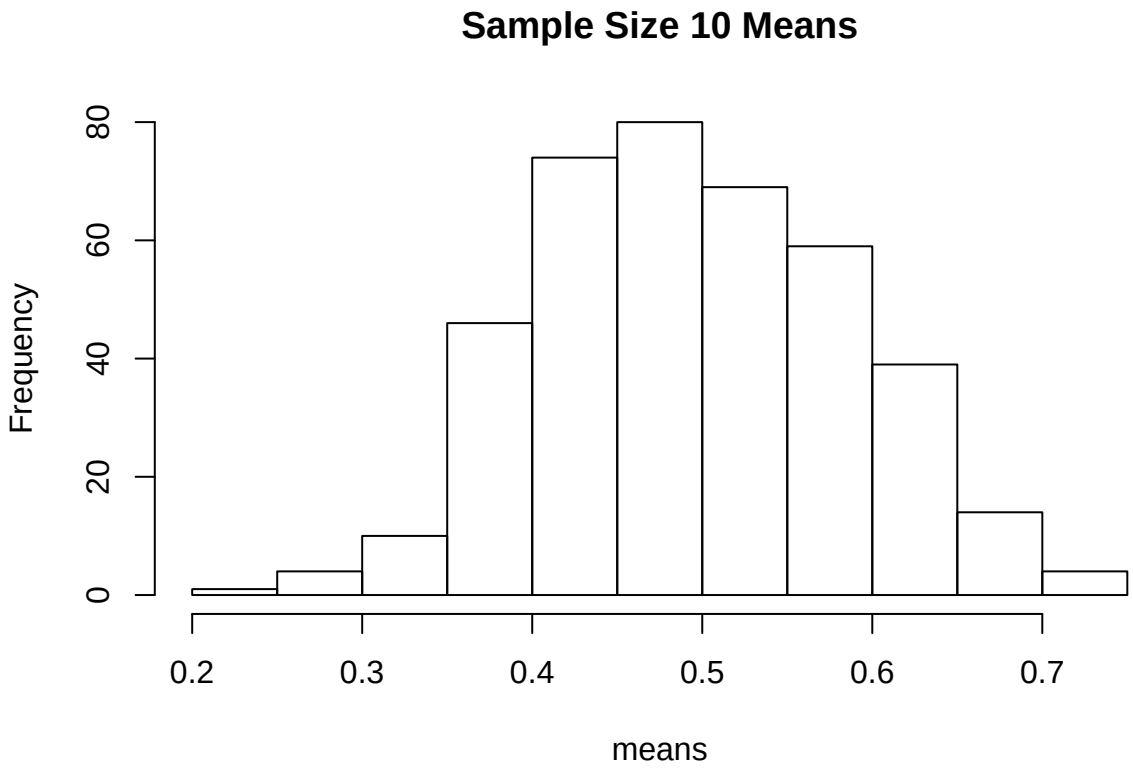
```r
var(mat)
```

```
##             [,1]        [,2]        [,3]        [,4]        [,5]
## [1,]  0.05555136  0.05158939  0.08545260 -0.08185362  0.03742794
## [2,]  0.05158939  0.08230414  0.06741533 -0.08604233  0.03744375
## [3,]  0.08545260  0.06741533  0.16450334 -0.12803908  0.07242707
## [4,] -0.08185362 -0.08604233 -0.12803908  0.12462034 -0.05899441
## [5,]  0.03742794  0.03744375  0.07242707 -0.05899441  0.03404666
```

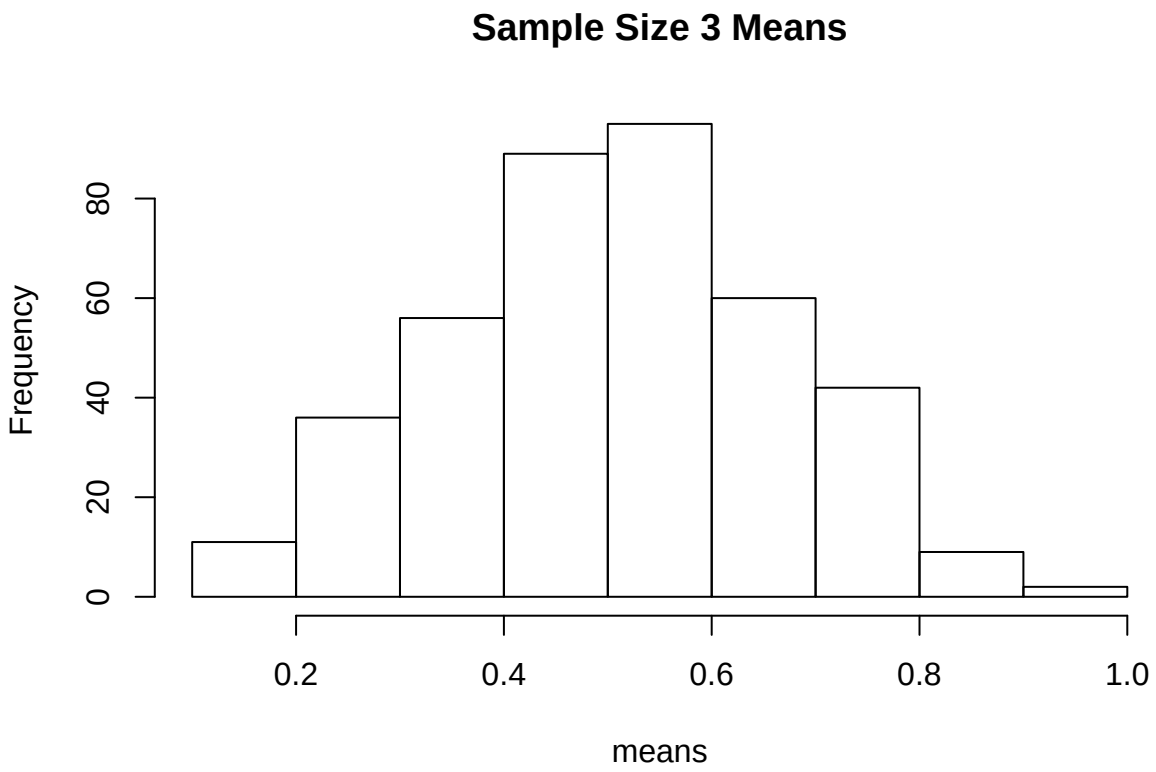
Problem 2.8

Refer to Example 2.6 and Exercise 2.7, where we computed sample means for each row of the data frame. Repeat the analysis in Exercise 2.7, but instead of sample size 3 generate a matrix that is 400 by 10 (sample size 10). Compare the histogram for sample size 3 and sample size 10. What does the Central Limit Theorem tell us about the distribution of the mean as sample size increases?

```r
mat2 <- matrix(runif(4000), ncol = 400)
means = apply(mat2, MARGIN=2, FUN=mean)
hist(means, main = "Sample Size 10 Means")
```



```
mat2 <- matrix(runif(1200), ncol = 400)
means = apply(mat2, MARGIN=2, FUN=mean)
hist(means, main = "Sample Size 3 Means")
```



The Central Limit Theorem tells us that as we take an increasing number of samples from a population, the sample means will tend towards the normal distribution centered on the true population mean. The histograms from the plot with sample size 10 shows a more even bell curve than the sample size 3 counterpart. We can see in this small example that the more sampling we take, the more normal / Gaussian the sample means appear to be.

Problem 2.12

Refer to Example 2.1. Using the full mammals data set, order the data by brain size. Which mammals have the largest brain sizes? Which mammals have the smallest brain sizes?

This shows the largest and smallest brains in the dataset.

```
mammals %>% mutate(names = row.names(mammals)) %>%
  arrange(desc(brain)) %>%
  filter(brain == max(brain) | brain == min(brain))
```

```
##       body   brain                     names
## 1 6654.000 5712.00          African elephant
## 2    0.005    0.14 Lesser short-tailed shrew
```

Problem 2.13

Refer to the mammals data in Example 2.7. Construct a scatterplot like Figure 2.19 on the original scale (Figure 2.19 is on the log-log scale.) Label the points and distances for cat, cow, and human. In this example, which plot is easier to interpret?

I believe the plot with log transformations on both axes is superior and more informative since we have less congenstion in the bottom left of the plot. In the plot with log transformations on both axes, there is greater separation between the points and we can get more insight on how the different species cluster and the relationship between the 2 variables.

```
plot(mammals$body, mammals$brain, xlab="body", ylab="brain")
y = mammals[c("Cat", "Cow", "Human"), ]
polygon(y)
text(y, rownames(y), adj=c(1, .5))
```