Stats 204 Homework 3

Jordan Berninger

10/27/2019

Chapter 4: 2, 6, 8

Chapter 6: 2, 3 (a, b and d), 5, 7

Chapter 7: 7, 8

Chapter 8: 2, 3, 4, 5

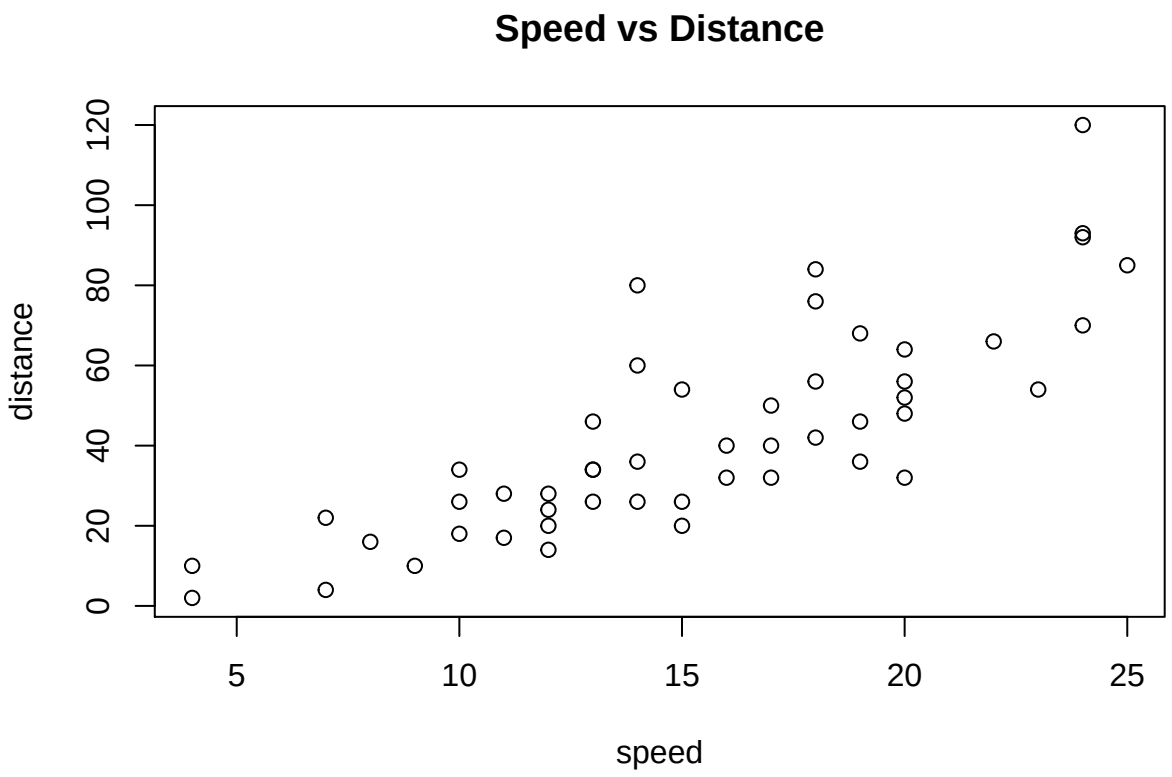
Problem 4.2

(Speed and stopping distance (continued)). Suppose that one wishes to compare linear and quadratic fits to the (speed,dist) observations. One can construct these two fits in R using the code

```
data(cars)
fit.linear = lm(dist ~ speed, data=cars)
fit.quadratic = lm(dist ~ speed + I(speed^2), data=cars)
```

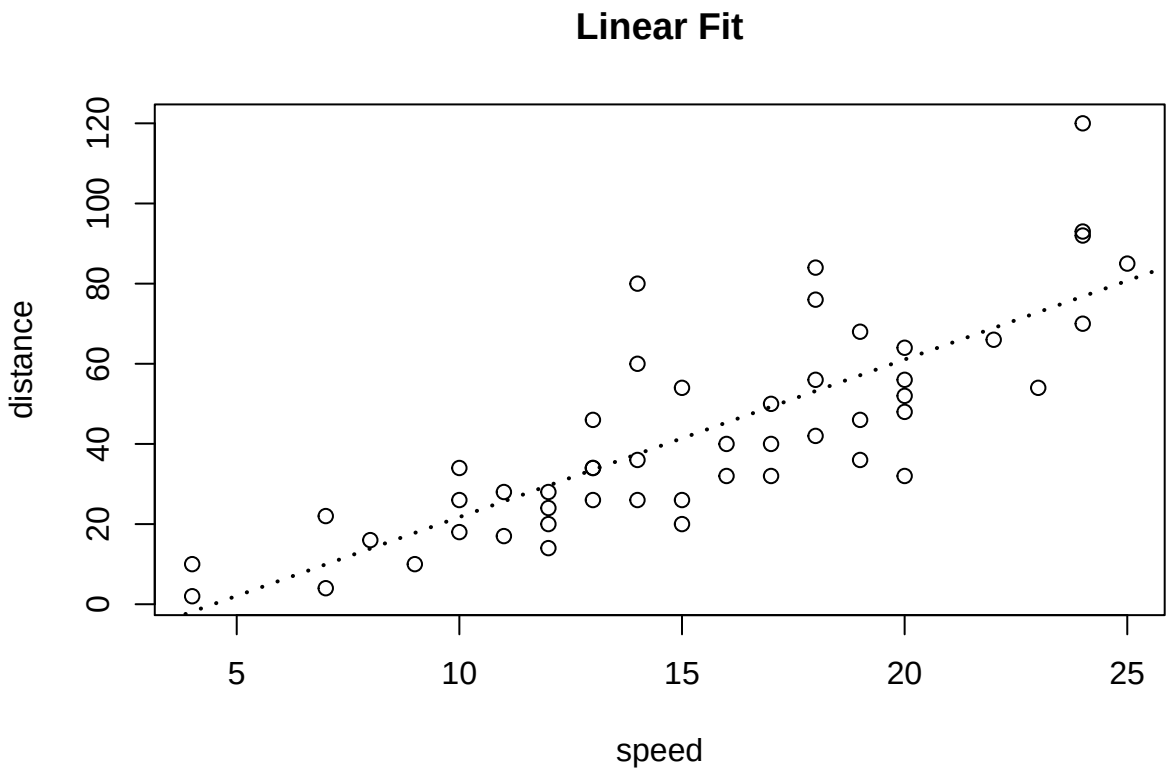
a. Construct a scatterplot of speed and stopping distance.

```
plot(cars$speed, cars$dist, main = "Speed vs Distance", xlab = "speed", ylab = "distance")
```



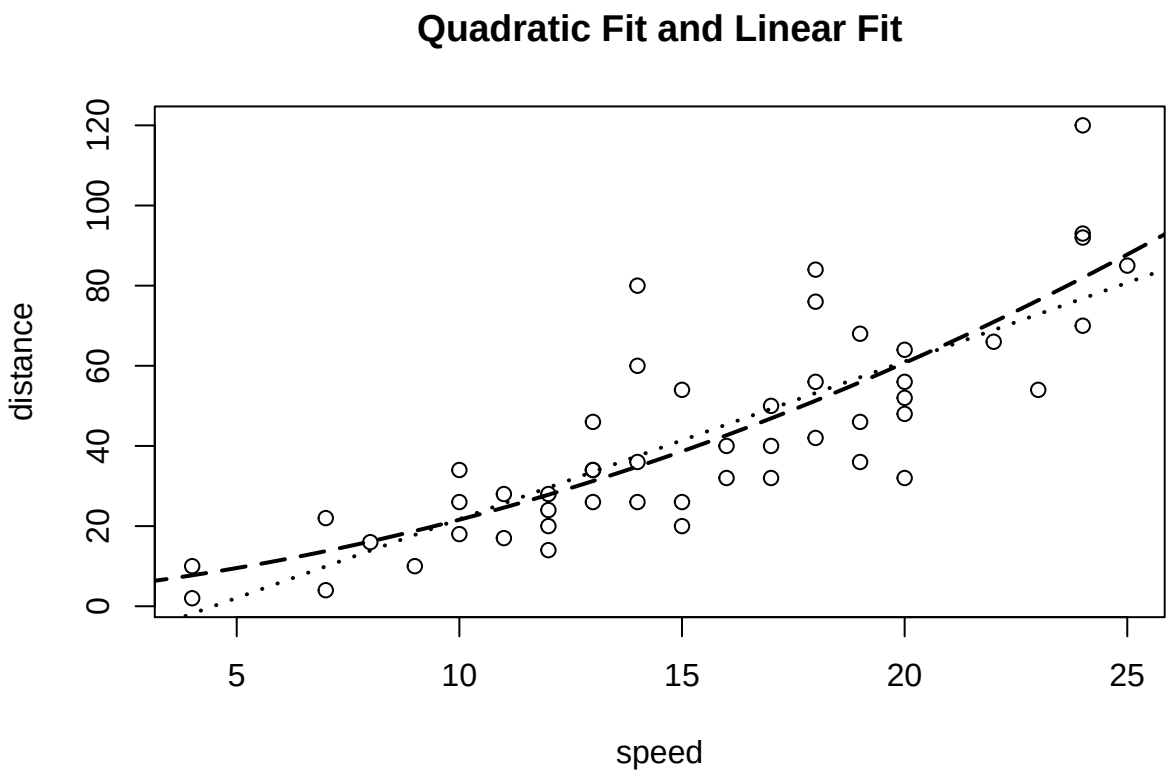
b. Using the abline function with argument fit.linear, overlay the best line fit using line type "dotted" and using a line width of 2.

```
plot(cars$speed, cars$dist, main = "Linear Fit", xlab = "speed", ylab = "distance")
abline(reg = fit.linear, lty = "dotted", lwd = 2)
```



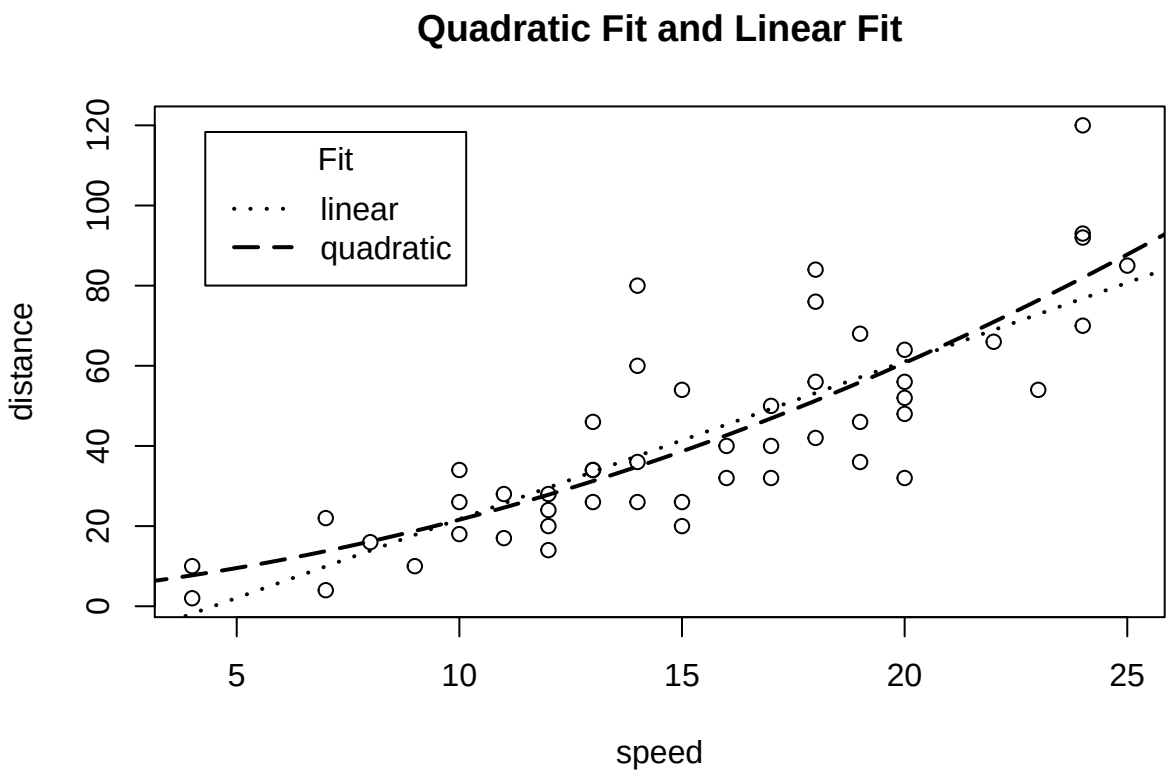
c. Using the lines function, overlay the quadratic fit using line type "long-dash" and a line width of 2.

```
newdat <- seq(from = min(cars$speed) - 2, to = max(cars$speed) + 2, length.out = 100)
quadratic.fitted <- predict(fit.quadratic, list(speed = newdat))
plot(cars$speed, cars$dist, main = "Quadratic Fit and Linear Fit", xlab = "speed", ylab = "distance")
abline(reg = fit.linear, lty = "dotted", lwd = 2)
lines(x = newdat, y = quadratic.fitted, lty = "longdash", lwd = 2)
```



d. Use a legend to show the line types of the linear and quadratic fits.

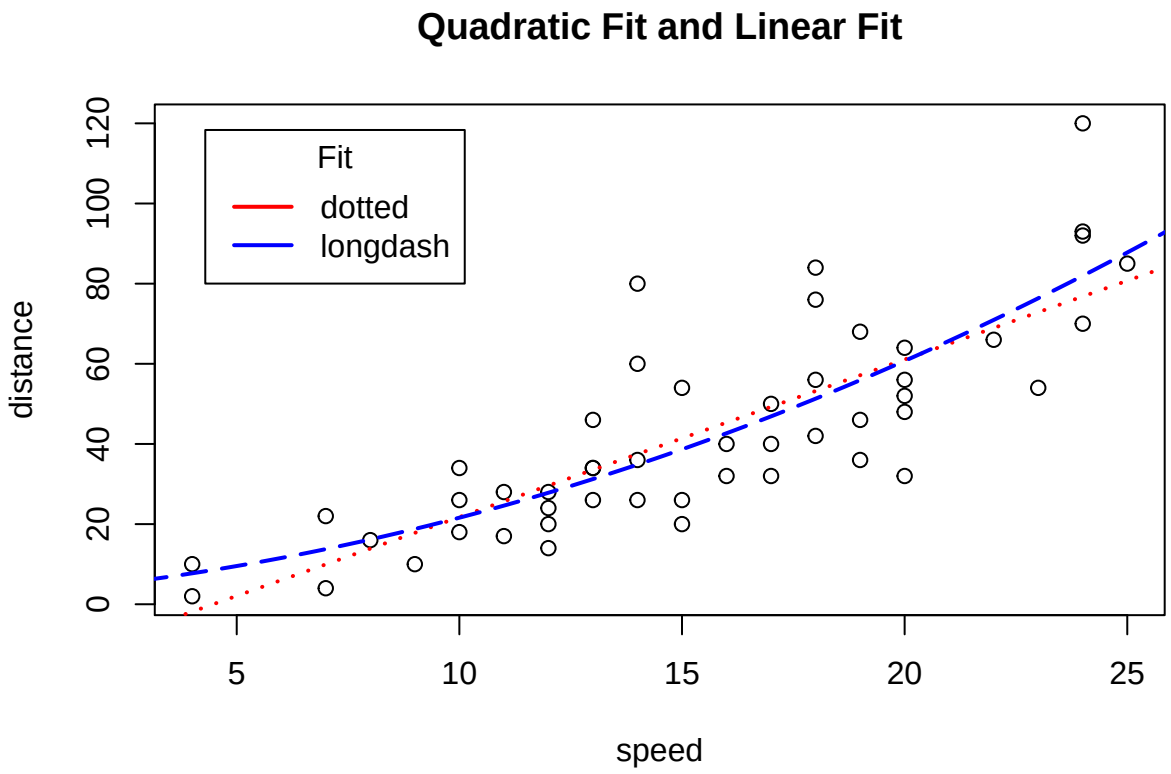
```
newdat <- seq(from = min(cars$speed) - 2, to = max(cars$speed) + 2, length.out = 100)
quadratic.fitted <- predict(fit.quadratic, list(speed = newdat))
plot(cars$speed, cars$dist, main = "Quadratic Fit and Linear Fit", xlab = "speed", ylab = "distance")
abline(reg = fit.linear, lty = 3, lwd = 2)
lines(x = newdat, y = quadratic.fitted, lty = 5, lwd = 2)


legend("topleft", c("linear", "quadratic"), lty = c("dotted", "longdash"),
       lwd = 2, inset = 0.05, title = 'Fit')
```



e. Redo parts (a) - (d) using two contrasting colors (say red and blue) for the two different fits.

Since I was just adding layers, I will only produce one additional plot.

```
newdat <- seq(from = min(cars$speed) - 2, to = max(cars$speed) + 2, length.out = 100)
quadratic.fitted <- predict(fit.quadratic, list(speed = newdat))
plot(cars$speed, cars$dist, main = "Quadratic Fit and Linear Fit", xlab = "speed", ylab = "distance")
abline(reg = fit.linear, lty = 3, lwd = 2, col = "red")
lines(x = newdat, y = quadratic.fitted, lty = 5, lwd = 2, col = "blue")


legend("topleft", c("linear", "quadratic"), c("dotted", "longdash"),
       lwd = 2, col = c("red", "blue"),
       inset = 0.05, title = 'Fit')
```

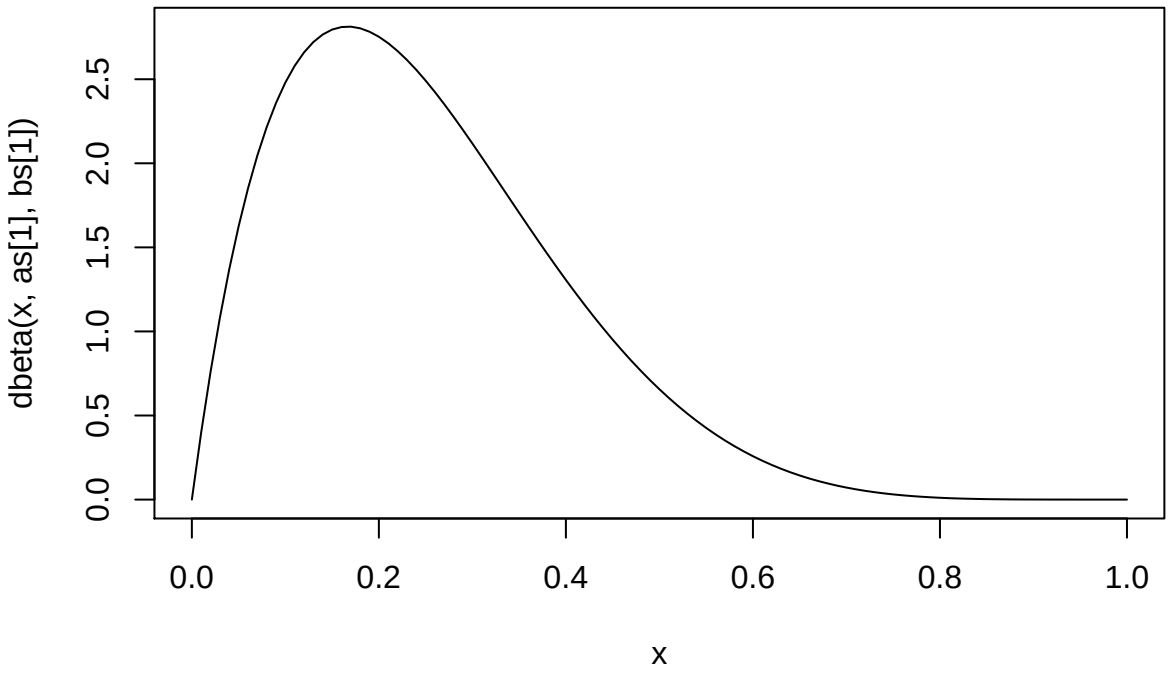


Problem 4.6

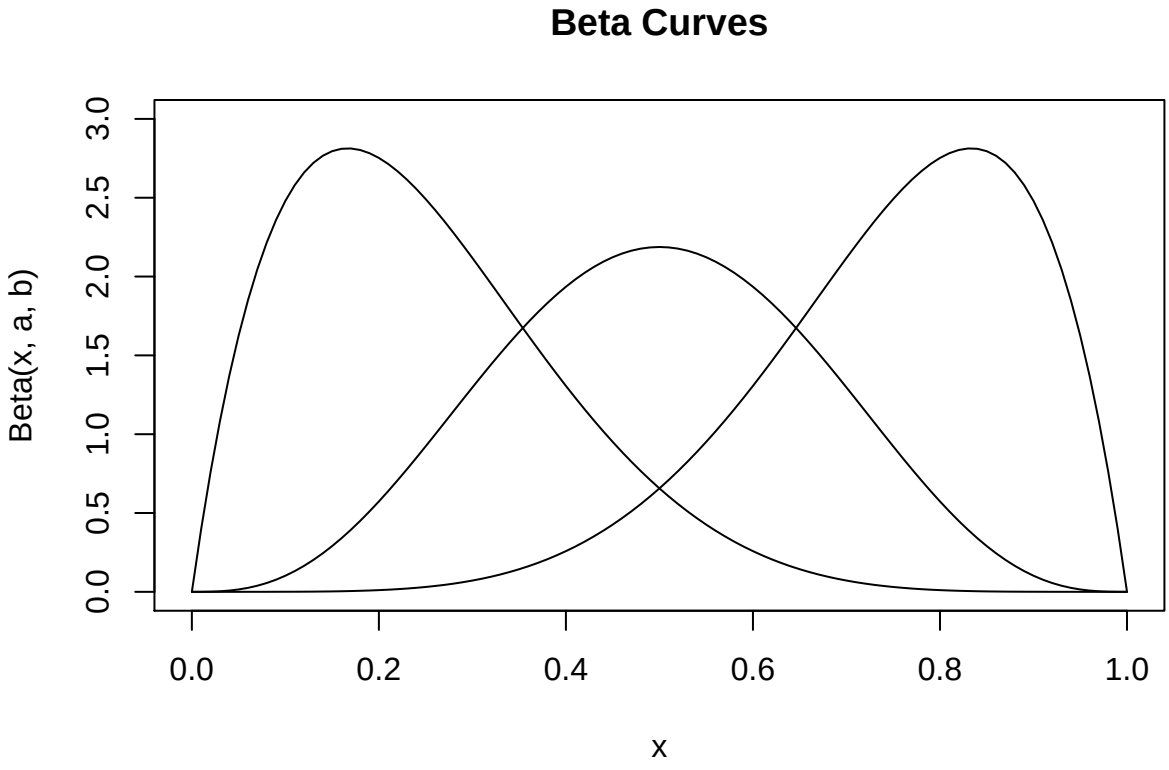
Suppose one is interesting in displaying three members of the beta family of curves, where the beta density with shape parameters a and b (denoted by Beta(a,b)) is given by $f(y) = \frac{1}{B(a,b)} y^{a-1}(1-y)^{b-1}, 0 < y < 1$ One can draw a single beta density, say with shape parameters a = 5 and b = 2, using the curve function: $curve(dbeta(x, 5, 2), from = 0, to = 1))$

a. Use three applications of the curve function to display the Beta(2, 6), Beta(4, 4), and Beta(6, 2) densities on the same plot. (The curve function with the add=TRUE argument will add the curve to the current plot.)

```
as <- c(2,4,6)
bs <- c(6,4,2)
plot(curve(dbeta(x, as[1], bs[1]), from = 0, to = 1), type = "l",
     xlab = "x", ylab = "Beta(x, a, b)",
     main = "Beta Curves",
     ylim = c(0, 3))
```

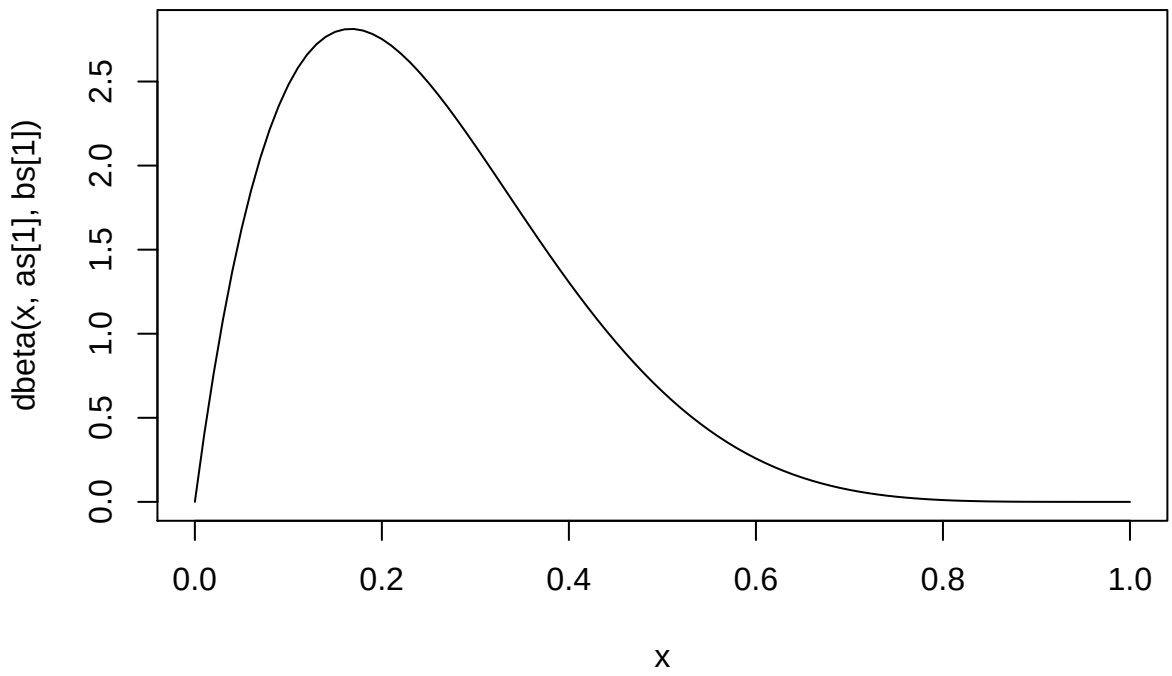



```
curve(dbeta(x, as[2], bs[2]), from = 0, to = 1, type = "l", add = TRUE)
curve(dbeta(x, as[3], bs[3]), from = 0, to = 1, type = "l", add = TRUE)
```



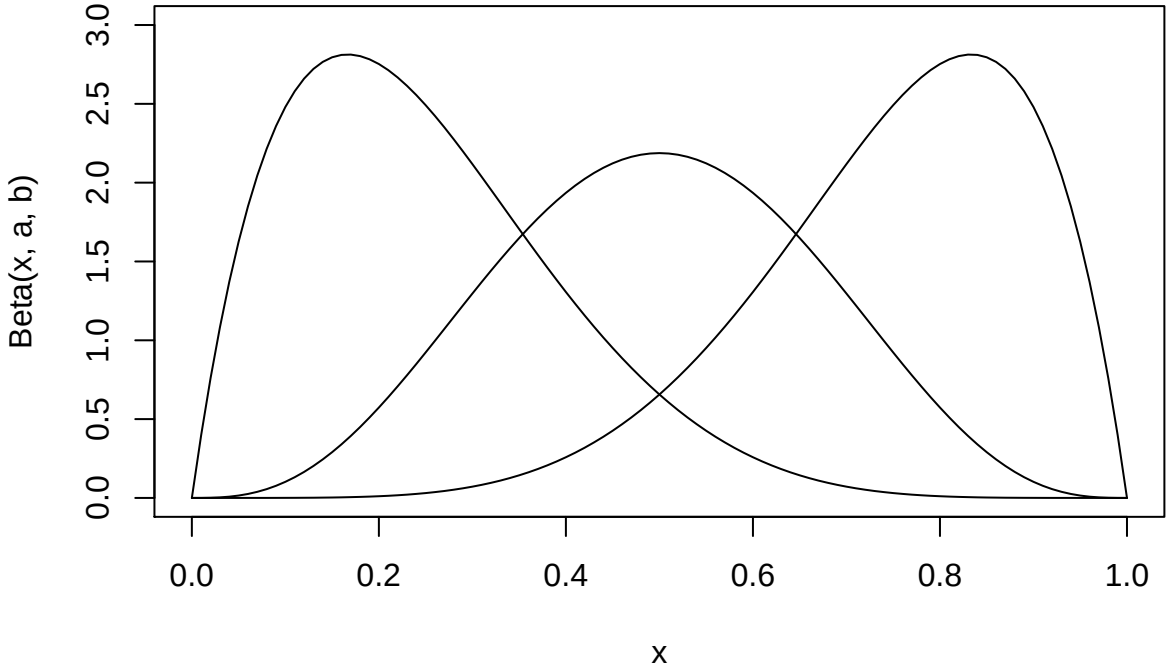
b. Use the following R command to title the plot with the equation of the beta density.

```r
as <- c(2,4,6)
bs <- c(6,4,2)
plot(curve(dbeta(x, as[1], bs[1]), from = 0, to = 1), type = "l",
     xlab = "x", ylab = "Beta(x, a, b)",
     ylim = c(0, 3))
```




```r
curve(dbeta(x, as[2], bs[2]), from = 0, to = 1, type = "l", add = TRUE)
curve(dbeta(x, as[3], bs[3]), from = 0, to = 1, type = "l", add = TRUE)
title(expression(f(y)==frac(1,B(a,b))*y^{a-1}*(1-y)^{b-1}))
```

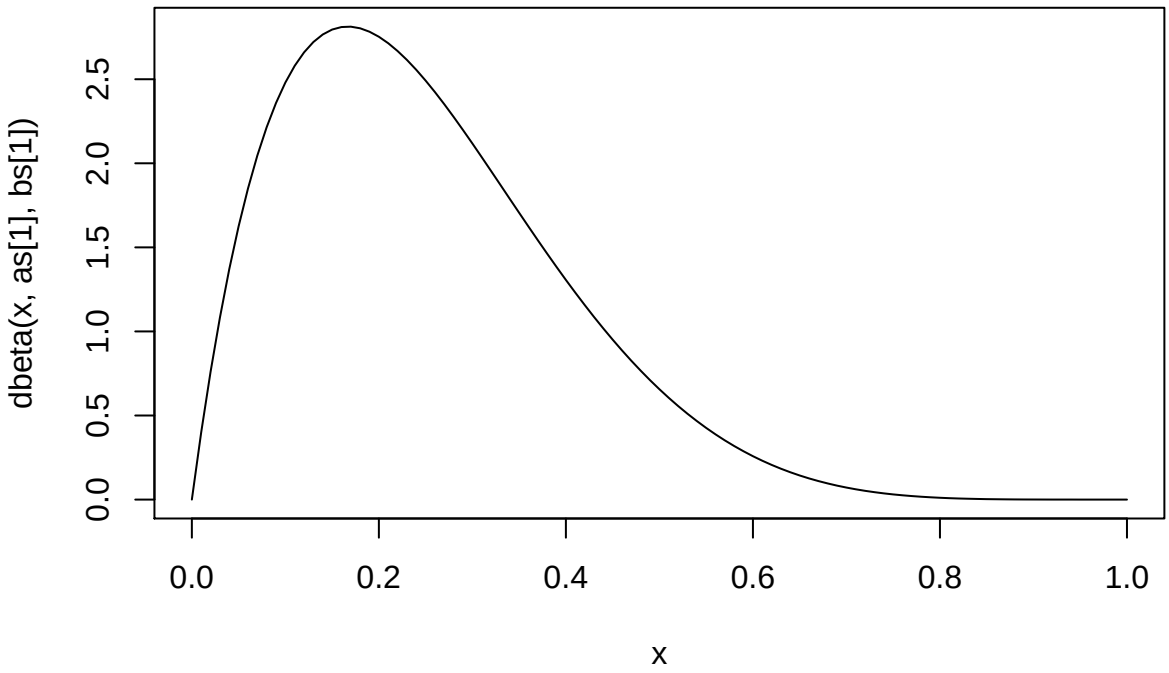
$$f(y) = \frac{1}{B(a, b)} y^{a-1}(1-y)^{b-1}$$



c. Using the text function, label each of the beta curves with the corresponding values of the shape parameters a and b.

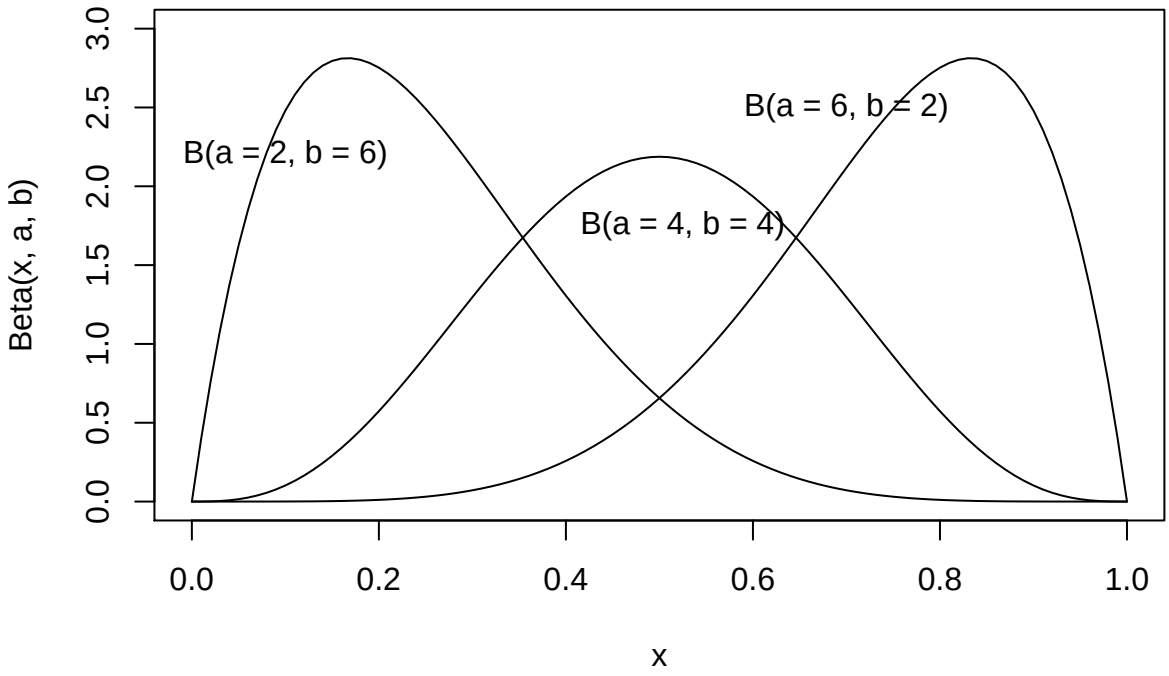
I used the following code to add the labels. RMarkdown does not allow me to use the locator

```
as <- c(2,4,6)
bs <- c(6,4,2)
plot(curve(dbeta(x, as[1], bs[1]), from = 0, to = 1), type = "l",
     xlab = "x", ylab = "Beta(x, a, b)",
     ylim = c(0, 3))
```



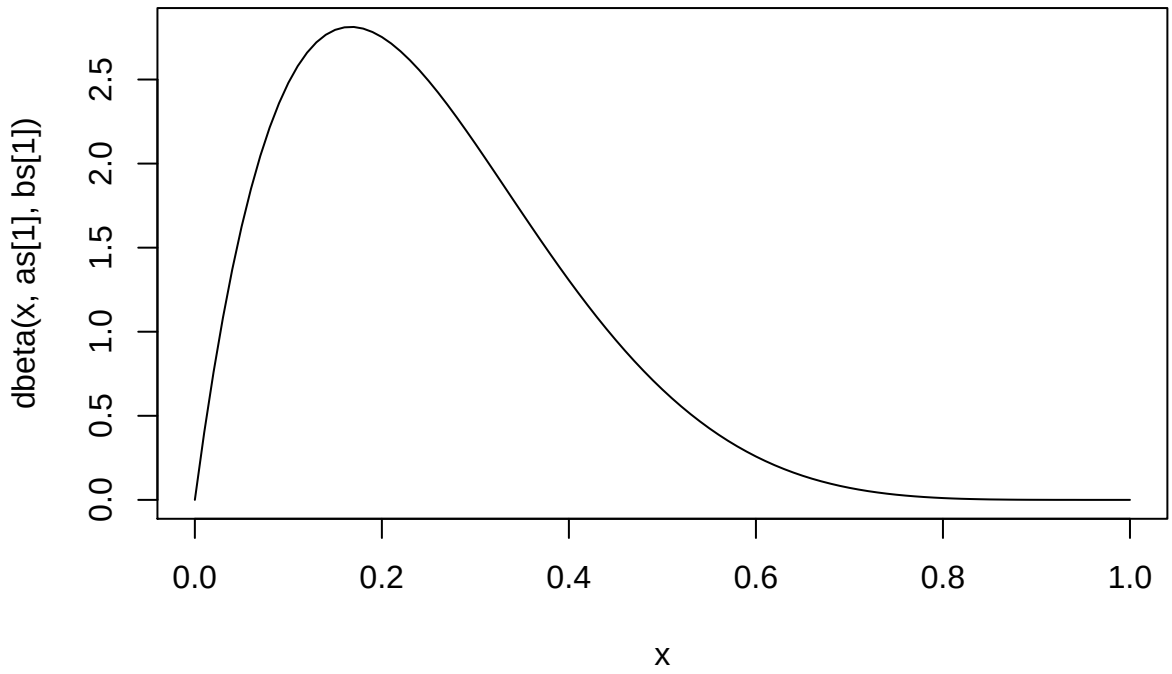

```
curve(dbeta(x, as[2], bs[2]), from = 0, to = 1, type = "l", add = TRUE)
curve(dbeta(x, as[3], bs[3]), from = 0, to = 1, type = "l", add = TRUE)
title(expression(f(y)==frac(1,B(a,b))*y^{a-1}*(1-y)^{b-1}))
#pos <- locator(3) # locatorr() does not work well with RMarkdown
text(x=c(0.1, 0.525, 0.7), y = c(2.2, 1.75, 2.5), labels = c("B(a = 2, b = 6)", "B(a = 4, b = 4)", "B(a
```

$$f(y) = \frac{1}{B(a, b)} y^{a-1}(1-y)^{b-1}$$



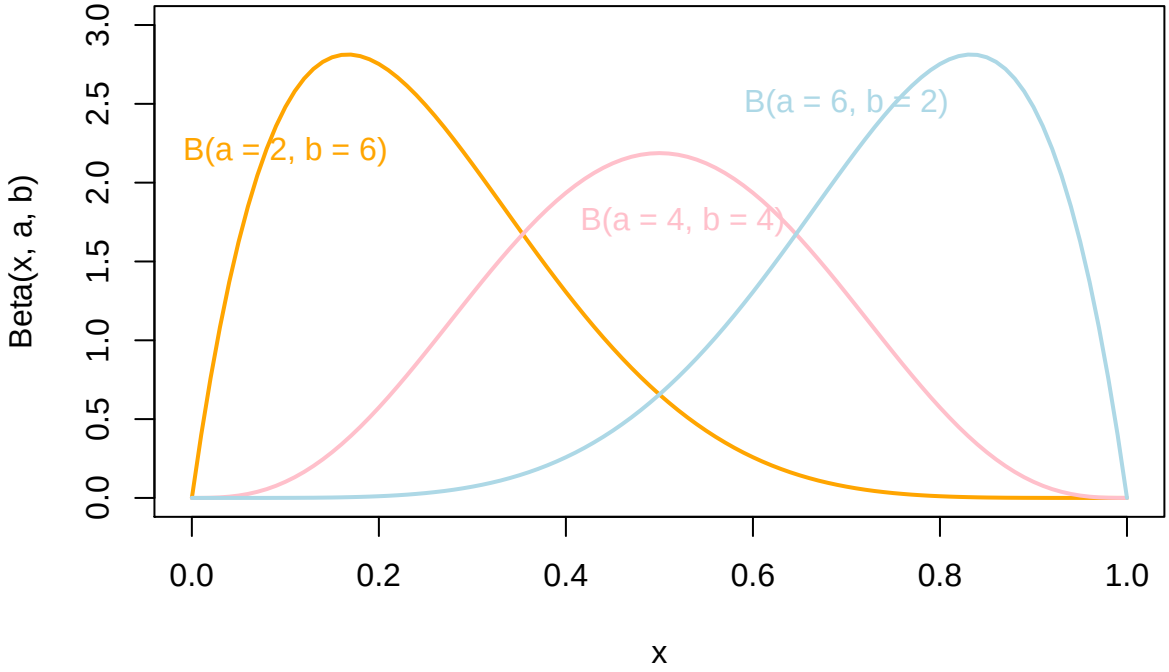
d. Redraw the graph using different colors or line types for the three beta density curves.

```r
as <- c(2,4,6)
bs <- c(6,4,2)
plot(curve(dbeta(x, as[1], bs[1]), from = 0, to = 1), type = "l",
     xlab = "x", ylab = "Beta(x, a, b)",
     ylim = c(0, 3), col = "orange", lwd = 2)
```



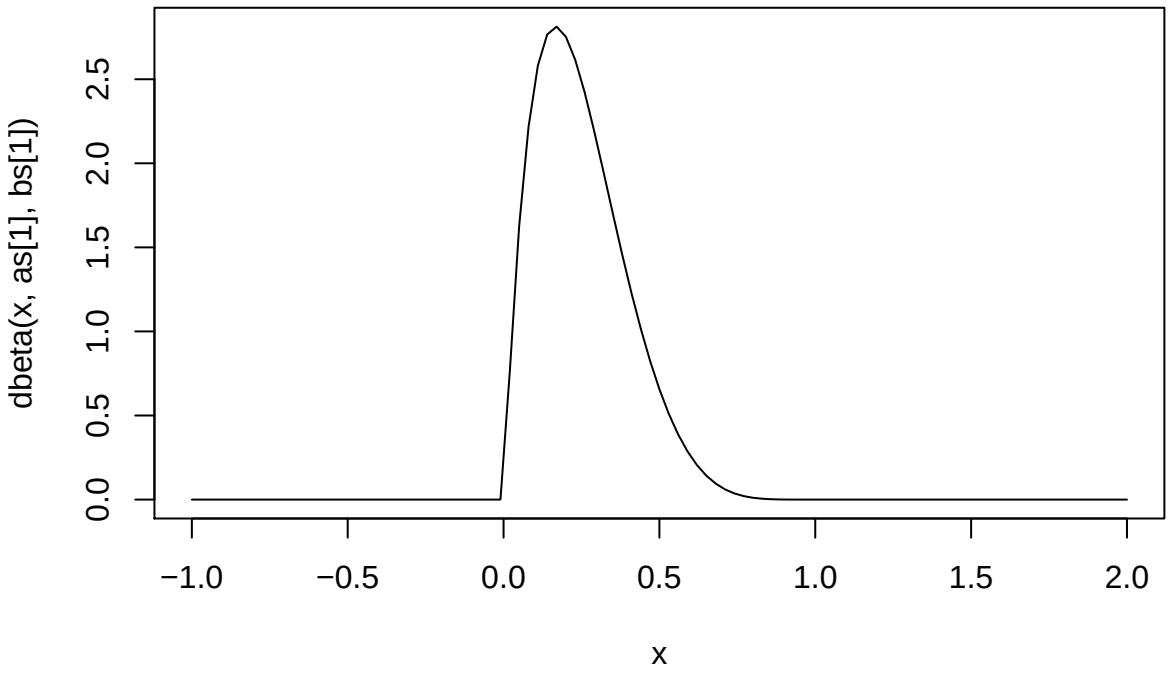

```r
curve(dbeta(x, as[2], bs[2]), from = 0, to = 1, type = "l", add = TRUE, col = "pink", lwd = 2)
curve(dbeta(x, as[3], bs[3]), from = 0, to = 1, type = "l", add = TRUE, col = "lightblue", lwd = 2)
title(expression(f(y)==frac(1,B(a,b))*y^{a-1}*(1-y)^{b-1}))
#pos <- locator(3) # locatorr() does not work well with RMarkdown
text(x=c(0.1, 0.525, 0.7), y = c(2.2, 1.75, 2.5), labels = c("B(a = 2, b = 6)", "B(a = 4, b = 4)", "B(a
```

$$f(y) = \frac{1}{B(a, b)} y^{a-1}(1-y)^{b-1}$$

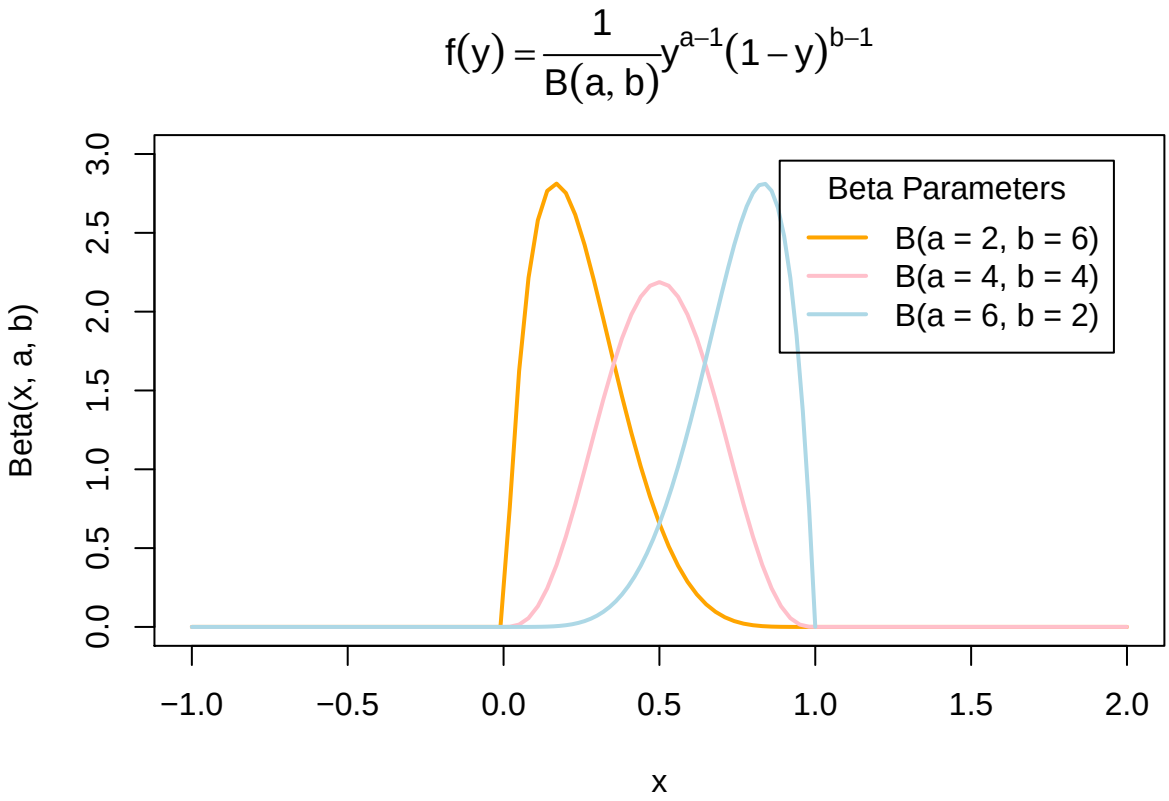


e. Instead of using the text function, add a legend to the graph that shows the color or line type for each of the beta density curves.

```
as <- c(2,4,6)
bs <- c(6,4,2)
plot(curve(dbeta(x, as[1], bs[1]), from = -1, to = 2), type = "l",
     xlab = "x", ylab = "Beta(x, a, b)",
     ylim = c(0, 3), col = "orange", lwd = 2)
```




```
curve(dbeta(x, as[2], bs[2]), from = -1, to = 2, type = "l", add = TRUE, col = "pink", lwd = 2)
curve(dbeta(x, as[3], bs[3]), from = -1, to = 1, type = "l", add = TRUE, col = "lightblue", lwd = 2)
title(expression(f(y)==frac(1,B(a,b))*y^{a-1}*(1-y)^{b-1}))
#pos <- locator(3) # locatorr() does not work well with RMarkdown
legend("topright", c("B(a = 2, b = 6)", "B(a = 4, b = 4)", "B(a = 6, b = 2)"),
       col = c("orange", "pink", "lightblue"),
       lwd = 2, inset = 0.05, title = 'Beta Parameters')
```



Problem 4.8

In Exercise 4.7, the waiting times for the Old Faithful geysers were compared for the short and long eruptions where the variable length in the faithful data frame defines the duration of the eruption.

a. Suppose a data frame dframe contains a numeric variable num.var and a factor factor.var. After the ggplot2 package has been loaded, then the R commands

$ggplot(dframe, aes(x = num.var, color = factor.var)) + geom\_density()$

will construct overlapping density estimates of the variable num.var for each value of the factor factor.var. Use these commands to construct overlapping density estimates of the waiting times of the geysers with short and long eruptions.

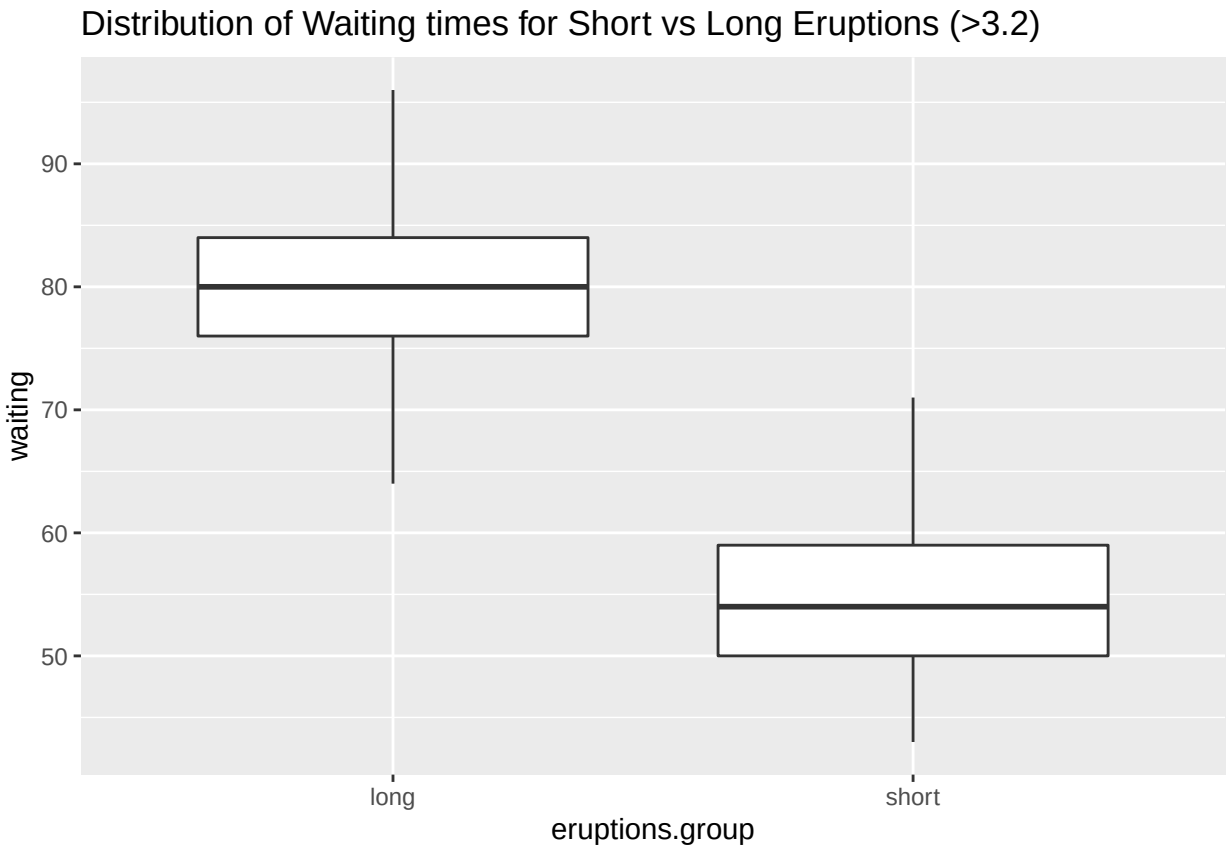
b. With a data frame dframe containing a numeric variable num.var and a factor factor.var, the ggplot2 syntax

$ggplot(dframe, aes(y = num.var, x = factor.var)) + geom\_boxplot()$

will construct parallel boxplots of the variable num.var for each value of the factor factor.var. Use these commands to construct parallel boxplots of the waiting times of the geysers with short and long eruptions.

```
data(faithful)
faithful %>% mutate(eruptions.group = ifelse(eruptions < 3.2, "short", "long")) %>%
    ggplot(aes(x = waiting, color = eruptions.group)) + geom_density() +
    ggtitle("Density of Waiting Times for Short vs Long Eruptions (>3.2)")
```


```
faithful %>% mutate(eruptions.group = ifelse(eruptions < 3.2, "short", "long")) %>%
    ggplot(aes(y = waiting, x = eruptions.group)) + geom_boxplot() +
    ggtitle("Distribution of Waiting times for Short vs Long Eruptions (>3.2)")
```



Problem 6.2

The datafile "nyc.marathon.txt" contains the gender, age, and completion time (in minutes) for 276 people who completed the 2010 New York City Marathon. It was reported that the mean ages of men and women marathoners in 2005 were respectively 40.5 and 36.1.

a. Create a new dataframe "women.marathon" that contains the ages and completion times for the women marathoners.

I downloaded the csv file from http://personal.bgsu.edu/~mrizzo/Rx/Rx-data/. Then I filtered it for females, selected the desired columns and saved the output as a new dataframe.

```
marathon <- read_csv("~/Desktop/Stat 204/nyc-marathon.csv")
```

```
## Parsed with column specification:
## cols(
##   Minutes = col_double(),
##   Gender = col_character(),
##   Age = col_double()
## )
```

```
women.marathon <- marathon %>% filter(Gender == "female") %>% dplyr::select(c("Minutes", "Age"))
```

b. Use the t.test function to construct a test of the hypothesis that the mean age of women marathoners is equal to 36.1.

We can put the value of 36.1 in as the μ parameter for the t.test. Alternatively, we could've tested whether the mean of $Age - 36.1$ is equal to zero. This t.test has an extremely low p-value, so we can reject the null hypothesis that the population mean is equal to 36.1.

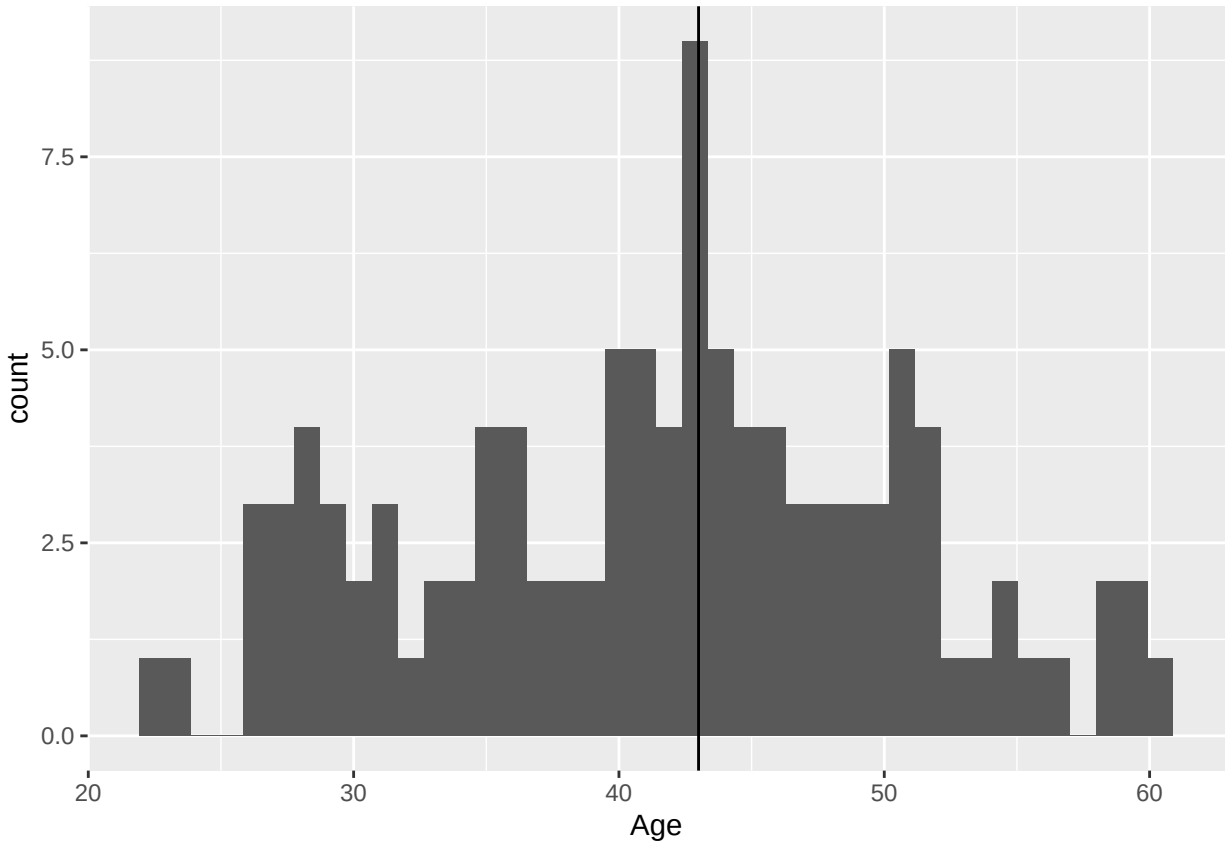
```
t.test(women.marathon$Age, mu = 36.1)
```

```
##
##  One Sample t-test
##
## data:  women.marathon$Age
## t = 6.1735, df = 106, p-value = 1.249e-08
## alternative hypothesis: true mean is not equal to 36.1
## 95 percent confidence interval:
##  39.80704 43.31446
## sample estimates:
## mean of x
##  41.56075
```

c. As an alternative method, use the wilcox.test function to test the hypothesis that the median age of women marathoners is equal to 36.1. Compare this test with the t-test used in part (b).

The Wilcox signed rank test places less assumptions on the underlying distribution than the t-test. The Wilcox test only assumes that the distribution is symmetric around is median. We produce a histogram of the ages below with a vertical line at the median and note that the distribution appears fairly symmetric around the median.

```
ggplot(data = women.marathon, aes(x = Age)) + geom_histogram(bins = 40) +
  geom_vline(xintercept = median(women.marathon$Age))
```



The text book also produces a QQ plot to as a diagnostic to confirm that the data is symetrically distributed around the median. We produce that plot below and see that the data appears to be symmetrically distributed around the median.

This Wilcox test returns another extremely low p-value, so we reject the null hypothesis that the population mean is equal to 36.1.

```
wilcox.test(women.marathon$Age, mu = 36.1)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  women.marathon$Age
## V = 4522, p-value = 3.879e-07
## alternative hypothesis: true location is not equal to 36.1
```

d. Construct a 90% interval estimate for the mean age of women marathoners.

Since we want a confidence interval for the mean (and not the median), we will use the t.test function with $conf.level = 0.90$ to get the 90% confidence interval for the mean age of women marathones, which we see to be $(40.09, 43.03)$

```
t.test(women.marathon$Age, conf.level = 0.90)
```

```
##
##  One Sample t-test
```

```
##
## data:  women.marathon$Age
## t = 46.985, df = 106, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  40.09296 43.02854
## sample estimates:
## mean of x
##  41.56075
```

Problem 6.3 (a, b)

From the information in the 2005 report, one may believe that men marathoners tend to be older than women marathons.

a. Use the t.test function to construct a test of the hypothesis that the mean ages of women and men marathoners are equal against the alternative hypothesis that the mean age of men is larger.

We filter the dataset by gender and then compare the ages with a t.test with the alternative hypothesis being "less", meaning the mean age from the first vector (women) is greater than the mean age from the second vector (men). The resulting t.test has a p-value = 0.007443, which in most cases, is enough to reject the null hypothesis. This t.test indicates male marathon runners are older.

```
t.test(data = marathon, Age ~ Gender,
       paired = FALSE,
       alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  Age by Gender
## t = -2.4519, df = 252.66, p-value = 0.007443
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf -0.974724
## sample estimates:
## mean in group female   mean in group male
##             41.56075             44.54438
```

b. Construct a 90% interval estimate for the difference in mean ages of men and women marathoners.

The confidence interval for the difference in mean ages between males and females is $(-\infty, 1.420086)$.

```
t.test(data = marathon, Age ~ Gender,
       paired = FALSE,
       alternative = "less",
       conf.level = 0.90)
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  Age by Gender
## t = -2.4519, df = 252.66, p-value = 0.007443
## alternative hypothesis: true difference in means is less than 0
## 90 percent confidence interval:
##       -Inf -1.420086
## sample estimates:
## mean in group female   mean in group male
##             41.56075             44.54438
```

Problem 6.5

The datafile "buffalo.cleveland.snowfall.txt" contains the total snowfall in inches for the cities Buffalo and Cleveland for the seasons 1968-69 through 2008-09.

a. Compute the differences between the Buffalo snowfall and the Cleveland snowfall for all seasons.

```
snow <- read.table("buffalo.cleveland.snowfall.txt", header = TRUE) %>%
    mutate(diff = Cleveland - Buffalo)
```

b. Using the t.test function with the difference data, test the hypothesis that Buffalo and Cleveland get, on average, the same total snowfall in a season.

The t.test on the difference in snowfall returns an extremely low p-value, so we reject the null hypothesis that the two cities have the same average snowfall. Given how we defined the difference variables, a negative value for the difference in means tells us that Buffalo has more snowfall.

```
t.test(snow$diff)
```

```
##
##  One Sample t-test
##
## data:  snow$diff
## t = -7.5692, df = 40, p-value = 3.061e-09
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -42.45731 -24.56221
## sample estimates:
## mean of x
## -33.50976
```

c. Use the t.test function to construct a 95% confidence interval of the mean difference in seasonal snowfall.

The default parameter for confidence level is 95%, but we will state that explicitly here anyways.

```
t.test(snow$diff, conf.level = 0.95)$conf.int
```

```
## [1] -42.45731 -24.56221
## attr(,"conf.level")
## [1] 0.95
```

Problem 6.7

In Example 1.2, the height of the election winner and loser were collected for the U.S. Presidential elections of 1948 through 2008. Suppose you are interested in testing the hypothesis that the mean height of the election winner is equal to the mean height of the election loser. Assuming that this data represent paired data from a hypothetical population of elections, use the t.test function to test this hypothesis. Interpret the results of this test.

We read in the data as we did before, run the paired t.test. The p-value = 0.2041 provides very weak evidence to reject the null hypothesis, indicating equivalent means of the winner's and opponent's heights. We also note that the confidence interval for the difference in means includes zero.

```
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)
pres <- data.frame(winner, opponent)

t.test(pres$winner, pres$opponent, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  pres$winner and pres$opponent
## t = 1.3279, df = 15, p-value = 0.2041
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.361429  5.861429
## sample estimates:
## mean of the differences
##                    2.25
```

Problem 7.7

For the cars data in Example 7.1, compare the coefficient of determination R2 for the two models (with and without intercept term in the model). Hint: Save the fitted model as L and use summary(L) to display R2. Interpret the value of R2 as a measure of the fit.

R-squared is a metric that tells us the percentage of variance in the response variablea (distance here) is explained by the model. A higher R-squared indicates a model that explains more variance in the response, however, R-squared generally increases when we add parameters to the model, so it is not the best metric for a just model comparison here. Adjusted R-squared only increases when we add a parameter if the parameter improves the model more than expected by chance. First we fit the model with an intercept term and return the summary noting the $MultipleR^2 = 0.6511, AdjustedR^2 = 0.6438$

```
data(cars)

L1 = lm(data = cars, dist ~ speed)

summary(L1)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

Now we fit the model without the intercept term, this is also called regression through the origin, and return the model summary. Here we see $MultipleR^2 = 0.8963, AdjustedR^2 = 0.8942$

```
L2 = lm(data = cars, dist ~ speed + 0)

summary(L2)
```

```
##
## Call:
## lm(formula = dist ~ speed + 0, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.183 -12.637  -5.455   4.590  50.181
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## speed   2.9091     0.1414   20.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.26 on 49 degrees of freedom
## Multiple R-squared:  0.8963, Adjusted R-squared:  0.8942
## F-statistic: 423.5 on 1 and 49 DF,  p-value: < 2.2e-16
```

This is an interesting result because the smaller model (L2) has a higher R^2 and $AdjustedR^2$. When we are performing regression through the orgin, we are stipulating that $E(Y|x = 0) = 0$ which changes how we compute SS_{model} and $SS_{residual}$ - the two main components of the formula for coefficient of determination. Accordingly, we are not comparing apples to apples with this metric here. I think anova() is a better way to compare these models, this indicates that the model without the the intercept term is better.

```
anova(L1, L2)
```

```
## Analysis of Variance Table
##
## Model 1: dist ~ speed
## Model 2: dist ~ speed + 0
```

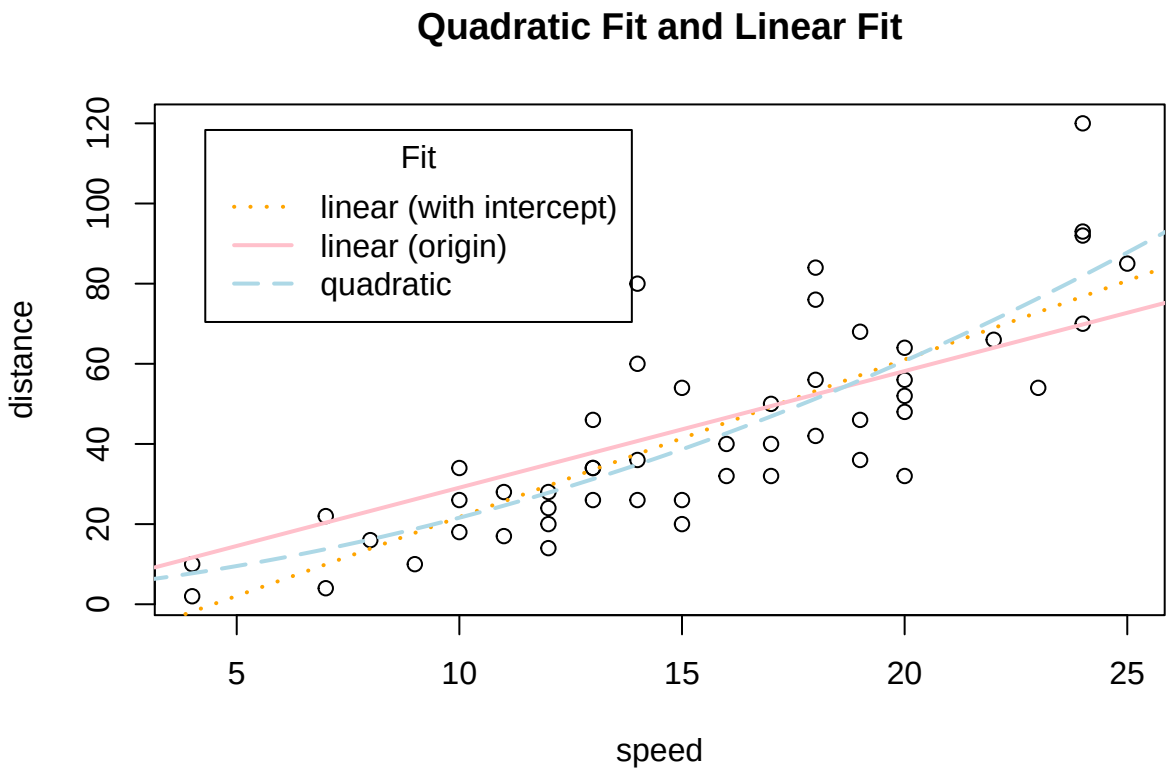
```
##   Res.Df   RSS Df Sum of Sq      F  Pr(>F)
## 1     48 11354
## 2     49 12954 -1   -1600.3 6.7655 0.01232 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Problem 7.8

Refer to the cars data in Example 7.1. Create a new variable speed2 equal to the square of speed. Then use lm to fit a quadratic model. The corresponding model formula would be dist ~ speed + speed2. Use curve to add the estimated quadratic curve to the scatterplot of the data and comment on the fit. How does the fit of the model compare with the simple linear regression model of Example 7.1 and Exercise 7.7?

This is very similar to Exercise 4.2, so I will bring some of the code from that problem down here. Visually, the quadratic model looks very good - it fits the low speed values very well, and is able to model some of the increased variance in distance as speed gets high (above 20). Visually, it looks better then the linear model,

```
L3 <- lm(data = mutate(cars, speed2 = speed^2), dist ~ speed + speed2)

newdat <- seq(from = min(cars$speed) - 2, to = max(cars$speed) + 2, length.out = 100)
quadratic.fitted <- predict(L3, list(speed = newdat, speed2 = newdat^2))

plot(cars$speed, cars$dist, main = "Quadratic Fit and Linear Fit", xlab = "speed", ylab = "distance")
abline(reg = L1, lty = 3, lwd = 2, col = "orange")
abline(reg = L2, lty = 1, lwd = 2, col = "pink")
lines(x = newdat, y = quadratic.fitted, lty = 5, lwd = 2, col = "lightblue")


legend("topleft", c("linear (with intercept)", "linear (origin)", "quadratic"),
       lty = c("dotted", "solid", "longdash"),
       col = c("orange", "pink", "lightblue"),
       lwd = 2, inset = 0.05, title = 'Fit')
```



To compare these models, it is also important to look at the anova table. Here, we see that the quadratic model has a lower RSS, but it is not significant enough (at say, alpha = 0.05) to conclude that the quadratic model performs significantly better than the linear model with intercept term.

```
anova(L1, L3)
```

```
## Analysis of Variance Table
##
## Model 1: dist ~ speed
## Model 2: dist ~ speed + speed2
##   Res.Df   RSS Df Sum of Sq     F Pr(>F)
## 1     48 11354
## 2     47 10825  1    528.81 2.296 0.1364
```

Problem 8.2

The PlantGrowth data is an R dataset that contains results from an experiment on plant growth. The yield of a plant is measured by the dried weight of the plant. The experiment recorded yields of plants for a control group and two different treatments. After a preliminary exploratory data analysis, use one-way ANOVA to analyze the differences in mean yield for the three groups. Start with the exploratory data analysis, and also check model assumptions. What conclusions, if any, can be inferred from this sample data?

First, we can get some basic info - for each group, the number of data points, the mean value and the sample variance. We want to do a one way anova test, so its important to know if the variances between the groups are approximately equal.

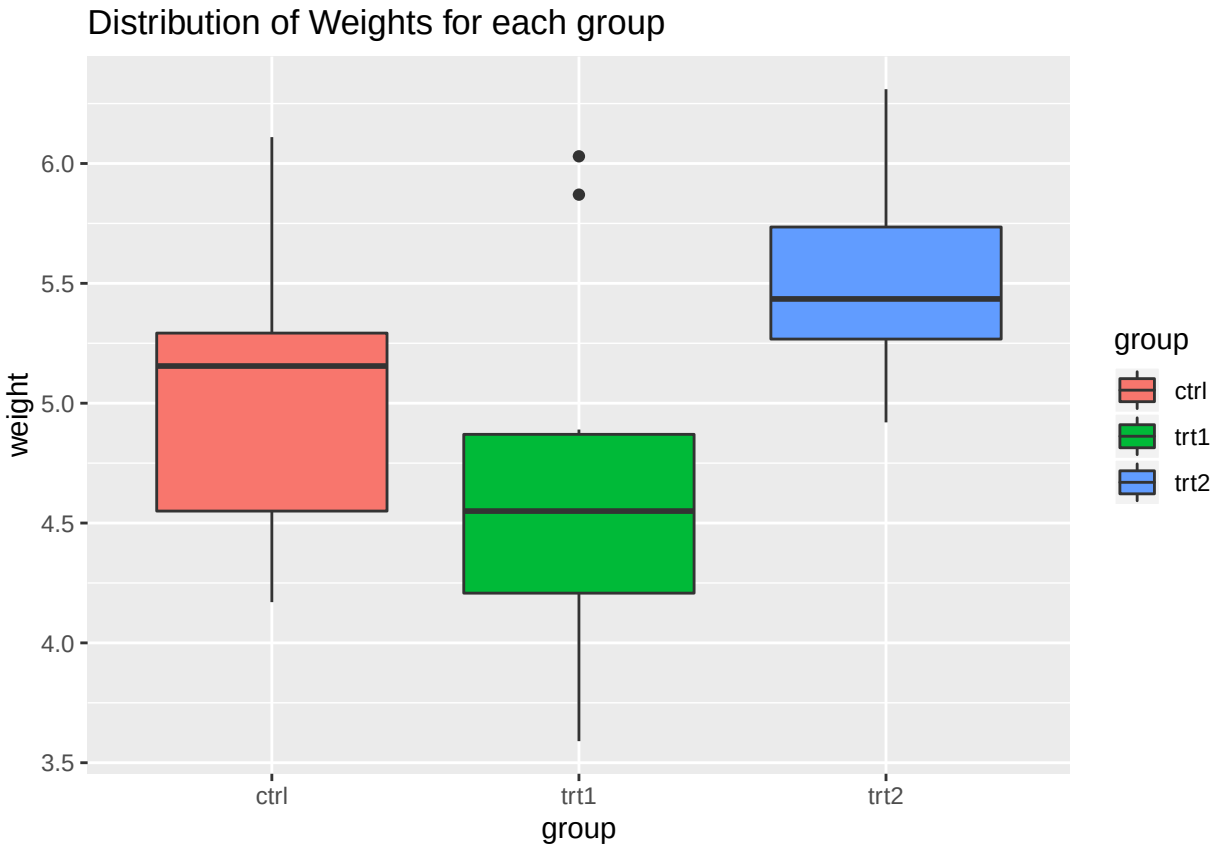
```
data(PlantGrowth)
PlantGrowth %>% group_by(group) %>%
    summarise(n.data.pts = n(),
```

```
              weight.mean = mean(weight),
              weight.var = var(weight))
```

```
## # A tibble: 3 x 4
##   group n.data.pts weight.mean weight.var
##   <fct>      <int>       <dbl>      <dbl>
## 1 ctrl          10        5.03      0.340
## 2 trt1          10        4.66      0.630
## 3 trt2          10        5.53      0.196
```

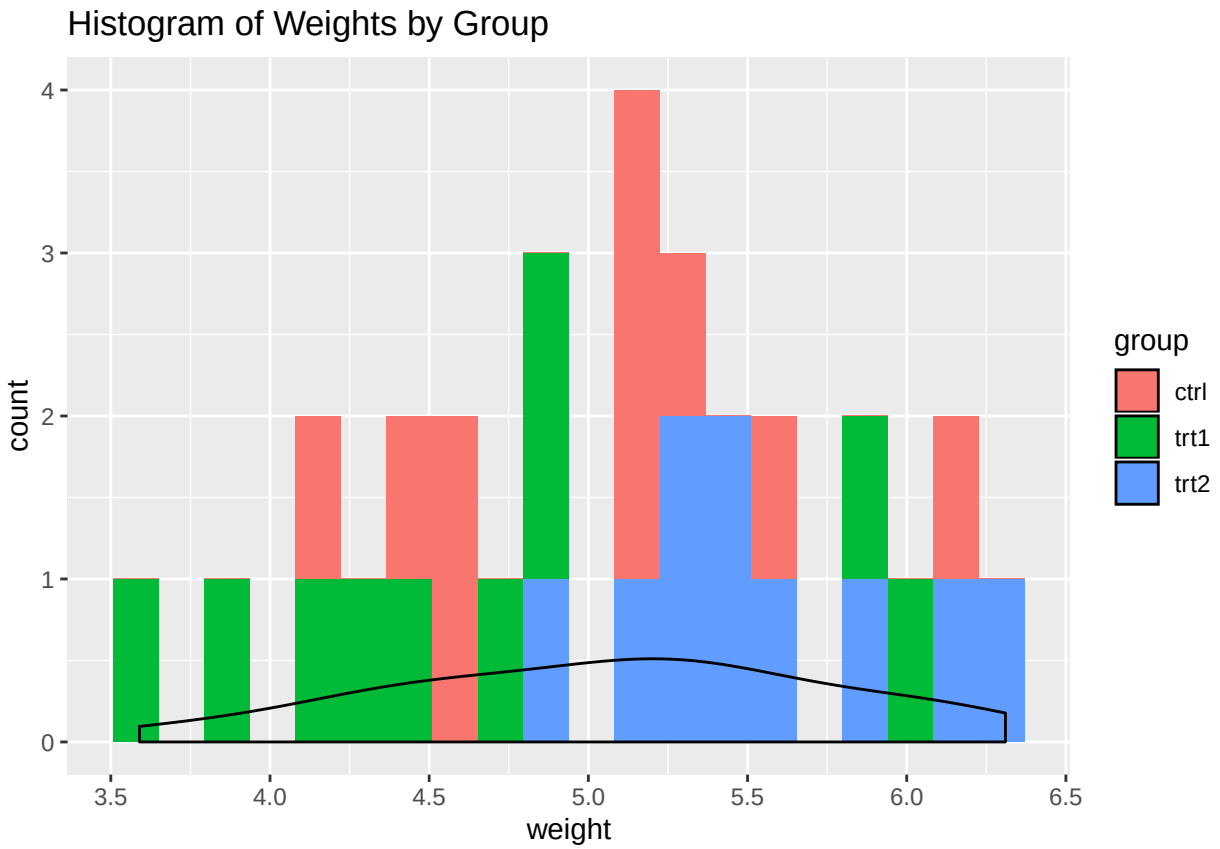
Data visualizations are better than boxes of numbers:

```
ggplot(data = PlantGrowth, aes(x = group, y = weight, fill = group)) + geom_boxplot() +
    ggtitle("Distribution of Weights for each group")
```



We can also visualize this with a histogram:

```
ggplot(data = PlantGrowth, aes(x = weight)) +
    geom_histogram(aes(fill = group), bins = 20) +
    geom_density() +
    ggtitle("Histogram of Weights by Group")
```



This box plot is interesting - the means vary from group to group and while the variances look similar, this is not conclusive. More rigorous methods of comparison are needed. We go to the one-way Anova test. There is no reason to assume the variance are equal here.

```
oneway.test(data = PlantGrowth, weight ~ group)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  weight and group
## F = 5.181, num df = 2.000, denom df = 17.128, p-value = 0.01739
```

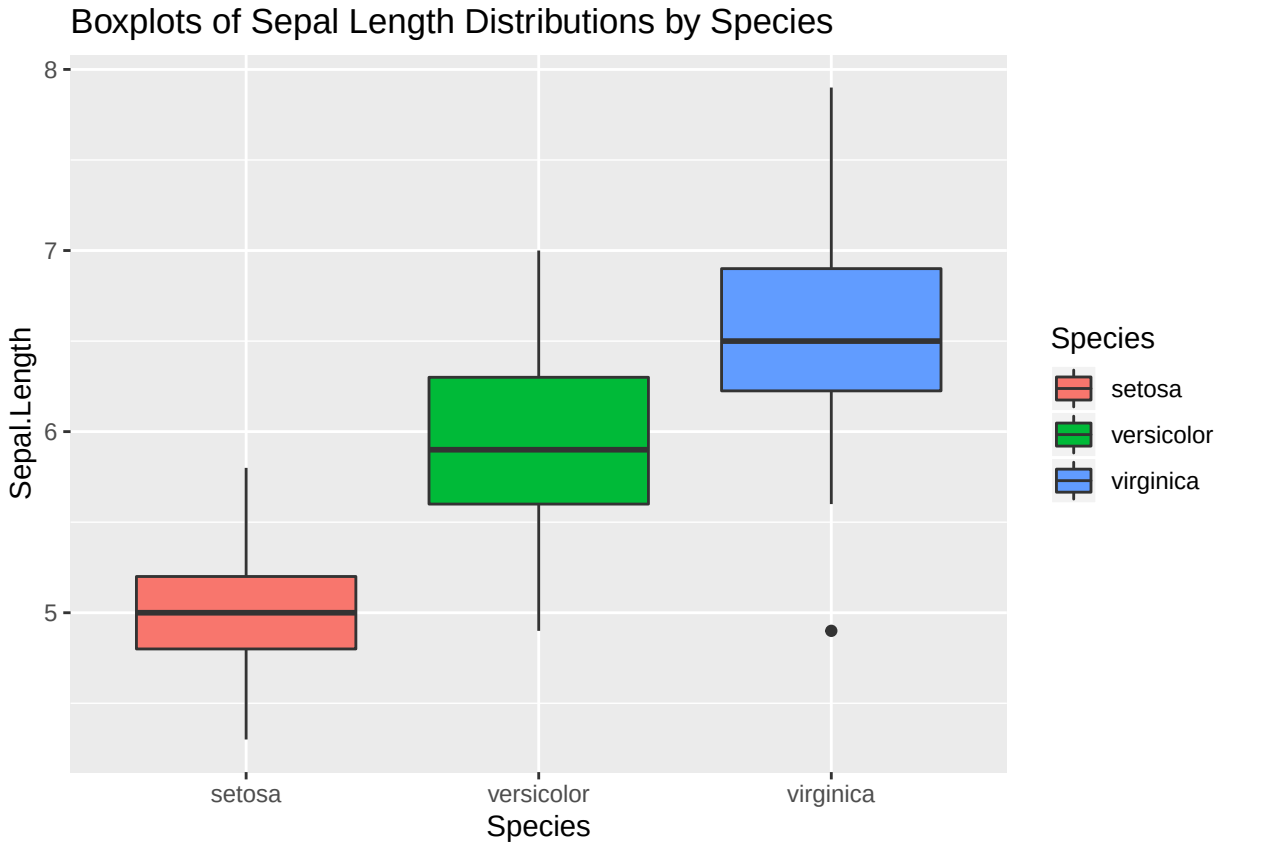
Using an alpha value of 0.05, which is standard, we can reject the null hypothesis in this case and conclude that not all of the group means are equal. However, the one-way anova test does not tell us which pair / pairs / of group means were detected to be different. This is good - the visual diagnostics were not totally conclusive, and the anova test gives us a p-value that can go either way (if we want to be really strict, at say alpha = 0.01, we would fail to reject the null hypothesis, which is a reasonable conclusion given the data).

Problem 8.3

The iris data has 50 observations of four measurements for each of three species of iris: setosa, versicolor, and virginica. We are interested in possible differences in the sepal length of iris among the three species. Perform a preliminary analysis as in Example 8.3. Write the effects model for a one-way ANOVA. What are the unknown parameters? Next fit a one-way ANOVA model for Sepal.Length by Species using lm. Display the ANOVA table. What are the parameter estimates?

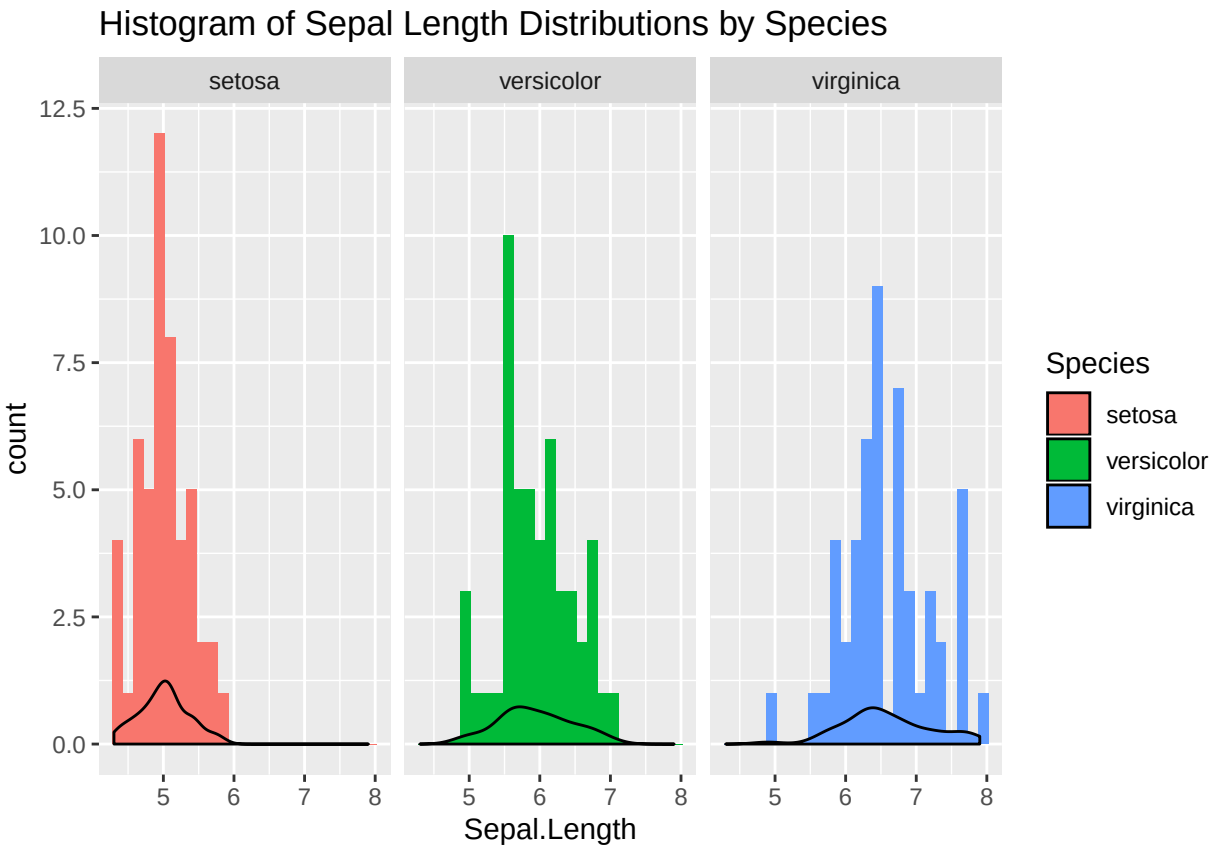
Since we are primarily interested in Sepal Length, we will begin with boxplots for each group:

```
data(iris)
ggplot(data = iris, aes(x = Species, y = Sepal.Length, fill = Species)) + geom_boxplot() +
    ggtitle("Boxplots of Sepal Length Distributions by Species")
```



Immediately, we can see evidence of a significant difference in group means. The boxes of the boxplots barely overlap, and it seems like the groups might have unequal variances. The same data in a histogram is interesting, but a little less insightful in my opinion. We note that the Speal Length's density estimation is approximately normal for each group

```
ggplot(data = iris, aes(x = Sepal.Length, fill = Species)) + geom_histogram(bins = 25) +
    facet_grid(~Species) +
    geom_density() +
    ggtitle("Histogram of Sepal Length Distributions by Species")
```



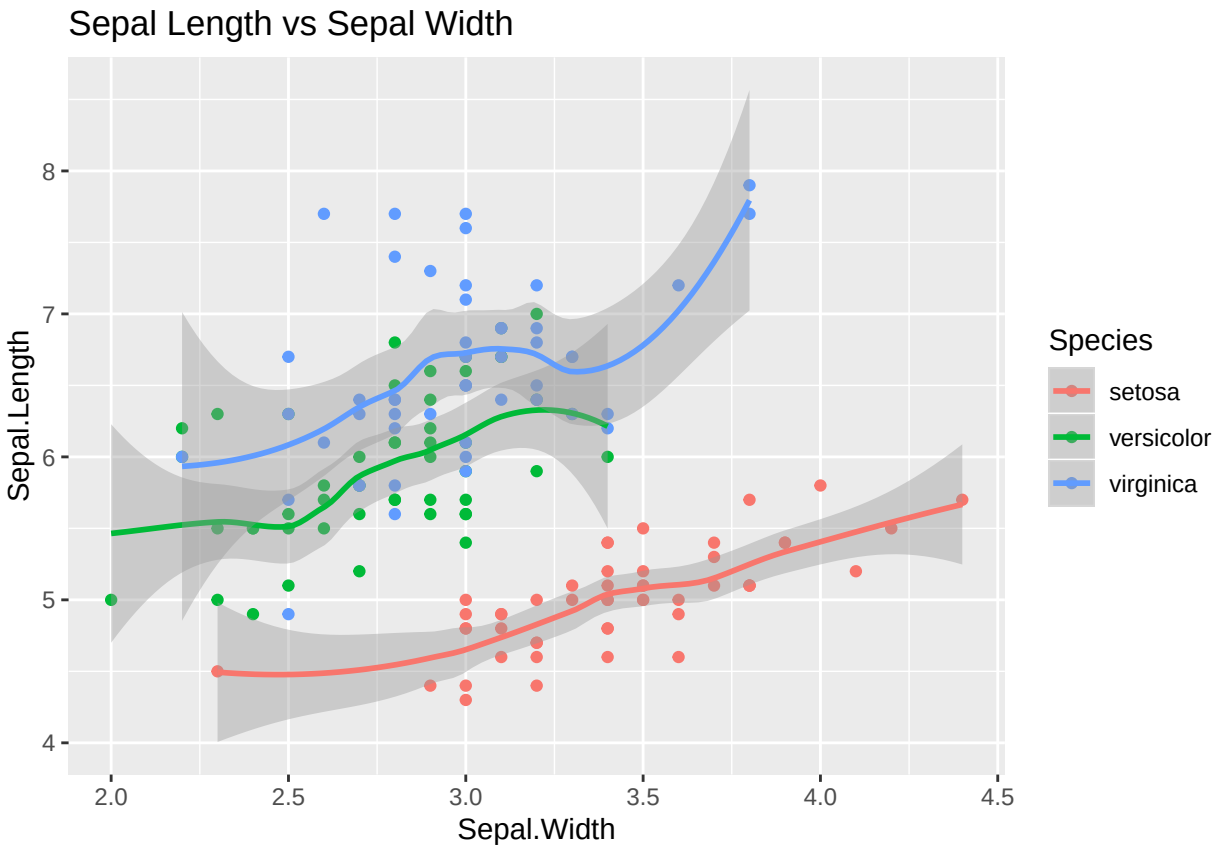
The other numerical variables are not asked about explicitly, but it is important to look at the correlations between these variables and visualize their relationships with scatterplots.

```
iris %>%
    dplyr::select(c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width")) %>%
    cor()
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
## Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
## Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
## Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

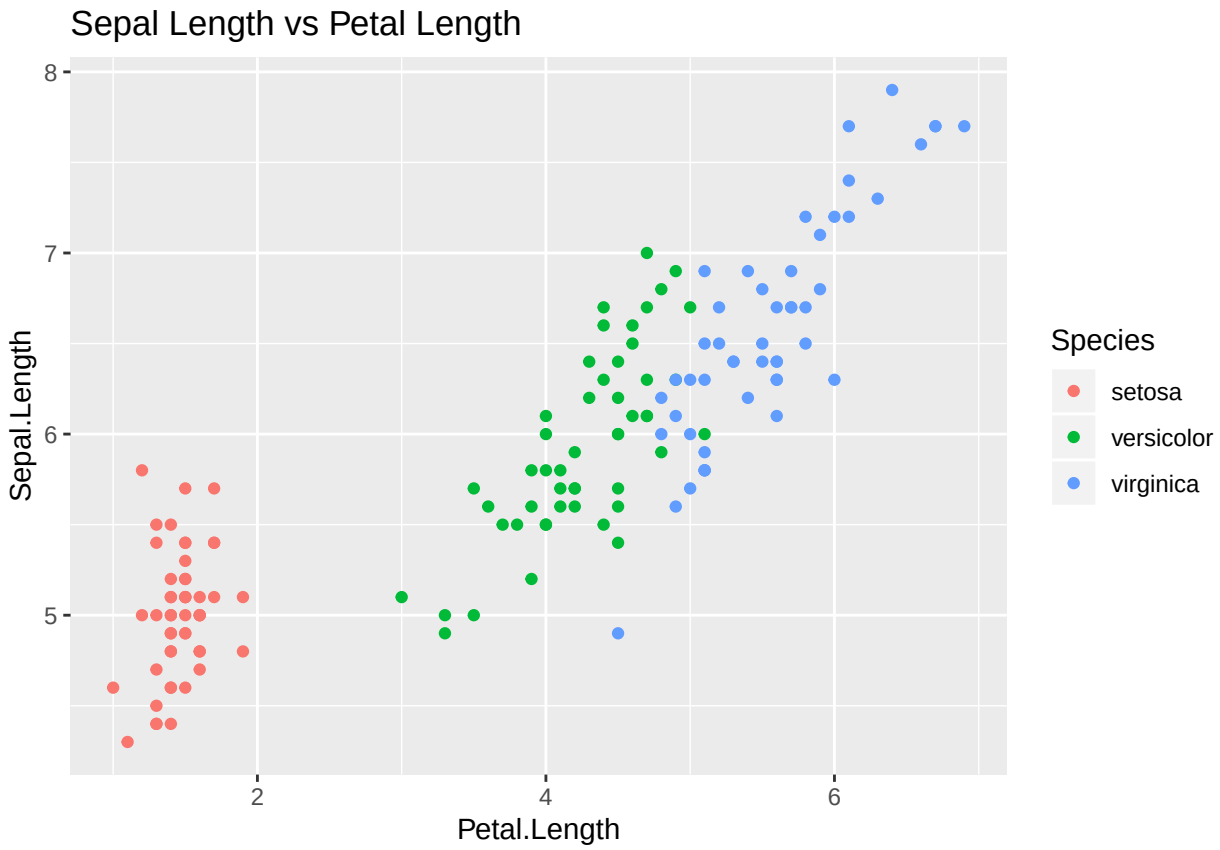
There are many high correlations here, but it is interesting to note the correlation between Speal Length and Sepal Width is negative and weak (correlation = -0.117). Scatterplots of Sepal Length vs each of the other variables seems useful.

```
ggplot(data = iris, aes(y = Sepal.Length, x = Sepal.Width, col = Species)) + geom_point() +
    geom_smooth(method = "loess", formula = "y ~ x") +
    ggtitle("Sepal Length vs Sepal Width")
```

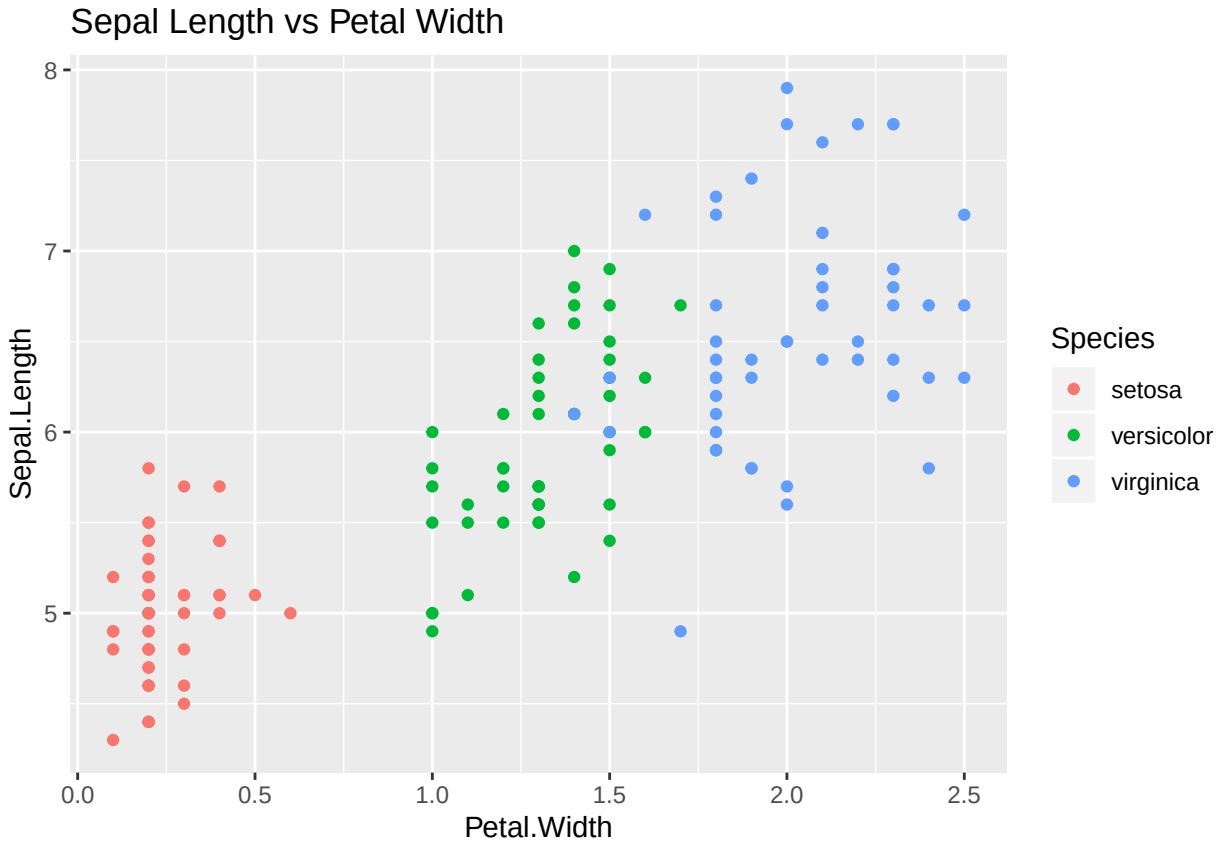


This plot is interesting because while we see a negative correlation between Sepal Length and Sepal Width overall, it looks like there is a weak positive correaltion between these variables, within each Species.

```
ggplot(data = iris, aes(y = Sepal.Length, x = Petal.Length, col = Species)) + geom_point() +
    ggtitle("Sepal Length vs Petal Length")
```




```
ggplot(data = iris, aes(y = Sepal.Length, x = Petal.Width, col = Species)) + geom_point() +
    ggtitle("Sepal Length vs Petal Width")
```



The Petal Length and Petal Width plots plot shows us that the mean petal length and width for the different species are dramatically different. Setosa's petals are, on average, much smaller in both dimensions than the other species. Virginica's petals are the biggest in both dimensions. Now we can do the one way anova test, there is no reason to assume equal variances.

```
oneway.test(data = iris, Sepal.Length ~ Species)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  Sepal.Length and Species
## F = 138.91, num df = 2.000, denom df = 92.211, p-value < 2.2e-16
```

The one-way test returns an extremely low p-value, so we have strong evidence to reject the null hypothesis. We conclude that not all of the groups have the same mean Sepal Length. Again, the one-way test does not tell us which groups have significantly different means, but the visualizations indicate that Setosa (smallest) and Virginica (largest) have significantly different means at the very least.

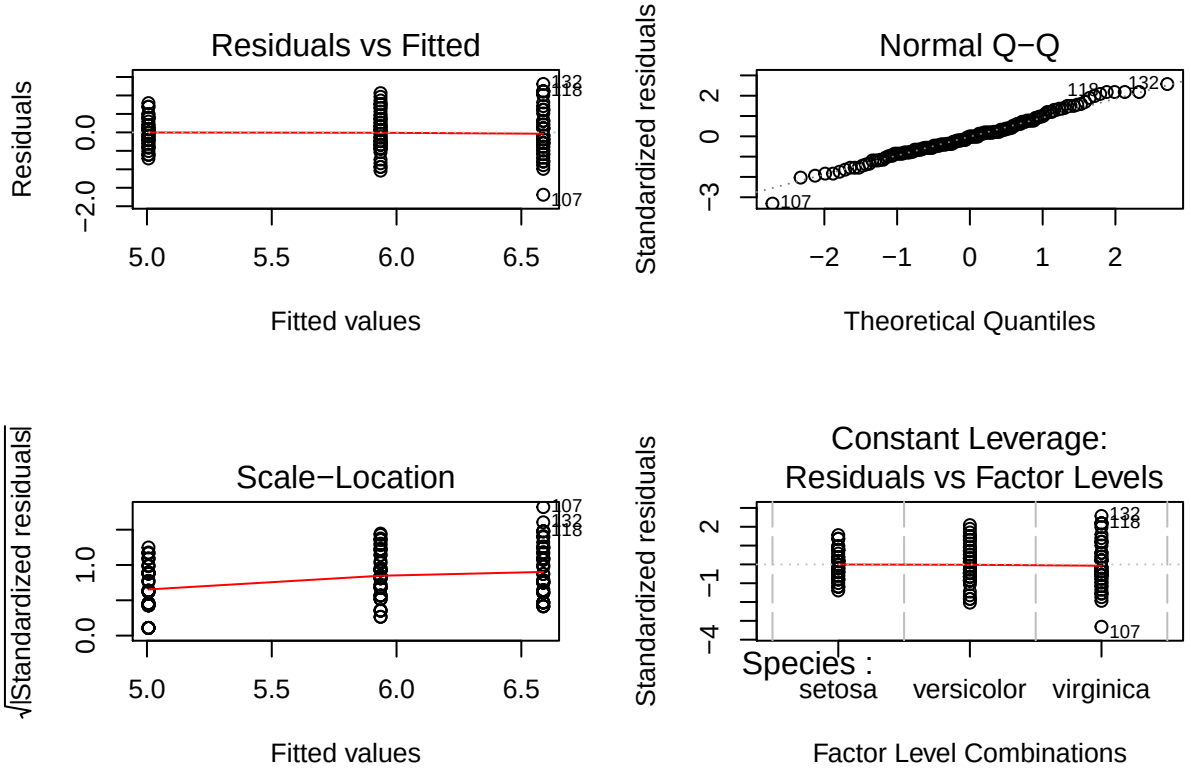
Problem 8.4

Refer to your results from Exercise 8.3. What are the assumptions required for inference? Analyze the residuals of the model to assess whether there is a serious departure from any of these assumptions. How can you check for normality of the error variable?

When performing a one-way anova test, we care assuming (1) the data points come from indpendent random samples and (2) the response variable follows a normal distribution. If we want, we can also assume that

the groups have equal variances, but we do not make that assumption here. The aov() model provides some in-built diagnostic plots that help us check for normality of the residuals. We display these visual disgnostics below:

```
L <- aov(data = iris, Sepal.Length ~ Species)
par(mfrow = c(2,2))
plot(L)
```



In the top-left, we see the residuals vs fitted values - and there is approximately equal residual variance across groups. In the top right, we see that the QQ plot hugs the $y = x$ line fairly tightly, indicating that the response variable does follow a normal distribution. The plots do not indicate a serious violation of our assumptions.

Problem 8.5

The cancer survival data "PATIENT.DAT" was introduced in Example 8.9. Start with the exploratory data analysis, and check for NID error model assumptions. Consider a transformation of the data if the assumptions for error are not satisfied. Complete a one-way ANOVA to determine whether mean survival times differ by organ. If there are significant differences, follow up with appropriate multiple comparisons to determine which means differ and describe how they differ.

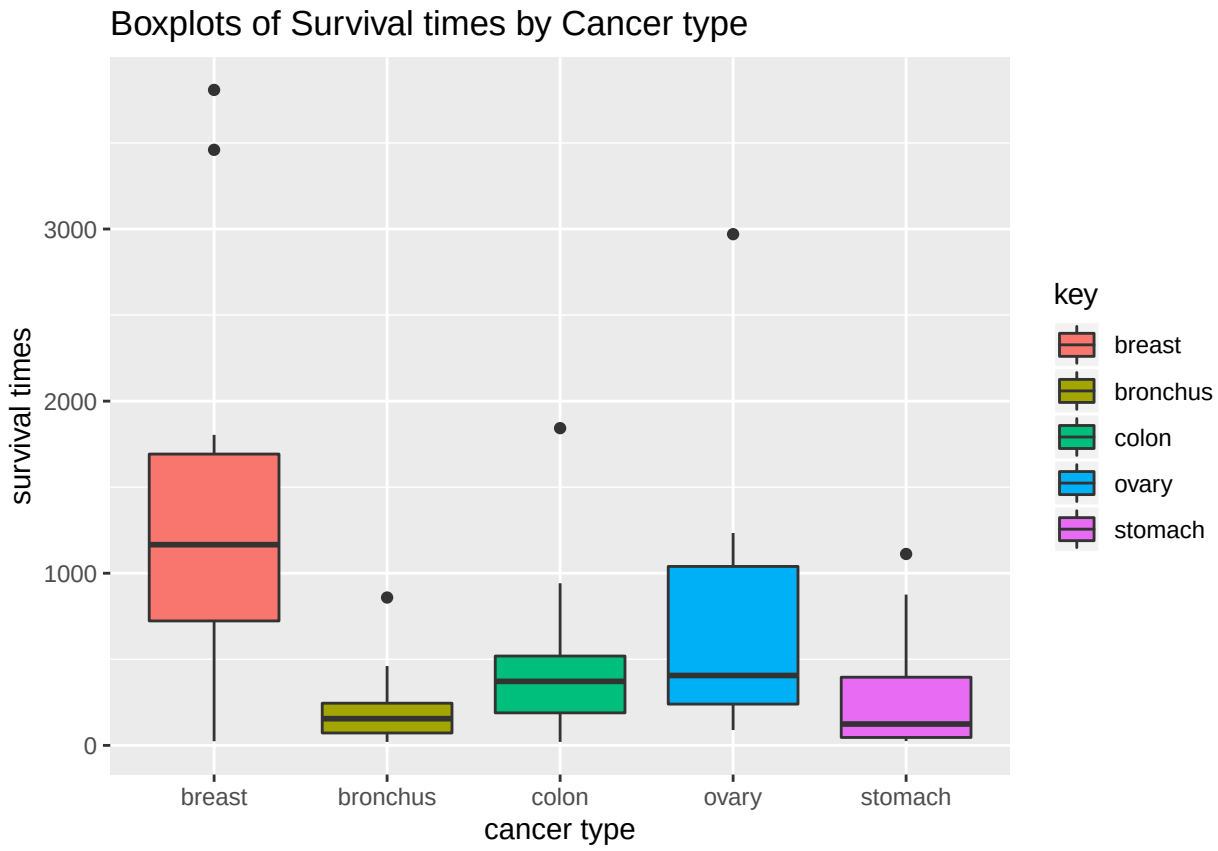
First we will read in the data and visualize the distribution of survival times by group with boxplots and histograms.

```
survive <- read_csv("PATIENT.csv")
```

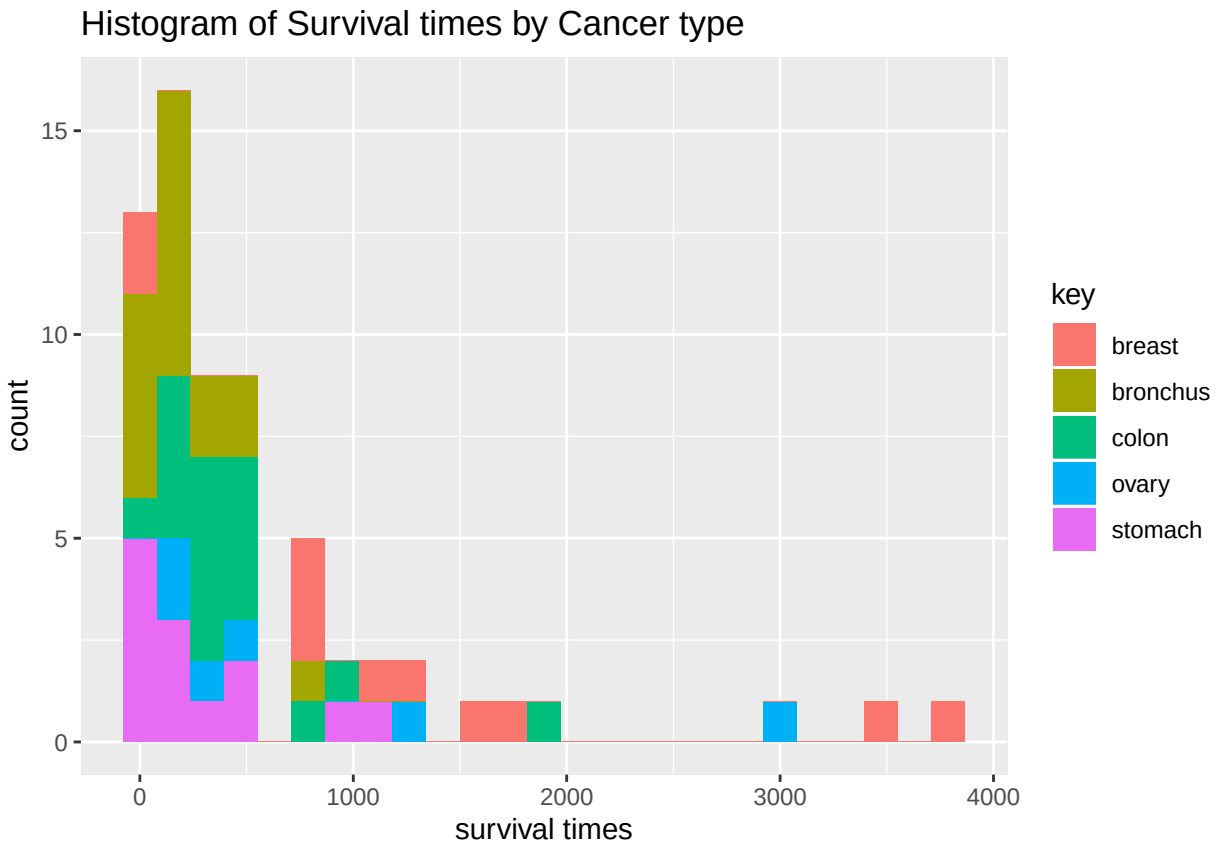
```
## Parsed with column specification:
## cols(
```

```
##   stomach = col_double(),
##   bronchus = col_double(),
##   colon = col_double(),
##   ovary = col_double(),
##   breast = col_double()
## )

## Warning: 2 parsing failures.
## row col  expected    actual         file
##  16  -- 5 columns 3 columns 'PATIENT.csv'
##  17  -- 5 columns 3 columns 'PATIENT.csv'
```

```
times1 <- survive %>% gather() %>% na.omit() %>%
    mutate(key = factor(key))

ggplot(data = times1, aes(x = key, y = value, fill = key)) + geom_boxplot() +
    xlab("cancer type") +
    ylab("survival times") +
    ggtitle("Boxplots of Survival times by Cancer type")
```

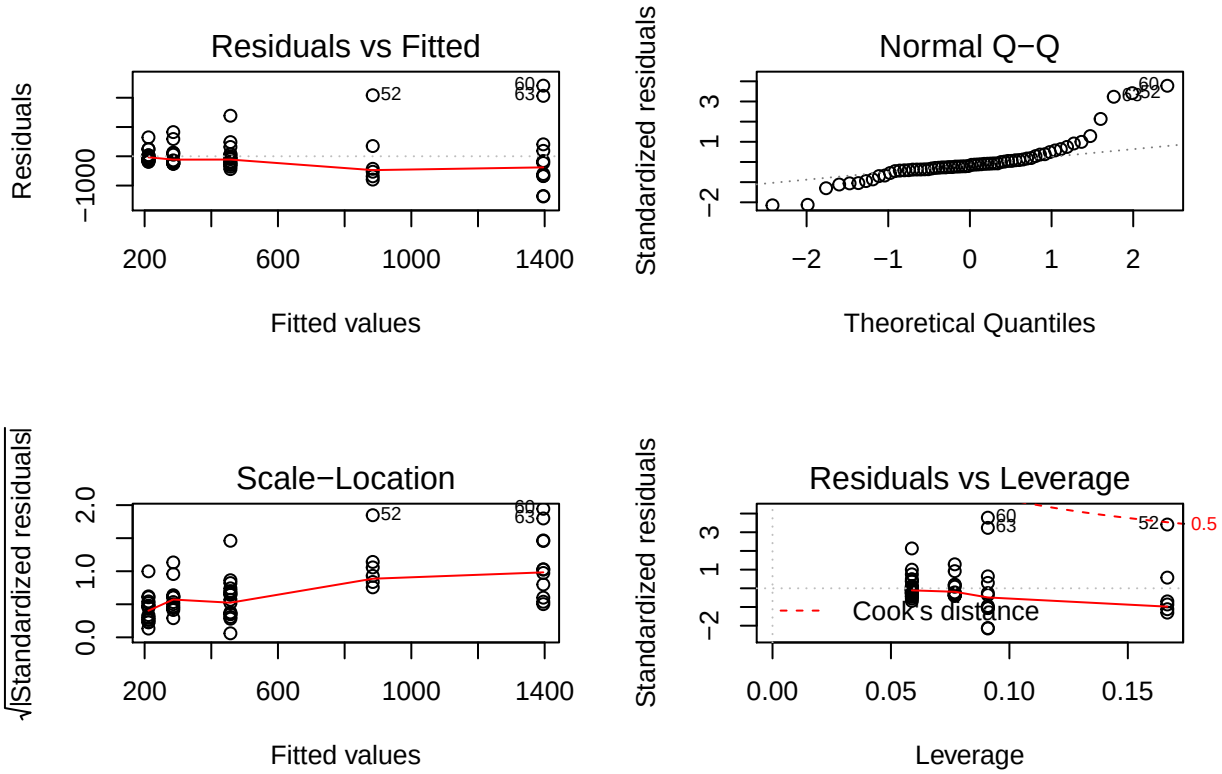



```
ggplot(data = times1, aes(x = value, fill = key)) + geom_histogram(bins = 25) +
    xlab("survival times") +
    ggtitle("Histogram of Survival times by Cancer type")
```



This histograms indicates the repsonse variable does not follow a normal distribution which violates one key assumption for the one-way anova test. This indicates that a transformation on the resposne variable is desirable. We will fit the model on the untransformed data and look at the residual distribution regardless.

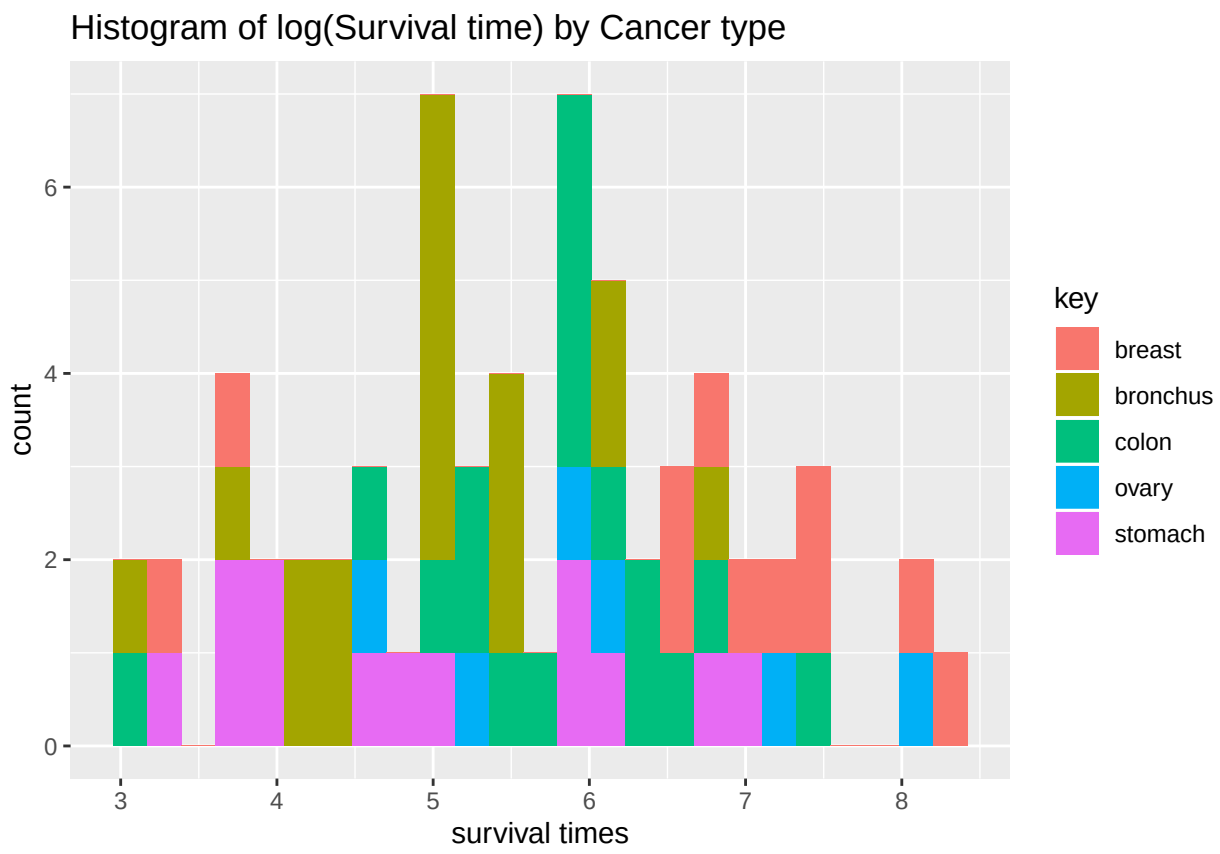
```
L2 <- aov(data = times1, value ~ key)
par(mfrow = c(2,2))
plot(L2)
```



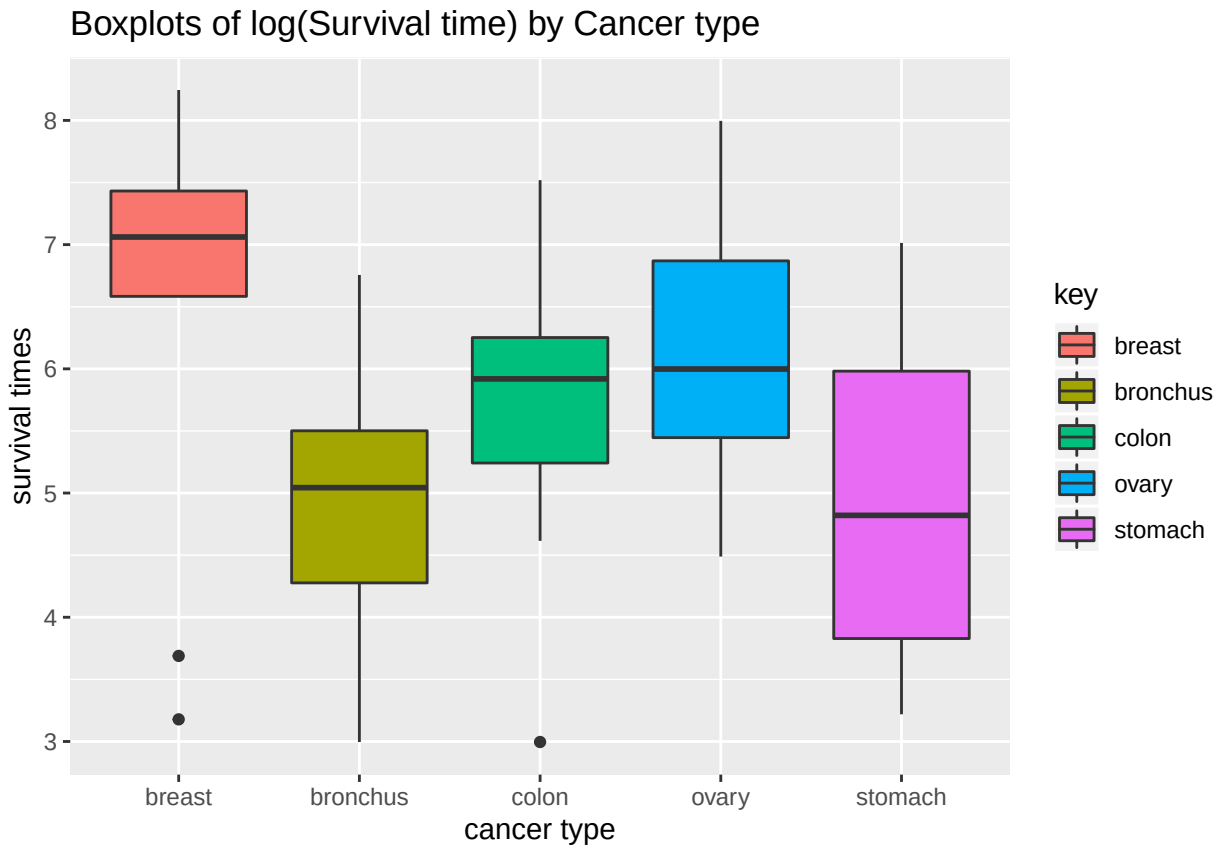
The residual plots are all over the place. The top left plot shows that there is unequal residual variance across groups. The Normal QQ plot has dramatic departures from the $y = x$ line, which indicates that the residuals from the model do not follow a normal distribution, which indicates that model isn't great.

Now, we will take the log transformatin of survival time and plot it's distribution across groups with analogous visualizations.

```
ggplot(data = times1, aes(x = log(value), fill = key)) + geom_histogram(bins = 25) +
    xlab("survival times") +
    ggtitle("Histogram of log(Survival time) by Cancer type")
```

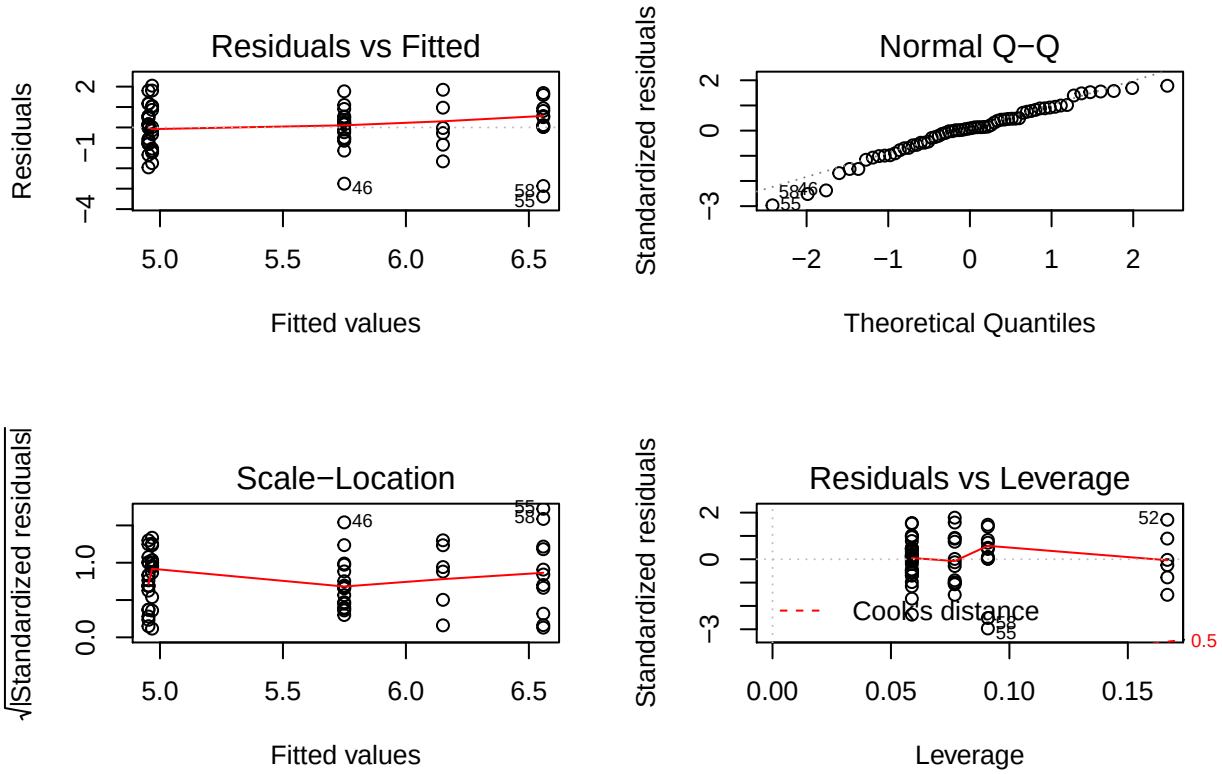



```
ggplot(data = times1, aes(x = key, y = log(value), fill = key)) + geom_boxplot() +
    xlab("cancer type") +
    ylab("survival times") +
    ggtitle("Boxplots of log(Survival time) by Cancer type")
```



The boxplots indicate that the log transformation on survival time results in a distribution that is closer to normal than the untransformed data. This means the log transformation fits the assumptions of the one-way anova test better. We fit the one-way anova model and inspect the residuals again.

```
L3 <- aov(data = times1, log(value) ~ key)
par(mfrow = c(2,2))
plot(L3)
```



These residual plots indicate that the log-transformed model's residuals follow an approximately normal distribution. In the top left, we see that the mean of rediuals is approximately zero for each group, with a tighter variances. The Normal QQ plot hugs the theoretical line more closely than in the untransformed model, which indicates that the log-transformed model's errors more closely follow a normal distribution.

Overall, the exploratoty boxplots and histograms indicate that the log-transformation on survival time results in a more normal distribution. The normality of response variable is a requisite for the proper implementation for the one-way anova test. Accordingly, we are not surprised to see that the log-transformed model's errors follow a normal distribution more closely.

Now, we can run the one-way anova test on the log-transformed values. Using an alpha level of 0.05, we can reject the null hypothesis and conclude that not all the organs have the same mean survival time.

```
oneway.test(data = times1, log(value) ~ key)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  log(value) and key
## F = 3.4103, num df = 4.000, denom df = 21.658, p-value = 0.02615
```

Now, we want to see which groups have a significant difference in mean survival time, using the $TukeyHSD()$ function in R. With a alpha level of 0.05, we conclude that only 2 pairs of cancer have significantly different mean survival times - the first pair is bronchus x breast cancer and the second pair is stomach x breast cancer. We also note that there is no significant difference between stomach and bronchus cancers. We can look at the p-values from the test, we can also look to see if the confidence interval for the difference in means includes zero or not.

```r
TukeyHSD(aov(data = times1, log(value) ~ key))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = log(value) ~ key, data = times1)
##
## $key
##                          diff       lwr        upr     p adj
## bronchus-breast  -1.60543320 -2.906741 -0.3041254 0.0083352
## colon-breast     -0.80948110 -2.110789  0.4918267 0.4119156
## ovary-breast     -0.40798703 -2.114754  1.2987803 0.9615409
## stomach-breast   -1.59068365 -2.968399 -0.2129685 0.0158132
## colon-bronchus    0.79595210 -0.357534  1.9494382 0.3072938
## ovary-bronchus    1.19744617 -0.399483  2.7943753 0.2296079
## stomach-bronchus  0.01474955 -1.224293  1.2537924 0.9999997
## ovary-colon       0.40149407 -1.195435  1.9984232 0.9540004
## stomach-colon    -0.78120255 -2.020245  0.4578403 0.3981146
## stomach-ovary    -1.18269662 -2.842480  0.4770864 0.2763506
```

```r
plot(TukeyHSD(aov(data = times1, log(value) ~ key)))
```