Homework 7

The purpose of this homework is to practice conducting inference and diagnostic plots for simple linear regression models. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through gradescope by 11:59pm on Sunday November 1st.

As always, if you need help with any of the homework assignments, please attend the TA office hours which are listed on Canvas and/or ask questions on Piazza. Also, if you have completed the homework, please help others out by answering questions on Piazza.

Part 1: Fitting a linear model and statistical inference on regression coefficients

On July 3rd 2015, my 1999 Toyota Corolla broke down on the side of the highway outside of Sturbridge MA. While I had the car repaired, I knew it was time to sell it and get a new car. I intended to sell my Corolla to the car dealership, the only catch was that I was not sure how much the used Corolla was worth. In the following exercises we will model how much a used Corolla is worth as a function of the number of miles it has been driven.

The data we will look at comes from Edmunds.com which is a website where you can buy new and used cars online. This data set is from the 2015 DataFest competition, which is an undergraduate data science competition that takes place at difference colleges across the United States. The data has been made available to this class for educational purposes, however please do not share this data outside of the class.

Note: for all plots on this homework, please use base R graphics

Part 1.1 (8 points): Let's start by loading the `car_transactions` and data set using the code below. Report how many cases and variables the full data set has. Then use the dplyr `select()` and `filter()` functions to create a reduced data frame object called `used_corollas` in which:

1. The only variables that should be in the `used_corollas` data frame are:

a) model_bought: the model of the car
b) new_or_used_bought: whether a car was new or used when it was purchased
c) price_bought: the price the car was purchased for
d) mileage_bought: the number of miles the car had when it was purchased

2. The only cases that should be in the `used_corollas` data frame are:

a) used cars
b) Toyota Corollas

c) cars that have been drive less than 150,000 miles

3. Finally use the `na.omit()` function on the `used_corollas` data frame to remove cases that have missing values.

If you have properly filtered the data, the resulting data set should have 248 cases, so check this is the case before going on to the next set of exercises.

```
# load the data set
load("car_transactions.rda")


# get the size of the original data set

(n_cases<-nrow(car_transactions))
```

```
## [1] 107832
```

```
(n_variables<-ncol(car_transactions))
```

```
## [1] 21
```

```
# use dplyr to reduce the data set to only used Corolla's with under 150,000 miles

used_corollas<- filter(car_transactions, mileage_bought<150000, model_bought == "Corolla",
                       new_or_used_bought == "U" )%>%
  select(model_bought, new_or_used_bought, price_bought, mileage_bought)%>%
    na.omit()




# check the size of the resulting data frame
(n_cases1<-nrow(used_corollas))
```

```
## [1] 248
```

```
(n_variables1<-ncol(used_corollas))
```

```
## [1] 4
```

Answers

The full data set has 107832 cases and 21 variables.

Part 1.2 (8 points):

Now that we have the relevant data, let's examine the relationship between a car's price and the number of miles driven! Let's begin analyzing the data by taking the following steps:

1. Plot the price as a function of the number of miles driven (use base R graphics for all plots on this homework).
2. Fit a linear model regression model that shows the predicted (expected) price as a function of the number of miles driven. Save this model to an object called `lm_fit` which you will use throughout the rest of this homework.
3. Add a red line to our plot showing the regression line fit.
4. Print the regression coefficients found.

Report how much does the price of a Corolla decrease for every additional mile it has been driven, and what this regression model suggests a car that has been driven 0 miles would be worth.

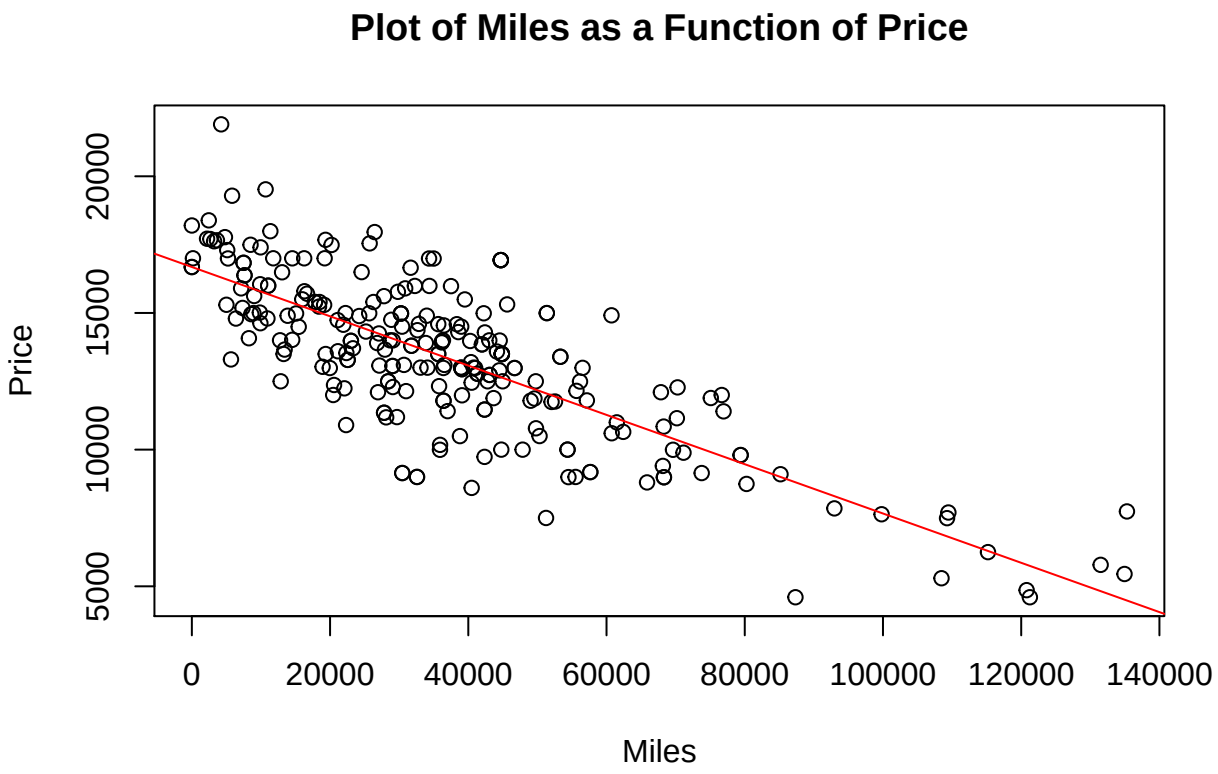
Also describe whether the the sign and magnitude of these regression coefficient values match what you might expect for car prices.

Finally, write out the regression equation.

```
# start by plotting the data


plot(used_corollas$mileage_bought, used_corollas$price_bought,
     main = "Plot of Miles as a Function of Price",
     xlab = "Miles",
     ylab = "Price")


# fit a regression model
(lm_fit<-lm(price_bought ~ mileage_bought, data = used_corollas))
```

```
##
## Call:
## lm(formula = price_bought ~ mileage_bought, data = used_corollas)
##
## Coefficients:
##    (Intercept)  mileage_bought
##    16681.91993        -0.09019
```

```
# add the regression line to the plot
abline(lm_fit, col= "red")
```




```
# display the regression coefficients

coef(lm_fit)
```

```
##    (Intercept) mileage_bought
## 16681.91992781    -0.09018627
```

Answers:

For every additional mile driven, the price decreases by -.09019. The regression model suggests that a car driven 0 miles would be worth 16681.92. The negative sign and the weak magnitude of the regression model are expected since (respectively) it would be expected for an increase in mileage to lower the value of the car and also, in terms of magnitude would not be expected for the prices of the cars to differ heavily.

$$\hat{y} = -0.09019x + 16681.92$$

$$\beta_0 = 16681.92$$

$$\beta_1 = -0.9019x$$

Part 1.3 (5 points): Now use R's `summary()` function to report whether there is statistically significant evidence that the price of a car decreases as a function of the number of miles driven. Also, write out the hypothesis that is being tested using the appropriate symbols/notation discussed in class.

```r
# get information about the statistical significance of the fit

summary(lm_fit)
```

```
##
## Call:
## lm(formula = price_bought ~ mileage_bought, data = used_corollas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4791.8 -1131.9    -0.3  1027.7  5600.7
##
## Coefficients:
##                    Estimate   Std. Error t value            Pr(>|t|)
## (Intercept)    16681.919928   204.459353   81.59 <0.0000000000000002 ***
## mileage_bought    -0.090186     0.004539  -19.87 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1816 on 246 degrees of freedom
## Multiple R-squared:  0.616,  Adjusted R-squared:  0.6145
## F-statistic: 394.7 on 1 and 246 DF,  p-value: < 0.00000000000000022
```

Answer

The p values are less than 0.05 (p-value < 0.001), therefore there is statistical evidence to reject the null hypothesis and assert that the price of the car decreases as a function of the number of miles driven.

The null hypothesis states that the the slope is 0 (there is no relationship between the price of the car and the number of miles driven).

$$H_0 : \beta_1 = 0$$
$$H_A : \beta_1 < 0$$

The alternative hypothesis is that the slope is less than 0 (there is a negative relationship between the price of the car and the number of miles driven)

Part 1.4 (5 points): We can create confidence intervals using a t-distribution via the `confint()` function. Report what the confidence interval for slope of the regression line is. Also, based on the confidence interval, explain why it seems likely that the price of a car is not independent of the number of miles driven.

```r
confint(lm_fit)
```

```
##                        2.5 %         97.5 %
## (Intercept)    16279.20570876 17084.63414685
## mileage_bought    -0.09912747    -0.08124508
```

Answer

The confidence interval lies from -0.09912747 and -0.08214508 for the slope of the regression line. Since 0 is not included in the confidence interval, we can conclude that the price of the car is not independent of

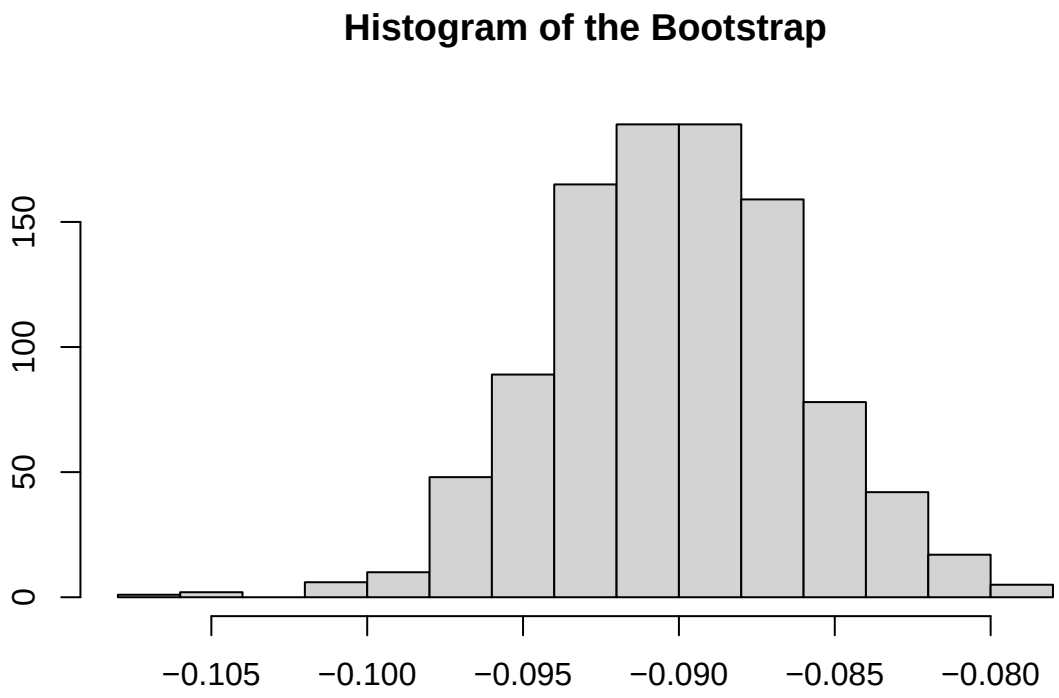
the number of miles driven (the relationship between the price of the car and the number of miles driven is statistically significant).

Part 1.5 (8 points): We can also use the bootstrap to create confidence intervals for the slope of the regression coefficient. To do this you can use the following procedure:

1. Create a bootstrap resampled data frame by sampling with replacement from the `used_corollas` data frame. You can do this using dplyr's `sample_n()` function with the sample size being the number of cases in the `used_corollas` data frame and setting the `replace = TRUE` argument.
2. Fit the regression model using the bootstrap data frame.
3. Extract the regression slope coefficient and save it to a vector object.
4. Repeat this process 1,000 times (this is less than what we normally use because it is computationally expensive to run this bootstrap procedure).
5. Plot the bootstrap distribution and use the percentile method to report a 95% confidence interval for the regression slope.

Report whether the bootstrap confidence interval is similar to the confidence interval using the t-distribution you calculated above.

```
n_cases<-nrow(used_corollas)
boot_fit_coeff<-NULL
for(i in 1:1000){

  boot_sample<-sample_n(used_corollas, size = n_cases, replace = TRUE)

  boot_fit<-lm(price_bought~mileage_bought, data = boot_sample)
  boot_fit_coeff[i]<-boot_fit$coefficients[2]


}

hist(boot_fit_coeff, xlab = "", ylab = "", main = "Histogram of the Bootstrap")
```




```
(quantile(boot_fit_coeff, c(.025, .975)))
```

```
##       2.5%      97.5%
## -0.09781155 -0.08239486
```

Answers:

According to our confidence interval in 1.4, we are 95 percent confident that

$$\beta_1$$

is contained between -0.09912747 and -0.08214508. The bootstrap confidence interval of -0.0978 to -0.082 is similar to the confidence interval of the t-distribution.

Part 1.6 (8 points): My Toyota had 180,000 miles at the time I wanted to sell it. Based on the regression model fit above, what is the predicted worth of this car? Does this seem like a reasonable estimate?

```
my_car<-data.frame(mileage_bought=180000)
(predicted_value<-predict.lm(lm_fit, my_car))
```

```
##        1
## 448.3911
```

Answer

The predicted worth of the car is $448.3911.This seems unreasonable from our actual data since 180,000 miles is a value outside of the 150,000 mile limit we created when we built this model. Therefore, we cannot say if it is a reasonable estimate because our model is not equipped to extrapolate from the data.

Part 2: Regression diagnostics

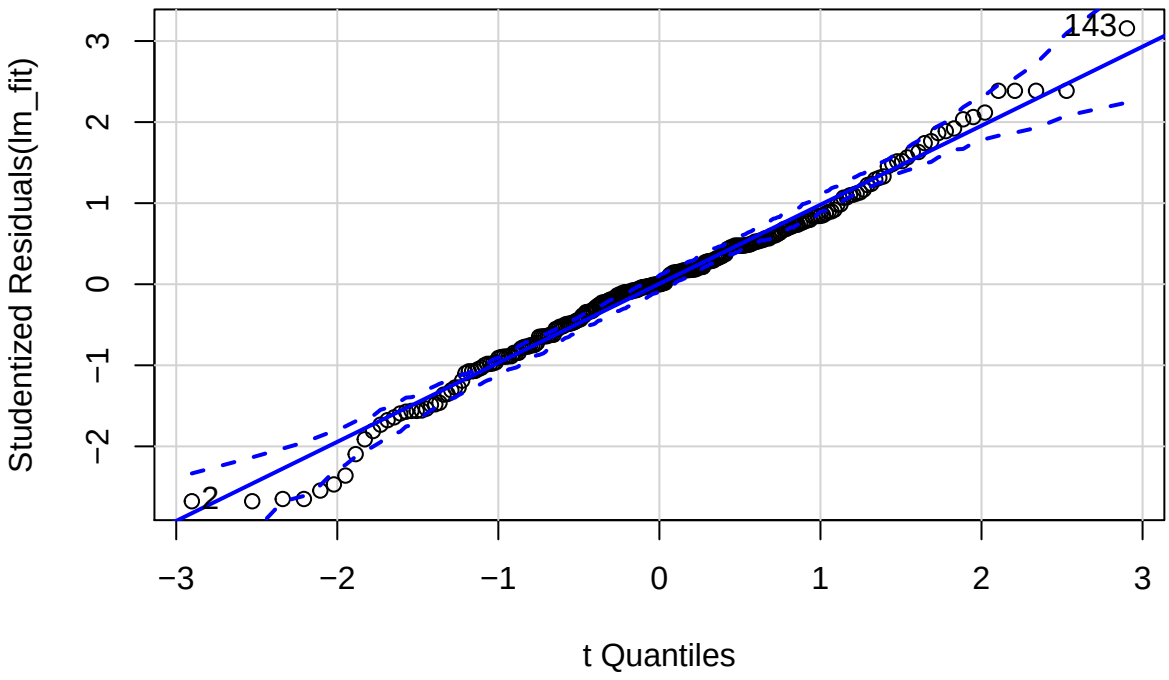
When making inferences about regression coefficients using most parametric methods, there are a number of assumptions that need to be met to make the mathematical derivations of tests/confidence intervals methods valid. The assumptions are:

1) **Normality**: residuals are normally distributed around the predicted value $\hat{y}$
2) **Homoscedasticity**: constant variance over the whole range of x values
3) **Linearity**: A line can describe the relationship between x and y
4) **Independence**: each data point is independent from the other points

We can check whether these assumptions are met by creating a set of diagnostic plots.

Part 2.1 (4 points): To check whether the residuals are normally distributed we can use create a Q-Q plot. The `car` package has a nice function to create these plots called `qqPlot()` to create these plots. If we pass the `lm_fit` object to the `qqPlot()` function it will create a Q-Q plot of the studentized residuals. Create this plot and report if the residuals seem normally distributed.

```
qqPlot(lm_fit)
```



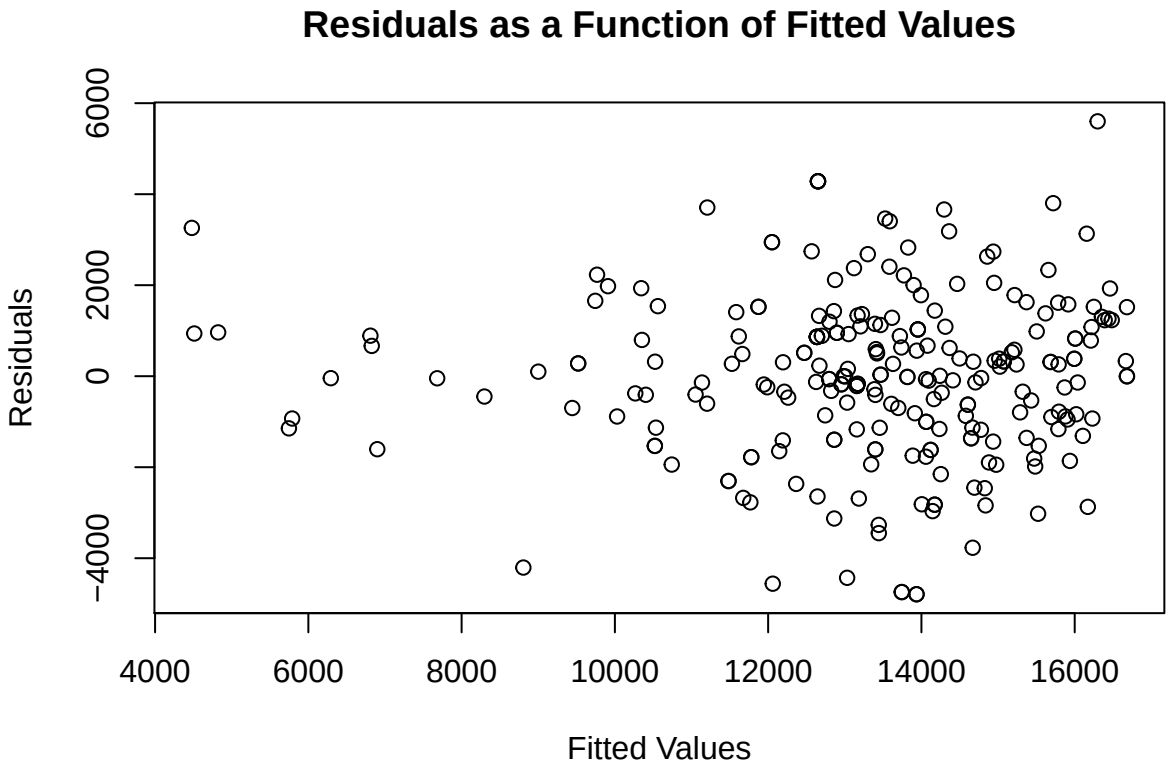

```
##    2 143
##    2 140
```

Answer:

Mostly all of the values fall on the diagonal so the residuals seem normally distributed.

Part 2.2 (5 points): To check for homoscedasticity and linearity, we can create a plot of the residuals as a function of the fitted values. Create such a plot below using information in the `lm_fit` object. Does it appear that homoscedasticity and linearity are met here? Are these results what you would expect from looking at plots above and from the nature of the type of data you are analyzing?

```
plot(lm_fit$fitted.values, lm_fit$residuals,
     xlab = "Fitted Values",
     ylab = "Residuals",
     main = "Residuals as a Function of Fitted Values")
```



Answers:

The requirement for linearity is met since there is no non linear pattern within our data.We know this because when looking a the plot, there does not seem to be any pattern or trends and the residuals seem to be randomnly distributed above and below the y axis. While the residuals are more spread out for larger values, there is not too much heteroscedasticity so the requirement for homoscedasticity is met. The results are expected given that we are working with data that seems approximately linear (it makes sense that an increased mileage would be inversely related to price).

Part 2.3 (5 points): To check if the data points are independent requires knowledge of how the data was collected. For example, if the data you have is from a time-series (e.g., recordings of the temperature in New Haven on consecutive days) then there is a high likelihood that the data points might not be independent. On the other hand, if you take a simple random sample from a population where every point is equally likely to be selected, then the data is going to be independent.

Unfortunately I do not know exactly how this data was collected so it is difficult to say if the data is independent here. However, there might be ways to investigate whether it seems plausible that it could be independent. Please describe some ways you might investigate whether the data could be independent (hint: think about the variables in the full `car_transactions` data set) Note: there is no exact 'right answer' here, just describe some possible ideas.

Answer:

To check if the data is independent, we could check variables like state_bought and city_bought and

date_sold to to check the geographical location and dates of the data to see if there are any clusters of data according to whether sales were concentrated around certain states or cities or particular dates.

Part 3: The effect of high leverage points on regression models

In the above example we fit the regression model using all used Toyotas that had less than 150,000 miles. Let's now examine the affects of high leverage points by including all used Toyotas regardless of how many miles that have been driven.

Part 3.1 (5 points):

Let's start again with the `car_transactions` data frame, and again use the dplyr `select()` and `filter()` functions to create a reduced data frame object called `used_corollas_all`. However this time **do not do any filtering related to the number of miles a car has been driven**; i.e., also keep in the data frame cars that have been driven more than 150,000 miles.

In particular, follow the steps below:

1. The only variables that should be in the `used_corollas_all` data frame are:

a) model_bought: the model of the car
b) new_or_used_bought: whether a car was new or used when it was purchased
c) price_bought: the price the car was purchased for
d) mileage_bought: the number of miles the car had when it was purchased

2. The only cases that should be in the `used_corollas_all` data frame are:

a) used cars
b) Toyota Corollas

3. Use the `na.omit()` function on the `used_corollas_all` data frame to remove cases that have missing values.

If you have properly filter the data, the `used_corollas_all` should have additional case in your data frame, or 249 cases in total. Please check this before going on to the next set of exercises.

```
# use dplyr to reduce the data set to only used Corolla's for all miles driven


used_corollas_all<- filter(car_transactions, model_bought == "Corolla",new_or_used_bought == "U" )%>%
  select(model_bought, new_or_used_bought, price_bought, mileage_bought)%>%
    na.omit()



# check the size of the resulting data frame

(n_cases2<-nrow(used_corollas_all))
```

```
## [1] 249
```

```
(n_variables2<-ncol(used_corollas_all))
```

```
## [1] 4
```

Part 3.2 (8 points):

Now fit a linear regression model to the `used_corollas_all` that shows the predicted (expected) price as a function of the number of miles driven. Save this model to an object called `lm_fit_all`, print the regression coefficients, and describe how much the cost of a car decreases for every mile driven in this model.

Also, create a scatter plot of the price as a function of the number of miles driven, and add a red line to our plot showing the regression line for the model with all used Corollas. Then add to the plot the regression a green line based on the model you fit in part 1 when only cars with less than 150,000 miles were used (i.e., when the one high leverage point was not included).

Finally, make a prediction for the price of a car that has been driven 150,000 miles using both the `lm_fit_all` based on the model using all the data, and the `lm_fit` based on using only cars with less than 150,000 miles. Do these models seem similar to you?

```
# fit a regression model

(lm_fit_all<-lm(price_bought ~ mileage_bought, data = used_corollas_all))
```

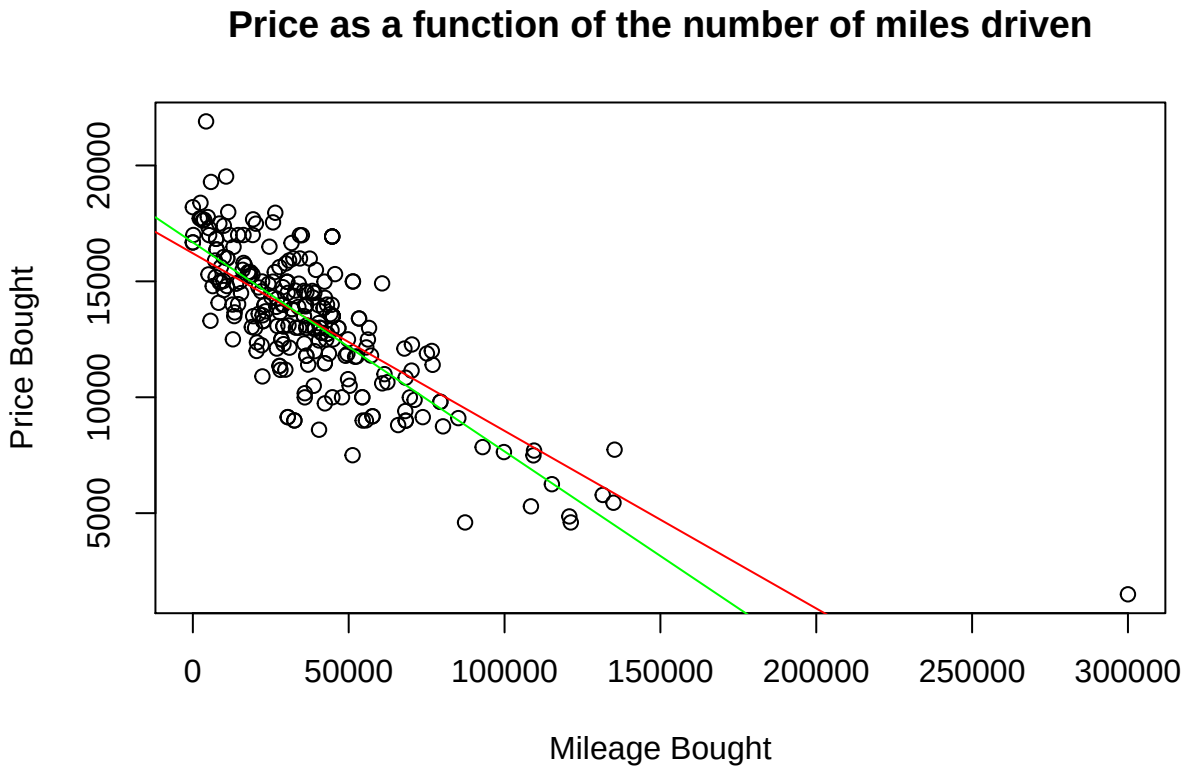
```
##
## Call:
## lm(formula = price_bought ~ mileage_bought, data = used_corollas_all)
##
## Coefficients:
##    (Intercept)  mileage_bought
##    16210.25517        -0.07661
```

```
# display the regression coefficients
coef(lm_fit_all)
```

```
##    (Intercept) mileage_bought
## 16210.25517005    -0.07660694
```

```
# let's start by plotting the data

plot(used_corollas_all$mileage_bought,used_corollas_all$price_bought,
     xlab = "Mileage Bought",
     ylab = "Price Bought",
     main = "Price as a function of the number of miles driven")




# add the regression line to the plot
```

```r
abline(lm_fit_all, col = "red")


# add a green line for the model that excludes cars with over 150k miles
abline(lm_fit, col = "green")
```




```r
# make a prediction for a car driven 150k miles using both models



car_150<-data.frame(mileage_bought = 150000)
print("lm_fit_all prediction:")
```

```
## [1] "lm_fit_all prediction:"
```

```r
predict.lm(lm_fit_all, car_150)
```

```
##        1
## 4719.215
```

```r
print("lm_fit prediction:")
```

```
## [1] "lm_fit prediction:"
```

```
predict.lm(lm_fit, car_150)
```

```
##        1
## 3153.979
```

Answers: The linear regression model lm_fit_all which included all of the data predicts that a car bought at 150,000 mileage would have a price of 4719.215 while the lm_fit model that is based on cars with less than 150,000 miles predicts that a car bought at 150,000 mileage would have an expected price of 3153.979. Based on intuition from the scatterplot, the two models seem similar in terms of their slope; however, the large difference of over 1,000 dollars between predicted values for a mileage of 150,000 (4719.215-313.979 = $1565.236) indicates that the models are not very similar.

Part 3.3 (5 points): Now look again at the 95% confidence interval for the value of the regression slope β_1 that you created in part 1.4 that was based on using only the cars that have fewer than 150,000 miles. If we were assuming that this confidence interval is reasonable, would the value for the estimated regression slope found in using all the data (i.e., the $\hat{\beta}_1$ using data in `used_corollas_all` that includes the car with 300,000 miles) be a plausible value for what the true parameter value β_1 that was found by using the `used_corollas` data set?

Answer

No, because using the coefficients from 3.2 , we find that our estimated linear regression slope using data in used_corollas_all is -.0766, which is not contained within our 95 percent confidence interval from -0.099127247 and -0.08214508.

Part 3.4 (8 points): Now sort the data frame `used_corollas_all` so that the rows are in the order from smallest number of miles driven to the most number of miles driven, and store it again the same object called `used_corollas_all`. Refit the `lm_fit_all` using this sorted data frame (as a sanity check, the coefficients found should be the same as before). Then, recreate the scatter plot based on this sorted `used_corollas_all` data and add to this plot both the 95% confidence intervals for **the regression line** in green, and the 95% prediction interval in blue (again using this sorted `used_corollas_all`).

```
# arrange the data and refit the model
used_corollas<-
  used_corollas_all%>%
  arrange((mileage_bought))
lm_fit_all<-lm(price_bought~mileage_bought,
               data = used_corollas_all)
coef(lm_fit_all)
```

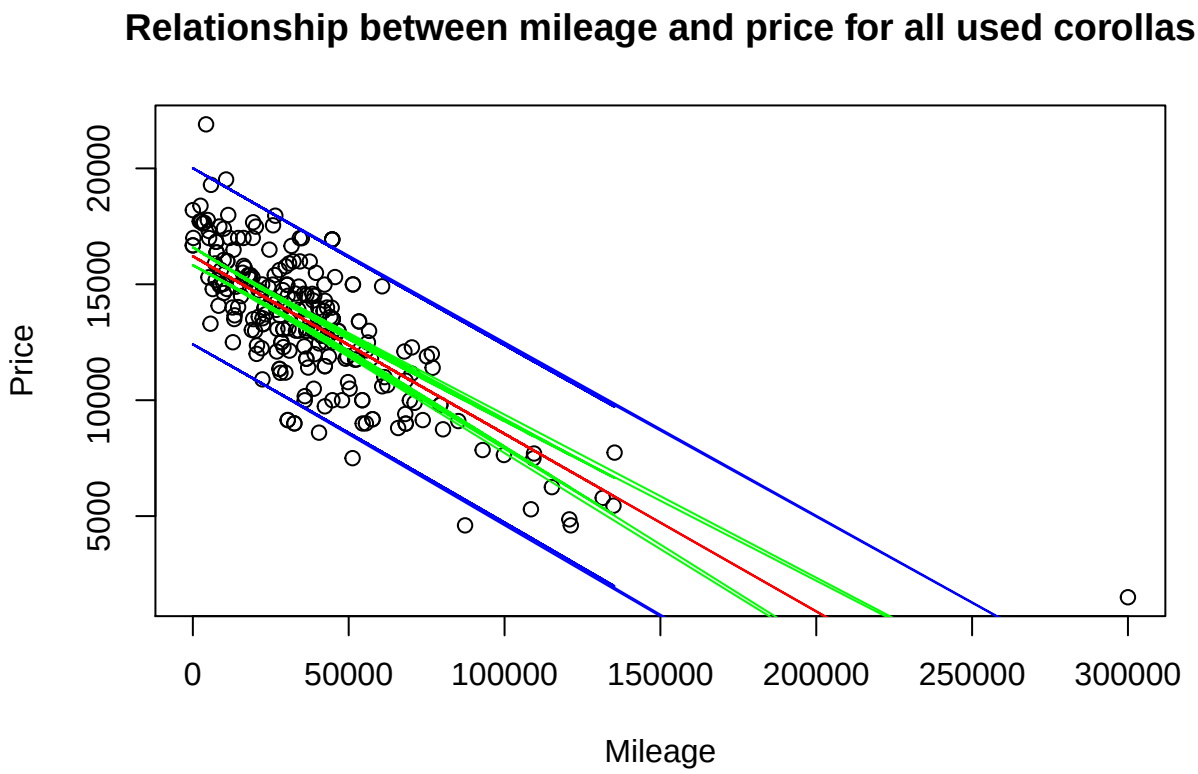
```
##    (Intercept) mileage_bought
## 16210.25517005    -0.07660694
```

```
# confidence intervals for the betas

CI_betas<-confint(lm_fit_all)

# confidence interval for the regression line mu_y
```

```r
CI_regression_line<-predict(lm_fit_all, interval = "confidence", level = 0.95)

# prediction interval for the regression line


prediction_interval<-predict(lm_fit_all, interval = "predict", level = 0.95)
```

```
## Warning in predict.lm(lm_fit_all, interval = "predict", level = 0.95): predictions on current data re
```

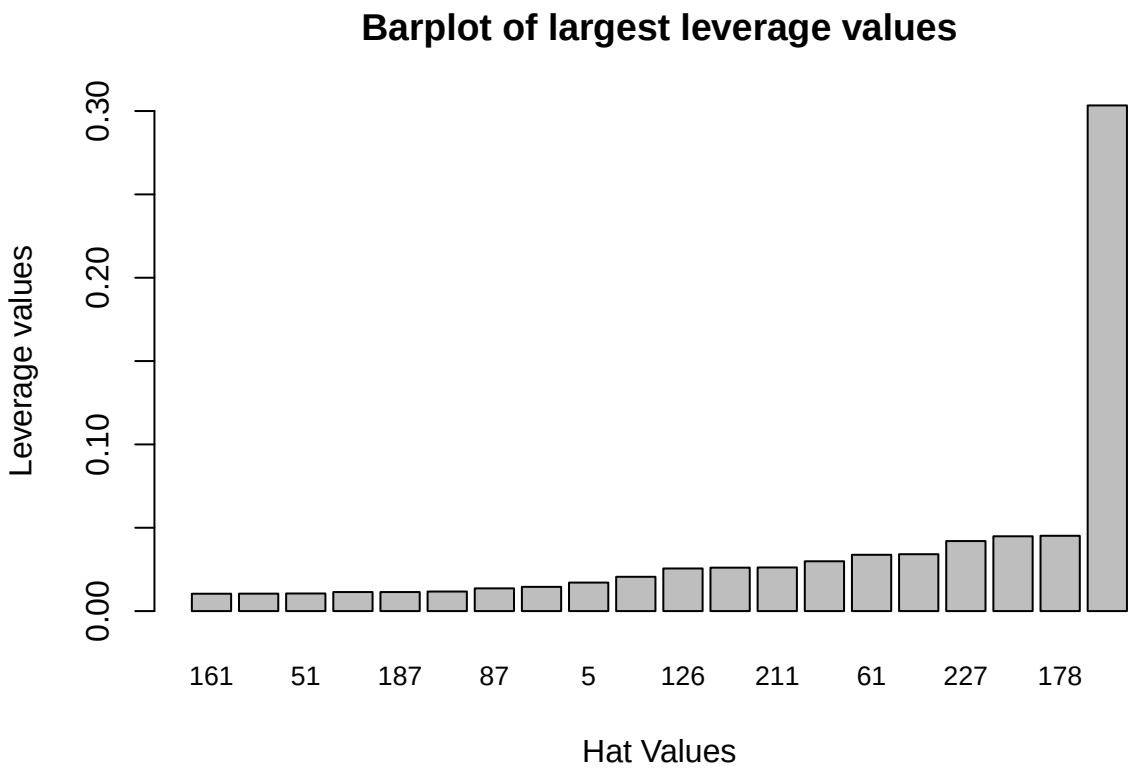
```r
# plot both confidence interval and the prediction interval

plot(used_corollas_all$mileage_bought,
     used_corollas_all$price_bought,
     xlab = "Mileage",
     ylab = "Price",
     main = "Relationship between mileage and price for all used corollas")



# plot confidence interval
points(used_corollas_all$mileage_bought, CI_regression_line[,1], col = "red", type = "l")
points(used_corollas_all$mileage_bought, CI_regression_line[,2], col = "green", type = "l")
points(used_corollas_all$mileage_bought, CI_regression_line[,3], col = "green", type = "l")
# plot prediction interval

points(used_corollas_all$mileage_bought, prediction_interval[,1], col = "red", type = "l")
points(used_corollas_all$mileage_bought, prediction_interval[,2], col = "blue", type = "l")
points(used_corollas_all$mileage_bought, prediction_interval[,3], col = "blue", type = "l")
```



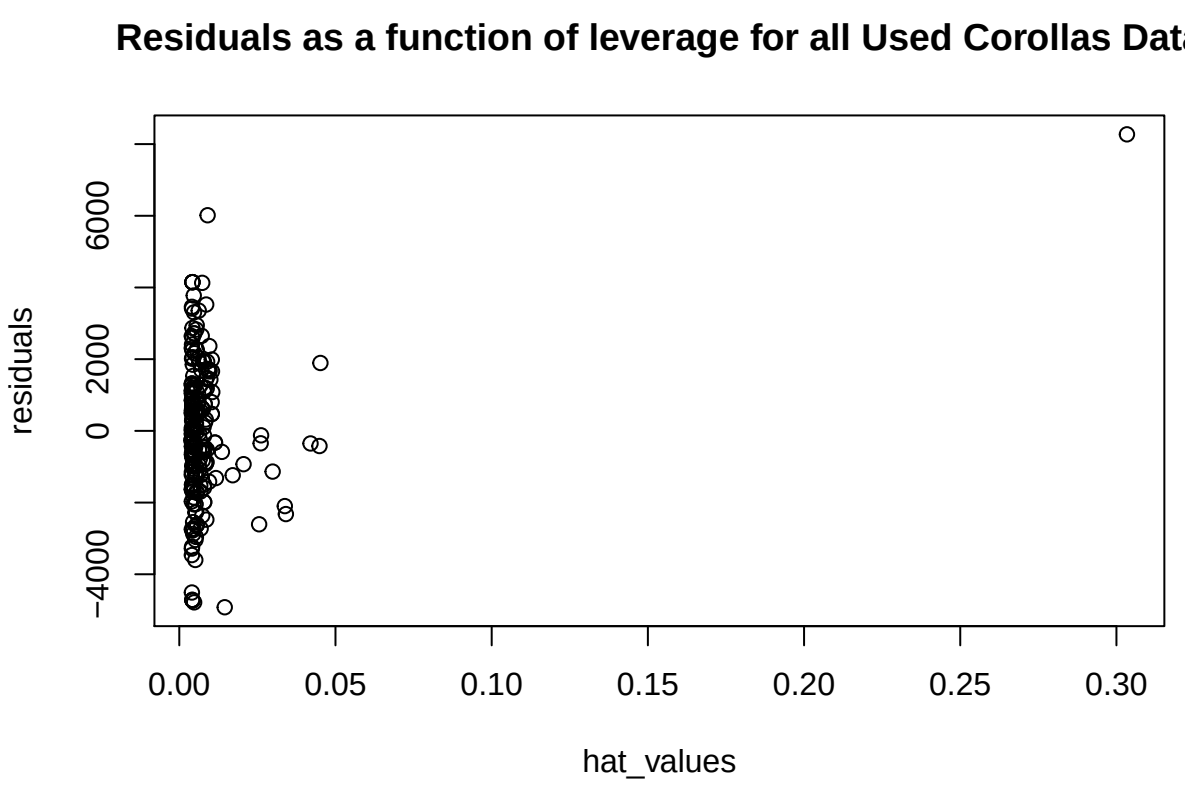
Part 3.5 (10 points): Let's analyze the leverage and Cook's distance for the data points in the `used_corollas_all`. Calculate the leverage for the data points in this model (i.e., the hat values) and plot the 20 largest leverage values found using a bar plot. Also plot the residuals as a function of the leverage for each point, and use R's built in plot functions to plot Cook's distance for each data point and the standardized residuals as a function of the leverage for each point. Based on the 'rules of thumb' we discussed in class, **how many points** are considered 'very unusual' for the different measures of:

a) Cook's distance
b) standardized residuals
c) studentized residuals
d) leverage

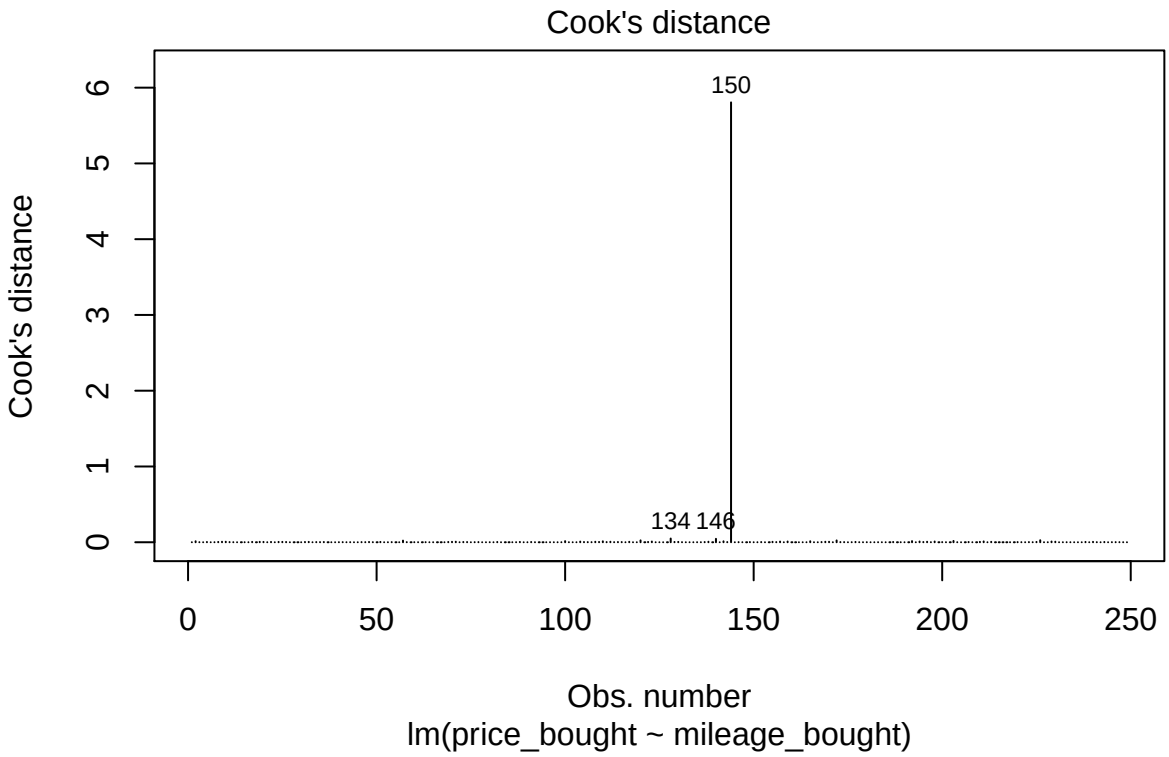
```
hat_values<-hatvalues(lm_fit_all)
top_20<-tail(sort(hat_values),20)
barplot(top_20,
        cex.names = 0.8,
        ylab = "Leverage values",
        xlab = "Hat Values",
        main ="Barplot of largest leverage values")
```



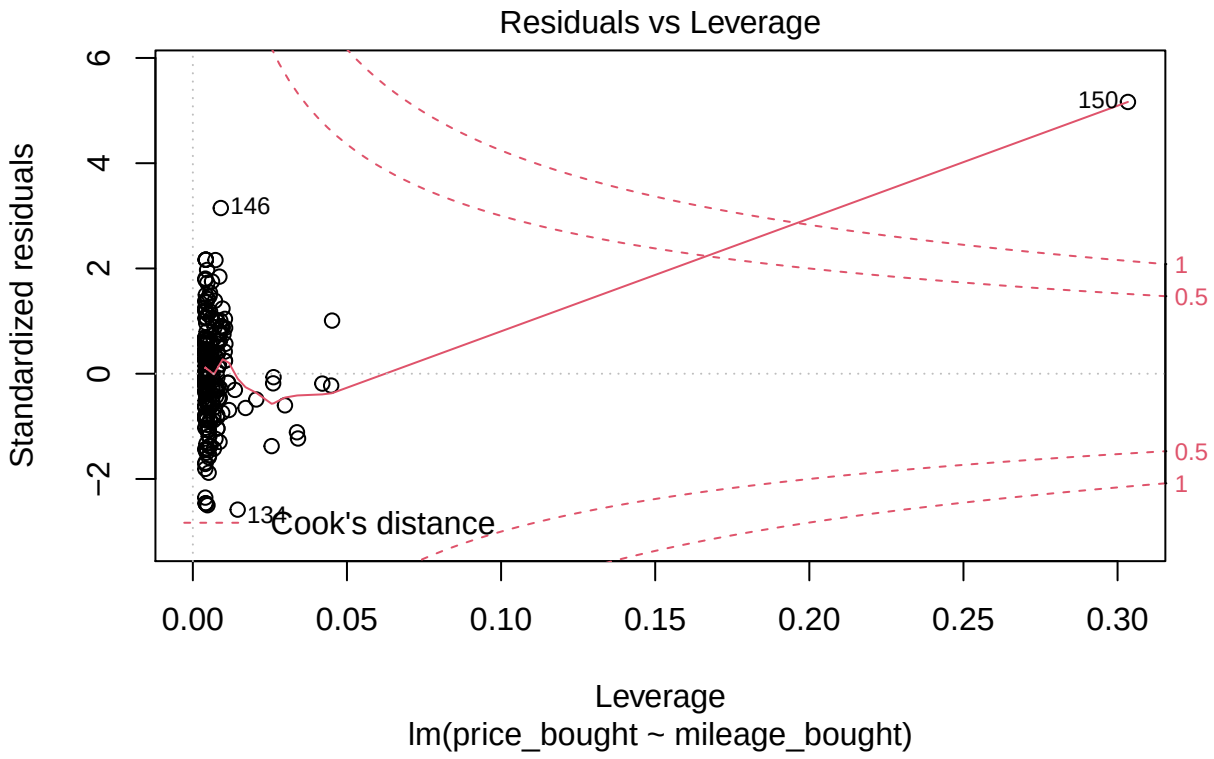

```
plot(hat_values, lm_fit_all$residuals,  ylab = "residuals",
     main = "Residuals as a function of leverage for all Used Corollas Data")
```




```
plot(lm_fit_all, 4)
```




```
plot(lm_fit_all, 5)
```




```
#cookes distance
cooks_d<-cooks.distance(lm_fit_all)
sum(cooks_d>1)
```

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

```
#leverage
sum((rstandard(lm_fit_all) > 3)| (rstudent(lm_fit_all)< -3))
```

```
## [1] 2
```

```
sum((rstudent(lm_fit_all)> 3)|(rstudent(lm_fit_all)< -3))
```

```
## [1] 2
```

```
n<-dim(used_corollas_all)[1]
sum(hat_values>=6/n)
```

```
## [1] 10
```

Answer Based on the rule's of thumb discussed in class and the `lm_fit_all model`, the number of 'highly unusual' points are:

a) 1
b) 2
c) 2
d) 10

Part 3.6 (5 points): Above you fit two models: `lm_fit_all` which contained all the used Corollas and `lm_fit` which did not contain the high leverage car with 300,000 miles driven. Describe below which you think is best and why? Also describe any limitation to these models.

Answer

I believe that the "lm_fit", the model that doesn't contain the high leverage car point is a better model than "lm_fit_all" which contains all of the used Corollas. This "lm_fit" model is better because it captures the majority of the data set while excluding the high leverage point representing a car driven for 300,000 miles that could have affected the model. The "lm_fit" model enables us to make more accurate for the cars that are within the normal miles range.

The lm_fit_all model is limited because it extrapolates too far and the outlier changes the prediction model. One limitation of the model is that a linear relationship is assumed between mileage and the price of the car. A limitation of the lm_fit model is that since predictions cannot be made outside of the data range, which means that predictions for very large values beyond 150,000 miles might not be accurate.

Reflection (3 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 7.