Homework 8

The purpose of this homework is to examine the analysis of variance for regression and to practice building multiple 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 8th.

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: Analysis of variance (ANOVA) for regression

In this homework we will continue to use the Edmunds.com car data to explore the ANOVA for regression and to practice building multiple linear regression models. As you will recall from homework 7, this data set has a large number of sales of cars from the website Edmunds.com. Also, remember that this data set has been made available to this class for educational purposes, so please do not share this data outside of the class.

Part 1.1 (3 points). To explore the ANOVA for regression, let's recreate our data frame that contains all used Toyota Corollas; i.e., recreate the `used_corollas_all` data frame, as you did on problem 1.1 of homework 7, using the following steps:

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 done this correctly you should have 249 cases in your `used_corollas` (i.e., make sure to include all used Corollas, **including** cars that have been driven over 150,000 miles).

```
load('car_transactions.rda')


# use dplyr to reduce the data set to only used Corollas

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



# check the size of the resulting data frame
dim(used_corollas_all)
```

```
## [1] 249   4
```

Part 1.2 (4 points): Now that you have created the used Corolla data frame, let's recreate our linear regression model object `lm_fit` by fitting the price as a function of the mileage bought. (I recommend that you also print the regression coefficients and check that you have fit the model correctly by looking at the answers to homework 7 part 3 so that any mistakes you make won't propagate to the rest of this problem).

Once you have fit this model, create an ANOVA table using the `anova()` function and report in the answer section the values for the model sum of squares (i.e., the SSModel) and the residual sum of squares (i.e., the SSResiduals, which we have also been calling the RSS and the SSE).

```
(lm_fit<-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
```

```
print("ANOVA TABLE")
```

```
## [1] "ANOVA TABLE"
```

```
coef(lm_fit)
```

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

```
anova(lm_fit)
```

```
## Analysis of Variance Table
##
## Response: price_bought
##                  Df     Sum Sq    Mean Sq F value                Pr(>F)
## mileage_bought    1 1343204901 1343204901  364.68 < 0.00000000000000022 ***
## Residuals       247  909748630    3683193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answers: The SSModel is 1343204901. The F value is 364.68. The SSResidual is 909748630.

Part 1.3 (10 points): We can also extract the SSTotal, SSModel and SSError from the original data and from values stored in the `lm_fit` object. Please use the following equations to calculate the SSTotal, SSModel and SSError values directly, and report these values:

1. Use the `used_corolla_all` data frame to calculate the SSTotal using the formula

$$\sum_{i=1}^{n}(y_i - \bar{y})^2$$

2. Use the `fitted.values` in the `lm_fit` object to calculate SSModel using the formula

$$\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$$

3. Use the `residuals` in the `lm_fit` object to calculate SSResidual using the formula

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

.

To check you have the right answers, look at the values you got in question 1.2 above. Also show that SSTotal = SSModel + SSResidual for the values you calculated.

```
# the total sum of squares
(SSTotal = sum((used_corollas_all$price_bought - mean(used_corollas_all$price_bought))^2))
```

```
## [1] 2252953531
```

```
# the model sum of squares
(SSModel = sum((lm_fit$fitted.values - mean(used_corollas_all$price_bought))^2))
```

```
## [1] 1343204901
```

```r
# the sum of squared error

(SSResidual = sum(lm_fit$residuals^2))
```

```
## [1] 909748630
```

```r
round(SSTotal) == round(SSModel + SSResidual)
```

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

Part 1.4 (5 points): As we discussed in class, the *coefficient of determination*, r^2, is equal to the percentage of the variance explained by the linear model, i.e.,

$$r^2 = SSModel/SSTotal$$

. For simple linear regression, r^2 is equal to the correlation coefficient squared (which it is why it is denoted r^2).

Please calculate the correlation coefficient between `mileage_bought` and `price_bought`, and square it. Then using the values calculated in part 1.3, show that this value is equal to SSModel/SSTotal, and that it is also equal to 1 - SSResidual/SSTotal. As always, be sure to show your work by printing these values to get credit for this problem.

```r
(CoefficientOD  = (cor(used_corollas_all$mileage_bought, used_corollas_all$price_bought))^2)
```

```
## [1] 0.5961973
```

```r
(CoefficientOD2 = SSModel / SSTotal)
```

```
## [1] 0.5961973
```

```r
(CoefficientOD3 = 1  - (SSResidual / SSTotal))
```

```
## [1] 0.5961973
```

```r
round(CoefficientOD) ==  round(CoefficientOD2)
```

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

```r
round(CoefficientOD2)  == round(CoefficientOD3)
```

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

Part 1.5 (4 points): Let's look at the relationship between the ANOVA F-statistic and the t-statistic. Use the `summary()` function on the `lm_fit` model to get the t-statistic (as you had previously done in homework 7, part 1.3). Show that the value of `t_stat` squared is (approximately) equal to the F-statistic found with the `anova()` function in part 1.2 above, by printing the t^2 value and reporting the t^2 value and the F-statistic value in the answer section below.

```r
summary(lm_fit)
```

```
##
## Call:
## lm(formula = price_bought ~ mileage_bought, data = used_corollas_all)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4919.9 -1135.7    11.7  1089.0  8271.7
##
## Coefficients:
##                   Estimate   Std. Error t value            Pr(>|t|)
## (Intercept)    16210.255170   195.782251    82.8 <0.0000000000000002 ***
## mileage_bought    -0.076607     0.004012   -19.1 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1919 on 247 degrees of freedom
## Multiple R-squared:  0.5962, Adjusted R-squared:  0.5946
## F-statistic: 364.7 on 1 and 247 DF,  p-value: < 0.00000000000000022
```

```r
(F_stat_anova <- 364.68)
```

```
## [1] 364.68
```

```r
t_stat<--19.1
(F_stat_manual<-(t_stat)^2)
```

```
## [1] 364.81
```

```r
round(F_stat_anova) == round(F_stat_manual)
```

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

Answers:

$$r^2 = 364.81$$

$$F - statistic = 364.68$$

F-stat is equal to 364.68 while t-stat squared is equal to 364.81.

Part 2: Multiple linear regression

Let's continue to use the Edmunds.com car data to explore multiple linear regression where we try to predict a response variable y, based on several explanatory variables x_1, x_2, ..., x_p.

Part 2.1 (6 points): Let's start by using dplyr to derive a new data set from the original Edmunds `car_transactions` data set. Please create this data set in an object called `car_transactions2` that has the following properties:

1. It contains a new variable called `years_old` which is the difference between the year the car was sold, and the model year of the car.
2. It only contains used cars.
3. It only contains the variables: `price_bought`, `mileage_bought`, `years_old`, and `msrp_bought`.

If you have created this data frame correctly it should have 17,134 cases.

Also, report what is the maximum and minimum value for the variable `years_old`. Does it make sense that the minimum value of `years_old` is negative? Please explain why. Finally, report the price that the least and most expensive used cars sold for.

```
car_transactions2<-car_transactions%>%
  filter(new_or_used_bought == "U")%>%
  mutate(years_old = year_sold - model_year_bought)%>%
  select(price_bought, mileage_bought, years_old, msrp_bought)
#min and max of years_old
min(car_transactions2$years_old)
```

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

```
max(car_transactions2$years_old)
```

```
## [1] 34
```

```
#min and max price
min(car_transactions2$price_bought, na.rm = TRUE)
```

```
## [1] 0
```

```
max(car_transactions2$price_bought, na.rm = TRUE)
```

```
## [1] 220000
```

```
dim(car_transactions2)
```

```
## [1] 17134     4
```

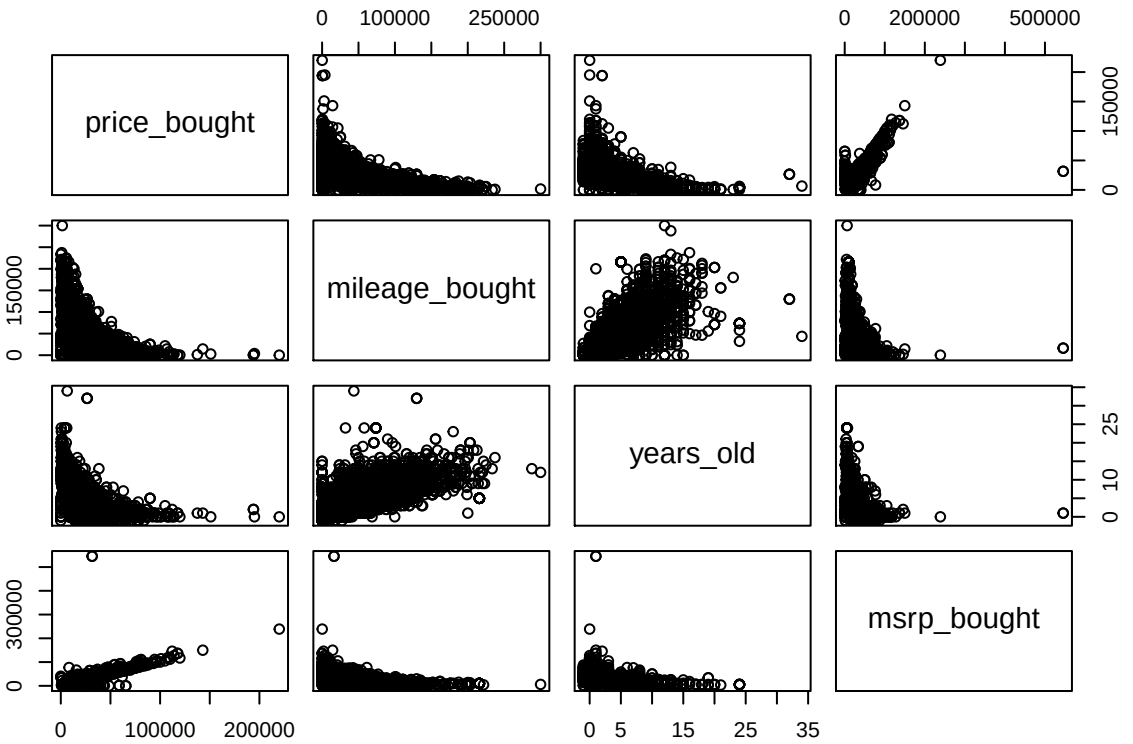
Answer:

The minimum value for years_old is -1 and the maximum value of years_old is 34. It makes sense that the minimum value is negative since for the smallest years_old value, this must mean that the year that you bought the model is a smaller number than the model year. Some companies release their car models ahead of the year that the car model corresponds to, which might cause a negative value in years_old. For instance, a customer might be able to buy a 2021 car model in 2020, making the year_sold - model_year_bought value negative.

The least expensive car was 0 dollars and the most expensive car was 220000.

Part 2.2 (5 points): Now use the `pairs()` function to visualize the correlation between all pairs of variables in the `car_transactions2` data frame. Report whether any variable looks like it has a particularly strong linear relationship with `price_bought` and whether it makes sense that there would be a strong relationship between these variables.

```
pairs(select(car_transactions2, price_bought, mileage_bought, years_old, msrp_bought))
```



Answer:

Price_bought and msrp_bought have a strong positive linear trend. It makes sense that this relationship would exist between price_bought and msrp_bought since it is logical that as the manufacturer's suggested retail price (indicative of the value of the car) goes up, the price that the car was actually sold for would go up with the msrp_bought value.

Part 2.3 (10 points): Next fit a multiple linear regression model predicting the price a car was bought for using the three variables `mileage_bought`, `years_old`, `msrp_bought` and save the linear fit to an object called `lm_cars`. Then use the `summary()` function to get information about the the linear regression model you fit by: a) saving the output of the `summary()` function to an object called `summary_lm_cars`, and b) print the output so you can see the result.

Report below the following information:

1. Are all the regression coefficients statistically significant at the $\alpha = 0.05$ level?
2. Do the signs for the regression coefficients make sense? Explain why.
3. Report what percentage of the total sum of squares is explained by this model by looking at the values stored in the `summary_lm_cars` object.

```
lm_cars<-lm(price_bought~mileage_bought +years_old + msrp_bought, data = car_transactions2)
(summary_lm_cars<-summary(lm_cars))
```

```
##
## Call:
## lm(formula = price_bought ~ mileage_bought + years_old + msrp_bought,
##     data = car_transactions2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -310279   -2934    -571    2238   63537
##
## Coefficients:
##                    Estimate   Std. Error t value            Pr(>|t|)
## (Intercept)    11155.428797   202.919846   54.98 <0.0000000000000002 ***
## mileage_bought    -0.046924     0.003563  -13.17 <0.0000000000000002 ***
## years_old       -447.222407    40.271434  -11.11 <0.0000000000000002 ***
## msrp_bought        0.608395     0.004918  123.71 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6759 on 8731 degrees of freedom
##   (8399 observations deleted due to missingness)
## Multiple R-squared:  0.7309, Adjusted R-squared:  0.7308
## F-statistic:  7904 on 3 and 8731 DF,  p-value: < 0.00000000000000022
```

Answers:

1. Yes, the results are statistically significant because all of the p values on the summary table for lm_cars are all less than 0.05.
2. It makes sense that the regression coefficient for msrp_bought is positive (0.608395) since as msrp increases, price increases (as explained in question 2.2)
3. The percentage of the total sum of squares explained by this model is 73.09%.

Part 3: Fitting linear regression models with categorical predictors

When I sold my car, I was interested in also buying a new car. Toward the end of my search, the car models I was considering were the Mazda 3 and the Subaru Impreza. When buying the new car, I knew that at some point I was going to have to sell the car, so I was interested in looking at how the prices of these car models decline as they are driven more miles.

Part 3.1 (5 points): In these exercises, let's examine regression models found from modeling car prices as a function of number of miles driven for the Mazda 3's, Subaru Imprezas, and BMW 5 series (one can dream, right?). Please create an object called `three_car_data` that has this data by using the following steps:

1. Get only the makes of Mazda 3's, Subaru Imprezas, and BMW 5 series car makes (i.e., `MAZDA3`, `Impreza` and `5 Series`)
2. Get the data for only used cars.
3. Use the `droplevels()` function on this data frame to remove all levels of the categorical data we are not using (i.e., remove levels for other makes of cars).

If you've filter the data correctly the data frame should have 497 cases.

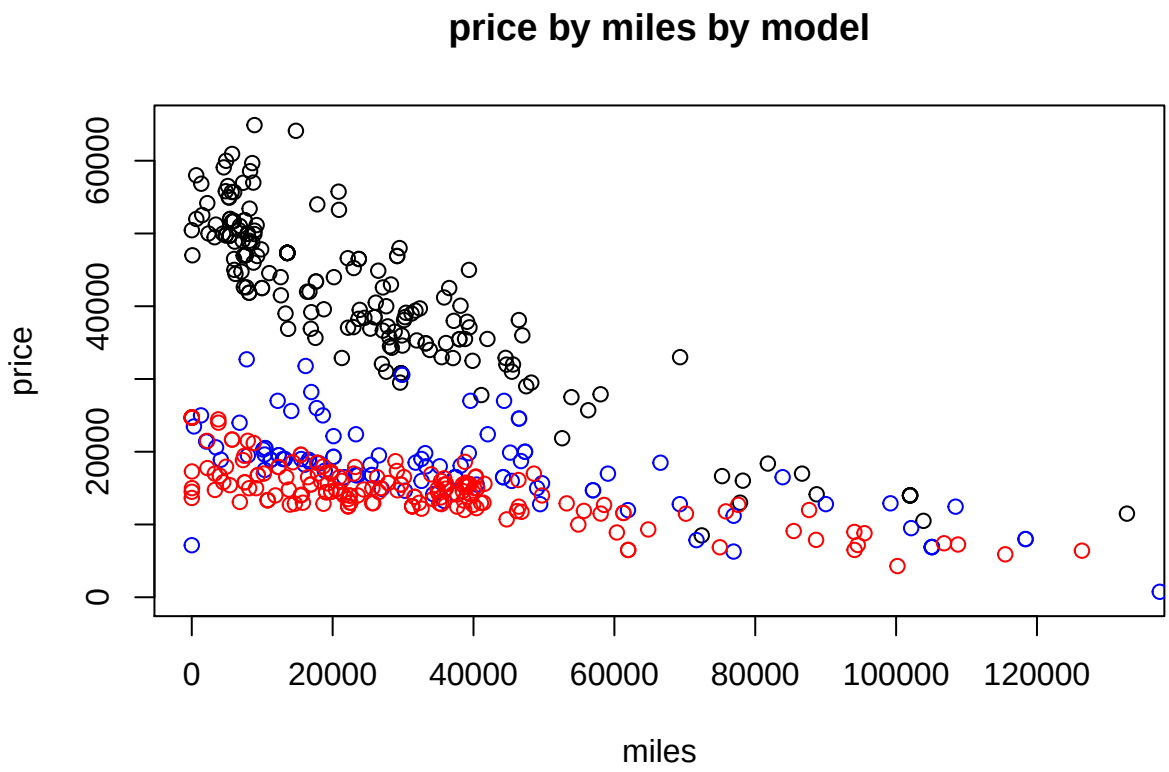
Once you have created the `three_car_data` data frame, use the `plot()` and `points()` functions to create a scatter plot of the the price of a car as a function of miles driven, where the BMW's are plotted as *black* circles, the Subaru's are plotted as *blue* circles, and the Mazda's are plotted as *red* circles. Be sure to set the `ylim` argument in the plot function to be from 0 to $65,000 so that it captures all the points in the plot.

Note: For the rest of this question the term `brand` will refer to the model/model of the different types of cars.

```
#filtered by models

three_car_data<-car_transactions%>%
  filter(model_bought == "MAZDA3" | model_bought ==  "Impreza" | model_bought == "5 Series")%>%
  filter(new_or_used_bought == "U")%>%
  droplevels()

dim(three_car_data)
```

```
## [1] 497  21
```

```
plot(price_bought~mileage_bought, data = filter(three_car_data, make_bought=="BMW"),
     col = "black",
    ylab = "price",
    xlab = "miles",
    main = "price by miles by model",
    ylim = c(0, 65000))
points(price_bought~mileage_bought,
data = filter(three_car_data, make_bought == "Subaru"), ylim = c(0,65000), col = "blue")
points(price_bought~mileage_bought,
data = filter(three_car_data, make_bought == "Mazda"), ylim = c(0,65000), col = "red")
```



Part 3.2 (12 points): Let's fit a linear model for predicting price as a function of miles driven *with a separate intercept for each brand*. Use the `summary()` function to extract information about the linear model, and then answer the following questions:

1. How much does the price of a car decrease for each additional mile driven?
2. What is the reference car brand that the other car brands are being compared to?
3. What is the difference in car prices for each of the other brands relative to the reference car brand?
4. Does there appear to be statistically significant differences between the y-intercept of reference brand and each of the other car brands?
5. How much of the total sum of squares of car prices is mileage and car brand accounting for in this model based on the R^2 statistic?

```
car_intercept_fit<-lm(price_bought~mileage_bought + make_bought, data = three_car_data)

summary(car_intercept_fit)
```

```
##
## Call:
## lm(formula = price_bought ~ mileage_bought + make_bought, data = three_car_data)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22653.3  -3635.8   -175.6   3207.9  20104.5
##
## Coefficients:
##                          Estimate    Std. Error t value            Pr(>|t|)
## (Intercept)         47307.312097    463.920519  101.97 <0.0000000000000002 ***
## mileage_bought         -0.223103      0.009718  -22.96 <0.0000000000000002 ***
## make_boughtMazda   -25472.300077    574.150783  -44.37 <0.0000000000000002 ***
## make_boughtSubaru -20849.590947    687.926849  -30.31 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5514 on 472 degrees of freedom
##   (21 observations deleted due to missingness)
## Multiple R-squared:  0.8667, Adjusted R-squared:  0.8659
## F-statistic:  1023 on 3 and 472 DF,  p-value: < 0.00000000000000022
```

```
the_coefs<- coef(car_intercept_fit)
```

Answers:

1. For each additional mile driven the price decreases by 22 cents for each additional mile driven.
2. The reference car brand is the BMW 5 series and it is being compared to the Subaru and Mazda.
3. The difference in car prices for each of the other brands relative to the BMW 5 series is: -25472.300077 for the Mazda and -20849.590947 for the Subaru.
4. Since the p-values are all much less than 0.05, we can infer that there is a statistically significant difference between the y intercept of the
5. The total sum of squares is accounting for this model by around 86.67%.

$$R^2 = 0.87$$

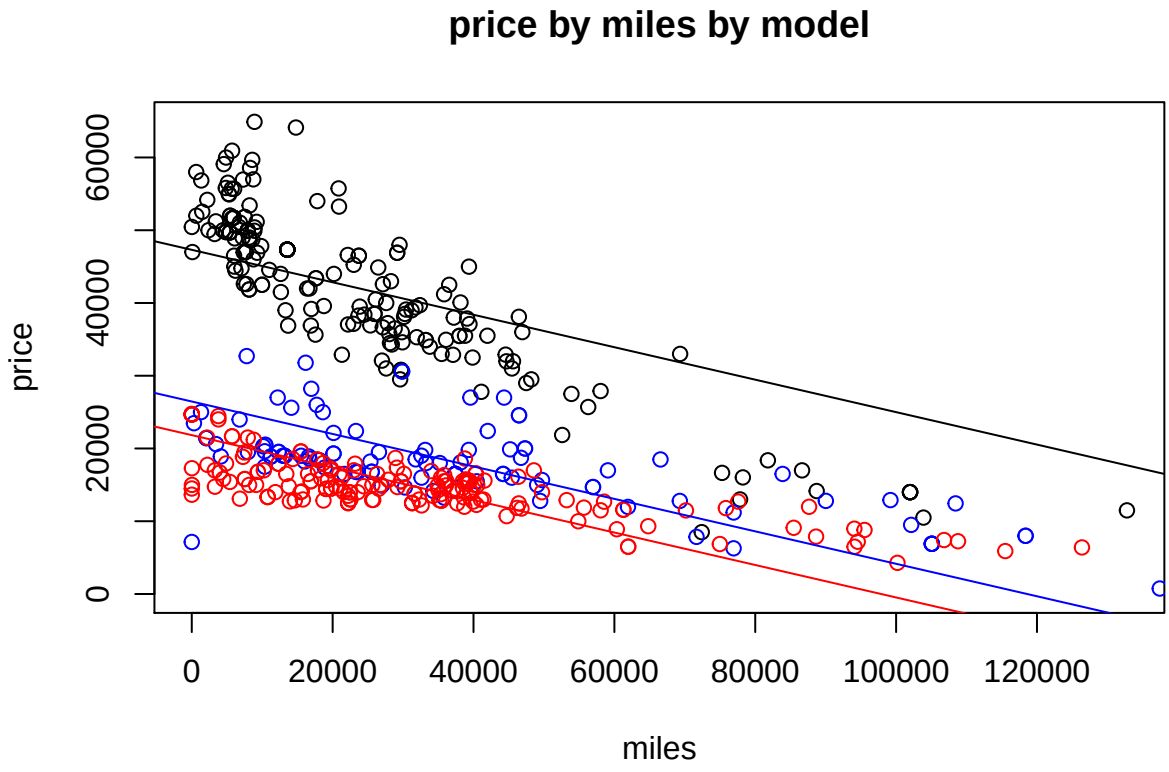
Part 3.3 (6 points): Now recreate the scatter plot you created in part 3.1 using the same colors but also add on the regression lines with different y-intercepts that you fit in part 3.2 (using the appropriate colors to match the colors of the points). Based on this visualization, does it appear the model you fit in part 3.2 is doing a good job capturing the trends in this data?

the_coefs[1] = 47307 the_coefs[2] = -0.22 the_coefs[3] = -25472.3 the_coefs[4] = -20849.59

```
t
```

```
## function (x)
## UseMethod("t")
## <bytecode: 0x7f8836e8dad8>
## <environment: namespace:base>
```

```r
plot(price_bought~mileage_bought, data = filter(three_car_data, make_bought=="BMW"),
     col = "black",
    ylab = "price",
    xlab = "miles",
    main = "price by miles by model",
    ylim = c(0, 65000))
points(price_bought~mileage_bought, data = filter(three_car_data,
               make_bought == "Subaru"), ylim = c(0,65000), col = "blue")
points(price_bought~mileage_bought, data = filter(three_car_data,
make_bought == "Mazda"), ylim = c(0,65000), col = "red")

abline(the_coefs[1], the_coefs[2])
abline(the_coefs[1] + the_coefs[3], the_coefs[2], col = "red")
abline(the_coefs[1]+ the_coefs[4], the_coefs[2], col = "blue")
```



Answer Based on the visualization, the model does an adequate job capturing the trends in the data. Still, it could do a better job if they were not constricted to one slope for all of the lines. For the 5 series BMW, the model seems to capture the data well (though the line could be steeper). For the Mazda 3' and the Subaru Imprezas, the model does not capture the data very well.

Part 3.4 (5 points): Please now fit a linear regression model for car price as a function of miles driven, but use separate y-intercepts **and slopes** for each of the 3 car brands. Once you have fit this model, please answer the following questions:

1. How much of the total sum of squares of car prices is this model capturing?
2. Based on this model, if a BMW 5 Series and Mazda 3 both had been driven 50,000 miles, what would be the difference in the car prices that the model predicts?

```
car_interaction_fit<-lm(price_bought~mileage_bought* make_bought, data =  three_car_data)

summary(car_interaction_fit)
```

```
##
## Call:
## lm(formula = price_bought ~ mileage_bought * make_bought, data = three_car_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -15670  -2416   -376   1798  18192
##
## Coefficients:
##                                    Estimate   Std. Error t value
## (Intercept)                       52067.23393    427.22099  121.87
## mileage_bought                       -0.41567      0.01240  -33.53
## make_boughtMazda                  -33605.59528    651.01035  -51.62
## make_boughtSubaru                 -29246.89438    782.73157  -37.37
## mileage_bought:make_boughtMazda       0.30011      0.01741   17.24
## mileage_bought:make_boughtSubaru      0.28728      0.01821   15.77
##                                           Pr(>|t|)
## (Intercept)                       <0.0000000000000002 ***
## mileage_bought                    <0.0000000000000002 ***
## make_boughtMazda                  <0.0000000000000002 ***
## make_boughtSubaru                 <0.0000000000000002 ***
## mileage_bought:make_boughtMazda   <0.0000000000000002 ***
## mileage_bought:make_boughtSubaru <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4135 on 470 degrees of freedom
##   (21 observations deleted due to missingness)
## Multiple R-squared:  0.9254, Adjusted R-squared:  0.9246
## F-statistic:  1166 on 5 and 470 DF,  p-value: < 0.00000000000000022
```

```
(difference<-predict(car_interaction_fit,
  data.frame(mileage_bought = 50000,
  make_bought = "Mazda")) -
predict(car_interaction_fit,data.frame(mileage_bought = 50000,
 make_bought = "BMW")))
```

```
##         1
## -18600.17
```

Answers

1. This model is capturing 92.54% of the total sum of squares.

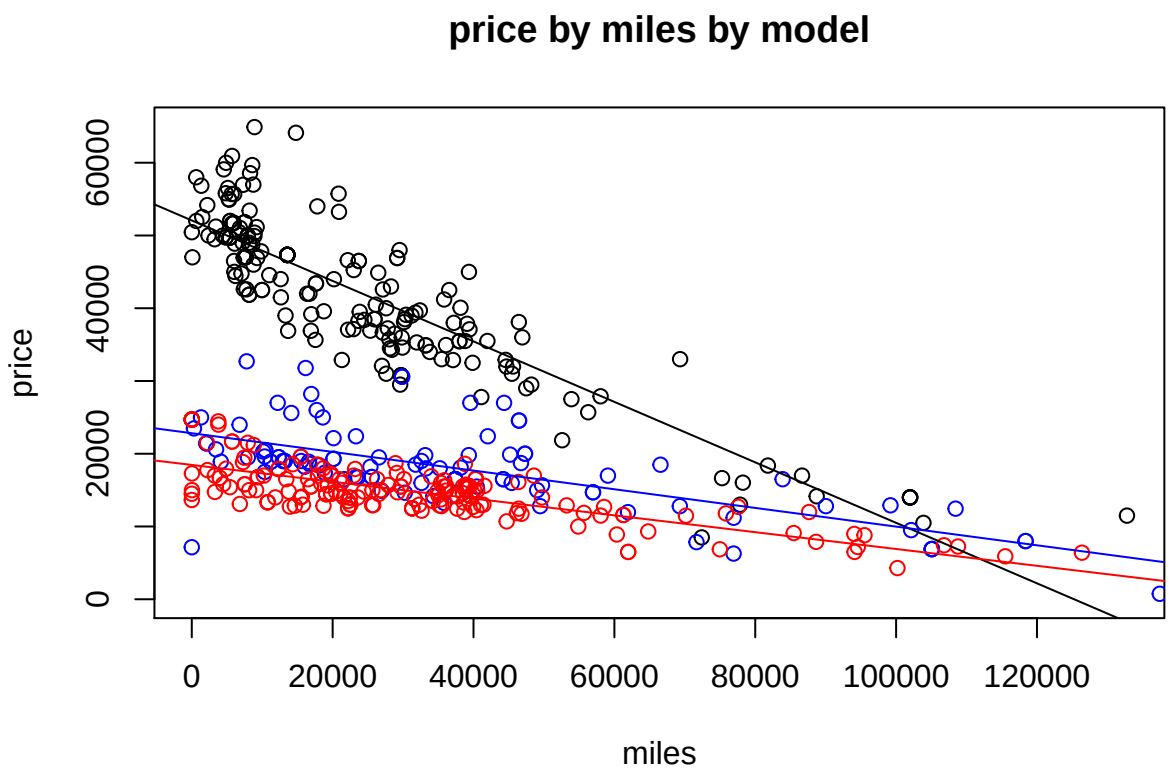
2. The difference in car prices predicted between BMW R series and Mazda 3 is 18600.17 dollars.

Part 3.5 (5 points): Now let's visualize these regression models. Start by recreating the scatter plot you created in part 3.1 using the same colors, but also add on the regression line with different y-intercepts and different slopes based on the model you fit in part 3.4 (again use the appropriate colors). Based on this visualization, does it seem that the slopes are different for all 3 car brands?

```
(the_coefs = coef(car_interaction_fit))
```

```
##                       (Intercept)                    mileage_bought
##                     52067.2339256                      -0.4156659
##                  make_boughtMazda                 make_boughtSubaru
##                    -33605.5952815                    -29246.8943770
##  mileage_bought:make_boughtMazda mileage_bought:make_boughtSubaru
##                         0.3001086                         0.2872814
```

```
plot(price_bought~mileage_bought, data = filter(three_car_data, make_bought=="BMW"),
     col = "black",
    ylab = "price",
    xlab = "miles",
    main = "price by miles by model",
    ylim = c(0, 65000))
points(price_bought~mileage_bought, data =
filter(three_car_data, make_bought == "Subaru"), ylim = c(0,65000), col = "blue")
points(price_bought~mileage_bought, data =
filter(three_car_data, make_bought == "Mazda"), ylim = c(0,65000), col = "red")
abline(the_coefs[1], the_coefs[2])
abline(the_coefs[1] + the_coefs[3], the_coefs[2] + the_coefs[5], col = "red")
abline(the_coefs[1]+ the_coefs[4], the_coefs[2] + the_coefs[6], col = "blue")
```



Answer

Based on the visualization, it seems like the slopes appear to be different now for all three lines.

Part 4: Polynomial regression

As discussed in class, we can create models that better fit our data by adding polynomial expanded terms; i.e., we can create models of the form: $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_1^3 + ...$ Let's explore fitting these models now!

Part 4.1 (7 points): Let's use polynomial regression to build models that predict the price that a car was purchased for (`price_bought`) as a function `mileage_bought` taken to different powers (i.e., *mileage*, $mileage^2$, etc.). To build these model, use the used Toyota Corolla data that you created in part 1 of the homework which you have stored in the object `used_corollas_all`.

Create polynomial fits for models of degree 1, degree 3 and degree 5, and for every model be sure to include the lower order terms as well; i.e., the model of degree 3 should be $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \hat{\beta}_3 x^3$. Save all these models to the object the names `model_1`, `model_3`, and `model_5`.

Then, use the `summary()` function to extract the r^2 values. You can do this in one line of code using the following syntax: `summary(model_k)$r.squared`, where `model_k` is one of our models. In the answer section below fill in a table reporting the different r^2 values. Based on these values, discuss which model you think has the best fit to the used Corolla data.

```
model_1<-
  lm(price_bought~mileage_bought, data = used_corollas_all)
model_3<-
  lm(price_bought~mileage_bought + I(mileage_bought^2)+ I(mileage_bought^3),
     data = used_corollas_all)
model_5<-
  lm(price_bought~mileage_bought + I(mileage_bought^2)+ I(mileage_bought^3)+I(mileage_bought^4)+
       I(mileage_bought^5), data = used_corollas_all)
summary(model_1)$r.squared
```

```
## [1] 0.5961973
```

```
summary(model_3)$r.squared
```

```
## [1] 0.6424999
```

```
summary(model_5)$r.squared
```

```
## [1] 0.6540874
```

Answer

Model degree	r^2
1	0.5961973
3	0.6424999
5	0.6540874

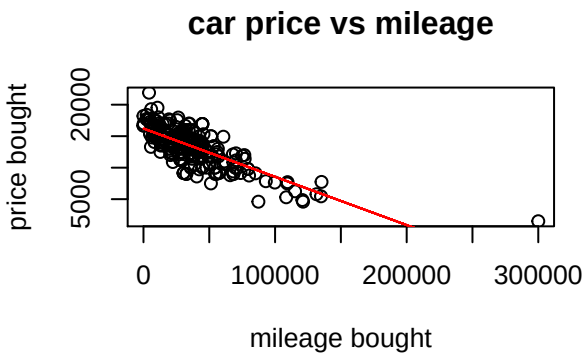
Degree 5 would appear to be the best fit since it has the highest r-squared value; however, this is expected since as as you increase the power of the model, it is supposed to account for more of the used Corolla data naturally. Because of this, it is useful to use other metrics beyond the r^2 value to determine which model has the best fit to the used Corolla data.

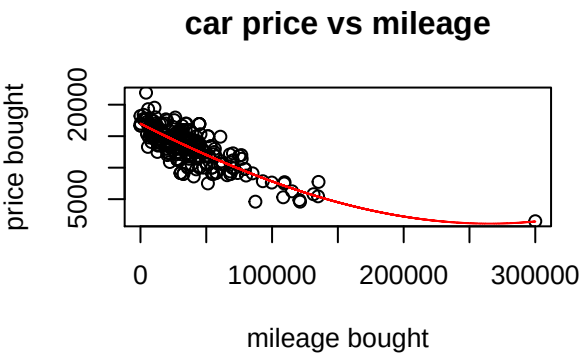
Part 4.2 (10 points): To gain better insight into which model to use, let's plot all three of these polynomial models using the following steps for each model:

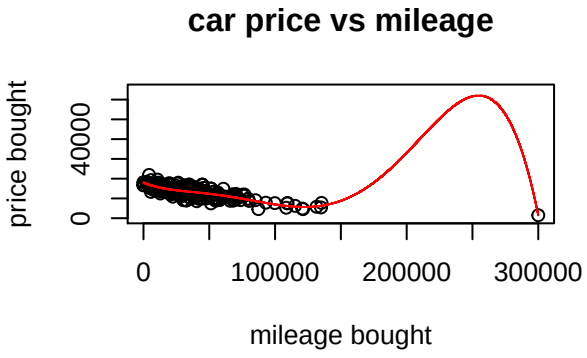
1. Create a scatter plot of the `price_bought` as a function of the `mileage_bought`.
2. Using the `predict_df` data frame created below to predict the y-hat values for all the x-values given in the `predict_df`. You can do this using the `predict()` function applied to the fit model.
3. Using the `points()` function to add a red line to the scatter plots that shows the predicted values. If the predicted values are being cut off by the edge of the figure, go back to step 1 and adjust the y-limits of the scatter plot using the `ylim = c(lower_lim, upper_lim)` argument.

Based on these plots, describe which model do you think is the best one and why.

```r
predict_df <- data.frame(mileage_bought = seq(0, 300000))
par(mfrow = c(2, 2)) # this create 4 subplots which makes the plots take up less space


prediction1<-predict(model_1, newdata = predict_df)
prediction3<-predict(model_3, newdata = predict_df)
prediction5<-predict(model_5, newdata = predict_df)

plot(price_bought~mileage_bought, data = used_corollas_all,
     xlab = "mileage bought",
     ylab = "price bought",
     main = "car price vs mileage")
points(prediction1, col = "red", type = "l")

plot(price_bought~mileage_bought, data = used_corollas_all,
     xlab = "mileage bought",
     ylab = "price bought",
     main = "car price vs mileage")
points(prediction3, col = "red", type = "l")
plot(price_bought~mileage_bought, data = used_corollas_all,
     xlab = "mileage bought",
     ylab = "price bought",
     main = "car price vs mileage",
     ylim =c(0,65000))
points(prediction5, col = "red", type = "l")
```







Answer

Model 1 does not seem to fit the data well since it abruptly goes to 0 at a certain number of miles. Model 5 does not make sense since it would be illogical that as more mileage would equate to a higher price beyond a certain points (Model 5 overfits the data). As a result, Model 3 makes the most sense intuitivelyt

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 8.