STAT 300: Used car data project

Sanjana Kuchipudi

# Introduction and directions

The purpose of this project is to give you experience sourcing, reading, and using real data to answer research questions using simple linear regression. You should refer back to your previous homework assignments and R notebooks used in lecture for the relevant R codes. You are also encouraged to get help from your instructor and TA during office hours.

## Collaboration rules:

You may consult with up to two classmates for help with this project, but *use your own data*. If you collaborate with someone and use the same make/model/zip code, you will both receive 50% of earned points if there are two of you and 33% if there are three of you. Please identify who you collaborate with here.

**List collaborators**

## Project premise

Let’s assume you are interested in purchasing a used car and you want to use data to help you research what you could consider a ‘fair price’. Obviously, the price of a car depends on many things, including the car’s age, mileage, condition, make, and model. At this time, we only have the tools to consider one predictor variable at a time so you will be using the variable ‘age’ to predict the price of used cars.

For this project you will source a new, never seen before dataset by scraping observations from autotrader.com for a make and model of your choosing. You’ll want to ultimately have a clean dataset of at least 50 cars. Because you will likely need to eliminate some observations that are clearly errors, make sure the zip code you choose has at least 60 cars downloaded to the dataset.

To get your data, go to <http://myslu.stlawu.edu/~clee/dataset/autotrader/>, choose the make and model, then input a zip code. If you are choosing a more rare type of car it might be difficult to get at least 60 observations for certain zip codes. Try a zip code close to a big city like Boston (02124), Los Angeles (90010), or Chicago (60176). Save the data and choose a name for the dataset with a .csv extension. After you save the data, you should check the spreadsheet for any cases that should be deleted. For example, sometimes new cars will be included (mileage of 0), or odd entries with a price of 0 will appear. Make sure that after cleaning the data you have at least 50 observations. Note that if you are more comfortable cleaning the data in R, you are welcome to filter your dataset as part of your code.

You should have a dataset with variables ‘year’, ‘price’ (in $1,000’s), and ‘mileage’ (in 1,000’s) ready to load into R. Run the front-matter below to load your data into the workspace and load the packages you are most likely to need for this project.

#add a variable that is named 'age', which is 2021 - year  
used\_cars$age <- 2021 - used\_cars$year

# filter observations  
UsedCars <- used\_cars %>%  
 filter(mileage > 0) %>%  
 filter(price >0)

glimpse(UsedCars)

## Rows: 278  
## Columns: 4  
## $ year <int> 2010, 2019, 2018, 2018, 2015, 2016, 2019, 2018, 2017, 2019, 20…  
## $ price <dbl> 16.998, 58.798, 46.604, 45.850, 26.961, 26.888, 59.900, 35.980…  
## $ mileage <dbl> 82.078, 33.714, 57.669, 20.807, 87.231, 96.695, 22.843, 94.844…  
## $ age <dbl> 11, 2, 3, 3, 6, 5, 2, 3, 4, 2, 2, 4, 3, 3, 3, 8, 6, 2, 4, 5, 8…

# get down to 50  
set.seed(4)  
UsedCars <- UsedCars[-sample(1:60,10,replace=F),]

glimpse(UsedCars)

## Rows: 268  
## Columns: 4  
## $ year <int> 2010, 2019, 2018, 2015, 2016, 2018, 2017, 2019, 2017, 2018, 20…  
## $ price <dbl> 16.998, 58.798, 45.850, 26.961, 26.888, 35.980, 48.995, 57.675…  
## $ mileage <dbl> 82.078, 33.714, 20.807, 87.231, 96.695, 94.844, 67.879, 42.919…  
## $ age <dbl> 11, 2, 3, 6, 5, 3, 4, 2, 4, 3, 3, 3, 8, 6, 2, 4, 5, 8, 4, 7, 7…

UsedCars

## year price mileage age  
## 1 2010 16.998 82.078 11  
## 2 2019 58.798 33.714 2  
## 4 2018 45.850 20.807 3  
## 5 2015 26.961 87.231 6  
## 6 2016 26.888 96.695 5  
## 8 2018 35.980 94.844 3  
## 9 2017 48.995 67.879 4  
## 10 2019 57.675 42.919 2  
## 12 2017 41.990 31.649 4  
## 13 2018 49.858 29.116 3  
## 14 2018 46.998 35.545 3  
## 15 2018 45.798 43.319 3  
## 16 2013 20.990 71.715 8  
## 17 2015 29.977 66.065 6  
## 18 2019 57.499 31.975 2  
## 19 2017 36.648 45.176 4  
## 20 2016 29.495 75.869 5  
## 21 2013 20.990 70.391 8  
## 22 2017 36.999 52.146 4  
## 23 2014 24.990 85.877 7  
## 24 2014 28.900 69.714 7  
## 25 2013 18.990 91.230 8  
## 26 2015 25.999 91.697 6  
## 27 2017 31.995 94.619 4  
## 28 2018 57.400 20.331 3  
## 29 2013 7.500 126.527 8  
## 31 2018 45.998 40.151 3  
## 32 2019 51.440 52.864 2  
## 33 2018 47.277 33.455 3  
## 34 2015 26.900 93.447 6  
## 35 2016 30.987 56.001 5  
## 36 2019 59.998 31.687 2  
## 37 2018 47.419 30.881 3  
## 38 2018 41.839 30.195 3  
## 39 2017 33.590 63.064 4  
## 40 2017 36.990 43.844 4  
## 41 2017 41.990 35.514 4  
## 42 2018 52.647 13.082 3  
## 43 2019 72.500 35.854 2  
## 45 2018 50.025 31.262 3  
## 46 2018 46.998 47.475 3  
## 47 2019 64.898 44.246 2  
## 48 2018 41.799 57.116 3  
## 49 2013 7.995 125.955 8  
## 50 2019 51.390 46.621 2  
## 52 2018 39.990 37.906 3  
## 55 2017 32.950 76.542 4  
## 57 2018 45.990 25.989 3  
## 59 2014 22.744 83.997 7  
## 60 2018 42.990 32.541 3  
## 61 2017 32.030 59.455 4  
## 62 2018 55.277 20.100 3  
## 63 2018 43.990 42.290 3  
## 64 2018 47.699 18.333 3  
## 65 2017 41.805 51.240 4  
## 66 2014 23.898 82.472 7  
## 67 2018 42.995 32.525 3  
## 68 2018 43.944 21.636 3  
## 69 2018 48.995 31.600 3  
## 70 2018 50.685 31.884 3  
## 71 2017 32.295 69.891 4  
## 72 2018 51.800 16.908 3  
## 73 2017 35.798 57.092 4  
## 74 2017 35.990 52.564 4  
## 75 2016 29.995 87.302 5  
## 76 2014 24.999 113.108 7  
## 77 2018 45.650 34.420 3  
## 78 2009 13.995 75.033 12  
## 79 2018 43.990 43.903 3  
## 80 2018 41.397 54.231 3  
## 81 2018 50.999 17.771 3  
## 82 2018 45.500 39.142 3  
## 83 2018 54.488 28.201 3  
## 84 2019 59.990 28.488 2  
## 85 2018 50.988 19.314 3  
## 86 2015 19.995 119.000 6  
## 87 2018 50.998 13.608 3  
## 88 2018 45.162 28.594 3  
## 89 2016 33.499 84.040 5  
## 90 2018 48.995 37.029 3  
## 91 2013 15.995 119.138 8  
## 92 2018 46.998 47.019 3  
## 93 2017 36.495 51.217 4  
## 94 2019 58.887 33.549 2  
## 95 2018 45.675 31.609 3  
## 96 2018 44.995 17.491 3  
## 97 2011 12.995 117.683 10  
## 98 2018 48.500 30.349 3  
## 99 2018 46.850 35.671 3  
## 100 2013 23.990 37.445 8  
## 101 2019 58.561 32.233 2  
## 102 2019 54.785 52.498 2  
## 103 2012 21.995 80.219 9  
## 104 2015 26.990 87.143 6  
## 105 2018 41.623 54.420 3  
## 106 2019 56.998 34.493 2  
## 107 2018 45.975 46.834 3  
## 108 2019 59.988 34.220 2  
## 109 2015 27.750 71.794 6  
## 110 2018 53.800 27.773 3  
## 111 2018 47.000 21.829 3  
## 112 2016 30.990 65.414 5  
## 113 2018 42.512 49.628 3  
## 114 2016 33.990 43.332 5  
## 115 2019 56.994 43.053 2  
## 116 2018 46.998 36.311 3  
## 117 2019 66.995 40.818 2  
## 118 2014 26.998 62.570 7  
## 119 2017 36.500 47.520 4  
## 120 2019 57.966 28.904 2  
## 121 2018 46.998 34.625 3  
## 122 2018 42.999 44.714 3  
## 123 2019 57.991 42.111 2  
## 124 2018 46.064 31.454 3  
## 125 2018 45.500 36.935 3  
## 126 2013 19.995 93.078 8  
## 127 2015 23.999 99.650 6  
## 128 2018 39.344 55.236 3  
## 129 2019 55.400 36.513 2  
## 130 2018 45.400 36.387 3  
## 131 2018 48.400 32.114 3  
## 132 2019 58.491 38.214 2  
## 133 2019 59.998 27.539 2  
## 134 2017 38.495 56.252 4  
## 135 2018 47.900 28.966 3  
## 136 2011 18.590 20.490 10  
## 137 2018 59.550 26.187 3  
## 138 2018 45.900 39.169 3  
## 139 2018 39.900 58.195 3  
## 140 2013 17.500 93.612 8  
## 141 2018 44.800 32.121 3  
## 142 2018 44.991 48.964 3  
## 143 2013 16.849 87.350 8  
## 144 2016 32.444 69.158 5  
## 145 2018 48.995 30.998 3  
## 146 2018 46.456 32.953 3  
## 147 2018 46.277 27.897 3  
## 148 2009 5.995 197.487 12  
## 149 2016 33.371 64.222 5  
## 150 2019 59.950 11.669 2  
## 151 2017 38.499 53.498 4  
## 152 2018 56.798 38.901 3  
## 153 2018 49.025 46.656 3  
## 154 2018 52.646 11.641 3  
## 155 2015 26.590 74.543 6  
## 156 2018 46.900 35.772 3  
## 157 2017 25.988 139.553 4  
## 158 2018 49.800 24.407 3  
## 159 2018 46.975 30.840 3  
## 160 2018 35.990 82.382 3  
## 161 2019 56.524 35.690 2  
## 162 2018 45.998 41.879 3  
## 163 2019 63.698 31.803 2  
## 164 2019 60.800 38.057 2  
## 165 2015 26.590 81.230 6  
## 166 2018 41.590 49.478 3  
## 167 2017 36.447 29.560 4  
## 168 2018 53.800 39.274 3  
## 169 2019 54.900 31.921 2  
## 170 2018 41.800 39.124 3  
## 171 2018 45.975 47.629 3  
## 172 2017 35.800 49.630 4  
## 173 2018 40.999 61.830 3  
## 174 2018 46.500 45.000 3  
## 175 2013 16.299 86.754 8  
## 176 2017 35.569 55.003 4  
## 177 2016 32.995 32.398 5  
## 178 2019 58.986 30.689 2  
## 179 2015 34.989 51.049 6  
## 180 2018 38.698 56.223 3  
## 181 2015 25.900 82.241 6  
## 182 2013 20.990 68.494 8  
## 183 2018 46.988 64.807 3  
## 184 2015 32.888 57.456 6  
## 185 2017 33.990 61.850 4  
## 186 2013 17.995 95.179 8  
## 187 2018 45.988 31.622 3  
## 188 2019 57.894 33.417 2  
## 189 2019 58.968 23.495 2  
## 190 2014 25.295 79.319 7  
## 191 2019 58.598 32.948 2  
## 192 2018 45.590 30.035 3  
## 193 2019 53.995 45.066 2  
## 194 2019 56.800 19.764 2  
## 195 2018 44.999 35.650 3  
## 196 2006 8.989 97.305 15  
## 197 2019 59.995 41.607 2  
## 198 2019 74.900 17.598 2  
## 199 2017 38.310 36.906 4  
## 200 2019 69.995 1.000 2  
## 201 2018 51.100 20.961 3  
## 202 2011 14.995 139.469 10  
## 203 2018 46.750 21.694 3  
## 204 2016 35.968 63.615 5  
## 205 2014 23.199 97.856 7  
## 206 2018 53.950 26.144 3  
## 207 2014 27.900 78.281 7  
## 208 2018 42.998 32.764 3  
## 209 2019 57.499 32.950 2  
## 210 2018 47.500 25.565 3  
## 211 2017 38.984 29.838 4  
## 212 2014 35.995 78.533 7  
## 213 2019 59.949 25.205 2  
## 214 2019 56.454 1.000 2  
## 215 2018 46.850 29.982 3  
## 216 2015 26.276 84.665 6  
## 217 2015 32.990 67.526 6  
## 218 2018 44.999 44.165 3  
## 219 2018 46.975 48.761 3  
## 220 2015 28.990 65.581 6  
## 221 2018 52.634 30.727 3  
## 222 2017 32.337 93.022 4  
## 223 2019 59.277 24.115 2  
## 224 2019 58.033 35.686 2  
## 225 2019 57.488 24.019 2  
## 226 2018 40.590 56.260 3  
## 227 2019 55.995 55.174 2  
## 228 2019 59.698 24.952 2  
## 229 2018 43.590 33.390 3  
## 230 2019 57.700 15.529 2  
## 231 2019 59.488 21.149 2  
## 232 2018 46.994 27.684 3  
## 233 2018 45.900 27.593 3  
## 234 2018 48.500 14.804 3  
## 235 2011 12.995 120.089 10  
## 236 2016 36.700 42.959 5  
## 237 2014 18.725 154.550 7  
## 238 2018 42.598 45.265 3  
## 239 2017 34.646 65.804 4  
## 240 2018 45.965 36.174 3  
## 241 2018 40.988 54.629 3  
## 242 2005 7.599 108.519 16  
## 243 2018 44.898 40.080 3  
## 244 2014 23.495 102.459 7  
## 245 2016 26.500 107.259 5  
## 246 2018 48.995 25.421 3  
## 247 2018 46.998 24.102 3  
## 248 2019 57.900 33.430 2  
## 249 2018 48.900 25.334 3  
## 250 2017 40.977 46.369 4  
## 251 2018 44.990 31.920 3  
## 252 2018 48.800 35.472 3  
## 253 2018 47.745 8.673 3  
## 254 2019 65.800 25.574 2  
## 255 2018 45.590 28.047 3  
## 256 2012 18.590 91.249 9  
## 257 2018 52.888 13.187 3  
## 258 2018 43.500 45.188 3  
## 259 2018 44.590 26.906 3  
## 260 2016 32.027 62.543 5  
## 261 2018 42.995 88.488 3  
## 262 2018 58.900 31.854 3  
## 263 2011 13.995 115.981 10  
## 264 2018 54.896 32.370 3  
## 265 2018 46.790 32.113 3  
## 266 2013 19.486 105.359 8  
## 267 2009 9.990 116.000 12  
## 268 2018 47.500 34.261 3  
## 269 2018 47.850 46.196 3  
## 270 2018 44.718 41.302 3  
## 271 2018 52.995 34.353 3  
## 272 2016 39.995 78.285 5  
## 273 2015 24.999 87.613 6  
## 274 2016 29.900 91.204 5  
## 275 2019 74.888 14.364 2  
## 276 2011 17.990 69.860 10  
## 277 2018 46.998 27.111 3  
## 278 2018 49.995 35.801 3

# Project

**Introduce your data using complete sentences.** What kind of car are you looking at? Where did these car listings come from (zip code and town)? Example:

I chose to study BMW X5s from the downtown Boston area and the zip code is 02101.

## Model: Choose

**Use R to compute each of the summary statistics below, writing them in the text next to their names.**

* average age: 4.1007463 4.101 years
* standard deviation of age: 2.3672863 2.367 years.
* average price: 41.7751306 41.775 thousands of dollars
* standard deviation of price: 13.8578111 13.858 thousands of dollars

## Average age

mean(UsedCars$age)

## [1] 4.100746

## Standard deviation of age

sd(UsedCars$age)

## [1] 2.367286

## Average price

mean(UsedCars$price)

## [1] 41.77513

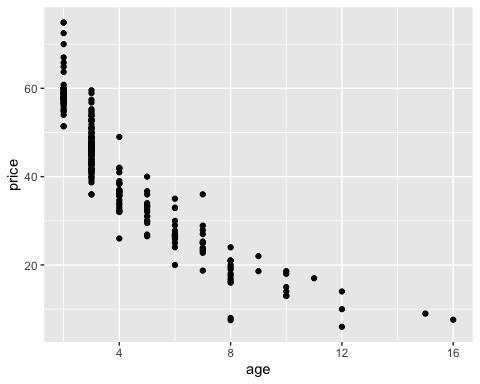
## Standard deviation of price

sd(UsedCars$price)

## [1] 13.85781

**Produce a scatterplot of the relationship between age and price.**

gf\_point(price ~ age, data = UsedCars)



**Using complete sentences, describe what you’ve learned from your exploratory data analysis. Make sure you are thorough, and include information about what you learned from the scatterplot.**

Most cars in the dataset are under 5 years old and the range goes from 0 to 16 years. The average age of the cars is 4.101 years and the standard deviation is 2.367 years. The majority of the cars are around 2 to 6 years old.. The average price of the cars is 41,775 dollars and the standard deviation of price is 13,858 dollars. It seems like the younger the car, the more expensive the car will be which means there a negative relationship between age and price of the car. It lools like a negative expoential relationship.

## Model: Fit

**Fit a simple linear model to your data.** Use R to compute each of the summary statistics below, writing them in the text next to their names.

MyCarModel <- lm(price ~ age, data = UsedCars)  
summary(MyCarModel)

##   
## Call:  
## lm(formula = price ~ age, data = UsedCars)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -16.3105 -4.2427 -0.6433 3.8433 27.6381   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 63.0777 0.7831 80.55 <2e-16 \*\*\*  
## age -5.1948 0.1655 -31.40 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.4 on 266 degrees of freedom  
## Multiple R-squared: 0.7875, Adjusted R-squared: 0.7867   
## F-statistic: 985.8 on 1 and 266 DF, p-value: < 2.2e-16

summary(MyCarModel)$coeff[1,1] ## estimated intercept

## [1] 63.0777

summary(MyCarModel)$coeff[2,1] ## estimated slope

## [1] -5.194802

summary(MyCarModel)$r.squared ## standard error of regression

## [1] 0.7874995

anova(MyCarModel)$Sum[1] ## SSModel

## [1] 40378.56

anova(MyCarModel)$Sum[2] ## SSError

## [1] 10895.83

anova(MyCarModel)$Sum[1]+anova(MyCarModel)$Sum[2] ## SSTotal

## [1] 51274.39

* degrees of freedom: 1 and 48

**Interpret, in context, what the slope estimate tells you about age and price in your used car model. Make sure you add a sentence about why the sign (positive or negative) makes sense.**

The slope is -5.195 In context, this tells me that as age increases by 1 year, the price of the car will decrease by 5.195 thousand dollar. The negative sign makes sense because as cars get older their value decreases so therefore the price will decrease over time.

## Model: Assess

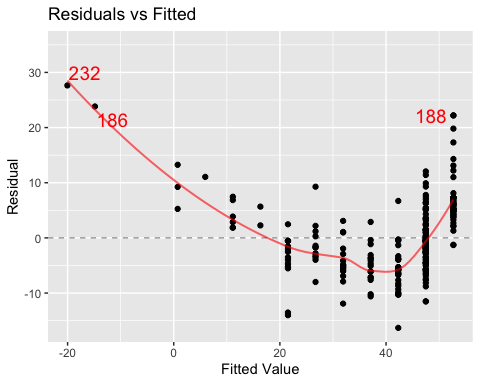
#### Residual plots

**Produce the appropriate residual plots**

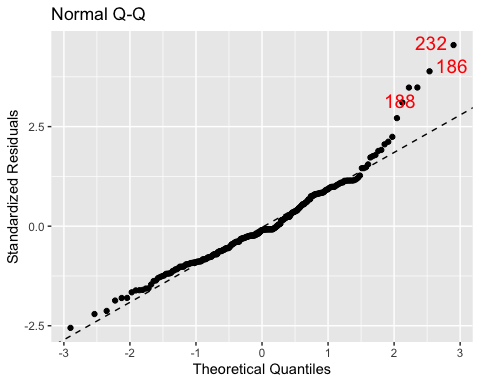
mplot(MyCarModel, which = c(1,2))

## [[1]]

## `geom\_smooth()` using formula 'y ~ x'



##   
## [[2]]



**Comment on how well your data appear to fit the conditions for a simple linear model.** At this point, don’t worry about doing any transformations if there are problems with the conditions, just mention them.

The residual vs fitted plot shows us that the data does not fit the conditions for a simple linear model. The data is not shapeless because there is clearly a curved pattern in the data. The data is clustered on the right side together instead of distributed symmetrically. The model also doesn’t meet the condition of zero mean. there are outliers that are skewing the data and it doesn’t meet the condition of normality. The normal q-q plot shows that the line fits well but there on both sides there are outliers that don’t fit the line so it is not noramlly distributed.

#### Unusual points

Find the car in your sample with the largest (in magnitude) residual. **What is the age and price of this car?** The car with the largest residual of 22.19 is 275 and the age of this car is 2 so it is from 2019. The price of this car is 74.888 thousand dollars.

mod1 = lm(price ~ age, data = UsedCars)  
mod1$residuals

## 1 2 4 5 6   
## 11.063126323 6.109908534 -1.643289489 -4.947883559 -10.215685536   
## 8 9 10 12 13   
## -11.513289489 6.696512487 4.986908534 -0.308487513 2.364710511   
## 14 15 16 17 18   
## -0.495289489 -1.695289489 -0.529279606 -1.931883559 4.810908534   
## 19 20 21 22 23   
## -5.650487513 -7.608685536 -0.529279606 -5.299487513 -1.724081583   
## 24 25 26 27 28   
## 2.185918417 -2.529279606 -5.909883559 -10.303487513 9.906710511   
## 29 31 32 33 34   
## -14.019279606 -1.495289489 -1.248091466 -0.216289489 -5.008883559   
## 35 36 37 38 39   
## -6.116685536 7.309908534 -0.074289489 -5.654289489 -8.708487513   
## 40 41 42 43 45   
## -5.308487513 -0.308487513 5.153710511 19.811908534 2.531710511   
## 46 47 48 49 50   
## -0.495289489 12.209908534 -5.694289489 -13.524279606 -1.298091466   
## 52 55 57 59 60   
## -7.503289489 -9.348487513 -1.503289489 -3.970081583 -4.503289489   
## 61 62 63 64 65   
## -10.268487513 7.783710511 -3.503289489 0.205710511 -0.493487513   
## 66 67 68 69 70   
## -2.816081583 -4.498289489 -3.549289489 1.501710511 3.191710511   
## 71 72 73 74 75   
## -10.003487513 4.306710511 -6.500487513 -6.308487513 -7.108685536   
## 76 77 78 79 80   
## -1.715081583 -1.843289489 13.254928300 -3.503289489 -6.096289489   
## 81 82 83 84 85   
## 3.505710511 -1.993289489 6.994710511 7.301908534 3.494710511   
## 86 87 88 89 90   
## -11.913883559 3.504710511 -2.331289489 -3.604685536 1.501710511   
## 91 92 93 94 95   
## -5.524279606 -0.495289489 -5.803487513 6.198908534 -1.818289489   
## 96 97 98 99 100   
## -2.498289489 1.865324347 1.006710511 -0.643289489 2.470720394   
## 101 102 103 104 105   
## 5.872908534 2.096908534 5.670522370 -4.918883559 -5.870289489   
## 106 107 108 109 110   
## 4.309908534 -1.518289489 7.299908534 -4.158883559 6.306710511   
## 111 112 113 114 115   
## -0.493289489 -6.113685536 -4.981289489 -3.113685536 4.305908534   
## 116 117 118 119 120   
## -0.495289489 14.306908534 0.283918417 -5.798487513 5.277908534   
## 121 122 123 124 125   
## -0.495289489 -4.494289489 5.302908534 -1.429289489 -1.993289489   
## 126 127 128 129 130   
## -1.524279606 -7.909883559 -8.149289489 2.711908534 -2.093289489   
## 131 132 133 134 135   
## 0.906710511 5.802908534 7.309908534 -3.803487513 0.406710511   
## 136 137 138 139 140   
## 7.460324347 12.056710511 -1.593289489 -7.593289489 -4.019279606   
## 141 142 143 144 145   
## -2.693289489 -2.502289489 -4.670279606 -4.659685536 1.501710511   
## 146 147 148 149 150   
## -1.037289489 -1.216289489 5.254928300 -3.732685536 7.261908534   
## 151 152 153 154 155   
## -3.799487513 9.304710511 1.531710511 5.152710511 -5.318883559   
## 156 157 158 159 160   
## -0.593289489 -16.310487513 2.306710511 -0.518289489 -11.503289489   
## 161 162 163 164 165   
## 3.835908534 -1.495289489 11.009908534 8.111908534 -5.318883559   
## 166 167 168 169 170   
## -5.903289489 -5.851487513 6.306710511 2.211908534 -5.693289489   
## 171 172 173 174 175   
## -1.518289489 -6.498487513 -6.494289489 -0.993289489 -5.220279606   
## 176 177 178 179 180   
## -6.729487513 -4.108685536 6.297908534 3.080116441 -8.795289489   
## 181 182 183 184 185   
## -6.008883559 -0.529279606 -0.505289489 0.979116441 -8.308487513   
## 186 187 188 189 190   
## -3.524279606 -1.505289489 5.205908534 6.279908534 -1.419081583   
## 191 192 193 194 195   
## 5.909908534 -1.903289489 1.306908534 4.111908534 -2.494289489   
## 196 197 198 199 200   
## 23.833334229 7.306908534 22.211908534 -3.988487513 17.306908534   
## 201 202 203 204 205   
## 3.606710511 3.865324347 -0.743289489 -1.135685536 -3.515081583   
## 206 207 208 209 210   
## 6.456710511 1.185918417 -4.495289489 4.810908534 0.006710511   
## 211 212 213 214 215   
## -3.314487513 9.280918417 7.260908534 3.765908534 -0.643289489   
## 216 217 218 219 220   
## -5.632883559 1.081116441 -2.494289489 -0.518289489 -2.918883559   
## 221 222 223 224 225   
## 5.140710511 -9.961487513 6.588908534 5.344908534 4.799908534   
## 226 227 228 229 230   
## -6.903289489 3.306908534 7.009908534 -3.903289489 5.011908534   
## 231 232 233 234 235   
## 6.799908534 -0.499289489 -1.593289489 1.006710511 1.865324347   
## 236 237 238 239 240   
## -0.403685536 -7.989081583 -4.895289489 -7.652487513 -1.528289489   
## 241 242 243 244 245   
## -6.505289489 27.638136206 -2.595289489 -3.219081583 -10.603685536   
## 246 247 248 249 250   
## 1.501710511 -0.495289489 5.211908534 1.406710511 -1.321487513   
## 251 252 253 254 255   
## -2.503289489 1.306710511 0.251710511 13.111908534 -1.903289489   
## 256 257 258 259 260   
## 2.265522370 5.394710511 -3.993289489 -2.903289489 -5.076685536   
## 261 262 263 264 265   
## -4.498289489 11.406710511 2.865324347 7.402710511 -0.703289489   
## 266 267 268 269 270   
## -2.033279606 9.249928300 0.006710511 0.356710511 -2.775289489   
## 271 272 273 274 275   
## 5.501710511 2.891314464 -6.909883559 -7.203685536 22.199908534   
## 276 277 278   
## 6.860324347 -0.495289489 2.501710511

**Use R to find this car’s studentized residual, leverage, and Cook’s distance.** Would any of these values be considered unusual? Why or why not? Again, use complete sentences.

The standardized residual, for this car(275) is 3.480305250. This value is definitely considered influential because the standardized residual value is much higher than all the other standardized residuals values.

rstandard(MyCarModel)

## 1 2 4 5 6 8   
## 1.760139539 0.957857404 -0.257343374 -0.775475178 -1.599586344 -1.803010843   
## 9 10 12 13 14 15   
## 1.048268428 0.781803399 -0.048290468 0.370319768 -0.077563612 -0.265486709   
## 16 17 18 19 20 21   
## -0.083278644 -0.302781529 0.754211676 -0.884524246 -1.191378634 -0.083278644   
## 22 23 24 25 26 27   
## -0.829578897 -0.270650019 0.343150154 -0.397965409 -0.926248152 -1.612902339   
## 28 29 31 32 33 34   
## 1.551416430 -2.205840878 -0.234166192 -0.195664738 -0.033871492 -0.785035627   
## 35 36 37 38 39 40   
## -0.957759185 1.145982787 -0.011633926 -0.885476324 -1.363221906 -0.830987752   
## 41 42 43 45 46 47   
## -0.048290468 0.807084365 3.105935737 0.396472399 -0.077563612 1.914161435   
## 48 49 50 52 55 57   
## -0.891740428 -2.127955904 -0.203503295 -1.175034496 -1.463407158 -0.235419013   
## 59 60 61 62 63 64   
## -0.623231909 -0.705226754 -1.607423458 1.218949151 -0.548624174 0.032214797   
## 65 66 67 68 69 70   
## -0.077250267 -0.442074518 -0.704443742 -0.555827893 0.235171741 0.499830102   
## 71 72 73 74 75 76   
## -1.565940502 0.674441979 -1.017582785 -0.987527209 -1.113087934 -0.269237180   
## 77 78 79 80 81 82   
## -0.288663890 2.119751169 -0.548624174 -0.954694665 0.549003312 -0.312154278   
## 83 84 85 86 87 88   
## 1.095389715 1.144728618 0.547280684 -1.867247049 0.548846710 -0.365085950   
## 89 90 91 92 93 94   
## -0.564426708 0.235171741 -0.869208841 -0.077563612 -0.908474783 0.971810036   
## 95 96 97 98 99 100   
## -0.284748826 -0.391238581 0.295465138 0.157653464 -0.100740794 0.388751505   
## 101 102 103 104 105 106   
## 0.920702640 0.328734768 0.894891324 -0.770930047 -0.919302482 0.675669329   
## 107 108 109 110 111 112   
## -0.237768052 1.144415075 -0.651816262 0.987647140 -0.077250407 -0.957289441   
## 113 114 115 116 117 118   
## -0.780082788 -0.487545241 0.675042244 -0.077563612 2.242910542 0.044570121   
## 119 120 121 122 123 124   
## -0.907692086 0.827423805 -0.077563612 -0.703817331 0.831343083 -0.223830422   
## 125 126 127 128 129 130   
## -0.312154278 -0.239835310 -1.239705480 -1.276199763 0.425149027 -0.327814536   
## 131 132 133 134 135 136   
## 0.141993206 0.909728660 1.145982787 -0.595395870 0.063691915 1.181706423   
## 137 138 139 140 141 142   
## 1.888111977 -0.249513245 -1.189128728 -0.632407050 -0.421776084 -0.391864991   
## 143 144 145 146 147 148   
## -0.734837592 -0.729620085 0.235171741 -0.162442211 -0.190474073 0.840377266   
## 149 150 151 152 153 154   
## -0.584469127 1.138457771 -0.594769712 1.457141676 0.239869818 0.806927762   
## 155 156 157 158 159 160   
## -0.833621512 -0.092910665 -2.553234857 0.361236818 -0.081165471 -1.801444817   
## 161 162 163 164 165 166   
## 0.601359802 -0.234166192 1.726036052 1.271713251 -0.833621512 -0.924470367   
## 167 168 169 170 171 172   
## -0.915988677 0.987647140 0.346763451 -0.891583825 -0.237768052 -1.017269706   
## 173 174 175 176 177 178   
## -1.017022492 -0.155551697 -0.821376453 -1.053430320 -0.643343734 0.987330380   
## 179 180 181 182 183 184   
## 0.482742534 -1.377365030 -0.941764290 -0.083278644 -0.079129638 0.153455611   
## 185 186 187 188 189 190   
## -1.300606123 -0.554522075 -0.235732218 0.816136282 0.984508500 -0.222770466   
## 191 192 193 194 195 196   
## 0.926503173 -0.298060045 0.204885557 0.644628641 -0.390612170 3.889019266   
## 197 198 199 200 201 202   
## 1.145512473 3.482186504 -0.624355669 2.713224000 0.564820173 0.612262737   
## 203 204 205 206 207 208   
## -0.116401052 -0.177827231 -0.551805034 1.011137527 0.186168013 -0.703973934   
## 209 210 211 212 213 214   
## 0.754211676 0.001050883 -0.518848075 1.456938448 1.138301000 0.590385822   
## 215 216 217 218 219 220   
## -0.100740794 -0.882834313 0.169441935 -0.390612170 -0.081165471 -0.457472720   
## 221 222 223 224 225 226   
## 0.805048531 -1.559365845 1.032950786 0.837927472 0.752487194 -1.081072947   
## 227 228 229 230 231 232   
## 0.518427863 1.098951441 -0.611265206 0.785722678 1.066029499 -0.078190022   
## 233 234 235 236 237 238   
## -0.249513245 0.157653464 0.295465138 -0.063209646 -1.254143136 -0.766614966   
## 239 240 241 242 243 244   
## -1.197916239 -0.239334078 -1.018745120 4.547816706 -0.406429031 -0.505338321   
## 245 246 247 248 249 250   
## -1.660339927 0.235171741 -0.077563612 0.817076909 0.220294496 -0.206864938   
## 251 252 253 254 255 256   
## -0.392021594 0.204634238 0.039418516 2.055569015 -0.298060045 0.357532548   
## 257 258 259 260 261 262   
## 0.844825587 -0.625359438 -0.454662626 -0.794914529 -0.704443742 1.786320300   
## 263 264 265 266 267 268   
## 0.453863938 1.159283568 -0.110136949 -0.319923092 1.479264609 0.001050883   
## 269 270 271 272 273 274   
## 0.055861786 -0.434617495 0.861582063 0.452726067 -1.082976816 -1.127963167   
## 275 276 277 278   
## 3.480305250 1.086667143 -0.077563612 0.391774321

hatvalues(MyCarModel)[265]

## 275   
## 0.006680748

2\*(2/275)

## [1] 0.01454545

3\*(2/275)

## [1] 0.02181818

The leverage is less than two or three times the average leverage that is according to the linear model.

## Model: Use

#### Confidence interval

**Compute and interpret a 95% confidence interval for the slope of your model.**

confint(MyCarModel, level = 0.95)

## 2.5 % 97.5 %  
## (Intercept) 61.535894 64.619497  
## age -5.520572 -4.869032

There is a 95% chance that the true slope of the data falls between -5.521 and -4.869.

#### Coefficient of determination

**Report the coefficient of determination (r-squared) and show how it can be computed using values from the ANOVA table.**

R^2 = 1 - SSE / SST 1- (10896/(40379+10896)) = 0.787

anova(MyCarModel)

## Analysis of Variance Table  
##   
## Response: price  
## Df Sum Sq Mean Sq F value Pr(>F)   
## age 1 40379 40379 985.76 < 2.2e-16 \*\*\*  
## Residuals 266 10896 41   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**Interpret the value in context using a complete sentence.**

The value of R^2 shows how well the regression line fits with the data. The closer it is to 1 the better of a fit it is. The value is 0.787 so the fit is fairly good.

summary(MyCarModel)

##   
## Call:  
## lm(formula = price ~ age, data = UsedCars)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -16.3105 -4.2427 -0.6433 3.8433 27.6381   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 63.0777 0.7831 80.55 <2e-16 \*\*\*  
## age -5.1948 0.1655 -31.40 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.4 on 266 degrees of freedom  
## Multiple R-squared: 0.7875, Adjusted R-squared: 0.7867   
## F-statistic: 985.8 on 1 and 266 DF, p-value: < 2.2e-16

#### Hypothesis tests

**Test the strength of the linear relationship between age and price using all three methods discussed in class.** For each of them, write the hypotheses (it’s fine to type them out without using special symbols), discuss how to calculate test statistic and show its value, indicate the reference distribution (t or F including degrees of freedom), and report the p-values. At the end, you can write one conclusion in context that reflects the conclusion based on all three p-values.

cor(UsedCars$price, UsedCars$age)

## [1] -0.8874117

1. Test for correlation H0: r; age = 0 HA: r; age =/ 0 The r value is -0.887 which means there is a strong negative relationship between age and price. As the age of the car goes up the price of the car does down.
2. Test for slope H0:βAge=0.0 HA:βAge>0.0 The p-value is really really low so it is approximately 0. This tells us that there is strong evidence to reject the null hypothesis. There is enough evidence to reject the null and conclude that there is a relationship between variables age and price.
3. ANOVA for regression H0: βAge=0 HA: βAge=/0 The test shows that the f value is 985.76 and the p-value is 2.2e-16 which is approximately 0. We can rehect the null and say that there is a relationship between varibles age and price.

anova(MyCarModel)

## Analysis of Variance Table  
##   
## Response: price  
## Df Sum Sq Mean Sq F value Pr(>F)   
## age 1 40379 40379 985.76 < 2.2e-16 \*\*\*  
## Residuals 266 10896 41   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**Conclusion in context:**

#### Predictions

Suppose you are interested in purchasing a car of this make and model that is five years old. For each of quantities below, show how to complete the calculations using formulas (with the correct numbers in the correct places). For the intervals, write a sentence that carefully interprets each in terms of car prices.

newdata=data.frame(age = 5)  
predict.lm(MyCarModel, newdata, interval = "confidence", level = 0.9)

## fit lwr upr  
## 1 37.10369 36.41323 37.79414

predict.lm(MyCarModel, newdata, interval = "prediction", level = 0.9)

## fit lwr upr  
## 1 37.10369 26.51706 47.69031

1. Predicted value for price of a car that is five years old
2. 90% confidence interval for the *mean price* of a car at this age is 37.104 thousand dollars and we are 90% confident that the mean price of a car of this model that is 5 years old falls between 36.413 and 37.794 thousand dollars.
3. 90% prediction interval for the price of an *individual* car of this age falls between 26.517 and 47.690 thousand dollars. This means we are 90% confident that the price of an individual car falls between 26.517 and 47.690 thousand dollars.

# Discussion

According to your model, is there an age at which the car should be free? If so, find out what this age is and comment on what the ‘free car phenomenon’ says about the appropriateness of your model.

63.0777/5.1948

## [1] 12.14247

0 = -5.1948(age) + 63.0777 age = 12.142

At approximately at 12.142 years the car should be free according to my model. This shows that my model is only useful for certain ages and after a certain age the model will start making no sense. Cars do not accurately follow the rate of depreciation in this model.