# STA 141A Final Project on Laptop Prices

Jared Duong, Brandon Sparks, and Jisung Park

December 15, 2023

## Contributions

Jared Duong: R coding
Brandong Sparks: R coding
Jisung Park: Written analysis

## Introduction

In today's world of technology, new devices are constantly being invented and revamped. However, every time you go to purchase some new device, whether it's a phone or computer, prices seem to have skyrocketed once again. You begin to wonder if there really are certain aspects or qualities within the device that seem to warrant such costly prices. Thus, we decided to focus on laptops and its prices by examining different specs/variables such as brand name, screen size, amount of memory(hard disk size), type of graphics card, and rating.

## Question of Interest

***What variable from the data set is best suited to explain the price of a laptop?***

## Dataset

Looking through Kaggle, we found a fairly recent and large dataset(October 2023 with approximately 4,500 data points) that lists not only the prices of laptops sold on the infamous Amazon.com, but also the specs of each laptop. Although there were 14 different types of variables listed in the dataset, some were unfit for our analysis so we chose to focus on these following variables:

**brand:** Quite self explanatory but corresponds to the maker/brand that made the laptop.
**screen size:** Also self explanatory but details the length of the screen.
**hard disk:** Measures the size of memory in the laptop.
**ram:** Measures the amount of ram memory(Random Access Memory which is the data stored by the computer while it's running)
**graphics:** Is in regards to whether the graphics card in the laptop is integrated or dedicated.
**rating:** Self explanatory but measures the ratings of the specific laptop sold.

## Exploratory Data Analysis

To start off, we decided to use the summary() function to examine the data first. Looking at the result, we see that many variables are non-numeric and need to be adjusted accordingly.

```
##   brand  screen_size harddisk   ram   graphics rating      price
## 1  ROKC    14 Inches  1000 GB  8 GB Integrated     NA   $589.99
## 2    HP  15.6 Inches  1000 GB 64 GB Integrated    4.5   $999.99
## 3   MSI 15.66 Inches          32 GB  Dedicated    5.0 $1,599.00
## 4 Apple  13.3 Inches   256 GB  8 GB Integrated    4.8   $689.99
## 5 Apple  15.3 Inches   256 GB  8 GB Integrated    4.8 $1,144.48
## 6  Acer  15.6 Inches   128 GB  8 GB Integrated    4.5   $299.99
```

```
##     brand           screen_size          harddisk            ram
##  Length:4446        Length:4446        Length:4446        Length:4446
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    graphics            rating         price
##  Length:4446        Min.   :1.000   Length:4446
##  Class :character   1st Qu.:4.000   Class :character
##  Mode  :character   Median :4.500   Mode  :character
##                     Mean   :4.087
##                     3rd Qu.:5.000
##                     Max.   :5.000
##                     NA's   :2272
```

One of the first steps to proceed with the analysis is to clean the data. In order to start, we need to make sure that all of our variables that are strings into numeric data. It is also imperative to keep the units of the values the same.For example, some values of the hard disk that were measured in terabytes had to be converted into gigabytes. Another change needed in the variables was in regards to the graphics since they are listed as dedicated or integrated, they needed to be in numeric value and so assigning binary values seemed valid.

## Parameter Selection and Model Fitting

After careful consideration, the approach we came up with was to initially create a full linear model and then utilize it to then create forward and backward step models. The forward and backward models seemed appropriate since it is a relatively simple model to interpret and it is objective, reproducible, and able to be transformed if needed.

```
## [1] "AIC of forward model =  12487.0842118767"
```

```
## [1] "AIC of backward model=  12485.5908845347"
```

```
##
## Call:
## lm(formula = price ~ brand + screen_size + harddisk + graphicsfixed +
##     rating, data = computers)
```

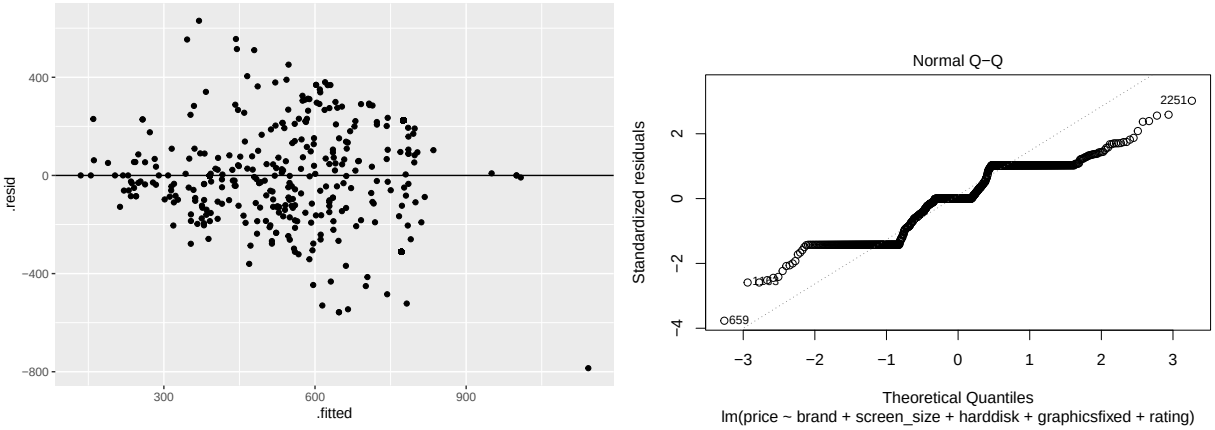
```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -785.5 -168.8   -1.3  224.5  630.2
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        -2.756e+02  1.418e+02  -1.943  0.05234 .
## brandalienware                      6.039e+02  2.234e+02   2.703  0.00701 **
## brandapple                          4.123e+02  1.313e+02   3.141  0.00174 **
## brandasus                           1.345e+02  4.700e+01   2.861  0.00433 **
## brandawow                           3.442e+01  2.228e+02   0.154  0.87728
## brandcarlisle foodservice products -1.149e+02  2.223e+02  -0.517  0.60552
## branddell                           2.397e+02  4.285e+01   5.594 2.96e-08 ***
## brandgateway                       -8.615e+01  2.224e+02  -0.387  0.69851
## brandgigabyte                       2.310e+02  1.611e+02   1.434  0.15194
## brandgoldengulf                     5.879e+01  2.241e+02   0.262  0.79311
## brandhp                             1.130e+02  4.315e+01   2.619  0.00898 **
## brandiview                         -9.734e+01  1.585e+02  -0.614  0.53935
## brandlatitude                      -4.965e+00  2.218e+02  -0.022  0.98214
## brandlenovo                         7.383e+01  4.207e+01   1.755  0.07963 .
## brandlg                             5.699e+02  2.215e+02   2.573  0.01023 *
## brandmicrosoft                      3.203e+02  1.155e+02   2.773  0.00567 **
## brandmsi                            2.468e+02  1.164e+02   2.120  0.03429 *
## brandonn                           -1.092e+02  2.215e+02  -0.493  0.62220
## brandpanasonic                      7.055e+01  8.940e+01   0.789  0.43024
## brandquality refurbished computers -1.708e+02  1.589e+02  -1.075  0.28262
## brandrokc                          -3.359e+02  5.113e+01  -6.569 8.64e-11 ***
## brandsamsung                       -1.649e+00  8.196e+01  -0.020  0.98395
## brandtocosy                        -5.528e+01  2.229e+02  -0.248  0.80425
## brandtoughbook                      7.202e+02  1.625e+02   4.432 1.05e-05 ***
## screen_size                         3.364e+01  8.059e+00   4.174 3.29e-05 ***
## harddisk                            2.834e-04  3.544e-05   7.997 3.96e-15 ***
## graphicsfixedIntegrated            -6.407e+01  3.340e+01  -1.918  0.05539 .
## rating                              4.312e+01  1.869e+01   2.307  0.02131 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 218.7 on 887 degrees of freedom
## Multiple R-squared:  0.4183, Adjusted R-squared:  0.4006
## F-statistic: 23.62 on 27 and 887 DF,  p-value: < 2.2e-16

##                   GVIF Df GVIF^(1/(2*Df))
## brand         6.967245 23        1.043104
## screen_size   1.644935  1        1.282550
## harddisk      3.989103  1        1.997274
## graphicsfixed 1.307746  1        1.143567
## rating        1.700522  1        1.304041
```

Looking at the results, the backwards model was a better fit for our analysis since the AIC(Akaike Information Criteria) was slightly lower(the AIC of the forwards model was 12487.08 while the AIC for the backwards model was 12485.59). This finding is reassuring because the backwards model is usually a more reliable model than the forward model as the forward can fail to recognize decent combinations of predictor variables while the backwards model can catch better predictors as it eliminates certain variables whilst testing.The

Variance Inflation Factor(VIF) is also calculated to check the multicollinearity of the variables.Although the GVIF values for brand and hard disk stray far from 1, the adjusted GVIF value which takes into consideration the degrees of freedom imply that all variables are low in multicollinearity resulting in accurate results.

## Residual Analysis

Our next step is to verify if the model is acceptable. In order to test this, residual analysis is used. By using the function ggplot() and plot(), we can see two plots(one that is residuals vs fits and the other which is a qqplot).



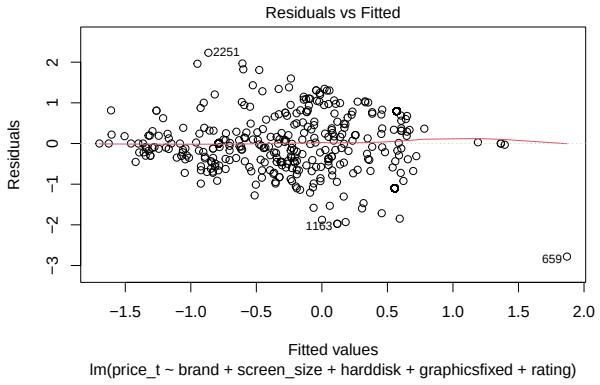

Looking at the results, it's clear there's a huge flaw since the data is extremely scattered in the residuals vs fits plot and the qqplot is not normal since the line is not straight and rather bends at six points. Based on these results, it's evident that transformations need to be made.

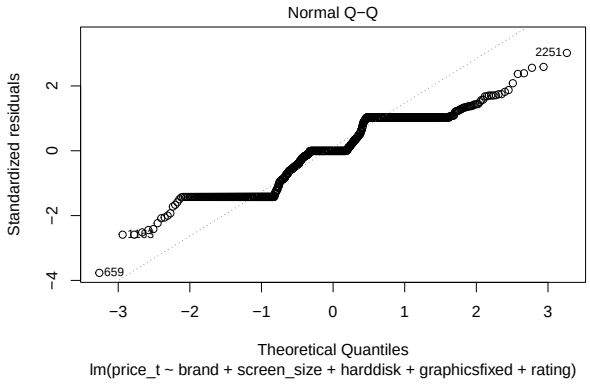
## Standardize the Price

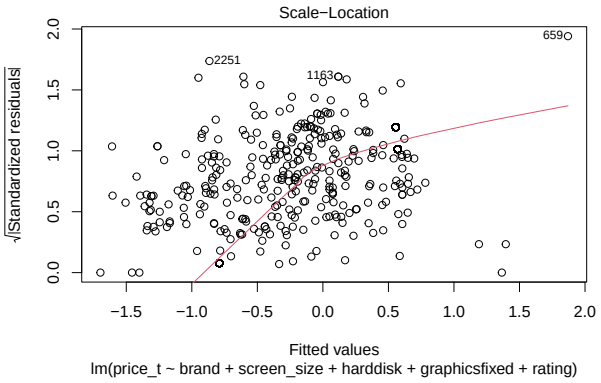
The first method to allow the data to be read more fluently and accurate is to standardize the price. This method of standardizing is effective as it allows a much better comparison of two variables while making the data internally consistent. Also, based on the data, standardization was imperative especially because the price of some laptops were as low as $70 while some were higher than $1000.

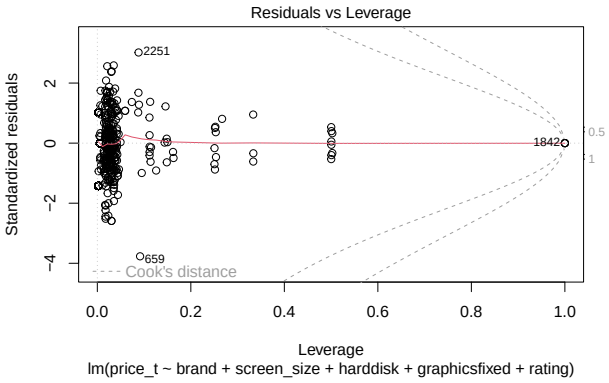
```
## [1] "AIC =  2157.97347703561"

##
## Call:
## lm(formula = price_t ~ brand + screen_size + harddisk + graphicsfixed +
##     rating, data = computers)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7809 -0.5976 -0.0046  0.7947  2.2313
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -3.151e+00  5.022e-01  -6.274 5.50e-10 ***
## brandalienware                  2.138e+00  7.910e-01   2.703  0.00701 **
## brandapple                      1.460e+00  4.647e-01   3.141  0.00174 **
## brandasus                       4.761e-01  1.664e-01   2.861  0.00433 **
## brandawow                       1.218e-01  7.889e-01   0.154  0.87728
```

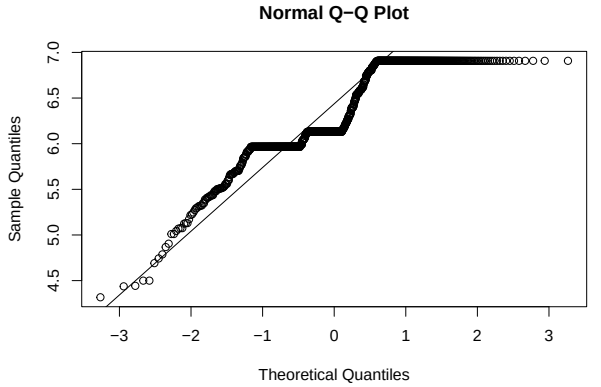
```
## brandcarlisle foodservice products -4.067e-01  7.872e-01  -0.517  0.60552
## branddell                           8.486e-01  1.517e-01   5.594 2.96e-08 ***
## brandgateway                       -3.050e-01  7.872e-01  -0.387  0.69851
## brandgigabyte                       8.179e-01  5.704e-01   1.434  0.15194
## brandgoldengulf                     2.081e-01  7.934e-01   0.262  0.79311
## brandhp                             4.001e-01  1.528e-01   2.619  0.00898 **
## brandiview                         -3.446e-01  5.613e-01  -0.614  0.53935
## brandlatitude                      -1.758e-02  7.852e-01  -0.022  0.98214
## brandlenovo                         2.614e-01  1.490e-01   1.755  0.07963 .
## brandlg                             2.018e+00  7.840e-01   2.573  0.01023 *
## brandmicrosoft                      1.134e+00  4.088e-01   2.773  0.00567 **
## brandmsi                            8.737e-01  4.122e-01   2.120  0.03429 *
## brandonn                           -3.865e-01  7.841e-01  -0.493  0.62220
## brandpanasonic                      2.498e-01  3.165e-01   0.789  0.43024
## brandquality refurbished computers -6.048e-01  5.626e-01  -1.075  0.28262
## brandrokc                          -1.189e+00  1.810e-01  -6.569 8.64e-11 ***
## brandsamsung                       -5.838e-03  2.902e-01  -0.020  0.98395
## brandtocosy                        -1.957e-01  7.893e-01  -0.248  0.80425
## brandtoughbook                      2.550e+00  5.754e-01   4.432 1.05e-05 ***
## screen_size                         1.191e-01  2.853e-02   4.174 3.29e-05 ***
## harddisk                            1.003e-06  1.255e-07   7.997 3.96e-15 ***
## graphicsfixedIntegrated            -2.268e-01  1.182e-01  -1.918  0.05539 .
## rating                              1.527e-01  6.619e-02   2.307  0.02131 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7742 on 887 degrees of freedom
## Multiple R-squared:  0.4183, Adjusted R-squared:  0.4006
## F-statistic: 23.62 on 27 and 887 DF,  p-value: < 2.2e-16
```






```
##                    GVIF Df GVIF^(1/(2*Df))
## brand          6.967245 23        1.043104
## screen_size    1.644935  1        1.282550
## harddisk       3.989103  1        1.997274
## graphicsfixed 1.307746  1        1.143567
## rating         1.700522  1        1.304041
```

By examining the results, one huge improvement that is noticed is with AIC as it lowered all the way from 12485.59 to 2157.963. However, the plots still look far from ideal and the only slight improvement was that the residuals vs fitted plot seemed to have gotten slightly more normal although the qqplot still looks like a disaster. The VIF values(adjusted with degrees of freedom) are still the same and show low multicollinearity.

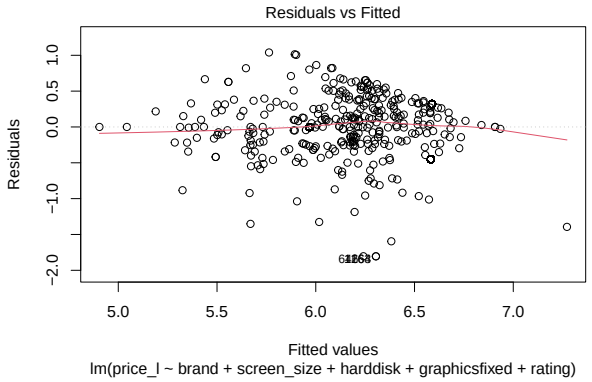
## Log Transformation

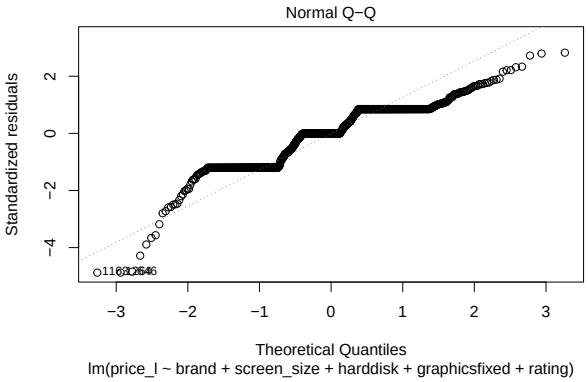
The last transformation we tried was a log transformation since it is apparently one of the most popular transformations to convert the data to normality.

```
## [1] "AIC =  835.695509632525"
```

```
##
## Call:
## lm(formula = price_l ~ brand + screen_size + harddisk + graphicsfixed +
##     rating, data = computers)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.80484 -0.30588 -0.00215  0.31949  1.03911
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       4.220e+00  2.438e-01  17.306  < 2e-16 ***
## brandalienware                    9.619e-01  3.841e-01   2.505 0.012438 *
## brandapple                        8.699e-01  2.256e-01   3.856 0.000124 ***
## brandasus                         2.397e-01  8.080e-02   2.966 0.003093 **
## brandawow                         1.642e-01  3.830e-01   0.429 0.668194
## brandcarlisle foodservice products -6.024e-01  3.822e-01  -1.576 0.115354
## branddell                         3.998e-01  7.365e-02   5.429 7.32e-08 ***
## brandgateway                      -4.520e-01  3.822e-01  -1.183 0.237248
## brandgigabyte                     4.282e-01  2.769e-01   1.546 0.122384
## brandgoldengulf                   8.556e-02  3.852e-01   0.222 0.824264
## brandhp                           1.820e-01  7.418e-02   2.453 0.014352 *
## brandiview                        -3.599e-01  2.725e-01  -1.321 0.186908
## brandlatitude                     2.600e-03  3.812e-01   0.007 0.994560
## brandlenovo                       1.689e-01  7.232e-02   2.336 0.019718 *
## brandlg                           9.193e-01  3.807e-01   2.415 0.015934 *
## brandmicrosoft                    7.584e-01  1.985e-01   3.820 0.000143 ***
## brandmsi                          4.821e-01  2.001e-01   2.409 0.016199 *
## brandonn                          -3.768e-01  3.807e-01  -0.990 0.322500
## brandpanasonic                    2.670e-01  1.537e-01   1.738 0.082605 .
## brandquality refurbished computers -6.312e-01  2.731e-01  -2.311 0.021066 *
## brandrokc                         -5.462e-01  8.789e-02  -6.214 7.92e-10 ***
## brandsamsung                      -1.277e-01  1.409e-01  -0.907 0.364771
## brandtocosy                       -1.191e-01  3.832e-01  -0.311 0.756129
## brandtoughbook                    1.429e+00  2.793e-01   5.114 3.86e-07 ***
## screen_size                       9.813e-02  1.385e-02   7.084 2.85e-12 ***
## harddisk                          5.231e-07  6.091e-08   8.588  < 2e-16 ***
## graphicsfixedIntegrated           -1.252e-01  5.741e-02  -2.180 0.029522 *
## rating                            5.732e-02  3.213e-02   1.784 0.074834 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3759 on 887 degrees of freedom
## Multiple R-squared:  0.4521, Adjusted R-squared:  0.4354
## F-statistic: 27.11 on 27 and 887 DF,  p-value: < 2.2e-16
```




```
##                   GVIF Df GVIF^(1/(2*Df))
## brand         6.967245 23        1.043104
## screen_size   1.644935  1        1.282550
## harddisk      3.989103  1        1.997274
## graphicsfixed 1.307746  1        1.143567
## rating        1.700522  1        1.304041
```

The log transformation appears to be quite successful as it once again lowers AIC all the way down to 835.6955. Also, the qqplot looks much, much better as one could arguably say that the plot somewhat matches the line of best fit. At this point, it is possible to utilize this transformed model to predict which variables correspond to price the best. The vif() numbers still remain same and yet again, there is low multicollinearity which is ideal.

## Conclusion

Based on the log transformation model, the summary() function depicts the p-value of each variable and the best predictor variables for price are the statistically significant ones(have *** next to them which implies their p-value is extremely close to 0). Thus, we can conclude that the best predictors are screen size and hard disk size along with some specific brands such as Apple, Dell, Microsoft, ROKC, and Toughbook.

During this report, we went through numerous steps to finally reach a model(log transformation) that was able to moderately analyze the best predictor for price. With this model, we were able to assess the best predictors which were brand, screen size, and hard disk capacity. Our focus was heavily on AIC which is a good determinant on how reliable a model is. Unfortunately, after some research online it appears that many statistician are opposed to the forward and backwards stepwise model as they doubt its accuracy. As a result, if this project were to be redone, a different model may have ended with more accurate predictors.

In conclusion, from a costumer's point of view, this project helped to shed light on what factors are important when buying a new laptop as they are likely the variable that affects its price. There is no surprise with these findings because it only makes sense that these specifications of the laptop allow its price to fluctuate up or down in the end.

## Appendix

```
library(MASS)
library(tidyverse)
library(caret)
library(leaps)
library(PerformanceAnalytics)
library(car)
computers <- read.csv('amazon_laptop_prices_v01.csv')
#performance <- read.csv('chip_dataset.csv')
computers <- computers %>%
  dplyr::select(c(brand,screen_size,harddisk,ram,graphics,rating,price))
head(computers)
summary(computers)
###convert data to usable datatypes
computers$brand <-tolower(computers$brand)
computers['screen_size']<-lapply(computers['screen_size'],function(data){as.numeric(substring(data,1,nch
computers['ram']<-lapply(computers['screen_size'],function(data){as.numeric(substring(data,1,nchar(data
computers['price']<-lapply(computers['price'],function(data){as.numeric(substring(data,2,nchar(data)))}
dim(computers)
computers['harddisk'] <- lapply(computers['harddisk'],function(data){as.numeric(substring(data,1,nchar(d

###converting the terabytes into gigabytes
computers[complete.cases(computers$harddisk < 10),'harddisk'] <- computers[complete.cases(computers$hard
naharddisk <- computers[is.na(computers$harddisk), ]
harddisk <- computers[-is.na(computers$harddisk), ]

#t.test(harddisk$price,naharddisk$price)
computers <- computers[complete.cases(computers),]
computers <- computers %>% #turning graphics into a binary variable
  mutate(graphicsfixed = ifelse(grepl('Integrated',graphics),'Integrated','Dedicated'))
fullmodel <- lm(price~ brand + screen_size  + harddisk  + ram  + graphicsfixed + rating,data = computers
step.modelf <- stepAIC(fullmodel, direction = "forward", trace = FALSE)
step.modelb <- stepAIC(fullmodel, direction = "backward", trace = FALSE)
faic <- AIC(step.modelf)
baic <- AIC(step.modelb)
print(paste("AIC of forward model = ", faic))
print(paste("AIC of backward model= ", baic))
summary(step.modelb)
vif(step.modelb)
ggplot(step.modelb, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0)
plot(step.modelb, which = 2) # should transform
computers$price_t <- scale(computers$price)
tranmodel <- lm(price_t~ brand + screen_size  + harddisk  + ram  + graphicsfixed + rating,data = compute
step.modelbt <- stepAIC(tranmodel, direction = "backward", trace = FALSE)
btaic <- AIC(step.modelbt)
print(paste("AIC = ", btaic))
summary(step.modelbt)
plot(step.modelbt)
{qqnorm(log(computers$price))
  qqline(log(computers$price))}
```

```r
vif(step.modelbt)
computers$price_l <- log(computers$price)
logmodel <- lm(price_l~ brand + screen_size  + harddisk  + ram  + graphicsfixed + rating,data = computer
step.modelbl <- stepAIC(logmodel, direction = "backward", trace = FALSE)
blaic <- AIC(step.modelbl)
print(paste("AIC = ", blaic))
summary(step.modelbl)
plot(step.modelbl, which = 1)
plot(step.modelbl, which = 2)
vif(step.modelbl)
pred_log <- predict(step.modelbl, computers[,c("brand", "screen_size", "harddisk", "ram", "graphicsfixed
pred_exp <- exp(pred_log)
head(pred_exp)
```