Analysis of retail price of laptops, using R

Nabhan Kazi

Introduction

Almost every household owns at least one laptop nowadays. Laptops are most common among students, who can carry it around with them easily. When it comes to buying a laptop there are many aspects that come into play. There are different specifications of a laptop that fits different people. Gamers require high end laptops with a lot of RAM, CPU power and graphic, which end up being quite high up on the price scale. Whereas, non-gamers require average specifications for their laptops, who prices are also around average. People who use their laptops just for school or office usually can get by with basic specifications, with laptops ranging from average to the lower end on the price scale. So what exactly determines a laptop's price? What makes one laptop super expensive and what makes another dirt cheap? In this study I will try to determine what some of those aspects may be. I will be looking at 210 different laptops from the BestBuy website, both Windows and Apple platforms. The goal of this study is to determine whether there is a way to predict the asking price of laptops from certain specifications.
Some of the features that I will be looking at in this study that may influence the asking price of laptops include- platform of the laptop, brand name, RAM size, storage capacity, type of storage, type of processor, number of processor cores, speed of the processor, screen size, screen resolution, type of graphics chipset and preloaded operating system.

Being a student myself, I have had owned two laptops so far. I had done quite some research both times before buying those laptops. So, from past experience, I expect that many of the variables in the study will play a significant role in determining the asking price of these laptops. Firstly, I expect Apple laptops to be higher in price than Windows laptops in general. Secondly, I expect there to be some correlation between screen size, resolution and the type of graphics with the retail price. Lastly, I

hypothesize that the higher the amount of RAM, storage capacity and processor speed, the higher the asking price will be.

Methods

Since I created the dataset myself, I made sure to include as many laptops as possible, from different brands and consisting of varieties of specifications. I did not include any of the open box or refurbished laptops, as I expected them to be lower in price than they usually are, which would have skewed my analysis. When I was making the dataset in a span of a month, I noticed that there were many laptops that went on sale during one week but then went off sale the week after, and vice versa. That is why the prices I entered for those laptops were the non-sale prices, as they tended to revert back to the original prices after a while. While creating the dataset I realized that I had to keep looking at the each laptop's description and features to check whether I had already entered it in my dataset. It ended up being very time consuming and monotonous. So I decided to make a new variable consisting of the item number of each laptop from the website. The item number of a laptop works as a fingerprint, where each item number belongs to only one specific laptop. Once I started using the item numbers, it became very easy for me to check whether I had already included a laptop in my dataset or not, by using the CTRL+F command.

In order to determine whether the retail price of laptops can be predicted from the given variables, I had to run some tests and analyses. I conducted all my tests and analyses in R, version 3.3.3, and used Rstudio as the IDE. For all my first and second stage analyses, I set alpha to be 0.05.

First and foremost, I installed the package "xlsx" and loaded it in order to be able to read in the excel file in R. Then I read in the data file into a data-frame, and checked to make sure that I had the right number of observations and variables.

In order to be able to create boxplots and scatterplots, I installed the “ggplot2” package and loaded it, which gives me access to several graphing functions.

After that, I created a function to convert my resolution variable from strings in multiplicative form to the products of the multiplication in numeric form. This will make it easier to make graphs and plots.

I started off by creating side-by-side boxplots of brands and platforms in order to examine my initial suspicion that Apple based laptops are generally more expensive than Windows based laptops. I set the platform of the laptops and the brands as the independent variable and the retail price as the dependent variable. As expected, in both boxplots, Apple laptops were higher in price range than the rest. There were some outliers in the Windows laptops, which I believe were due to the fact that some Windows gaming laptops go over and beyond on the features that they offer.

To further confirm my suspicion about the differences of prices between the two platforms, I ran a one-way ANOVA and Tukey (since significant differences exist). I found that significant differences existed only between Windows and Apple laptops, and not between Windows brands.

The above boxplots and ANOVA test made me realize that looking at Apple and Windows platforms together is going to skew my analyses. So, I decided to separate out the Windows and Apple laptops and conduct tests on them separately.

I wanted to get an initial idea of how each variable affects the retail prices, so I made plots for each variable against the retail prices, separated by platform. I set each specification as my independent variable and retail price as the dependent variable. Most of the variables like RAM, storage, processor_speed, etc. showed some positive interaction and made my list to do further analysis. Other variables however, like preloaded

operating system, graphics chipset and processor, did not seem to have any significant interaction. I realized this was because there were too many varieties of these variables. So I decided to make them more general and create two more variables, one that indicates the brand of the processor (Intel or AMD) and another that indicates the brand of graphics chipset (Intel, AMD, or NVIDIA). I decided not to use the preloaded operating system variable because it would basically have been a distinction between Windows and Apple, since almost all Windows laptops came with Windows 10, and we already know how that distinction went.

Once I determined my most correlated variables, I fitted them in a regression model and checked for significant interaction with retail price. This is also called an "analysis of covariance model". I added an interaction between the variables to see whether it altered my prediction in anyway, and then compared both models using ANOVA.

After that I added other variables to the model and checked to see if all the variables were still significant. I made a function that displayed the individual significant and non-significant variables. I removed the variables that were non-significant and came up with two final models, one for each platform.

Results and Statistical Analyses

Code:

```
>install.packages("xlsx")
>library(xlsx)
>laptops=read.xlsx("laptops.xlsx", header=T, sheetIndex=1)
>str(laptops)
```

Output:

```
'data.frame': 210 obs. of  17 variables:
 $ laptop_name     : Factor w/ 162 levels " ENVY Touchscreen Convertible Laptop",
..: 161 103 162 128 22 116 14 28 66 128 ...
 $ item_number     : num  10562713 10481595 10490599 10483535 10584954 ...
 $ brand           : Factor w/ 8 levels "Acer","Apple",..: 3 6 3 6 1 6 1 4 3 6 ..
.
 $ platform        : Factor w/ 2 levels "Mac OS","Windows": 2 2 2 2 2 2 2 2 2 2 .
..
 $ processor       : Factor w/ 76 levels " AMD A10-9600P ",..: 34 25 50 29 65 13
37 51 15 69 ...
 $ processor_type  : Factor w/ 2 levels "AMD","Intel": 2 2 2 2 2 1 2 2 1 2 ...
 $ processor_cores : num  2 2 2 2 2 4 2 4 4 4 ...
 $ processor_speed : num  2.3 1.6 2.5 2.3 2.3 1.8 2.5 2.6 2.4 1.6 ...
 $ ram             : num  8 4 8 8 8 4 12 16 12 4 ...
 $ storage         : num  1024 32 1024 1024 256 ...
 $ storage_type    : Factor w/ 4 levels "eMMC","HDD","SSD",..: 2 3 2 2 3 2 2 4 2
2 ...
 $ screen_size     : num  15.6 14 15.6 15.6 14 15.6 15 17.3 15.6 15.6 ...
 $ resolution      : Factor w/ 19 levels " 1920 x 1080 ",..: 5 5 5 5 10 5 5 10 5
7 ...
 $ graphics_chipset: Factor w/ 86 levels " Intel Core i7-7500U ",..: 24 31 56 2 3
6 7 42 68 8 25 ...
 $ graphics        : Factor w/ 3 levels "AMD","Intel",..: 2 2 3 2 2 1 2 3 1 2 ...
 $ os              : Factor w/ 33 levels " Windows 10 (64bit) ",..: 7 14 9 14 6 1
4 15 22 7 14 ...
 $ retail_price    : num  550 350 1000 700 850 ...
```

Analysis:

Everything seems to be correctly inputted. I see that I have 210 observations and 17 variables for each.

Code:

```
>install.packages("ggplot2")
>library(ggplot2)
#gets rid of the x between the two numbers and input the strings in a list
>resolution= strsplit(as.character(laptops$resolution),' x ')
#converts the strings in the list to numeric
>res=lapply(resolution,as.numeric)
#function to make the numbers in the list multiply with each other
>f=function(x){
  x[1]*x[2]
}
>res2=lapply(res,f)
#gets rid of the list
>resolution2=unlist(res2)
```

```
#displays the first six rows of the newly converted resolution column
>head(resolution2)
```

Output:

```
[1] 1049088 1049088 1049088 1049088 2073600 1049088
```

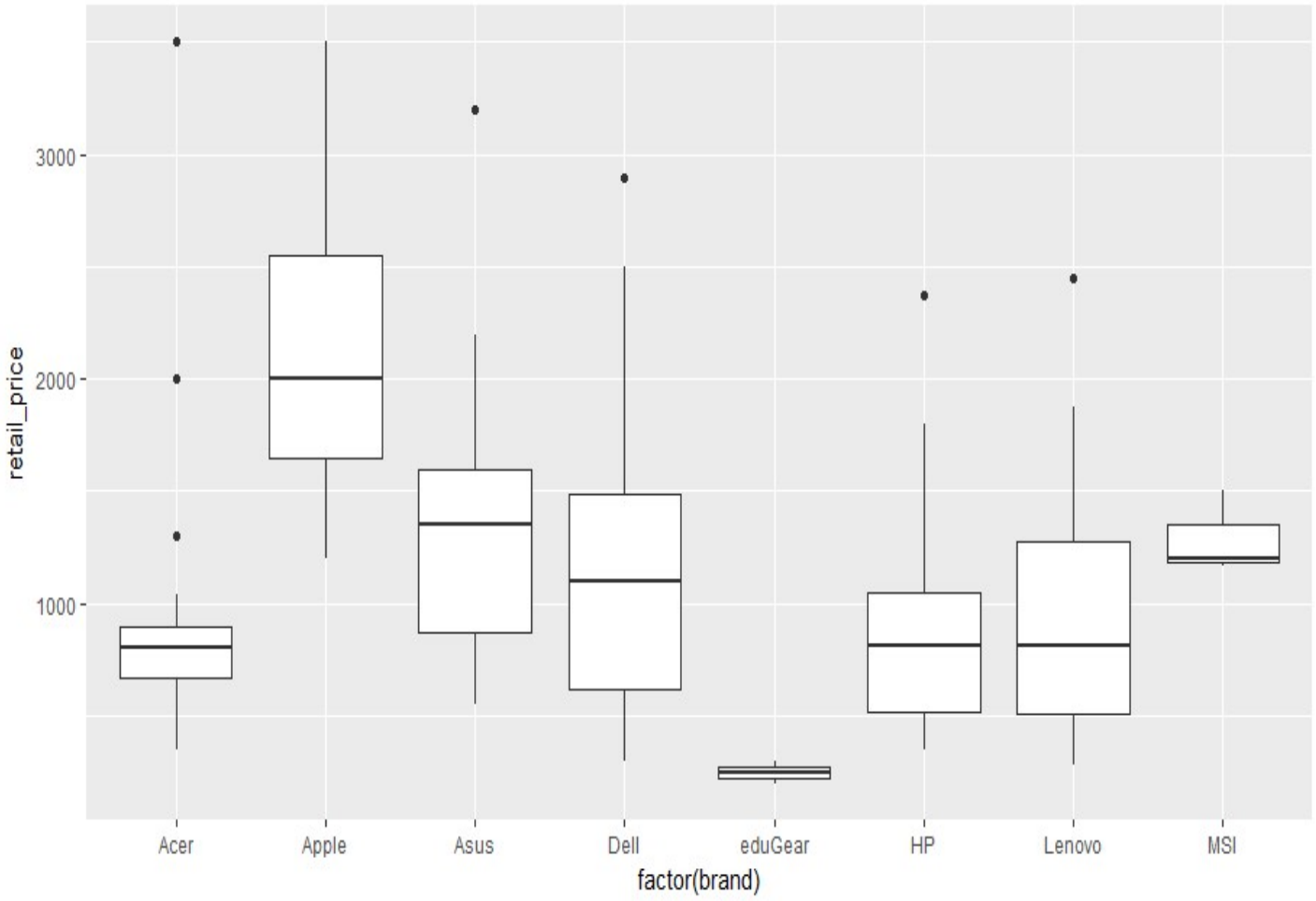
Analysis:

The new resolution variable seems to be displayed properly. The contents are numeric instead of strings.

Code:

```
>ggplot(laptops,aes(x=factor(brand),y=retail_price))+geom_boxplot()
```

Output:



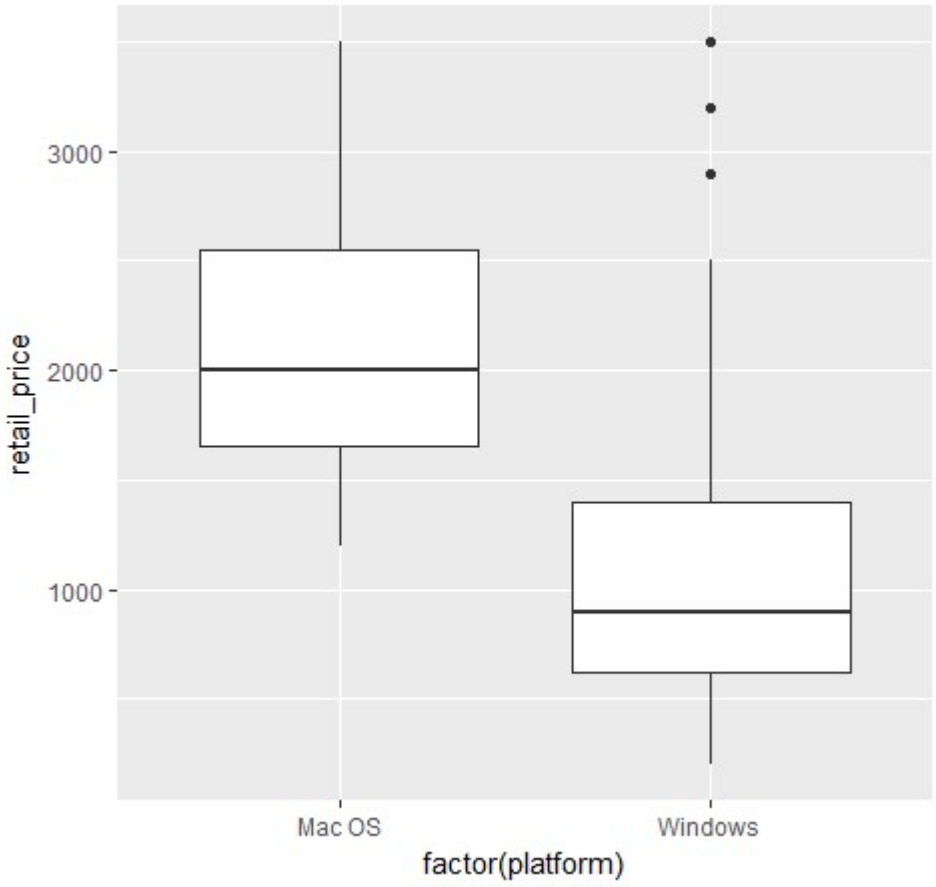
Analysis:

The boxplot shows that eduGear brand of laptops are the lowest in price, because they consist of the most basic specifications and can basically be considered of as tablets. Most of the Windows laptops have outliers due to some gaming computers having over-the-line specifications. MSI laptops are predominantly built as gaming laptops, which is why its mean is slightly higher than the other Windows laptops. The other Windows brands range from basic to professional level, as are their prices. Overall, Apple laptops seem to be the most expensive, with the highest mean among all the other brands.

Code:

```
>ggplot(laptops,aes(x=factor(platform),y=retail_price))+geom_boxplot()
```

Output:



Analysis:

This boxplot is further proof that Macs and Windows have to be dealt with separately. It shows that Macs are more expensive than windows laptops in general. The mean for Macs are more than twice as much as Windows laptops. Some outliers exist among the Windows laptops, which are due to the high-end gaming laptops.

Code:

```
>price0=aov(retail_price~brand,data=laptops)
>summary(price0)
```

Output:

```
             Df   Sum Sq Mean Sq F value Pr(>F)
brand         7 50163764 7166252   21.03 <2e-16 ***
Residuals   202 68823641  340711
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Analysis:

The P-value is $2*10^{-16}$, which is definitely less than the alpha level of 0.05. Thus, there are significant differences among the laptop brands, so running Tukey would be appropriate to determine where the differences lie.

Code:

```
>TukeyHSD(price0)
```

Output:

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = retail_price ~ brand, data = laptops)

$brand
                      diff         lwr         upr     p adj
Apple-Acer      1281.588333   845.39078 1717.785886 0.0000000
Asus-Acer        375.025598   -59.13839  809.189584 0.1459995
Dell-Acer        330.874048  -166.46768  828.215777 0.4592173
eduGear-Acer    -662.954286 -1971.54694  645.638372 0.7781280
HP-Acer          -25.753233  -471.03921  419.532745 0.9999997
```

```
Lenovo-Acer       38.066048  -431.73309  507.865189 0.9999970
MSI-Acer         377.540714  -708.58243 1463.663861 0.9632710
Asus-Apple      -906.562735 -1294.43476 -518.690712 0.0000000
Dell-Apple      -950.714286 -1408.20214 -493.226433 0.0000000
eduGear-Apple  -1944.542619 -3238.51367 -650.571567 0.0001957
HP-Apple       -1307.341566 -1707.62389 -907.059240 0.0000000
Lenovo-Apple   -1243.522286 -1670.90686 -816.137713 0.0000000
MSI-Apple       -904.047619 -1972.50905  164.413812 0.1650765
Dell-Asus        -44.151550  -499.70089  411.397786 0.9999898
eduGear-Asus   -1037.979884 -2331.26684  255.307070 0.2196907
HP-Asus         -400.778831  -798.84416   -2.713506 0.0471279
Lenovo-Asus     -336.959550  -762.26842   88.349320 0.2342560
MSI-Asus           2.515116 -1065.11773 1070.147961 1.0000000
eduGear-Dell    -993.828333 -2309.67087  322.014200 0.2914907
HP-Dell         -356.627281  -822.78866  109.534099 0.2756546
Lenovo-Dell     -292.808000  -782.43854  196.822538 0.5992650
MSI-Dell          46.666667 -1048.18050 1141.513834 1.0000000
HP-eduGear       637.201053  -659.86193 1934.264035 0.8042082
Lenovo-eduGear   701.020333  -604.66110 2006.701769 0.7226162
MSI-eduGear     1040.495000  -591.60679 2672.596794 0.5165083
Lenovo-HP         63.819281  -372.83719  500.475747 0.9998334
MSI-HP           403.293947  -668.90992 1475.497811 0.9441411
MSI-Lenovo       339.474667  -743.13919 1422.088521 0.9793143
```

Analysis:

Upon careful observation it can be seen that most of the significant differences lie among the Apple and Windows brands. The only time that a difference between Apple and a windows brand does not exist is between Apple and MSI. This is because it was previously stated that MSIs are mostly gaming laptops, so their prices are in the higher end of the scale as well. This test is a clear indication that Apples and Windows differ from each other significantly and should be analyzed separately.

Code:

```
>windows=laptops[1:169,]
>apple=laptops[169:211,]
>head(windows$platform)
>head(apple$platform)
```

Output:

```
[1] Windows Windows Windows Windows Windows Windows
Levels: Mac OS Windows
[1] Mac OS Mac OS Mac OS Mac OS Mac OS Mac OS
Levels: Mac OS Windows
```

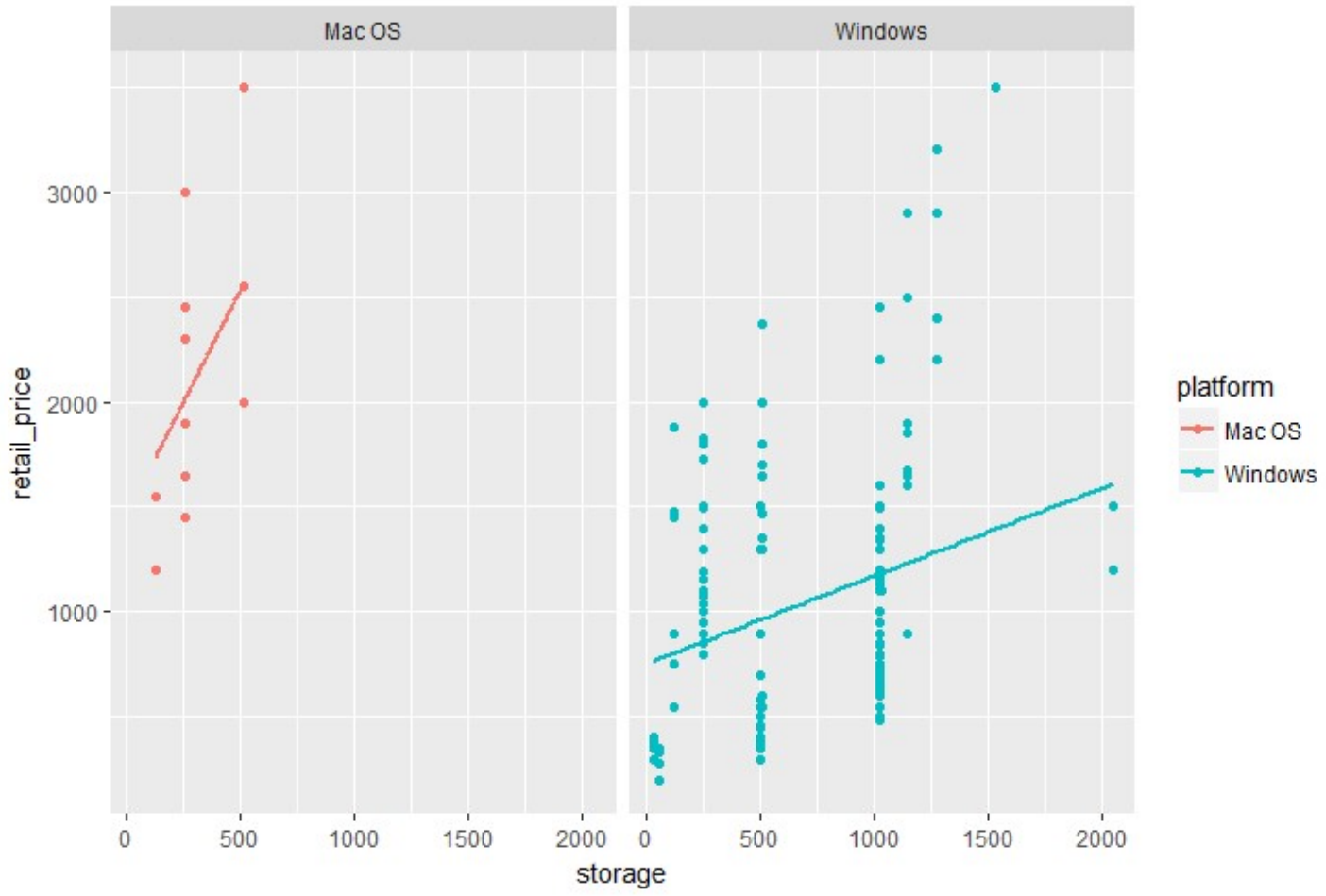
Analysis:

After all the clear indications, Windows and Apple laptops were separated. The output shows that there are two types of platforms, and the new data-frames windows consist of Windows laptops and apple consist of Apple laptops.

Code:

```
>ggplot(laptops,aes(x=storage,y=retail_price,colour=platform))+geom_point(
)+geom_smooth(method="lm",se=F)+facet_wrap(~platform)
```

Output:



Analysis:

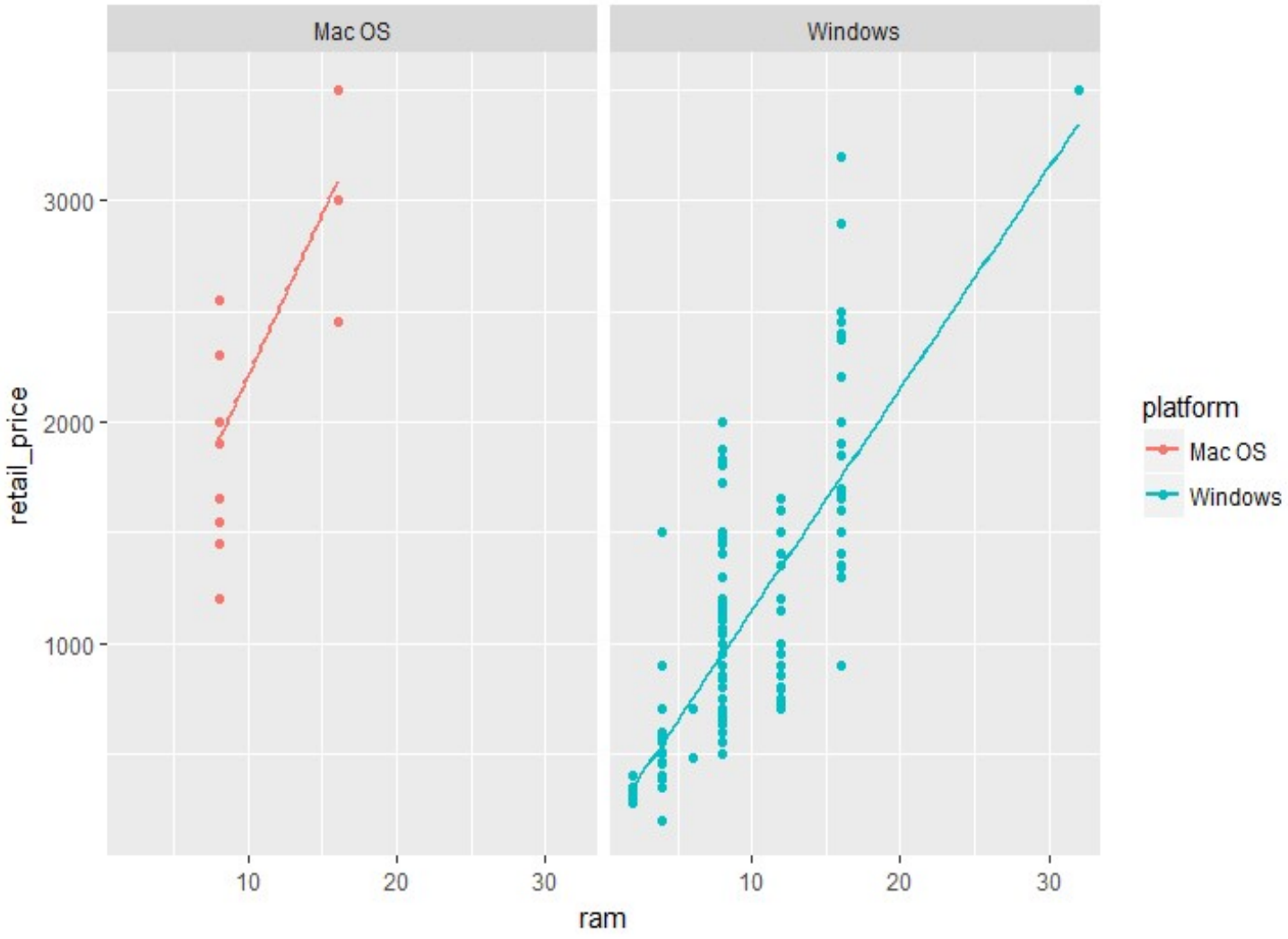
The scatterplot of storage against retail price shows us that prices for both Windows laptops and Apple laptops increase as the storage capacity increase. Macs only use SSD, which is why their capacity is in the lower

range than Windows laptops, whose capacity ranges from very low to very high. Either way, both platforms seem to have a positive linear trend, which will be analyzed further later. There does not seem to be any outliers that stand out.

Code:

```
>ggplot(laptops,aes(x=ram,y=retail_price,colour=platform))+geom_point()+geom_smooth(method="lm",se=F)+facet_wrap(~platform)
```

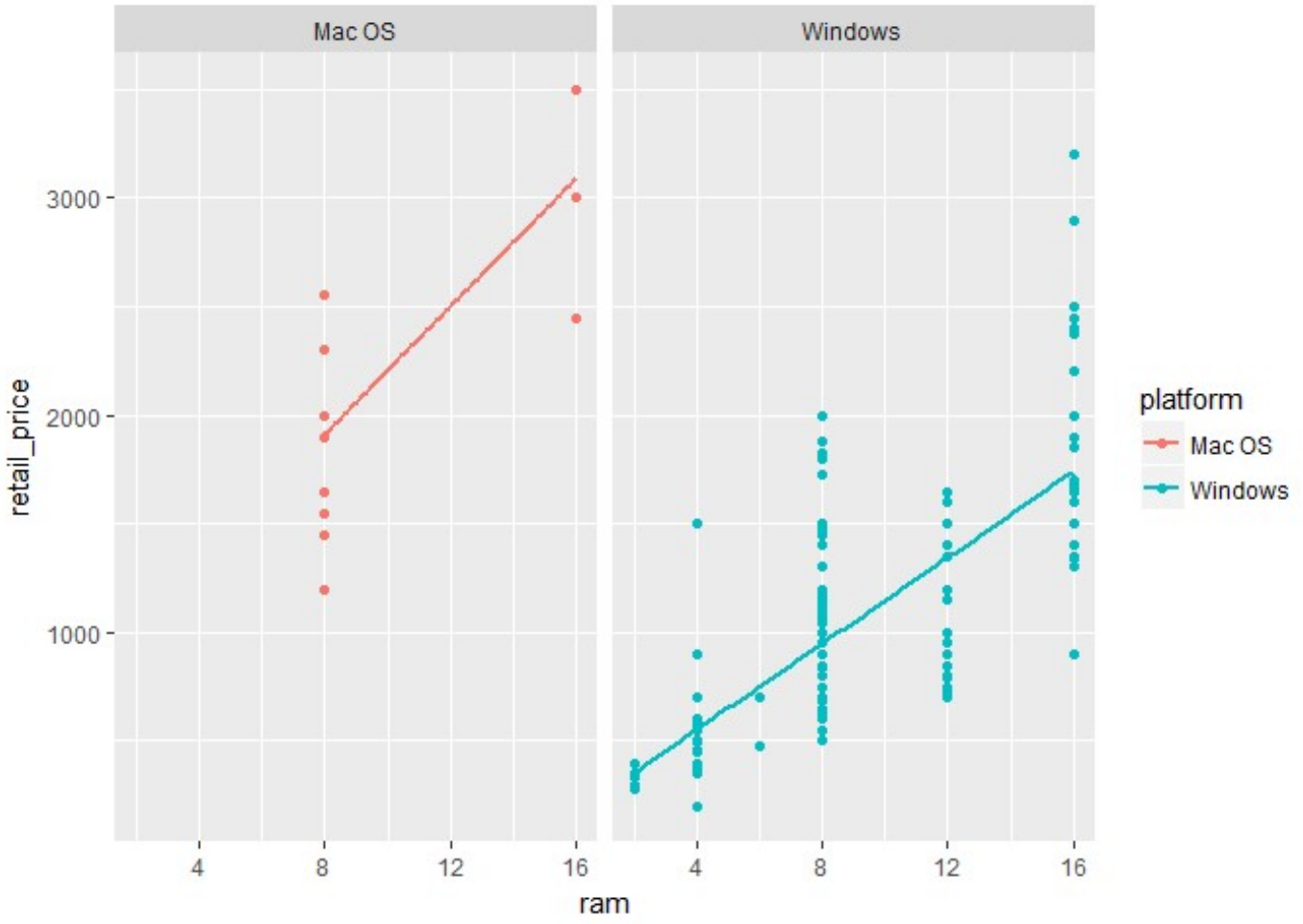
Output:



Analysis:

The scatterplot of amount of RAM against retail price shows us that prices for both Windows laptops and Apple laptops increase quite steeply as the

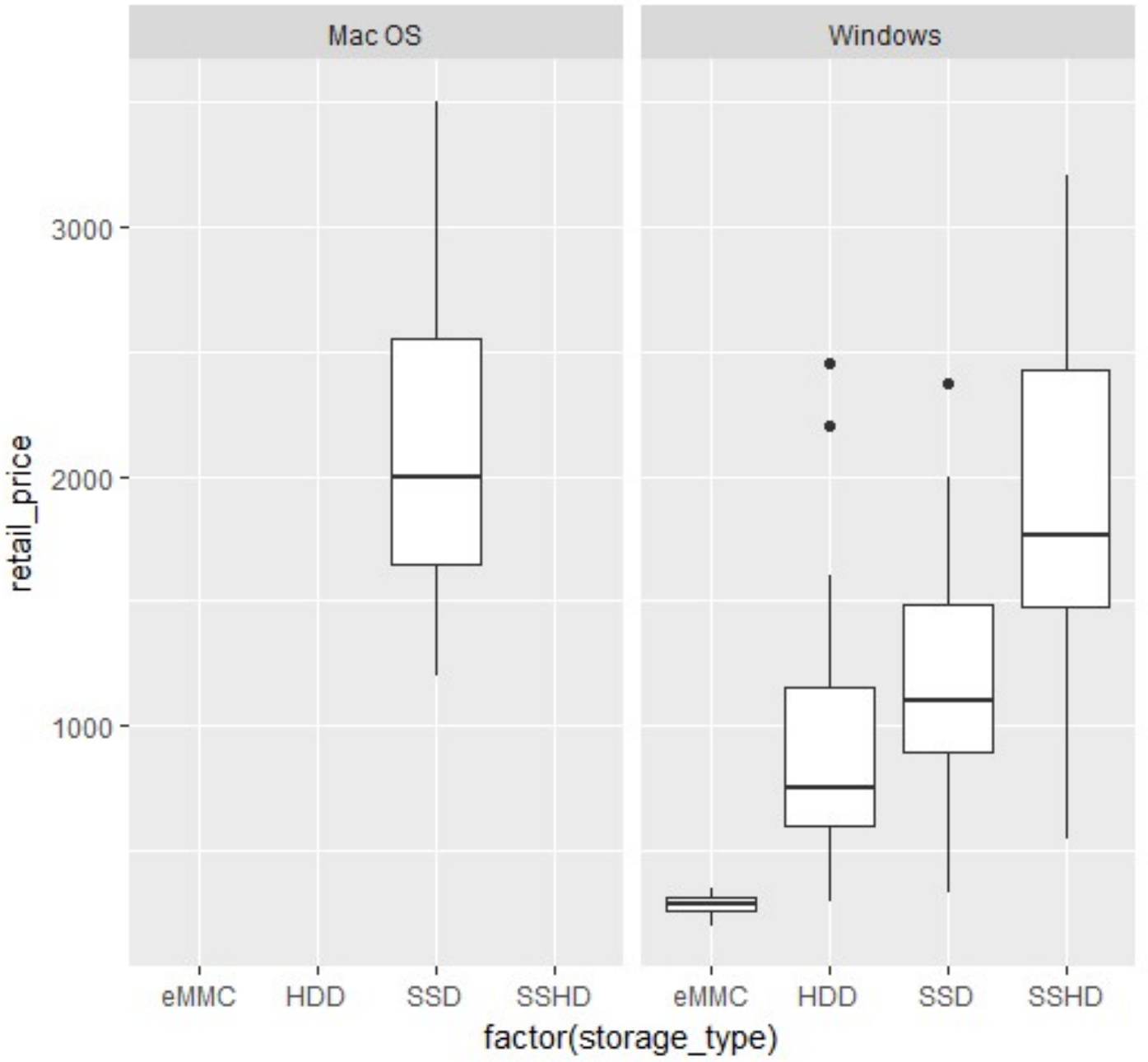
amount of RAM increases. Both platforms seem to have a positive linear trend, which will be analyzed further later. There seems to be one outlier that stands out in the Windows platform, but even without it the slope is still positive, as seen below.



Code:

```
>ggplot(laptops,aes(x=factor(storage_type),y=retail_price))+geom_boxplot()
+facet_wrap(~platform)
```

Output:



Analysis:

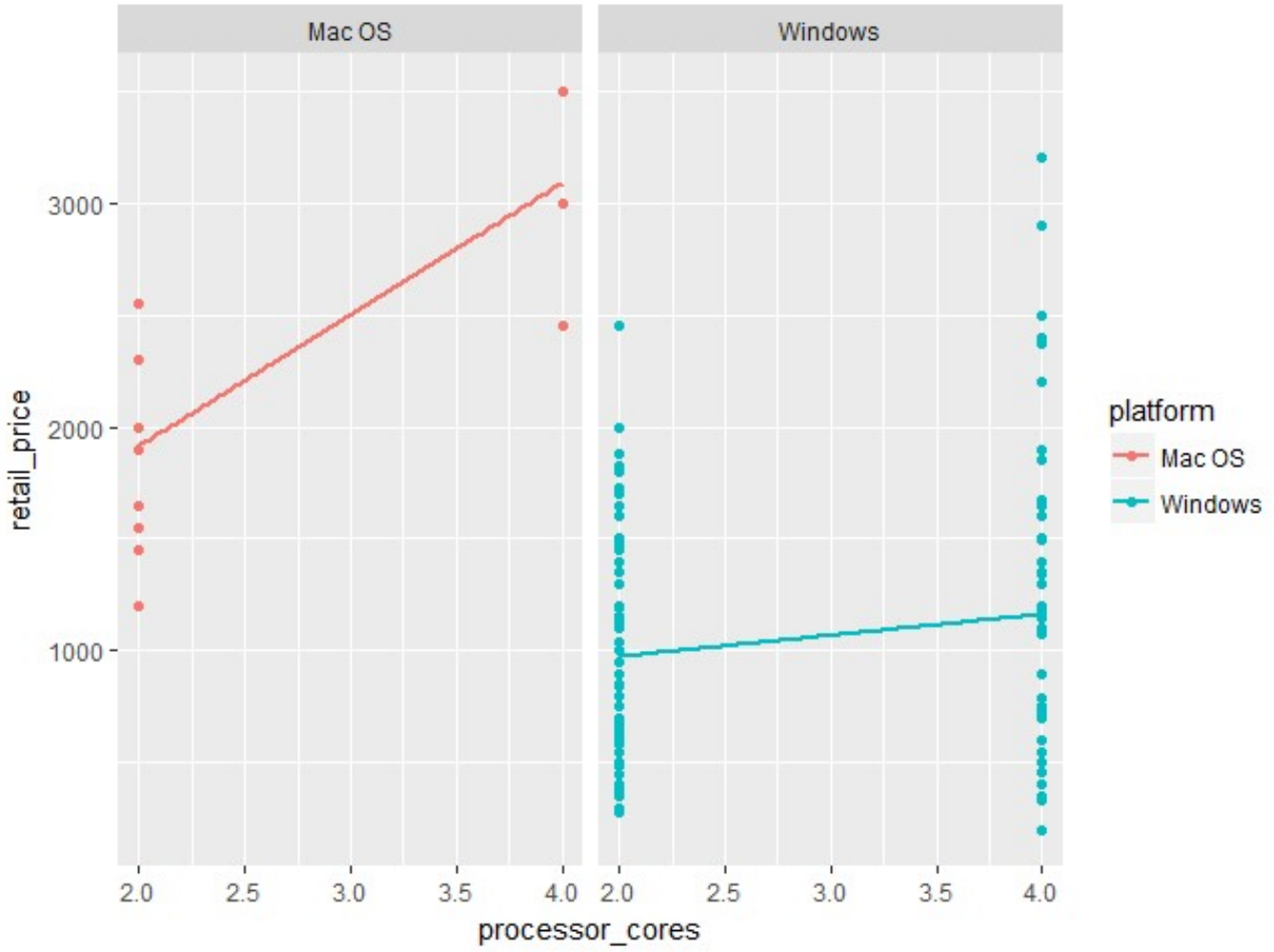
The boxplot above shows that when SSDs are compared amongst Windows and Macs, the mean price of Macs is almost twice as much. Among the Windows laptops, eMMC are the cheapest among the storage types, since they are typically low capacity flash memory storage. SSDs are slightly more expensive than HDD due to their versatility and fluidity. SSHD are understandably the

most expensive type of storage, because they contain the best of both worlds- the capacity and reliability of a HDD, and the speed and performance of a SSD. A few outliers exists among the Windows laptops, which are again due to some high-end gaming laptops.

Code:

```
>ggplot(laptops,aes(x=processor_cores,y=retail_price,colour=platform))+geo
m_point()+geom_smooth(method="lm",se=F)+facet_wrap(~platform)
```

output:



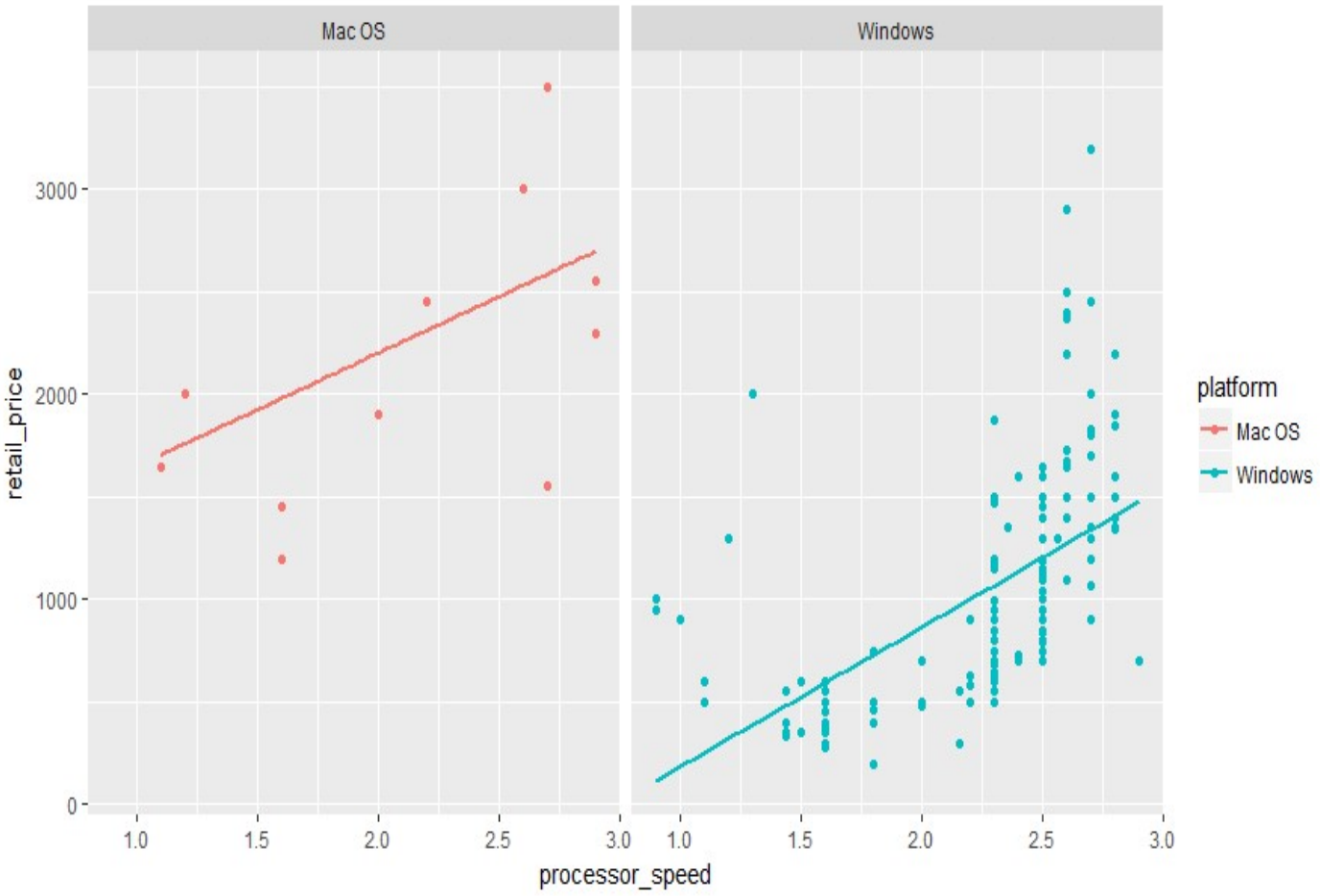
Analysis:

Macs seem to have a steeper slope than Windows as the number of processor cores increases. Windows laptops have a weak positive slope, only a slight increase as processor cores increase.

Code:

```
>ggplot(laptops,aes(x=processor_speed,y=retail_price,colour=platform))+geom_point()+geom_smooth(method="lm",se=F)+facet_wrap(~platform)
```

Output:



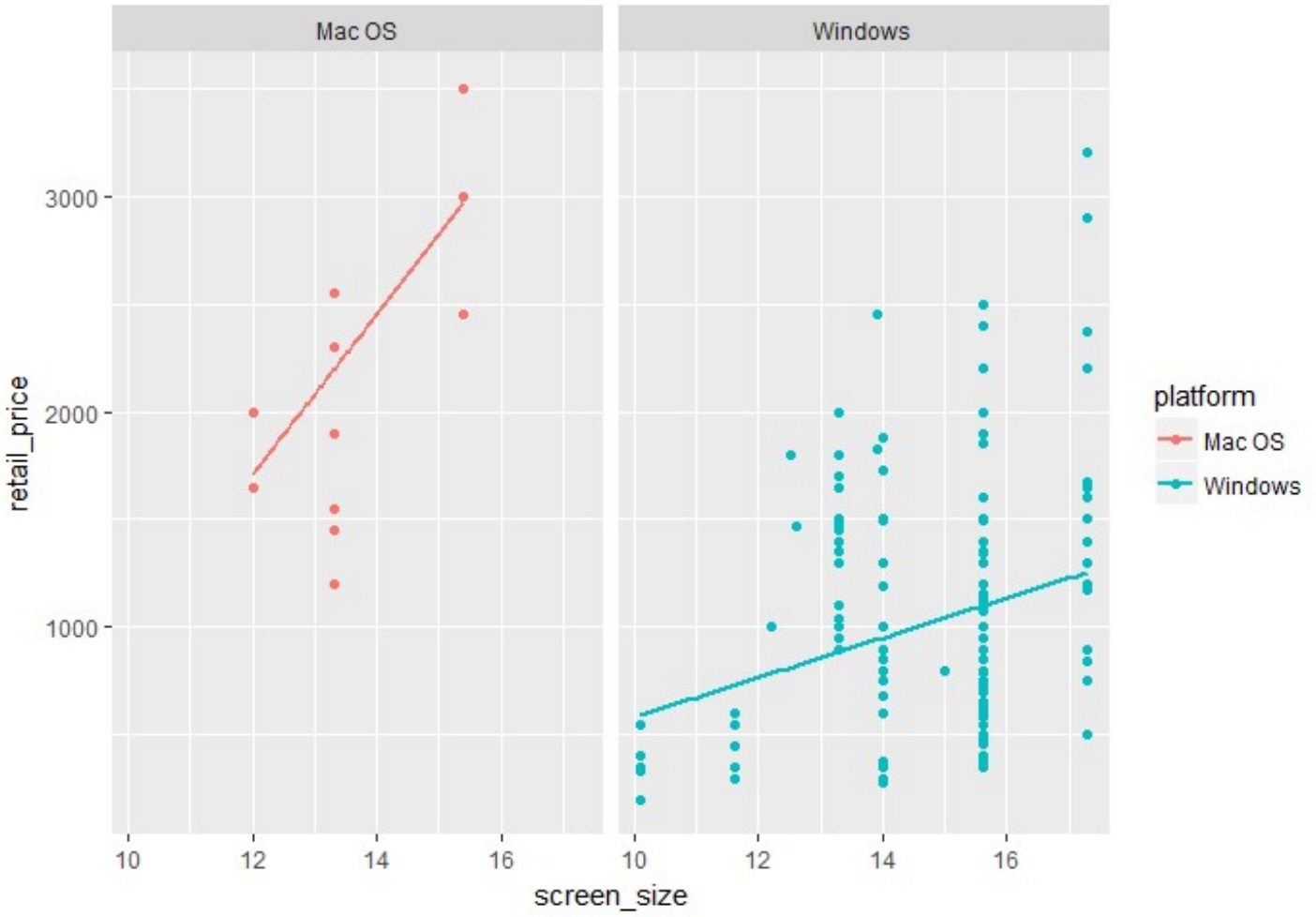
Analysis:

Both platforms seem to have almost equally positive slopes in regards to increase of the processor speed. Macs do not seem to have any noticeable outliers. Windows have some outliers, which are caused by high-end gaming laptops.

Code:

```
>ggplot(laptops,aes(x=screen_size,y=retail_price,colour=platform))+geom_point()+geom_smooth(method="lm",se=F)+facet_wrap(~platform)
```

Output:



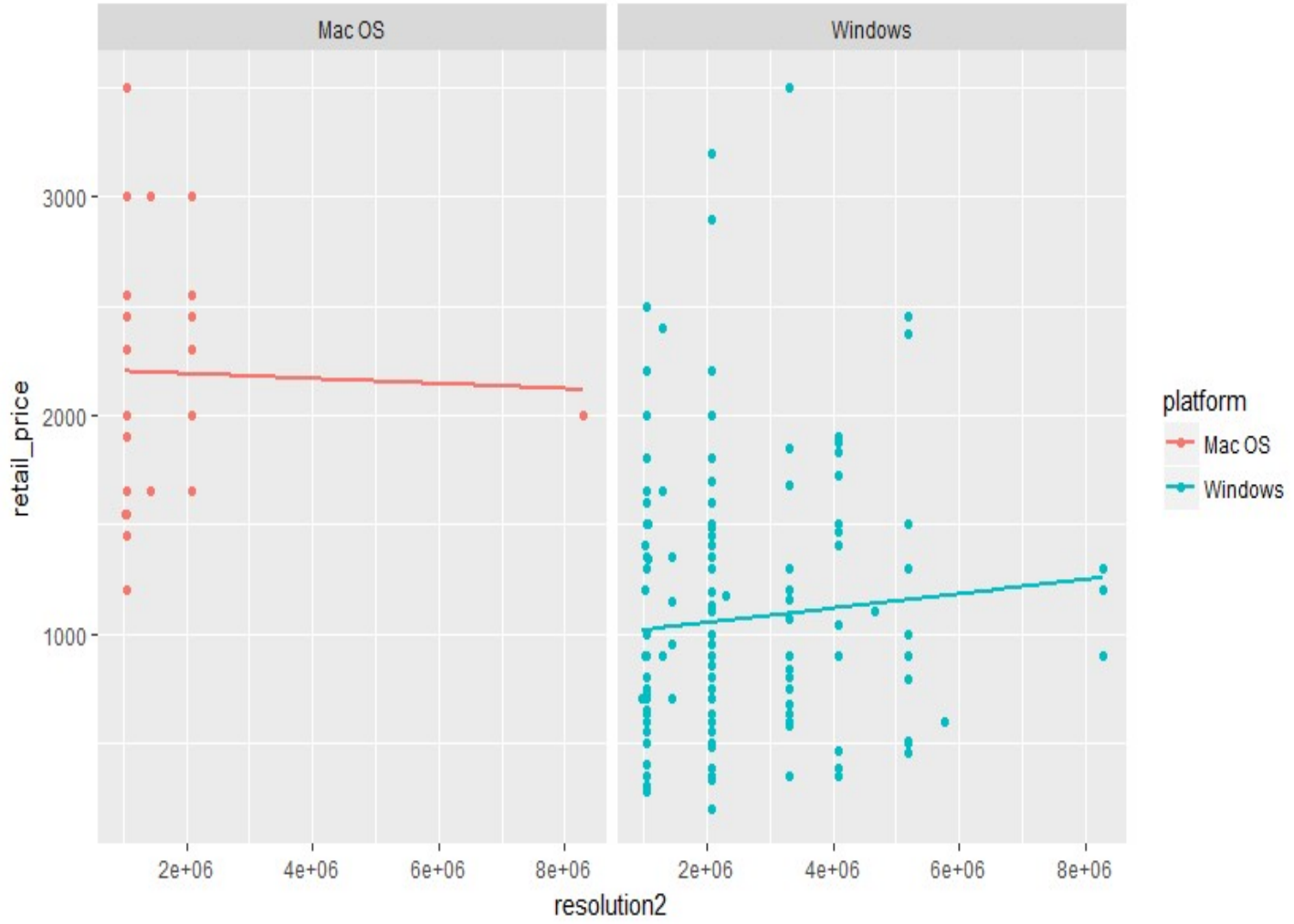
Analysis:

As screen size increases, Macs seem to have a steeper positive slope than Windows laptops. Either way, in both cases, there seem to be a positive association.

Code:

```
>ggplot(laptops,aes(x=resolution2,y=retail_price,colour=platform))+geom_point()+geom_smooth(method="lm",se=F)+facet_wrap(~platform)
```

Output:



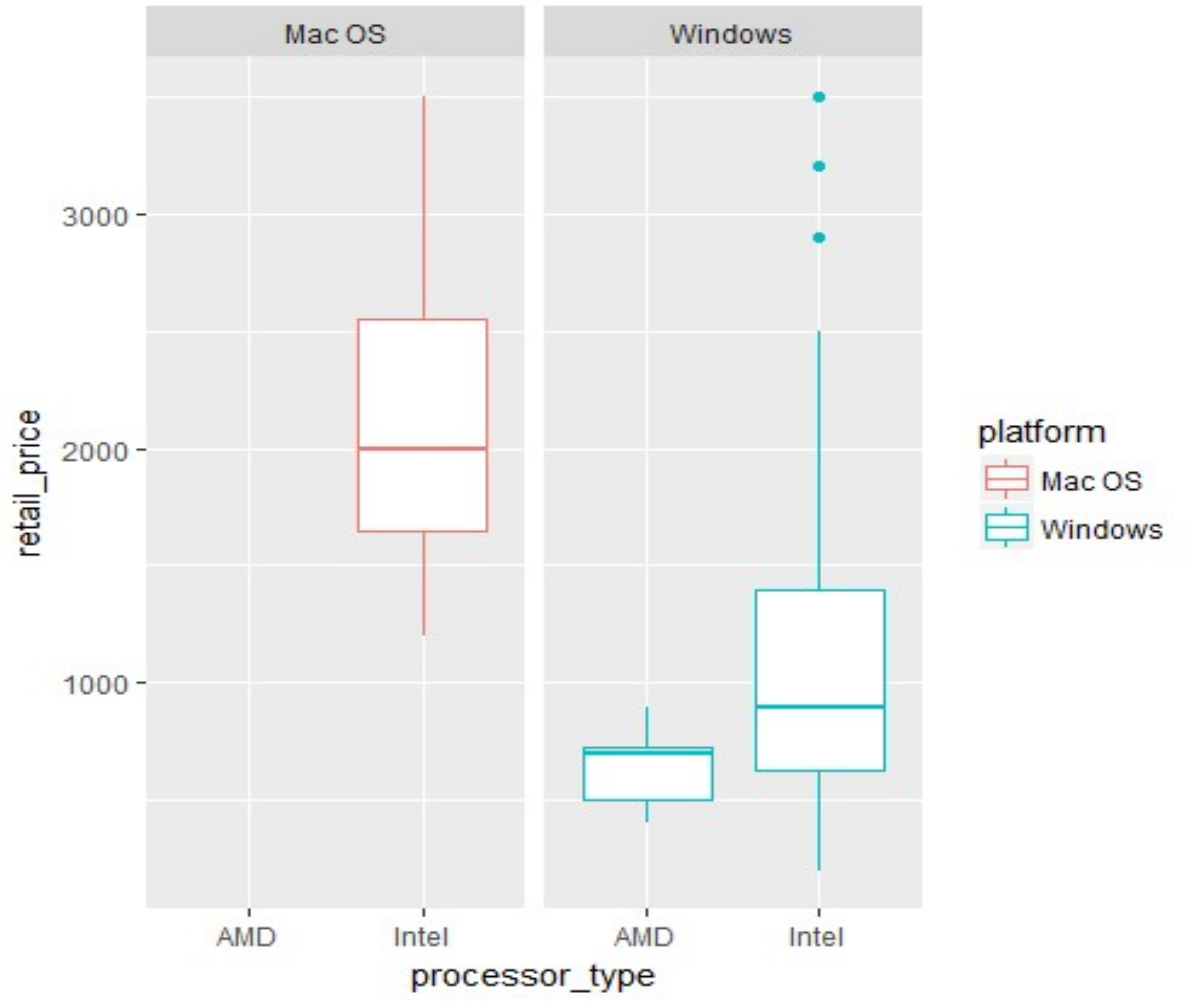
Analysis:

Surprisingly, Apple laptops have a slightly negative slope in regards to the increase in resolution. At first it was suspected that the negative slope in Macs was caused by the rightmost point in the scatterplot. However, upon careful observation of the dataset, it was determined that the rightmost point in the Mac OS scatterplot is not just one point, it is a collection of several points in the same spot. Hence, the point was not removed in order to manually alter the slope. Windows laptops seem to have a more positive slope than Macs, but still not steep enough to peak an interest for further analysis. The only thing worth noticing is the fact that Apple laptops have higher resolutions in general than Windows laptops.

Code:

```
>ggplot(laptops,aes(x=processor_type,y=retail_price,colour=platform))+geom
_boxplot()+facet_wrap(~platform)
```

Output:



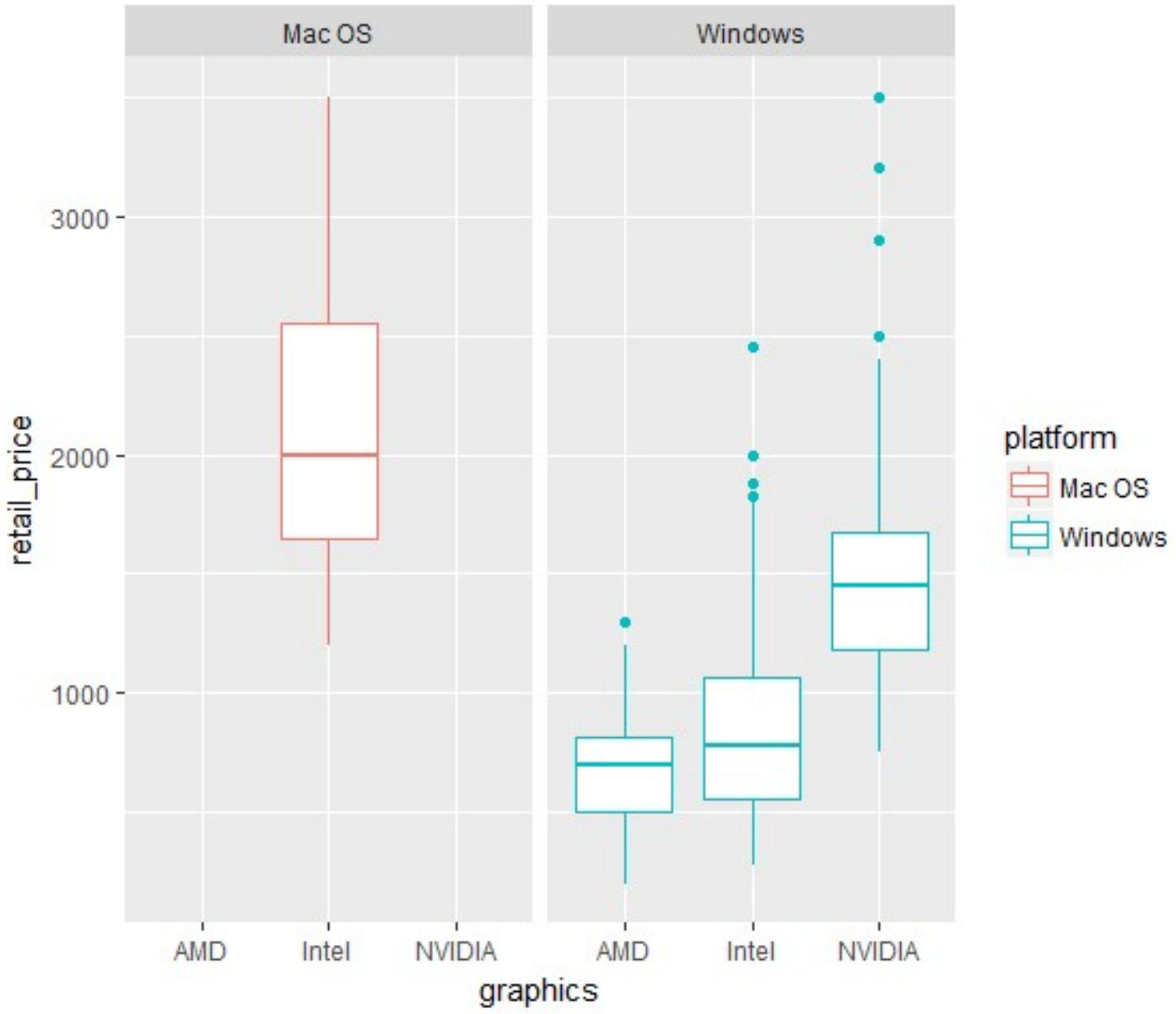
Analysis:

There is not much to talk about the Apple laptops here all the Apple laptops use Intel processors. The boxplot seems to be approximately normal with no outliers, with the top whisker being slightly longer than the bottom. Intel seems to dominate the Windows laptops with prices ranging from really cheap to really expensive and some outliers as well. Most Windows laptops consist of Intel as they are the popular choice among consumers. There are some cheap Windows laptops with basic features that come with AMD processors.

Code:

```
>ggplot(laptops,aes(x=graphics,y=retail_price,colour=platform))+geom_boxplot()+facet_wrap(~platform)
```

Output:



Analysis:

As seen before, Apple seem to be a huge fan of Intel, so both their processors and graphics chipsets are only by Intel. Similarly to their processors, the boxplot of graphics seems to be approximately normally distributed with no outliers or anything out of the blue. Windows however show some variation

when it comes to their graphics chipsets. Unlike Macs, Windows laptops tend to favor NVIDIA graphics more than Intel. Mean prices for AMD and Intel seem to be approximately same, with Intel prices going higher than AMD. NVIDIA, however, id quite high in the price range. Their lowest priced laptop is the same as the mean of Intel's. Overall, NVIDIA seems to be directed towards gamers and professionals, and AMD and Intel towards average users.

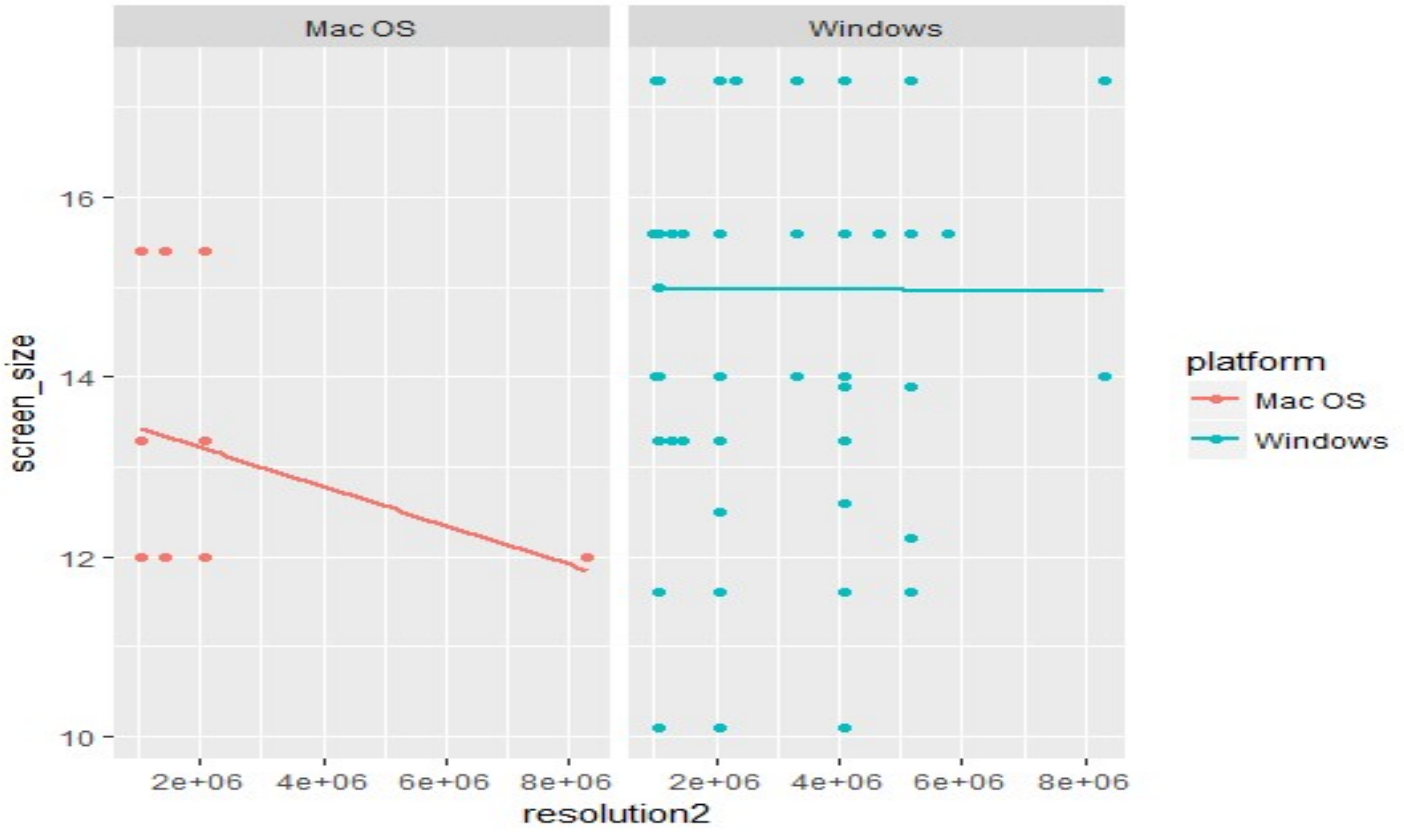
Code:

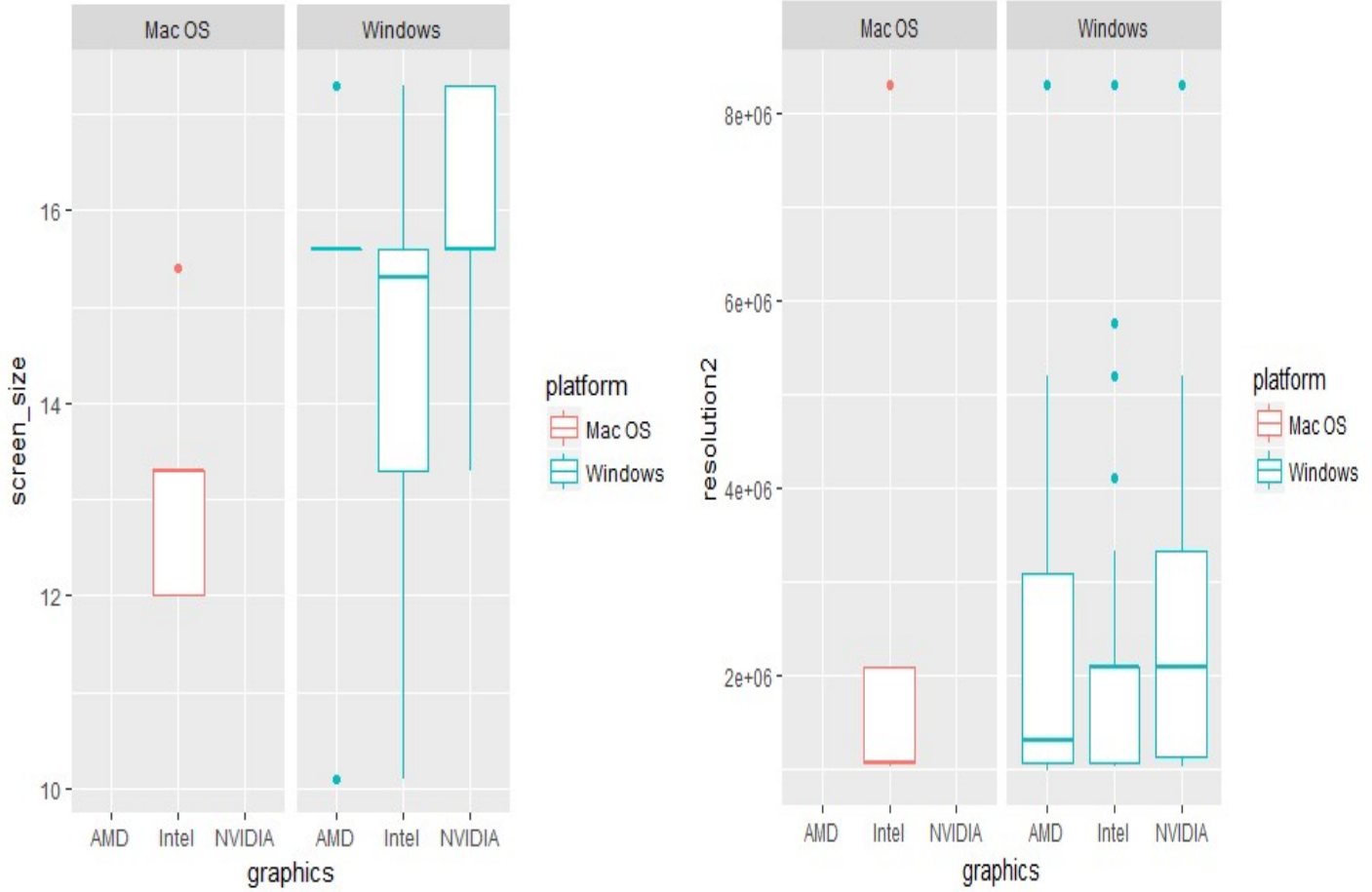
```
>ggplot(laptops,aes(x=resolution2,y=screen_size,colour=platform))+geom_poi
nt()+geom_smooth(method="lm",se=F)+facet_wrap(~platform)

>ggplot(laptops,aes(x=graphics,y=resolution2,colour=platform))+geom_boxplo
t()+facet_wrap(~platform)

>ggplot(laptops,aes(x=graphics,y=screen_size,colour=platform))+geom_boxplo
t()+facet_wrap(~platform)
```

Output:





Analysis:

All these plots show quite some unexpected results. Resolution and screen size of laptops was expected to correlate with each other, but it turns out to be the opposite of that, with Macs having a strong negative slope and Windows having almost no slope at all.

Similarly, there does not seem to be much interaction between the type of graphics and the screen size of laptops, and the type of graphics and resolution, for both platforms. The only noticeable thing is that laptops with NIVIDIA graphics seems to have slightly higher screen size and resolution compared to AMD and Intel, though the difference is not that huge.

Code:

```
>price1=lm(retail_price~ram+storage,data=windows)
```

```
>summary(price1)
>price2=lm(retail_price~ram+storage,data=apple)
>summary(price2)
```

Output:

```
Call:
lm(formula = retail_price ~ ram + storage, data = windows)

Residuals:
    Min      1Q  Median      3Q     Max
-869.93 -212.71  -73.55  163.42 1507.02

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 223.25029   69.43866   3.215 0.001567 **
ram         115.90490    7.96077  14.560  < 2e-16 ***
storage      -0.30059    0.08589  -3.500 0.000598 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 378.8 on 166 degrees of freedom
Multiple R-squared:  0.5968,    Adjusted R-squared:  0.5919
F-statistic: 122.8 on 2 and 166 DF,  p-value: < 2.2e-16


Call:
lm(formula = retail_price ~ ram + storage, data = apple)

Residuals:
    Min      1Q  Median      3Q     Max
-446.13 -247.28  -41.43  132.85  537.40

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 144.3892   162.6631   0.888     0.38
ram         141.6911    12.5302  11.308 7.04e-14 ***
storage       1.8933     0.3078   6.152 3.19e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 276 on 39 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.8187,    Adjusted R-squared:  0.8094
F-statistic: 88.06 on 2 and 39 DF,  p-value: 3.457e-15
```

Analysis:

From the regression models and their outputs above, it can be seen that both RAM and storage are very significant in determining the prices of both

platforms of laptops. For the Windows model, RAM has a p-value of $2*10^{-16}$ and storage has a p-value of 0.000598, both of which are lower than the alpha value of 0.05. Similarly, the p-values of RAM and storage in the Apple model are $7.04*10^{-14}$ and $3.19*10^{-7}$, which are also lower than 0.05.

The R^2 values for both the models are approximately 59% and 81%. This means that we can be assured RAM and storage will affect Windows laptops' prices 59% of the time and Apple laptops 81% of the time,

Code:

```
>price3=lm(retail_price~ram*storage,data=windows)
>summary(price3)
>price4=lm(retail_price~ram*storage,data=apple)
>summary(price4)
```

Output:

```
Call:
lm(formula = retail_price ~ ram * storage, data = windows)

Residuals:
   Min     1Q Median     3Q    Max
-868.7 -203.7  -68.2  154.3 1489.4

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 288.859128 125.927246   2.294   0.0231 *
ram         107.014162  16.307595   6.562 6.55e-10 ***
storage      -0.381039   0.154821  -2.461   0.0149 *
ram:storage   0.009629   0.015405   0.625   0.5328
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 379.5 on 165 degrees of freedom
Multiple R-squared:  0.5977,       Adjusted R-squared:  0.5904
F-statistic: 81.72 on 3 and 165 DF,  p-value: < 2.2e-16



Call:
lm(formula = retail_price ~ ram * storage, data = apple)

Residuals:
    Min      1Q  Median      3Q     Max
-366.67 -214.08  -68.59  183.33  522.31

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 549.2516   361.4396   1.520  0.13688
ram          99.0045    36.2916   2.728  0.00959 **
storage       0.7400     0.9704   0.763  0.45040
ram:storage   0.1206     0.0963   1.252  0.21820
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 274 on 38 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.8259,       Adjusted R-squared:  0.8121
F-statistic: 60.08 on 3 and 38 DF,  p-value: 1.725e-14
```

Analysis:

Like before, windows prices seem to be significantly affected by RAM and storage. However, unlike before, in the new model only RAM seems to affect the prices of Macs. The R-squared values of both the models are approximately the same as before.

Code:

```
>anova(price1,price3)
>anova(price2,price4)
```

Output:

```
Analysis of Variance Table

Model 1: retail_price ~ ram + storage
Model 2: retail_price ~ ram * storage
  Res.Df      RSS Df Sum of Sq      F Pr(>F)
1    166 23817660
2    165 23761399  1     56260 0.3907 0.5328
> anova(price2,price4)


Analysis of Variance Table

Model 1: retail_price ~ ram + storage
Model 2: retail_price ~ ram * storage
  Res.Df     RSS Df Sum of Sq      F Pr(>F)
1     39 2971664
2     38 2853927  1    117737 1.5677 0.2182
```

Analysis:

There is no significant improvement between the two sets of models by adding the interactions, so there is no evidence that having different slopes for RAM and storage is necessary.

Code:

```
>price5=lm(retail_price~processor_speed+processor_cores+screen_size+ram+storage+graphics+processor_type+storage_type,data=windows)
>summary(price5)


>price6=update(price5,.~.-processor_cores)
>summary(price6)
>drop1(price6,test="F")


>price7=update(price6,.~.-screen_size)
>summary(price7)
>drop1(price7,test="F")


>price8=update(price7,.~.-storage)
>summary(price8)
>drop1(price8,test="F")
```

Output:

```
> price5=lm(retail_price~processor_speed+processor_cores+screen_size+ram+storage+graphics+processor_type+storage_type,data=windows)
> summary(price5)

Call:
lm(formula = retail_price ~ processor_speed + processor_cores +
    screen_size + ram + storage + graphics + processor_type +
    storage_type, data = windows)

Residuals:
    Min      1Q  Median      3Q     Max
-559.41 -191.42  -48.05  139.27 1079.30

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         -892.54353  372.91327  -2.393  0.01787 *
processor_speed      210.79894   75.91729   2.777  0.00616 **
```

```
processor_cores       40.43922   33.56279   1.205  0.23006
screen_size           17.94143   22.46578   0.799  0.42572
ram                   66.58032    9.76992   6.815 1.91e-10 ***
storage               -0.09673    0.13760  -0.703  0.48312
graphicsIntel       -202.59635  154.58049  -1.311  0.19190
graphicsNVIDIA       -72.59420  165.62947  -0.438  0.66178
processor_typeIntel  466.27556  169.68998   2.748  0.00670 **
storage_typeHDD      133.89914  198.49540   0.675  0.50094
storage_typeSSD      472.48368  181.45474   2.604  0.01010 *
storage_typeSSHD     642.02760  214.28459   2.996  0.00318 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 320.1 on 157 degrees of freedom
Multiple R-squared:  0.7277,    Adjusted R-squared:  0.7086
F-statistic: 38.14 on 11 and 157 DF,  p-value: < 2.2e-16

>
> price6=update(price5,.~.-processor_cores)
> summary(price6)

Call:
lm(formula = retail_price ~ processor_speed + screen_size + ram +
    storage + graphics + processor_type + storage_type, data = windows)

Residuals:
    Min      1Q  Median      3Q     Max
-588.09 -178.22  -51.68  135.31 1082.99

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         -673.48467  326.05640  -2.066  0.04050 *
processor_speed      183.22732   72.48982   2.528  0.01247 *
screen_size           17.11549   22.48739   0.761  0.44772
ram                   68.10392    9.70158   7.020 6.19e-11 ***
storage               -0.08985    0.13768  -0.653  0.51494
graphicsIntel       -209.41660  154.69750  -1.354  0.17776
graphicsNVIDIA       -48.33752  164.63626  -0.294  0.76945
processor_typeIntel  422.54124  165.99920   2.545  0.01187 *
storage_typeHDD      119.70652  198.42864   0.603  0.54719
storage_typeSSD      446.01986  180.37787   2.473  0.01447 *
storage_typeSSHD     650.82109  214.46620   3.035  0.00282 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 320.5 on 158 degrees of freedom
Multiple R-squared:  0.7252,    Adjusted R-squared:  0.7078
F-statistic:  41.7 on 10 and 158 DF,  p-value: < 2.2e-16

> drop1(price6,test="F")
Single term deletions

Model:
retail_price ~ processor_speed + screen_size + ram + storage +
    graphics + processor_type + storage_type
               Df Sum of Sq      RSS    AIC F value    Pr(>F)
<none>                      16231339 1960.9
processor_speed  1    656332 16887671 1965.6  6.3889   0.01247 *
screen_size      1     59511 16290850 1959.5  0.5793   0.44772
ram              1   5062400 21293739 2004.7 49.2787 6.189e-11 ***
```

```
storage          1     43756 16275095 1959.3  0.4259   0.51494
graphics         2    576493 16807832 1962.8  2.8059   0.06347 .
processor_type   1    665616 16896955 1965.7  6.4793   0.01187 *
storage_type     3   4131166 20362505 1993.2 13.4046 7.719e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
> price7=update(price6,.~.-screen_size)
> summary(price7)

Call:
lm(formula = retail_price ~ processor_speed + ram + storage +
    graphics + processor_type + storage_type, data = windows)

Residuals:
    Min      1Q  Median      3Q     Max
-605.82 -185.70  -58.71  140.80 1102.10

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         -490.01641  219.26959  -2.235  0.02683 *
processor_speed      192.15432   71.43996   2.690  0.00791 **
ram                   69.03760    9.61097   7.183 2.48e-11 ***
storage               -0.07835    0.13666  -0.573  0.56725
graphicsIntel       -211.88419  154.45877  -1.372  0.17206
graphicsNVIDIA       -39.55509  164.01399  -0.241  0.80973
processor_typeIntel  419.07835  165.71717   2.529  0.01242 *
storage_typeHDD      166.31446  188.49342   0.882  0.37893
storage_typeSSD      471.43358  177.02580   2.663  0.00854 **
storage_typeSSHD     685.25018  209.36392   3.273  0.00131 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 320.1 on 159 degrees of freedom
Multiple R-squared:  0.7242,    Adjusted R-squared:  0.7086
F-statistic: 46.39 on 9 and 159 DF,  p-value: < 2.2e-16

> drop1(price7,test="F")
Single term deletions

Model:
retail_price ~ processor_speed + ram + storage + graphics + processor_type
+
    storage_type
               Df Sum of Sq      RSS    AIC F value    Pr(>F)
<none>                     16290850 1959.5
processor_speed  1    741250 17032100 1965.0  7.2347  0.007914 **
ram              1   5286686 21577536 2005.0 51.5985 2.480e-11 ***
storage          1     33675 16324525 1957.8  0.3287  0.567252
graphics         2    660431 16951281 1962.2  3.2229  0.042455 *
processor_type   1    655243 16946093 1964.1  6.3952  0.012417 *
storage_type     3   4077341 20368191 1991.2 13.2651 8.987e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
> price8=update(price7,.~.-storage)
> summary(price8)

Call:
lm(formula = retail_price ~ processor_speed + ram + graphics +
```

```
    processor_type + storage_type, data = windows)

Residuals:
    Min      1Q  Median      3Q     Max
-601.58 -190.21  -64.79  156.61 1095.04

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        -471.19     216.34  -2.178  0.03087 *
processor_speed     183.71      69.76   2.634  0.00928 **
ram                  66.13       8.15   8.115 1.22e-13 ***
graphicsIntel      -217.68     153.80  -1.415  0.15892
graphicsNVIDIA      -44.64     163.43  -0.273  0.78510
processor_typeIntel 422.52     165.26   2.557  0.01150 *
storage_typeHDD     122.67     172.07   0.713  0.47696
storage_typeSSD     477.48     176.34   2.708  0.00751 **
storage_typeSSHD    644.12     196.27   3.282  0.00127 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 319.4 on 160 degrees of freedom
Multiple R-squared:  0.7236,    Adjusted R-squared:  0.7098
F-statistic: 52.36 on 8 and 160 DF,  p-value: < 2.2e-16

> drop1(price8,test="F")
Single term deletions

Model:
retail_price ~ processor_speed + ram + graphics + processor_type +
    storage_type
               Df Sum of Sq      RSS    AIC F value    Pr(>F)
<none>                     16324525 1957.8
processor_speed  1    707613 17032138 1963.0  6.9355  0.009279 **
ram              1   6718523 23043049 2014.1 65.8496 1.215e-13 ***
graphics         2    674679 16999205 1960.7  3.3063  0.039170 *
processor_type   1    666941 16991466 1962.6  6.5368  0.011497 *
storage_type     3   6172676 22497201 2006.0 20.1666 3.882e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Analysis:

I started out with a regression model with the variables, which, I found previously from the plots, interacted with the retail price the most. I included processor speed, processor cores, screen size, ram, storage, graphics, processor type and storage type in my model. I looked at the model and noticed right off the bat that the number of processor cores were not affecting the retail price. So I updated my model by taking processor cores out of the equation. In my new model I found screen size to be insignificant so I took it out and updated the model again. This time I found storage to

be non-significant, so I took that out as well. In my new model I found all my remaining variables to be significant so I left them as it is. Thus, I came up with my model for Windows laptops, with R^2 approximately 71%, to be:

```
lm(formula = retail_price ~ processor_speed + ram + graphics +
    processor_type + storage_type, data = windows).
```

Code:

```
>price9=lm(retail_price~processor_speed+processor_cores+screen_size+ram+st
orage,data=apple)
>summary(price9)


>price10=update(price9,.~.-processor_cores)
>summary(price10)
>drop1(price10,test="F")
```

Output:

```
> price9=lm(retail_price~processor_speed+processor_cores+screen_size+ram+s
torage,data=apple)
> summary(price9)

Call:
lm(formula = retail_price ~ processor_speed + processor_cores +
    screen_size + ram + storage, data = apple)

Residuals:
    Min      1Q  Median      3Q     Max
-322.73  -61.09   54.95   77.35  202.98

Coefficients: (1 not defined because of singularities)
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     2588.9928   791.3491   3.272  0.00232 **
processor_speed  549.0702    67.2905   8.160 8.59e-10 ***
processor_cores  771.1064    96.3109   8.006 1.35e-09 ***
screen_size     -296.9631    82.9307  -3.581  0.00098 ***
ram                   NA         NA      NA       NA
storage            1.6300     0.1743   9.353 2.75e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 133 on 37 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.9601,    Adjusted R-squared:  0.9558
```

```
F-statistic: 222.6 on 4 and 37 DF,  p-value: < 2.2e-16

>
> price10=update(price9,.~.-processor_cores)
> summary(price10)

Call:
lm(formula = retail_price ~ processor_speed + screen_size + ram +
    storage, data = apple)

Residuals:
    Min      1Q  Median      3Q     Max
-322.73  -61.09   54.95   77.35  202.98

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     2588.9928   791.3491   3.272  0.00232 **
processor_speed  549.0702    67.2905   8.160 8.59e-10 ***
screen_size     -296.9631    82.9307  -3.581  0.00098 ***
ram              192.7766    24.0777   8.006 1.35e-09 ***
storage            1.6300     0.1743   9.353 2.75e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 133 on 37 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.9601,    Adjusted R-squared:  0.9558
F-statistic: 222.6 on 4 and 37 DF,  p-value: < 2.2e-16

> drop1(price10,test="F")
Single term deletions

Model:
retail_price ~ processor_speed + screen_size + ram + storage
               Df Sum of Sq     RSS    AIC F value    Pr(>F)
<none>                      654061 415.44
processor_speed  1   1176968 1831029 456.67  66.581 8.589e-10 ***
screen_size      1    226668  880729 425.94  12.823 0.0009798 ***
ram              1   1133167 1787228 455.66  64.103 1.353e-09 ***
storage          1   1546310 2200371 464.39  87.474 2.752e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Analysis:

While making the regression model for Apple, I realized that some of the variables like processor_type, graphics and storage_type only had one level. I kept running into errors, so I had to remove the single level variables. When I regressed a model with the remaining variables, I noticed that my RAM variable kept showing “NA” for all its values in the coefficients. After some research online, I found out that this was due to complete

multicollinearity between two or more variables. After some back-and-forth and trial-and-error I discovered that my RAM variable was collinear with processor_cores. So I removed processor_cores and made a new model. All my variables were significant in this model, there was no multicollinearity anymore, and all the coefficient values of RAM were displayed. Hence, I decided to stick to this model as my final model for Apple laptops, with R^2 approximately 96%:

```
lm(formula = retail_price ~ processor_speed + screen_size + ram + storage,
data = apple)
```

Conclusion

In summary, I can confidently confirm my first expectation that Apple laptops are generally more expensive than Windows laptops, as shown by my first two boxplots and Tukey's HSD.

I found, from my plots, many of the variables (ram, storage, processor_speed, processor_cores, screen_size) to have positive linear slopes with the retail price of the laptops. Resolution was the only variable that had negative slopes with both the platforms. I determined Intel processors to be higher in price than AMD processors due to their demand and longevity. Among the graphics chipsets, NVIDIA came out to be most expensive due to their performance in games and professional software. SSHD took the throne as most expensive storage types due to their dependability and durability. There was little to no correlation between resolution, graphics and screen size. The only correlation between these three worth mentioning is that some laptops with NVIDIA and AMD graphics had better resolution than Intel processors.

From my fitted regression models I was able to determine which variables significantly affected the retail price of laptops from each platform. I was surprised to find out that both the regression models were not the same,

i.e. the same variables did not affect both platforms. Nevertheless, most of my assumed variables did make it into the list.

My final repression model for Windows laptops is:

```
lm(formula = retail_price ~ processor_speed + ram + graphics +
    processor_type + storage_type, data = windows).
```

And my model for the Apple laptops is:

```
lm(formula = retail_price ~ processor_speed + screen_size + ram + storage,
data = apple).
```