Analysis for Laptop Price Variation in eBay

1 Introduction

We aim to understand the impact of various factors such as the processor speed and hard drive storage space on the prices of laptops sold on eBay, and provide recommendations on choosing an appropriate laptop with a good price. We will also focus on how the factors such as whether the laptop has solid state drive or not, and whether the laptop can be bought now or not will influence the price of listed laptops. Regression models is a powerful tool we use to analyze the association between one variables and some other variables. Our most important finding is that three factors will significantly influence the price of laptops are: whether the laptop has solid state drive or not, whether the laptop can be bought right now or not, and the storage space of the hard drive.

2 Methods

The goal of our analysis is to quantify the association of various factors with the prices of a list of laptops sold on eBay. Moreover, recommendations on choosing a laptop with large hard drive space need to be provided from the unsold list of laptops with a preference for a solid state drive.

Our approach for modeling how the laptop price will be influenced by its abilities, such as the speed of processor and the hard drive storage, and whether the laptop is at auction or can be bought right now, is to use multiple linear regression. Regression is a useful tool when we aims to explore the linear relationship between the predictor (independent) variables and response (dependent) variables. One powerful application is to predict the outcome of a response based on a new set of predictors.

2.1 Ordinary Least Squares (OLS)

We start with the introduction to ordinary least squares (OLS) where there is only one independent variable. The OLS model is described as below:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \tag{1}$$

where β_0 and β_1 are regression coefficients, i indicates the i-th observation, and ϵ is a random error assuming it follows a normal distribution with mean zero.

The best coefficients can be estimated by minimizing the squared errors between the actual value y_i and the fitted value $\hat{y}_i = \beta_0 + \beta_1 x_i$ as shown in Figure 1. The least squares criterion is expressed by:

$$\min_{\beta_0,\beta_1} \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_i\right)^2, \tag{2}$$

where n is the total number of observations.

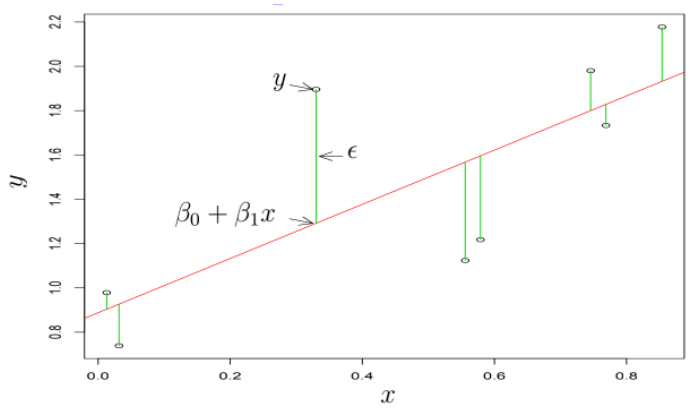


Figure 1: Least Squares Estimate

2.2 Multiple Linear Regression

Multiple linear regression is an extension from the OLS where there are more than one independent variables. Hence, it can be described as the following:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i, \tag{3}$$

where p is the number of independent variables. Similarly, the best coefficients can be estimated by minimizing the sum of squared residuals, $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$. The difference between y_i and $\hat{y}_i$ is called residuals.

Importantly, the multiple linear regression assumes there is no multicollinearity which means there should be no relationship between the independent variables. In the result section, we need to check our data does not violate this assumption. One useful tool to check the multicollinearity is to use the correlation matrix. A correlation matrix is a table that captures the Pearson correlation coefficients between different variables. The correlation coefficients between variable X and variable Y is calculated by:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^{n} \left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{\sqrt{\sum_{i=1}^{n} \left(X_i - \bar{X}\right)^2} \sqrt{\sum_{i=1}^{n} \left(Y_i - \bar{Y}\right)^2}}, \tag{4}$$

where $\bar{X}$ and $\bar{Y}$ are average of X and Y. If $\rho_{X,Y}$ gets close to 1 or -1, it indicates there is a strongly positive or negative linear relationship between two variables. If the coefficient gets close to 0, it indicates there is no tendency in other variable to either increase or decrease as one variable increases. In addition, it is necessary to check our normality assumption is satisfies which means our errors are normally distributed.

In our analysis, our response is the price of laptops and other variables could be our independent variables. The multiple linear regression model can tell us how strong the relationship is between the two or more independent variables and the price of laptops.

3 Results

3.1 Overview of the Data

The data set we analyzed later was scraped from eBay. It contains 200 listed laptops with 8 attributes (variables) including information related to laptops' abilities, sale status

and auction status. Each row of the data set represents a list laptop and its corresponding 8 attributes: listed id, whether it was sold or not, sales (listing) price, speed of processor, amount of ram, amount of hard drive space, whether the hard drive is solid state or not, and whether it could be bought now or was at auction.

3.1.1 Data Pre-processing

The whole data set contains 200 rows of observations, however, 72 rows contains missing values. There are 49, 43 and 70 missing values from variables speed of processor, amount of ram and amount of hard drive space respectively. Figure 2 is the comparison of distribution of prices if we remove all missing values.

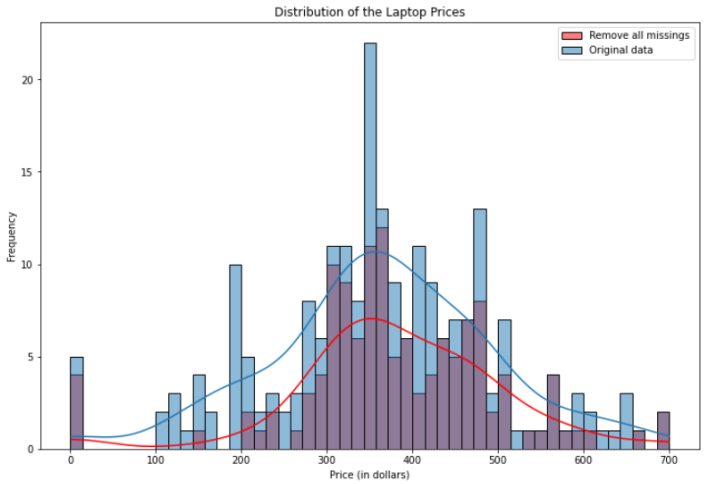


Figure 2: Comparing Distributions of Prices

The distribution does not change a lot in overall, but we can find many of the missing values come from the laptops with relative low price. Therefore, imputation strategy is used to fill out the missing ones. It is noticeable that those three variables are numeric but with categorical meanings, which means a more appropriate strategy is to use the mode other than median or mean for imputation. A mode is the most frequent number that shows in a list of numbers.

For our purpose of analysis, the variable 'ID' is excluded temporarily in the modeling part since it does not provide any information related to the prices. Moreover, there is a causal relationship between the sale status and the price which means the price of laptops will influence whether it could be sold or not. Therefore, the variable 'Sold or Not' is excluded.

3.1.2 Pairwise Correlation

Figure 3(a) is the correlation table between variables. We find there is no strong relationship between variables and there is a weak positive relationship between whether laptop is solid state drive (SSD) or not and the price. It means if the laptop has ssd, it is likely to have a higher price than those who do not have.

From Figure 3 (b), we can see the cost of storage differs by SSD or not. The price of laptop with SSD could be higher than that of laptop that is not SSD when the you buy

	Price	Speed (in GHz)	RAM (in GB)	Hard Drive Space (in GB)	Solid State Drive or Not
Price	1.00	0.18	0.26	0.01	0.44
Speed (in GHz)	0.18	1.00	0.23	-0.05	0.17
RAM (in GB)	0.26	0.23	1.00	0.09	0.37
Hard Drive Space (in GB)	0.01	-0.05	0.09	1.00	-0.26
Solid State Drive or Not	0.44	0.17	0.37	-0.26	1.00

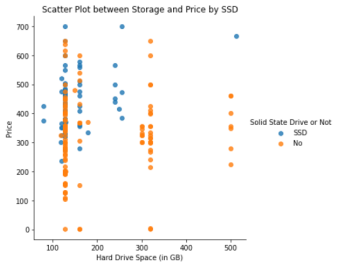


Figure 3: (a) Correlation Table (b) Scatter Plot between Storage and Price by SSD

a laptop with relative small storage. However, it seems there are bare laptops with SSD having storage larger than 300 GB. The blue point on the upper right could be an outlier.

3.2 Baseline Data Description

The basic statistic for cleaned data is shown in Table 1:

Table 1: Baseline Table

Variable	Median (IQR) or Percentage
Price (in dollars)	357.5(299.98, 449.99)
Speed (in GHz)	2.5(2.5, 2.6)
RAM (in GB)	4.0(4.0, 8.0)
Hard Drive Space (in GB)	128(128, 165)
Solid State Drive or Not (%)	36.36
Buy It Now or Auction (%)	54.09

3.3 Results and Analysis

Firstly, we used the multiple linear regression on the cleaned data set. Moreover, we would like to investigate the interaction between SSD and hard drive storage. All measurements are under the level of confidence of 5%. If the P-value is under 0.05, we would say this variable is statistically significant. This indicates that the changes in independent variables correlates with the shifts in the response variable which is the price in our modelling.

Table 2 and Table 3 show the results of estimated coefficients. We found SSD and storage become insignificant after we added the interaction term, while the interaction term is insignificant. This indicates there is no cost difference of storage no matter it is SSD or not. Moreover, we did not detect the multicollinearity issue. We can conclude that any change in whether the laptop is SSD or not, whether it can be bought now or at auction and the storage space will influence the price. For example, if the laptop is solid state drive, the price will increase 120.47 dollars when other conditions are all the same.

Table 2: Coefficients without Interaction term

Variable	Coefficient	P-value	Confidence Interval
Intercept	-81.79	0.670	(-460.11, 296.53)
Solid State Drive or Not (True)	120.47	0.000	(84.58, 156.35)
Buy It Now or Auction (True)	55.42	0.000	(24.62, 86.25)
Speed (in GHz)	119.26	0.118	(-30.31, 268.83)
RAM (in GB)	5.42	0.200	(-2.89, 13.74)
Hard Drive Space (in GB)	0.21	0.018	(0.04, 0.38)

Table 3: Coefficients with Interaction term

Variable	Coefficient	P-value	Confidence Interval
Intercept	-0.33	0.999	(-387.95, 387.28)
Solid State Drive or Not (True)	53.80	0.205	(-29.55, 137.14)
Buy It Now or Auction (True)	57.69	0.000	(26.91, 88.48)
Speed (in GHz)	91.78	0.235	(-60.29 243.85)
RAM (in GB)	5.15	0.221	(-2.89, 13.74)
Hard Drive Space (in GB)	0.15	0.108	(-0.03, 0.33)
Interaction	0.44	0.082	(-0.06, 0.94)

From exploratory data analysis in the Appendix, we can find there are few observations in some levels from the variables storage, SSD or not and the amount of ram. Therefore, we discretized those variables and the results of coefficients are shown below:

Table 4: Coefficients after Discretizing

Variable	Coefficient	P-value	Confidence Interval
Intercept	264.19	0.000	(235.93, 294,46)
Solid State Drive or Not (True)	117.48	0.000	(82.02, 152.95)
Buy It Now or Auction (True)	58.11	0.000	(27,70, 88.52)
Speed (in GHz) greater than 2.6	80.44	0.002	(30.02 130.86)
RAM (in GB) greater than 6	13.94	0.472	(-24.22, 52.09)
Hard Drive Space (in GB) greater than 200	44.89	0.018	(7.79, 81.99)

We found the variable SSD or not and auction status remains significant, and also the processor speed greater than 2.6 GHz becomes significant. It means if the processor speed is greater than 2.6 GHz, the price will increase by 80.44 dollars compared to those whose processor speed is less or equal to 2.6 GHz when other conditions remain the same.
We used AIC method to select our model. A smaller AIC means a better fitted model. R-squared is also a useful indicator telling how many of the variations of response variable can be explained by our model's inputs. A larger R-squared means a better fitted model. The summary is shown in Table 5.

Table 5: AIC and R-squared Comparisons

Model	AIC	R-squared
Regression without Interaction	2715	0.271
Regression with Interaction	2714	0.281
Regression without Interaction after Discretizing	2708	0.294

The regression model after discretizing could be the best model. However, its R-squared is small and less than 50%. This could result from the influence from outliers.

3.4 Recommendations

From above analysis, we can find if the laptop is solid state drive, the price will be higher than the laptop that is not SSD. Moreover, there is no significant difference in the cost of storage no matter the laptop is SSD or not. We also find whether the laptop can be bought now (BIN) will significantly influence the price. Therefore, I would give the following recommendations according to our client's request:

Table 6: Recommendations on Buying Laptops

Listed ID	Price ($)	Storage (in GB)	SSD or Not	BIN or Not
23	565	240	Yes	Yes
54	560	160	Yes	No
92	410	128	Yes	Yes

4 Conclusion

In our analysis, we utilized the multiple linear regression to investigate the relationship between the price of laptops and other attributes. We figured out factors that would significantly influence the price of laptops on eBay and those factors are: the hard drive storage space, whether the laptop has SSD or not, and whether it can be bought now or not. Moreover, We found that the cost of storage would not differ by whether the laptop has SSD or not. Based on our client's requirement, we recommended the 23rd, 54th and 92nd listed laptops with SSD and relative large storage space from the laptops that have not sold.

Our models all achieved relatively low R-squared scores that were less than 50 %. This might result from the impact of outliers. Therefore, more efforts should be taken in exploring outliers to figure out a more fitted model. Diagnostics should also be included in our analysis if time permitted. In addition, we might use more complicated model such as Gamma generalized linear model to see whether it is a good choice.

Appendix

September 10, 2022

[37]:
```
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

0.1 Explorary Data Analysis (EDA)

0.1.1 Read Data

[ ]:
```
file = open('laptopData.csv')
for _ in range(10):
    print(file.readline())
```

[38]:
```
# read data
df = pd.read_csv('laptopData.csv')
df.head(10)
```

[38]:

	Unnamed: 0	sale	price	ghz	ram	hd	ssd	BIN
0	1	SOLD	404.99	2.7	8.0	NaN	SSD	False
1	2	SOLD	355.00	2.5	8.0	128.0	SSD	False
2	3	SOLD	449.99	2.6	4.0	128.0	No	True
3	4	NOT SOLD	499.99	2.5	4.0	320.0	No	True
4	5	NOT SOLD	199.99	NaN	NaN	NaN	No	True
5	6	NOT SOLD	699.95	2.5	4.0	128.0	SSD	True
6	7	NOT SOLD	437.71	2.5	4.0	128.0	SSD	True
7	8	SOLD	449.00	2.5	4.0	128.0	SSD	True
8	9	SOLD	128.00	2.7	NaN	NaN	No	False
9	10	SOLD	512.96	2.7	8.0	160.0	No	True

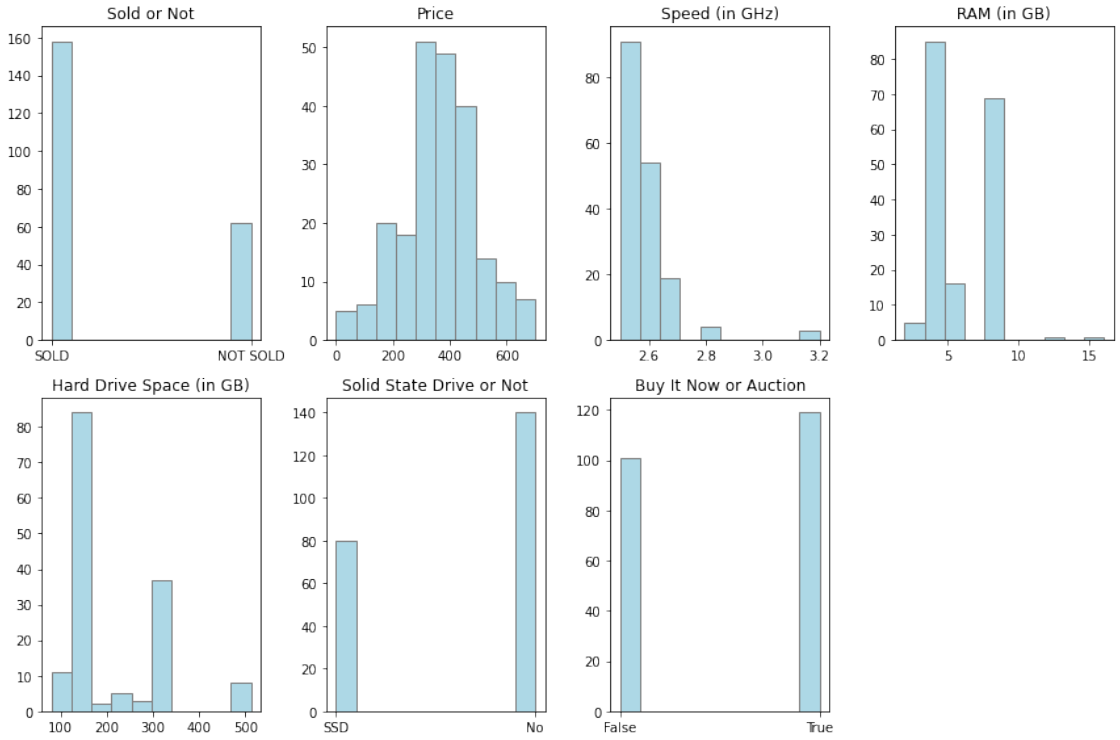
[39]:
```
# check rows and columns
df.shape
```

[39]:
```
(220, 8)
```

[40]:
```
# Change boolean to string
df['BIN'] = df['BIN'].astype(str)
df.columns = ['ID', 'Sold or Not', 'Price', 'Speed (in GHz)',
              'RAM (in GB)', 'Hard Drive Space (in GB)', 'Solid State Drive or␣
 ↪Not', 'Buy It Now or Auction']
```

[41]:
```
# histogram
plt.rcParams['figure.figsize'] = [12, 8]
def draw_histogram(df, variables, n_rows, n_cols):
    fig = plt.figure()
    for i, var_name in enumerate(variables):
        ax = fig.add_subplot(n_rows,n_cols,i+1)
        df[var_name].hist(ax=ax, color='lightblue', ec='grey')
        ax.set_title(var_name)
        ax.grid(False)
    fig.tight_layout()
    plt.show()

draw_histogram(df.iloc[:,1:8], df.iloc[:,1:8].columns, 2, 4)
```



0.1.2 Cleaning Data & Imputation

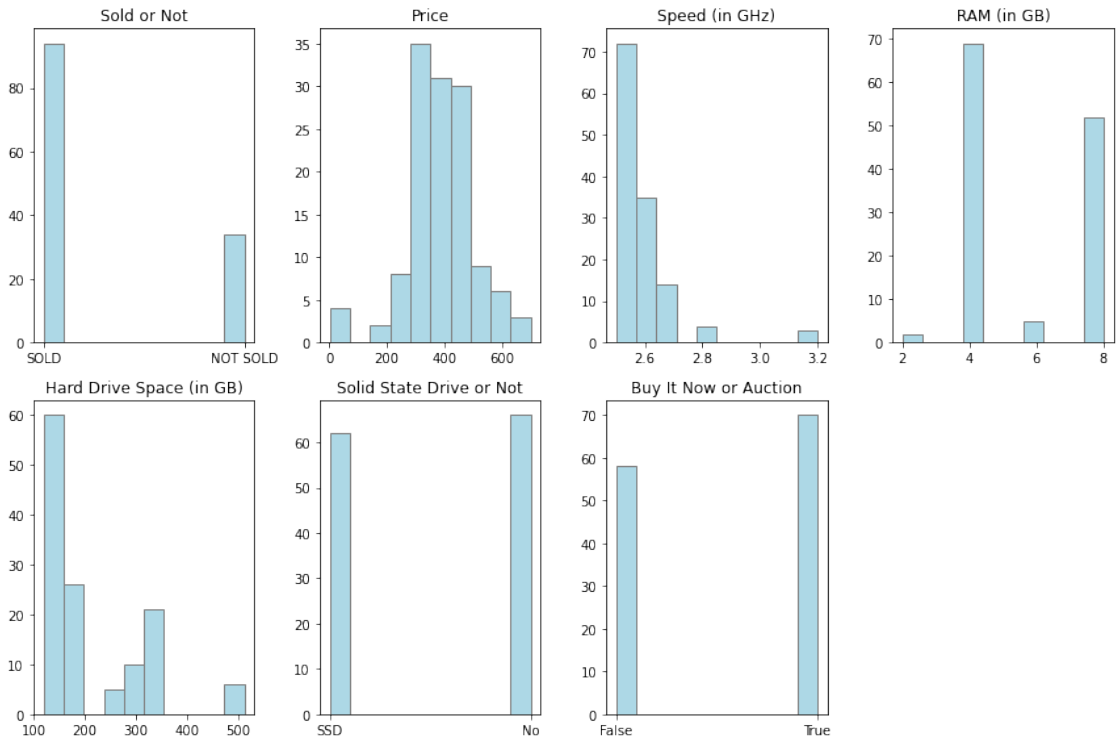
[42]:
```
# check missing values
df.isna().sum()
```

[42]:
```
ID                           0
Sold or Not                  0
Price                        0
Speed (in GHz)              49
RAM (in GB)                 43
Hard Drive Space (in GB)    70
Solid State Drive or Not     0
Buy It Now or Auction        0
dtype: int64
```

[43]:
```
df_no_missing = df.copy().dropna()
df_no_missing.shape # more than 1/3 rows includes missing values
```

[43]:
```
(128, 8)
```

[44]:
```
# compare the distributions after removing all missing values
draw_histogram(df_no_missing.iloc[:,1:], df_no_missing.iloc[:,1:].columns, 2, 4)
```

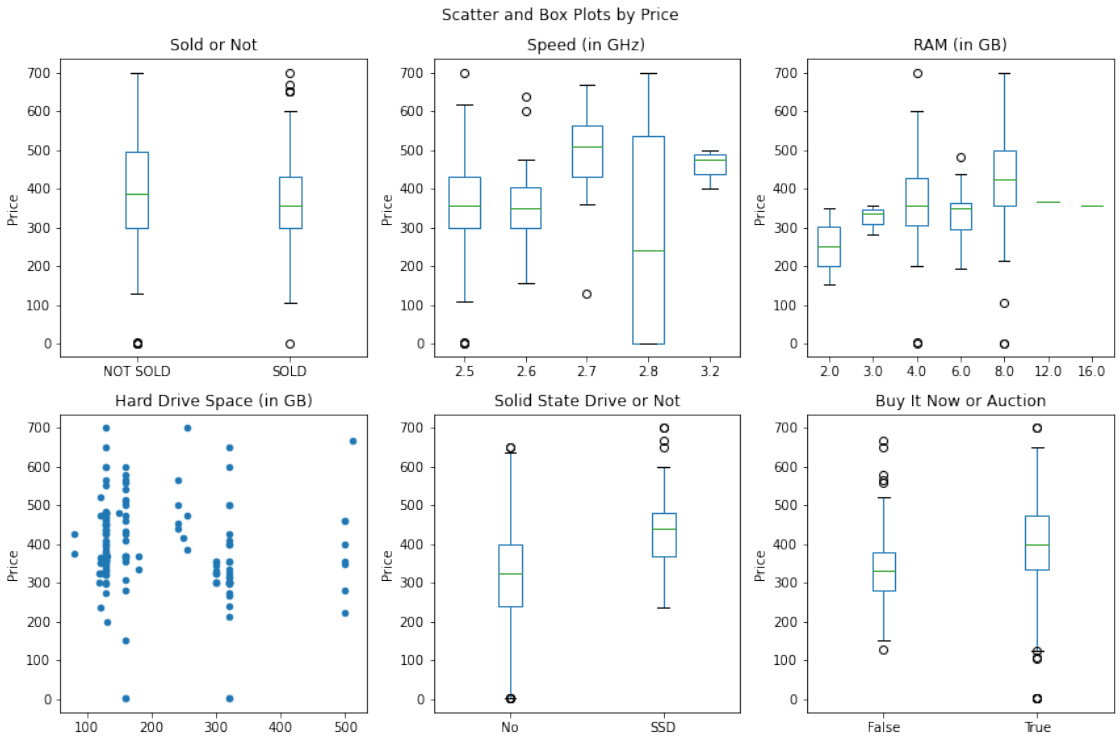


It seems it is not a good idea to remove all rows with missing values since the distribution of price

changes, especially when the price is low.

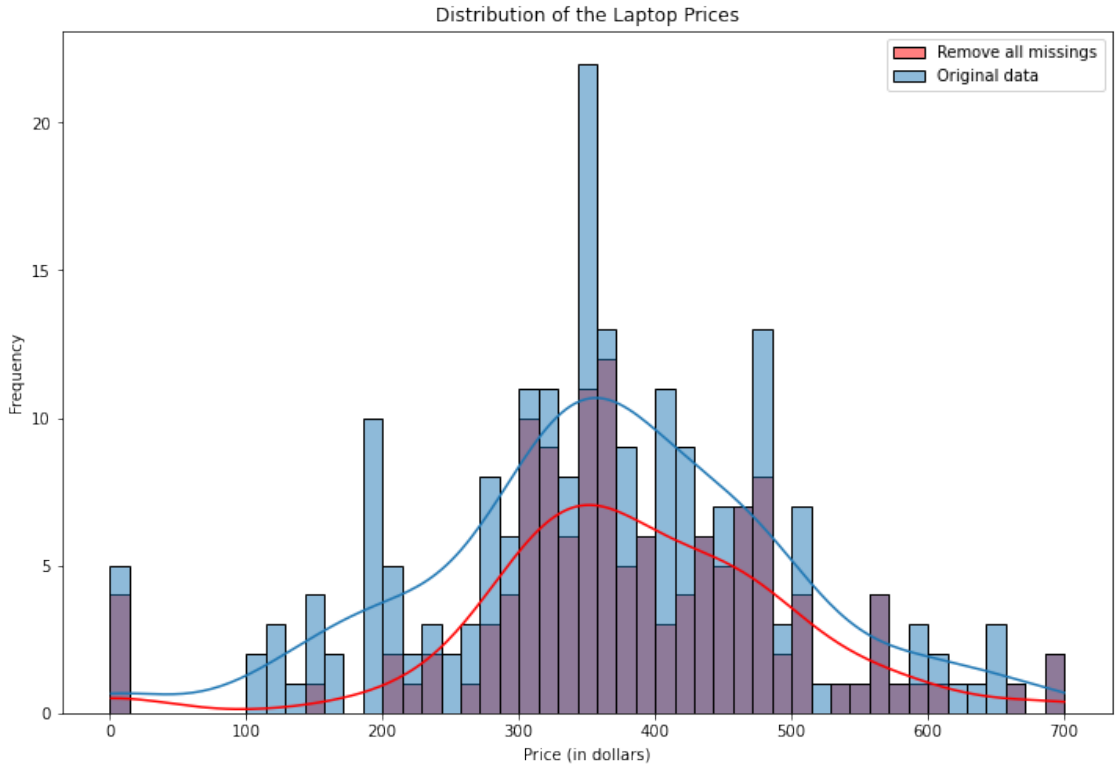
[45]:
```
# we will use boxplot to see whether there are any outliers,
def draw_boxplot(df, outcome, n_rows, n_cols):
    fig = plt.figure()
    variables = df.columns.drop(outcome)
    for i, var_name in enumerate(variables):
        ax = fig.add_subplot(n_rows, n_cols, i+1)
        if len(df[var_name].unique()) > 8:
            df.plot.scatter(x=var_name, y=outcome, ax=ax)
        else:
            df.boxplot(column=outcome, by=var_name, grid=False, ax=ax)
            ax.set(ylabel=outcome)
        ax.set_title(var_name)
        ax.set(xlabel=None) # hide x-axis labels
    fig.suptitle('Scatter and Box Plots by '+outcome)
    fig.tight_layout()
    plt.show()

draw_boxplot(df.iloc[:,1:], 'Price', 2, 3)
```



There are some outliers for processor speed, ram and drive space. Mean is not appropriate to use here. Also, those numerical variables have categorical meanings. Hence we use the **most frequent** number to fill in the missings. #### Mode for Imputation

[46]:
```
bins = np.linspace(0, 700)
ax = sns.histplot(df_no_missing['Price'], bins=bins, label='Remove all␣
 ↪missings', kde=True, color='red')
sns.histplot(df['Price'], bins=bins, label='Original data', kde=True)
ax.set(xlabel='Price (in dollars)', ylabel='Frequency')
plt.title('Distribution of the Laptop Prices')
plt.legend()
plt.show()
```



[47]:
```
df['Speed (in GHz)'].fillna(df['Speed (in GHz)'].mode().iloc[0], inplace=True)
df['Hard Drive Space (in GB)'].fillna(df['Hard Drive Space (in GB)'].mode().
 ↪iloc[0], inplace=True)
df['RAM (in GB)'].fillna(df['RAM (in GB)'].mode().iloc[0], inplace=True)
```

- We use the mode to fill out all missing values.
- There is a causual relationship between the sale price and whether the laptop was sold or not. The sale price would influence whether the laptop was sold or not. Hence we need to exclude the variable `Sold or Not`.

[48]:
```
dfc = df.drop(['ID', 'Sold or Not'], axis=1) # for modelling purposes we␣
 ↪exclude the listing id here
```

0.1.3 Pairwise Correlations

Correlation matrix

[49]:
```
dfc_num = dfc.copy()
# convert categorical variables to 1 or 0
dfc_num.iloc[:,4] = dfc.iloc[:,4].map(dict(SSD=1, No=0))
mapping = {'True' : 1, 'False' : 0}
dfc_num.replace({'Buy It Now or Auction': mapping})
dfc_num.corr().round(2) # did not see much multicollinearity
```

[49]:

	Price	Speed (in GHz)	RAM (in GB)	\
Price	1.00	0.18	0.26	
Speed (in GHz)	0.18	1.00	0.23	
RAM (in GB)	0.26	0.23	1.00	
Hard Drive Space (in GB)	0.01	-0.05	0.09	
Solid State Drive or Not	0.44	0.17	0.37	

	Hard Drive Space (in GB)	Solid State Drive or Not
Price	0.01	0.44
Speed (in GHz)	-0.05	0.17
RAM (in GB)	0.09	0.37
Hard Drive Space (in GB)	1.00	-0.26
Solid State Drive or Not	-0.26	1.00

Scatter Plot between Storage and Price by SSD

[50]:
```
sns.lmplot(x='Hard Drive Space (in GB)', y='Price', data=dfc, hue='Solid State␣
↪Drive or Not', fit_reg=False)
ax.set(xlabel='Hard Drive Space (in GB)', ylabel='Price (in dollars)')
plt.title('Scatter Plot between Storage and Price by SSD')
plt.show()
```

0.1.4 Baseline Table

[19]:
```
dfc.describe().round(2).transpose()
```

[19]:

	count	mean	std	min	25%	50%	75%	max
Price	220.0	364.16	132.30	1.0	299.98	357.5	449.99	699.99
Speed (in GHz)	220.0	2.56	0.11	2.5	2.50	2.5	2.60	3.20
RAM (in GB)	220.0	5.46	2.05	2.0	4.00	4.0	8.00	16.00
Hard Drive Space (in GB)	220.0	180.61	94.32	80.0	128.00	128.0	165.00	512.00

[20]:
```
# categorical variables
def get_categorical_percentages(df):
    df_cat = df.select_dtypes(exclude=np.number)
```

```
    for var in df_cat.columns:
        perc = df[var].value_counts() / df[var].count()
        print(var)
        print(perc)

get_categorical_percentages(dfc)
```

```
Solid State Drive or Not
No     0.636364
SSD    0.363636
Name: Solid State Drive or Not, dtype: float64
Buy It Now or Auction
True     0.540909
False    0.459091
Name: Buy It Now or Auction, dtype: float64
```

0.2 Modelling

0.2.1 Linear Regression

Without Interaction

[24]:
```
dfc.columns = ['price', 'ghz', 'ram', 'hd', 'ssd', 'bin']
# df_no_missing.columns = ['id', 'sale', 'price', 'ghz', 'ram', 'hd', 'ssd',␣
 ↪'bin']
```

[25]:
```
expr = '''price ~ ghz + ram + hd + ssd + bin'''
fit_ols = smf.ols(formula=expr, data=dfc).fit()
fit_ols.summary()
```

[25]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.271
Model:                            OLS   Adj. R-squared:                  0.254
Method:                 Least Squares   F-statistic:                     15.91
Date:                Fri, 09 Sep 2022   Prob (F-statistic):           2.56e-13
Time:                        19:34:47   Log-Likelihood:                -1351.6
No. Observations:                 220   AIC:                             2715.
Df Residuals:                     214   BIC:                             2736.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -81.7918    191.933     -0.426      0.670    -460.113     296.530
ssd[T.SSD]   120.4692     18.206      6.617      0.000      84.584     156.354
bin[T.True]   55.4246     15.637      3.544      0.000      24.602      86.248
```

```
ghz           119.2603     75.883      1.572      0.118     -30.313     268.834
ram             5.4238      4.219      1.286      0.200      -2.892      13.739
hd              0.2084      0.087      2.381      0.018       0.036       0.381
==============================================================================
Omnibus:                       13.968   Durbin-Watson:                   2.032
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               29.814
Skew:                          -0.247   Prob(JB):                     3.36e-07
Kurtosis:                       4.735   Cond. No.                     5.45e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 5.45e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

With Interaction between Storage and SSD

[26]:
```
expr = '''price ~ ghz + ram + hd + ssd + bin + hd:ssd'''
fit_ols_int = smf.ols(formula=expr, data=dfc).fit()
fit_ols_int.summary()
```

[26]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.281
Model:                            OLS   Adj. R-squared:                  0.261
Method:                 Least Squares   F-statistic:                     13.89
Date:                Fri, 09 Sep 2022   Prob (F-statistic):           2.58e-13
Time:                        19:34:50   Log-Likelihood:                -1350.1
No. Observations:                 220   AIC:                             2714.
Df Residuals:                     213   BIC:                             2738.
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
=
                    coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-
Intercept        -0.3309    196.643     -0.002      0.999    -387.946
387.284
ssd[T.SSD]       53.7967     42.283      1.272      0.205     -29.550
137.143
bin[T.True]      57.6919     15.617      3.694      0.000      26.908
88.476
```

```
ghz              91.7831     77.146      1.190      0.235     -60.285
243.851
ram               5.1578      4.201      1.228      0.221      -3.124
13.439
hd                0.1505      0.093      1.616      0.108      -0.033
0.334
hd:ssd[T.SSD]     0.4400      0.252      1.745      0.082      -0.057
0.937
==============================================================================
Omnibus:                       14.866   Durbin-Watson:                   2.042
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               32.440
Skew:                          -0.266   Prob(JB):                     9.03e-08
Kurtosis:                       4.804   Cond. No.                     5.77e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 5.77e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

0.2.2 Linear Regression with Discretized Variables

Storage, RAM and processor speed are numerical variables with categorical meanings. From EDA, we also notice that there are few observations in some levels from those variables.According to the histograms and boxplots, we discretize those variables as below:

[31]:
```
df_discretized = dfc.copy()
df_discretized['ghz'] = df_discretized['ghz'].apply(lambda x: '>2.6' if x > 2.6␣
 ↪else '<=2.6')
df_discretized['ram'] = df_discretized['ram'].apply(lambda x: '>6' if x > 6␣
 ↪else '<=6')
df_discretized['hd'] = df_discretized['hd'].apply(lambda x: '>200' if x > 200␣
 ↪else '<=200')
```

Withou Interaction

[33]:
```
expr = '''price ~ ghz + ram + hd + ssd + bin'''
fit_ols_dis = smf.ols(formula=expr, data=df_discretized).fit()
fit_ols_dis.summary()
```

[33]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.294
Model:                            OLS   Adj. R-squared:                  0.278
```

```
Method:                 Least Squares   F-statistic:                     17.82
Date:                Fri, 09 Sep 2022   Prob (F-statistic):           9.27e-15
Time:                        19:44:39   Log-Likelihood:                -1348.1
No. Observations:                 220   AIC:                             2708.
Df Residuals:                     214   BIC:                             2729.
Df Model:                           5
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept      265.1926     14.846     17.863      0.000     235.929     294.456
ghz[T.>2.6]     80.4375     25.578      3.145      0.002      30.020     130.855
ram[T.>6]       13.9353     19.358      0.720      0.472     -24.222      52.093
hd[T.>200]      44.8910     18.821      2.385      0.018       7.793      81.989
ssd[T.SSD]     117.4827     17.992      6.530      0.000      82.019     152.947
bin[T.True]     58.1077     15.428      3.766      0.000      27.697      88.519
==============================================================================
Omnibus:                       25.449   Durbin-Watson:                   1.978
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               64.552
Skew:                          -0.487   Prob(JB):                     9.61e-15
Kurtosis:                       5.468   Cond. No.                         4.85
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

With Interaction

[35]:
```
expr = '''price ~ ghz + ram + hd + ssd + bin + ssd:hd'''
fit_ols_int_dis = smf.ols(formula=expr, data=df_discretized).fit()
fit_ols_int_dis.summary()
```

[35]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.295
Model:                            OLS   Adj. R-squared:                  0.275
Method:                 Least Squares   F-statistic:                     14.86
Date:                Fri, 09 Sep 2022   Prob (F-statistic):           3.54e-14
Time:                        19:46:43   Log-Likelihood:                -1347.9
No. Observations:                 220   AIC:                             2710.
Df Residuals:                     213   BIC:                             2734.
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
```

```
=========
                              coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
---------
Intercept                  267.4130     15.344     17.428      0.000     237.167
297.659
ghz[T.>2.6]                 78.7857     25.772      3.057      0.003      27.985
129.587
ram[T.>6]                   13.4342     19.407      0.692      0.490     -24.820
51.689
hd[T.>200]                  39.2970     21.130      1.860      0.064      -2.354
80.948
ssd[T.SSD]                 113.7355     19.121      5.948      0.000      76.045
151.426
bin[T.True]                 57.6658     15.471      3.727      0.000      27.171
88.161
ssd[T.SSD]:hd[T.>200]       26.5419     45.302      0.586      0.559     -62.756
115.840
==============================================================================
Omnibus:                       25.395   Durbin-Watson:                   1.983
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               64.149
Skew:                          -0.488   Prob(JB):                     1.18e-14
Kurtosis:                       5.459   Cond. No.                         8.17
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

0.3 Recommendations

[53]:
```
# recommendations of unsold laptops with SSD
df_unsold = df.loc[df['Sold or Not'] == 'NOT SOLD']
df_unsold_ssd = df_unsold.loc[df_unsold['Solid State Drive or Not'] == 'SSD']
```

[58]:
```
df_unsold_ssd.sort_values(by='Price', ascending=False)
```

[58]:

	ID	Sold or Not	Price	Speed (in GHz)	RAM (in GB)	\
5	6	NOT SOLD	699.95	2.5	4.0	
99	100	NOT SOLD	579.00	2.7	8.0	
22	23	NOT SOLD	565.00	2.5	8.0	
74	75	NOT SOLD	565.00	2.7	8.0	
52	53	NOT SOLD	564.95	2.5	8.0	
53	54	NOT SOLD	560.00	2.7	8.0	
29	30	NOT SOLD	500.00	2.7	8.0	

38	39	NOT SOLD	499.99	2.7	8.0	
6	7	NOT SOLD	437.71	2.5	4.0	
35	36	NOT SOLD	437.71	2.5	4.0	
75	76	NOT SOLD	425.00	2.5	8.0	
91	92	NOT SOLD	410.00	2.6	8.0	
36	37	NOT SOLD	373.00	2.5	8.0	
42	43	NOT SOLD	373.00	2.5	8.0	

	Hard Drive Space (in GB)	Solid State Drive or Not	Buy It Now or Auction
5	128.0	SSD	True
99	160.0	SSD	False
22	240.0	SSD	True
74	160.0	SSD	False
52	128.0	SSD	True
53	160.0	SSD	False
29	160.0	SSD	False
38	128.0	SSD	False
6	128.0	SSD	True
35	128.0	SSD	True
75	80.0	SSD	True
91	128.0	SSD	True
36	128.0	SSD	False
42	128.0	SSD	False