Econ-UB 251 - Assignment 1

Shalem Sumanthiran - sps9893

Theory

Question 1

In [ ]:

```
from IPython.display import Image
Image("C:/Users/shale/OneDrive/Pictures/Screenshots/Q1.png")
```

Out[ ]:

1. $y_i = \beta_i X_i + u_i$

$$TSS = \sum_{i=1}^{n} (y_i - \beta_1 x_i)^2$$

Set $\frac{\partial TSS}{\partial \beta_1} = 0$,

$$\frac{\partial TSS(\beta_1)}{\partial \beta_1} = \sum_{i=1}^{n} -2X_i(Y_i - \beta_1 X_i)$$

$$= -2\sum_{i=1}^{n} X_i y_i + 2\sum_{i=1}^{n} \beta_1 X_i^2$$

$$\therefore \sum_{i=1}^{n} \beta_1 X_i^2 = \sum_{i=1}^{n} X_i y_i$$

$$\Rightarrow \beta_1 \sum X_i^2 = \sum_{i=1}^{n} X_i y_i$$

$$\therefore \hat{\beta}_1 = \frac{\sum_{i=1}^{n} X_i Y_i}{\sum_{i=1}^{n} X_i^2}$$

Question 2

In [ ]:
```python
from IPython.display import Image
Image("C:/Users/shale/OneDrive/Pictures/Screenshots/Q2a.png")
```

Out[ ]:

2. $(100 \cdot y_i) = \beta_0^* + \beta_1^* X_i + u_i$

$$\beta_1' = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(100 y_i - 100 \bar{y})}{\sum_{i=1}^{n} (X_i - \bar{X})^2}$$

$$= \frac{\sum (X_i - \bar{x}) \cdot 100 (y_i - \bar{y})}{\sum (X_i - \bar{X})^2}$$

$$= 100 \cdot \frac{Cov(X, Y)}{Var(X)}$$

$$\beta_0^* = \bar{y}' - \beta_1' \bar{x} \Rightarrow 100 \cdot \bar{y} - 100 \cdot \beta_1 \bar{X}$$

$$= 100 (\bar{y} - \beta_1 \bar{x})$$

Therefore, the intercept and slope both increase by a factor of 100 when the dependent variable is multiplied by 100.

In [ ]:
```python
from IPython.display import Image
Image("C:/Users/shale/OneDrive/Pictures/Screenshots/Q2b.png")
```

Out[ ]:

$$y_i = \beta_0^* + \beta_1'(100\cdot x_i) + u_i$$

$$\beta_1^* = \frac{Cov(x,y)}{Var(x)}$$

$$X = 100\cdot x_i \quad , \quad \beta_1' = \frac{Cov(100\cdot x, y)}{Var(100\cdot x)}$$

$$= \frac{100\cdot Cov(x,y)}{10000\cdot Var(x)}$$

$$\frac{1}{100}\cdot\frac{Cov(x,y)}{Var(x)}$$

$$\beta_0^* = \bar{y} - \beta_1'\bar{x} \Rightarrow \bar{y} - \frac{\beta_1}{\cancel{100}}\cdot \cancel{100}\cdot\bar{x}$$

$$\Rightarrow \bar{y} - \beta_1\cdot\bar{x}$$

Therefore, the intercept is divided by a factor of 100 when the independent variable is multiplied by 100, and the intercept remains unchanged.

Empirical

Question 3

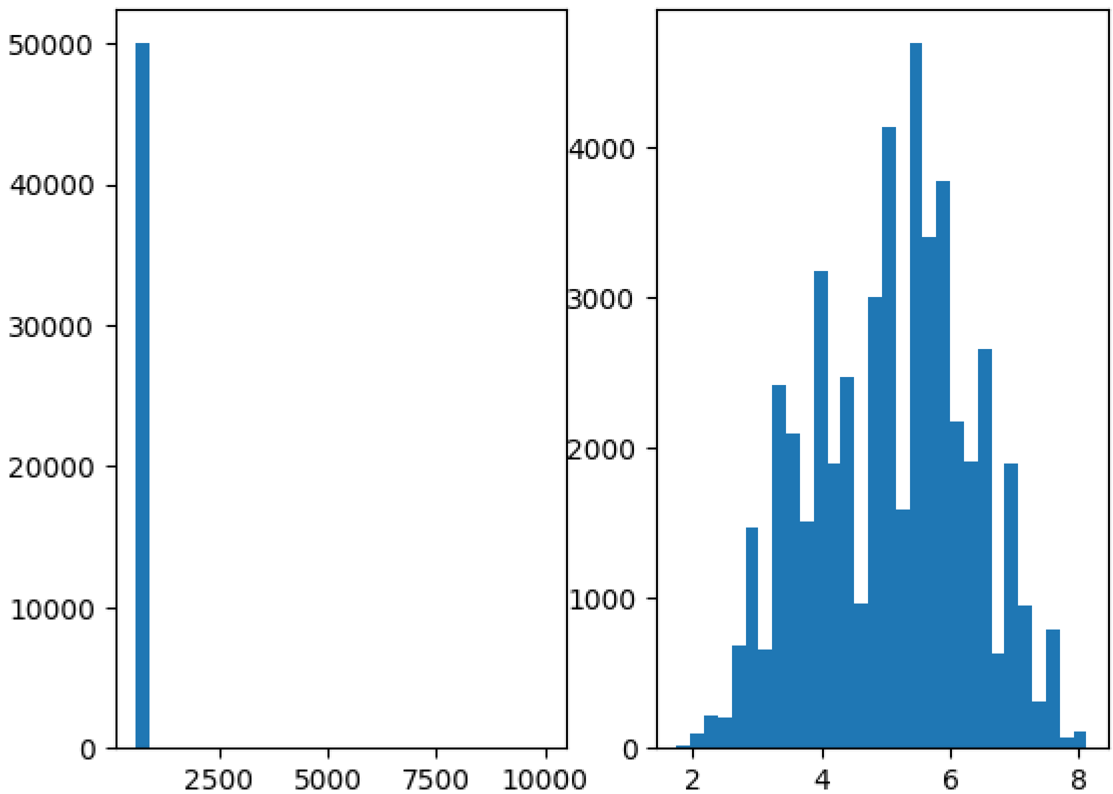
In [ ]:

```
import pandas as pd

var_names = ["Credit_Score","First_Payment_date","First_Time_Homebuyer", "Maturi
"MSA","Mortgage_Insurance_Percentage","Number_Units","Occupancy_Status","CLTV",
"DTI","UPB","LTV","Interest_Rate","Channel","Prepayment_Penalty",
"Amortization_Type","State","Property_Type","Postal_Code","Sequence_Number",
"Purpose","Loan_Term","Number_Borrowers","Seller_Name","Servicer_Name",
"Super_Conforming","Pre-HARP_Loan","Program_Indicator","HARP_Indicator",
"Valuation_Method","Interest_Only", "Insurance_cancellation"]

mysample = pd.read_table("C:/Users/shale/Downloads/sample_2022/sample_orig_2022.

print(mysample[0: 2])
```

```
   Credit_Score  First_Payment_date First_Time_Homebuyer  Maturity_Date  MSA  \
0           768              202203                    N         203702  NaN
1           781              202203                    N         205202  NaN

   Mortgage_Insurance_Percentage  Number_Units Occupancy_Status  CLTV  DTI  \
0                              0             1                P    57   28
1                              0             1                S    80   44

   ...  Number_Borrowers    Seller_Name    Servicer_Name Super_Conforming  \
0  ...                 1  Other sellers  Other servicers              NaN
1  ...                 2  Other sellers  Other servicers              NaN

  Pre-HARP_Loan Program_Indicator HARP_Indicator Valuation_Method  \
0           NaN                 9            NaN                2
1           NaN                 9            NaN                2

   Interest_Only Insurance_cancellation
0              N                      7
1              N                      7

[2 rows x 32 columns]
```

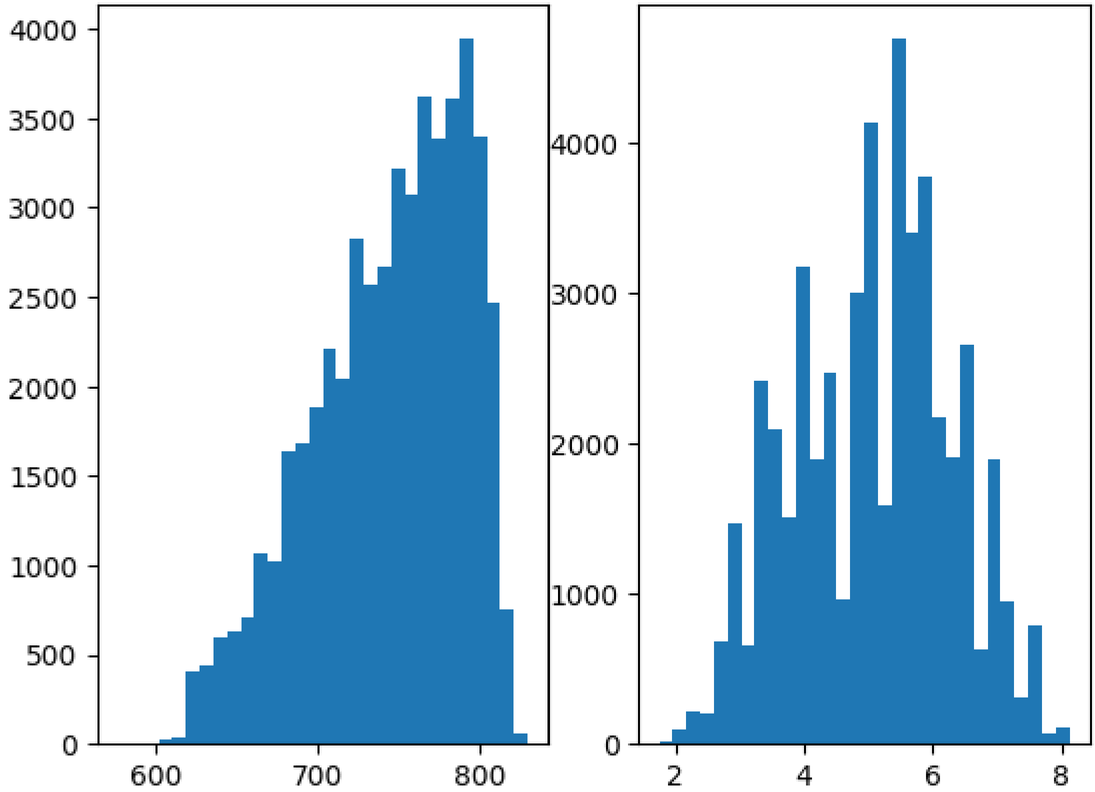
Question 4

In [ ]:
```
import matplotlib.pyplot as plt

figure, myplot = plt.subplots(1,2)

myplot[0].hist(mysample['Credit_Score'], bins = 30)
myplot[1].hist(mysample['Interest_Rate'], bins = 30)
plt.show()
```



a. There are outliers in the distribution of the first histogram showing the credit score, to the point where they dwarf the rest of the observations in the histogram. The second histogram does not appear to have outliers in the distribution.

In [ ]:
```
# b.

mysample = mysample[mysample['Credit_Score'] != 9999]
figure, myplot = plt.subplots(1,2)
myplot[0].hist(mysample['Credit_Score'], bins = 30)
myplot[1].hist(mysample['Interest_Rate'], bins = 30)
plt.show()
```



b. As seen above, the Freddie Mac convention does create a problem when creating histograms of the data. From the above histograms we can identify that neither distributions of the data contain outliers. The Credit Score distribution appears to be left-skewed, while the Interest Rate distribution follows a normal distribution, although with a few drops in the frequency of observations at some points and there are large spikes near the center of the distribution.

In [ ]:
```
import numpy as np
cs_std = np.std(mysample['Credit_Score'])
cs_mean = np.mean(mysample['Credit_Score'])
ir_std = np.std(mysample['Interest_Rate'])
ir_mean = np.mean(mysample['Interest_Rate'])

print("Sample Standard Deviation of Credit Score = " + str(cs_std))
print("Sample Mean of Credit Score = " + str(cs_mean))
print("Sample Standard Deviation of Interest Rate = " + str(ir_std))
print("Sample Mean of Interest Rate = " + str(ir_mean))
```

```
Sample Standard Deviation of Credit Score = 46.44319235495786
Sample Mean of Credit Score = 744.1436688474081
Sample Standard Deviation of Interest Rate = 1.2262844957900607
Sample Mean of Interest Rate = 5.083297121021148
```

c. We can observe that the credit score data has a moderately high standard deviation, indicating a relatively wide spread or dispersion of credit scores around the mean. The data being negatively skewed indicates that there are more observations with relatively lower values.

The sample standard deviation of the interest rate distribution is relatively low, indicating that interest rates in the sample are relatively close to the mean. This suggests that there is less variability in interest rates compared to credit scores. The data appears to follow a normal distribution.

For both distributions, the absence of outliers suggests that there are no extreme values that significantly deviate from the general pattern of the credit scores.

Question 5

In [ ]:
```
import scipy as sci
cs_k = sci.stats.kurtosis(mysample['Credit_Score'])
cs_skew = sci.stats.skew(mysample['Credit_Score'])

print("Sample Standard Deviation = " + str(cs_std))
print("Sample Mean = " + str(cs_mean))
print("Sample Skewness = " + str(cs_k))
print("Sample Kurtosis = " + str(cs_skew))
```

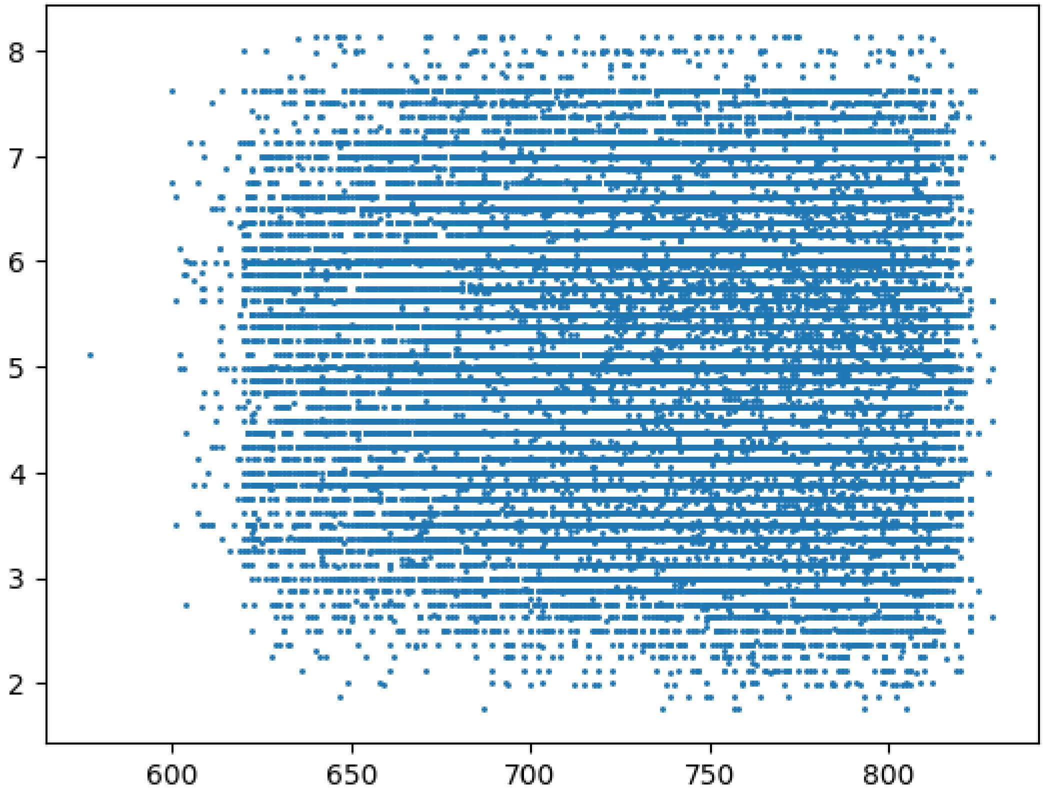
```
Sample Standard Deviation = 46.44319235495786
Sample Mean = 744.1436688474081
Sample Skewness = -0.44345816763045764
Sample Kurtosis = -0.5751367560552632
```

The data has a moderate spread around the mean, as indicated by the standard deviation and as was discussed previously.

The data is negatively skewed, which means there is a longer left tail, suggesting a higher density of lower values. The kurtosis value is negative, indicating lighter tails compared to a normal distribution and a less extreme distribution, although not extremely flat as the kurtosis is not very low.

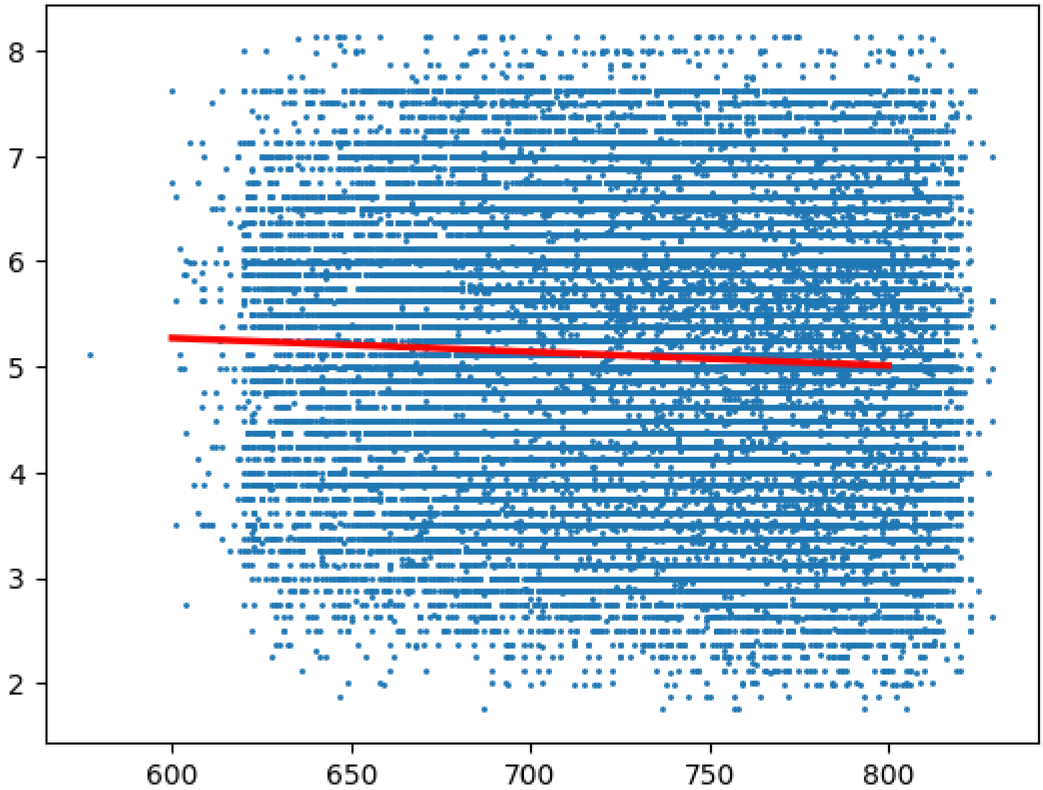
Question 6

In [ ]:
```
fig, ax = plt.subplots()

ax.scatter(mysample['Credit_Score'], mysample['Interest_Rate'], marker=".", s=5)
plt.show()
```



The plot does not indicate that there is any dependence between the two variables. We can plot a line of best fit and calculate the correlation to further validate this conclusion:

In [ ]:
```
fig, ax = plt.subplots()
b, a = np.polyfit(mysample['Credit_Score'], mysample['Interest_Rate'], deg=1)

ax.scatter(mysample['Credit_Score'], mysample['Interest_Rate'], marker=".", s=5)
xseq = np.linspace(600, 800, num=200)
ax.plot(xseq, a + b * xseq, color='red', lw=2.5)
plt.show()
corr = np.corrcoef(mysample['Credit_Score'], mysample['Interest_Rate'])
corrco = corr[0,1]

print("Correlation coeeficient = " + str(corrco))
```




```
Correlation coeeficient = -0.049363292630207146
```

From these statistics we can conclude that there is an extremely weak negative correlation between the two variables, although we might need to run a regression model to determine the statistical significance of the model.

Question 7

In [ ]:
```
s_xy = np.cov(mysample['Credit_Score'], mysample['Interest_Rate'])
s2_x = np.var(mysample['Credit_Score'])
beta1_hat = s_xy[0,1] / s2_x

y_bar = np.mean(mysample['Interest_Rate'])
x_bar = np.mean(mysample['Credit_Score'])
beta0_hat = y_bar - beta1_hat * x_bar

print("Intercept = " + str(beta0_hat))
print("Slope = " + str(beta1_hat))
```

```
Intercept = 6.05322350611977
Slope = -0.0013034128028004666
```

Question 8

The slope coefficient is negative, which indicates that when credit scores increase, the interest rate decreases. This is consistent with economic reasoning, a it is generally expected that there is an inverse relationship between an individual's creditworthiness (as represented by their credit score) and the interest rate they are offered on loans or credit products. This is because a high credit score shows that the likelihood of a debt default is relatively low. The lender may decide to give a lower interest rate because a borrower with a high score statistically poses less risk to the lender.

Question 9

In [ ]:
```
import statsmodels.formula.api as smf
results = smf.ols('Interest_Rate ~ Credit_Score', data=mysample).fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          Interest_Rate   R-squared:                       0.002
Model:                            OLS   Adj. R-squared:                  0.002
Method:                 Least Squares   F-statistic:                     122.1
Date:                Thu, 28 Sep 2023   Prob (F-statistic):           2.38e-28
Time:                        09:54:38   Log-Likelihood:                -81058.
No. Observations:               49983   AIC:                         1.621e+05
Df Residuals:                   49981   BIC:                         1.621e+05
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept        6.0532      0.088     68.825      0.000       5.881       6.226
Credit_Score    -0.0013      0.000    -11.049      0.000      -0.002      -0.001
==============================================================================
Omnibus:                     2954.745   Durbin-Watson:                   0.359
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1173.635
Skew:                          -0.113   Prob(JB):                    1.41e-255
Kurtosis:                       2.284   Cond. No.                     1.20e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.2e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

The R-squared in this regression model is 0.002, which indicates that approximately 0.2% of the variability in the data of the dependent variable can be explained by the independent variable. In this case, it means that credit scores have a very weak explanatory power when it comes to understanding interest rates, which means this model is not very practically significant, even if it is statistically significant. Most likely, there are other significant variables that also explain the variability in interest rates which should be included in the model to increase the R-squared value.