Multivariate Modeling
DATS 6450-15
Lab # 2
Taisha Ferguson
February 5, 2020 (TF)

Abstract

The primary purpose of this lab was the test the relation of different variables by calculating there correlation coefficients. The relationship was tested for given points X, Y, Z, G, and H and as well as with the variables from company sales dataset.

Introduction

The purpose of this lab was to examine the relationship between given variables by calculating the correlation coefficient. The correlation coefficient is a mathematical means to calculate relationships between variables by quantifying the strength as a value between 0 and 1 with stronger relationships being closer to 1 and weaker relationships being closer to 0. A correlation of 0 signifies no correlation and a correlation of 1 means that the variables are identical. The sign of the coefficient signifies the direction of the relationship. Variables with a negative sign are negatively correlated, meaning that as one variable increases the other variable decreases. Variables with a positive correlation coefficient have a positive relationship, meaning as one variable increases the other variable also increases.

Methods, theory, and procedures

As stated in the Introduction, the correlation coefficient of variables was used to examine the relationship between them. The following formula, which was given in the assignment instructions, was used to calculate the correlation coefficients:

$$r = \frac{\sum(x_t - \overline{x})(y_t - \overline{y})}{\sqrt{\sum(x_t - \overline{x})^2}\sqrt{\sum(y_t - \overline{y})^2}}$$

This formula was translated into Python function with the assistance of the Numpy and Math Python Libraries. After the correlation coefficient function was created it was applied to the 3 different variable pairings and then compared to scatter plots of the same variables.

The correlation function was also applied to a dataset of company Sales, Advertising Budget, and Product Pricing. As with the first set of variable pairs, the correlation coefficients were calculated and then compared to scatter plots of the same variables.

Lastly, the Sales data for the company data was tested to determine if it was stationary. This test was done using the ADF test from the Stats Model Python Library. The test results were also compared the histogram of the Sales data.

Answers to Asked Questions

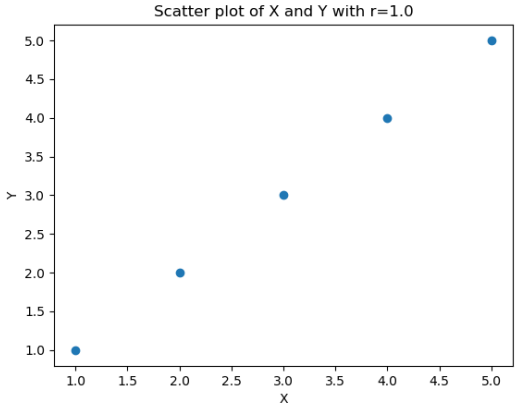
Q1: Compare the answer in section d and e. Any difference in value?

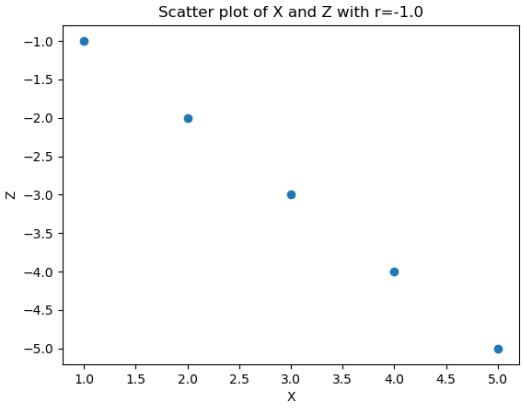
A1: There is no difference in my hand calculations of the correlation coefficients and the results from the python code. My values for both were:

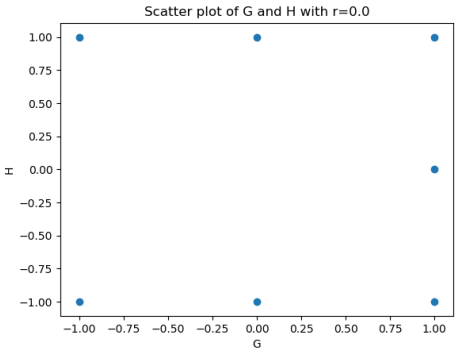
- R_xy = 1
- R_xz = -1
- R_gh = 0

Q2: Does the calculated r_xy, r_xz and r_gh make sense with respect to the scatter plots? Explain why?

A2: Yes the r values make sense with respect to the scatter plot. For X and Y the values are exactly the same meaning that they are positively correlated. For X and Z are inverse values and the graph is decreasing. For G and H there is no pattern and the correlation is zero.

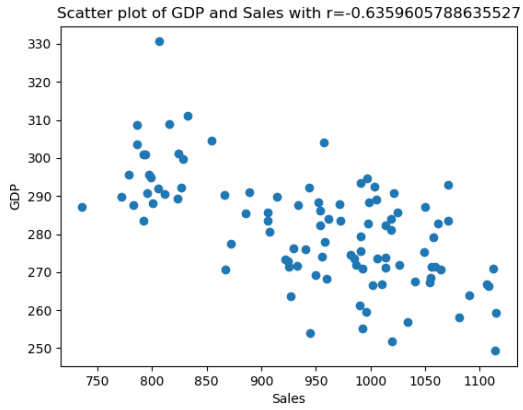






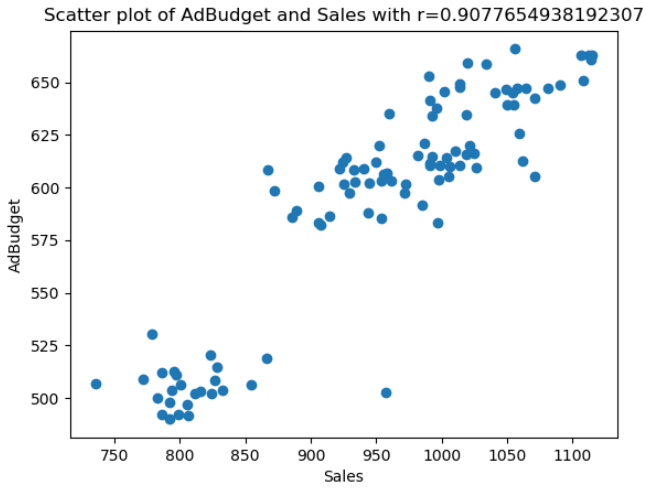
Q3: Does the r_xy value make sense with respect to the scatter plot graphed in step 7(Sales and GDP). Explain why?

A3: Yes the r value of -.64 makes sense with respect to the scatter plot of Sales and GDP. You can see from the plot that the correlation is negative because the graph is decreasing. You can also see that there is a strong pattern but it's not an exact match which corresponds to a value of .64 which is closer to 1 than 0.



Q4: Does the r_xy value make sense with respect to the scatter plot graphed in step 8(Sales and AdBudget). Explain why?

A4: Yes the r value of -.9 makes sense with respect to the scatter plot of Sales and AdBudget. You can see from the plot that the correlation is positive because the graph is increasing. You can also see that there is a very strong relationship which results in a value very close to 1.

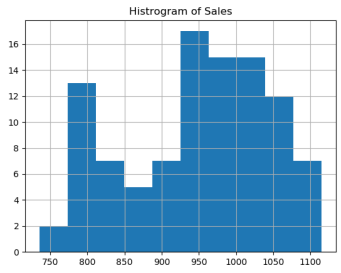


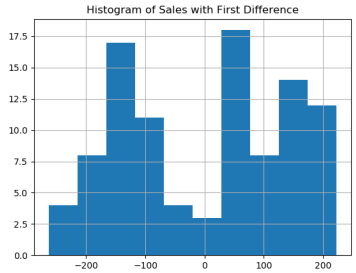
Q5: By looking at the correlation coefficients, write down your observation about the effect of AdBudget data and GDP data on the Sales revenue?

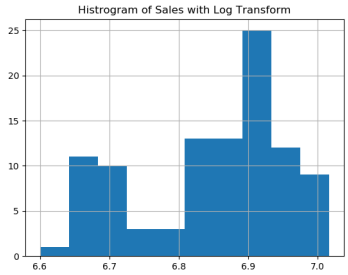
A5: Both AdBudget and GDP have a correlation on Sales revenue. AdBudget has a very strong positive relationship with Sales so as AdBudget increases Sales also tend to increase. GDP has a negative correlation and the strength of that relationship is moderate. As GPD increases Sales tends to decrease

Q6: Which Sales dataset is stationary and which Sales dataset is non-stationary?Justify your answer according to the ADF-Statistics and the histogram plot.

A6: Based on the results of the ADF tests and the histograms all of the different forms of the sales data are stationary. The most stationary form of the data is the data after the First difference because the p value of the ADF test is less than 1%







Sales Data	
ADF Statistic:	:-3.262755
p-value:	0.016628
Critical Values:	
	1%: -3.505
	5%: -2.894
	10%: -2.584

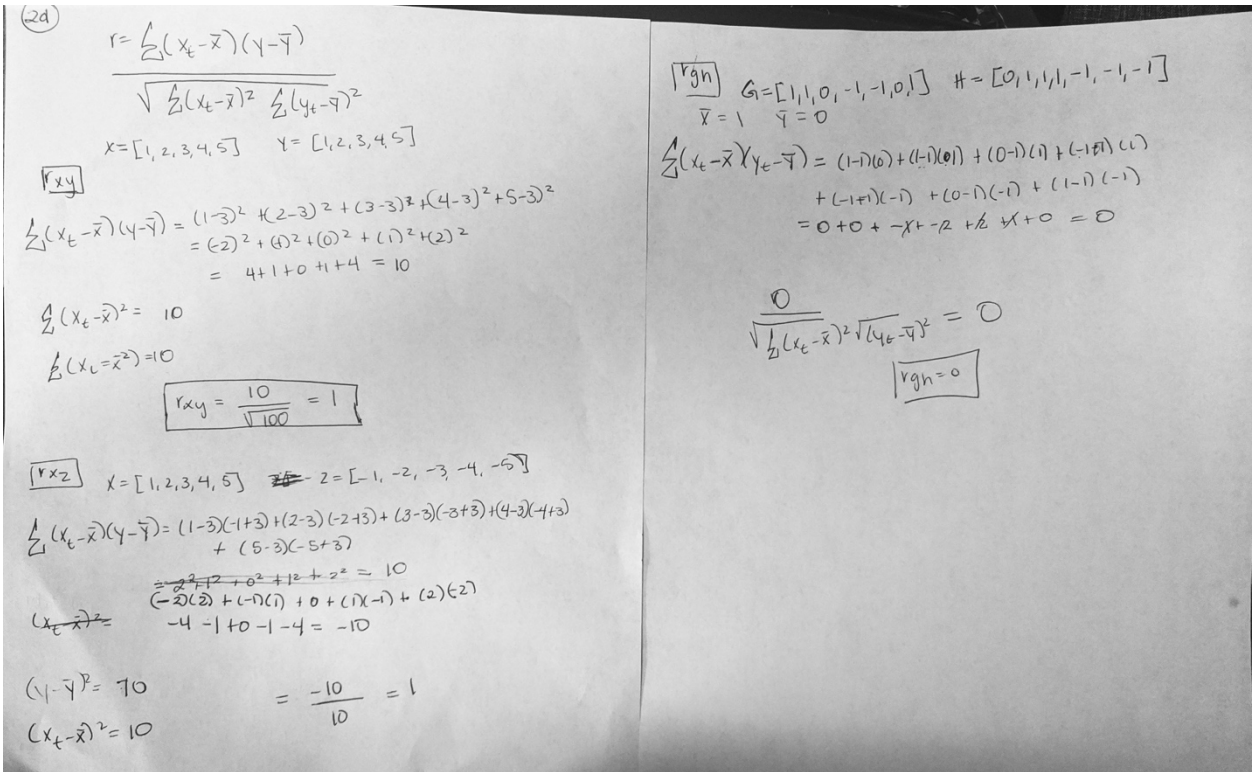
Sales Data After First Difference	
ADF Statistic:	-5.100287
p-value:	0.000014
Critical Values:	
	1%: -3.507
	5%: -2.895
	10%: -2.585

Sales Data After Log Transform	
ADF Statistic:	-2.944119
p-value:	0.040453
Critical Values:	
	1%: -3.504
	5%: -2.894
	10%: -2.584

Conclusion

The main objective of this lab was to test the relationship of different variables by calculating their correlation coefficients. This method was used on given data points as well as on variables from a time series data on a companies sales. In both cases the correlation coefficients corresponded with scatter plots of the corresponding variables.

Appendix A – Handwritten Correlation Computation



Appendix B – Python Code

```
# Import Libraries
import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller

# 1 - Write a python function called " correlation_coefficent_cal(x,y)"
# that implement the correlation coefficient.
# The formula for correlation coefficientis given below. The function should be
written
# in a general form than can work for any dataset x and dataset y.
# The return value for this function is r.

def correlation_coefficent_cal(x,y):
    n= len(x)
    x_bar = np.sum(x) / len(x)
    y_bar = np.sum(y) / len(y)
    numerator = 0
    denominator1 = 0
    denominator2 = 0
    i = 0
    while i < n:
        numerator = numerator + ((x[i] - x_bar) * (y[i] - y_bar))
        denominator1 = denominator1 + (x[i] - x_bar)*(x[i] - x_bar)
        denominator2 = denominator2 + (y[i] - y_bar)*(y[i] - y_bar)
        i=i+1
    r = numerator /(math.sqrt(denominator1*denominator2))
    return r
```

```
# 2 - Test the " correlation_coefficent_cal(x,y)"functionwith the following simple
dataset.
# The x and y here are dummy variable and should be replaced by any other dataset.

X = [1,2,3,4,5]
Y = [1,2,3,4,5]
Z = [-1,-2,-3,-4,-5]
G = [1,1,0,-1,-1,0,1]
H = [0,1,1,1,-1,-1,-1]

# a - Plot the scatter plot between X and Y
r_xy = correlation_coefficent_cal(X, Y)
plt.scatter(X,Y)
plt.xlabel("X")
plt.ylabel("Y")
plt.title("Scatter plot of X and Y with r={}".format(r_xy))
plt.show()

# b - Plot the scatter plot between X and Z

r_xz = correlation_coefficent_cal(X, Z)
plt.scatter(X,Z)
plt.xlabel("X")
plt.ylabel("Z")
plt.title("Scatter plot of X and Z with r={}".format(r_xz))
plt.show()


# c - Plot the scatter between G and H

r_gh = correlation_coefficent_cal(G, H)
plt.scatter(G,H)
plt.xlabel("G")
plt.ylabel("H")
plt.title("Scatter plot of G and H with r={}".format(r_gh))
plt.show()

# e - Calculate r_xy , r_xz  and r_gh using the written function "
correlation_coefficent_cal(x,y)".

print('The correlation coefficient between X and Y is ', r_xy)
print('The correlation coefficient between X and Z is ', r_xz)
print('The correlation coefficient between G and H is ', r_gh)


# 3 - Load the time series data tute1.

df= pd.read_csv('tute1.csv', index_col="Date")
print(df.head())
print(df.info())
df=df[['Sales', 'AdBudget', 'GDP']]
print(df.head())
print(df.tail())


# 6 - Plot Sales, AdBudget and GDP versus time steps
ax = plt.gca()
df["Sales"].plot(y='Sales',ax=ax, legend=True, title= "Sales, AdBUdget, and GDP (March
1, 1981- June 8, 1981)")
df["AdBudget"].plot(y='AdBudget',ax=ax, legend=True)
```

```
df['GDP'].plot(y='GDP', ax=ax, legend=True)
plt.show()


# 7 - Graph the scatter plot for Sales and GDP. (y-axis plot Sales and x-axis plot
GDP).
#Add the appropriate x-label and y-label. Don't add any title i this step. This needs
to be updated in step 11.
plt.scatter(df['Sales'],df['GDP'])
plt.xlabel("Sales")
plt.ylabel("GDP")
plt.show()

# 8 - Graph the scatter plot for Sales and AdBudget. (y-axis plot Sales and x-axis
plot AdBudget).
#Add the appropriate x-label and y-label. Don't add any title i this step. This needs
to be updated in step 11.
plt.scatter(df['Sales'],df['AdBudget'])
plt.xlabel("Sales")
plt.ylabel("AdBudget")
plt.show()

# 9 - Call the function correlation_coefficient_cal(x,y) with the y as the Sales data
and the x as the GDP data.
#Save the correlation coefficient between these two variables as r_xy.

r_xy = correlation_coefficent_cal(df['GDP'], df['Sales'])
print("The correlation coefficient between Sales value and GDP is ",r_xy)



# 10 - Call the function correlation_coefficient_cal(y,z) with the y as the Sales data
and the z as the AdBudget data.
#Save the correlation coefficient between these two variables as r_yz.

r_yz = correlation_coefficent_cal(df['AdBudget'], df['Sales'])
print("The correlation coefficient between Sales value and AdBuget is ",r_yz)


# 11 - Include the r_xy and r_yz in the title of the graphs developed in step 5 and 6.
# Write your code in a way that anytime r_xy and r_yz value changes it automatically
updated on the figure title.

plt.scatter(df['Sales'],df['GDP'])
plt.xlabel("Sales")
plt.ylabel("GDP")
plt.title("Scatter plot of GDP and Sales with r={}".format(r_xy))
plt.show()


plt.scatter(df['Sales'],df['AdBudget'])
plt.xlabel("Sales")
plt.ylabel("AdBudget")
plt.title("Scatter plot of AdBudget and Sales with r={}".format(r_yz))
plt.show()

# 13 - Performthe ADF-test and plot the histogram plot on the raw Salesdata,
# first order difference Sales dataand the logarithmictransformationof the Sales data.
# Which Sales dataset is stationary and which Sales dataset is non-stationary?Justify
your answer
# according tothe ADF-Statistics and the histogramplot.
```

```
#Raw Sales Data
def ADF_Cal(x):
    result = adfuller(x)
    print("ADF Statistic: %f" %result[0])
    print("p-value: %f" % result[1])
    print("Critical Values:")
    for key, value in result[4].items():
        print('\t%s: %.3f' %(key, value))

df['Sales'].hist()
plt.title("Histrogram of Sales")
plt.show()
ADF_Cal(df['Sales'])

#Sales First Difference
firstorderdifference = []
for i in range(len(df['Sales'])):
    difference=df['Sales'][i]-df['Sales'][i-1]
    firstorderdifference.append(difference)
df['FirstOrderDifference'] = firstorderdifference
df1 = df.drop(index='3/1/1981')
print(df1.head())
df1['FirstOrderDifference'].hist()
plt.title("Histogram of Sales with First Difference")
plt.show()
ADF_Cal(df1['FirstOrderDifference'])

#LogTransform
df["LogSales"] = np.log(df['Sales'])
print(df.head())
df['LogSales'].hist()
plt.title("Histrogram of Sales with Log Transform")
plt.show()
ADF_Cal(df['LogSales'])
```