Multivariate Modeling
DATS 6450-15
Lab # 1
Taisha Ferguson
January 29, 2020 (TF)

Abstract

The primary purpose of this lab was the test the whether or not the variables in given time series dataset was stationary or non-stationary. The data set contained daily company data from March 1 1981 through June 8 1981. In order for a variable to be stationary the statistical measures such as mean and variance should be mostly constant through time. After performing graphical and statistical tests it was determined that the Sales and GDP variables were stationary but AdBuget variable was non-stationary.

Introduction

The purpose of this experiment was to determine the whether or not three variables (Sales, AdBudget, and GDP) in the given were stationary or non-stationary. To determine this, I graphed all variables against time, graphed their averages overtime, variance over time, and performed the Augmented Dickey-Fuller (ADF) test.

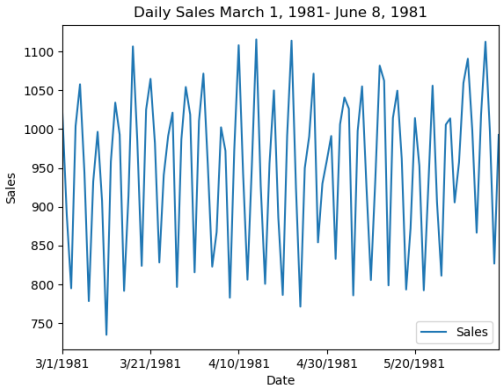
Methods, theory, and procedures

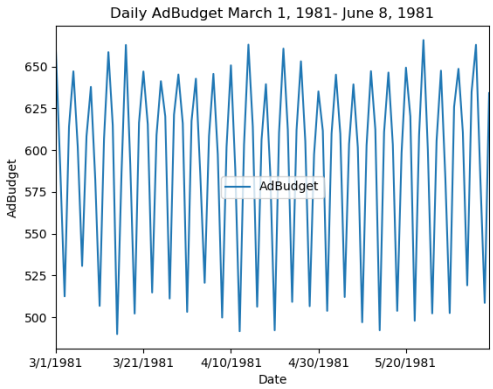
In this lab 3 variables from given dataset were tested to determine if they were stationary or non-stationary. Stationary data is a requirement for many time series models and is categorized as having statistical measures such as mean and variance that are mostly constant through time. The first step in testing the variables was plotting each one against time to see if there were any specific trends present. After plotting each variable they all seemed have fairly stationary data because there didn't appear to be any significant fluctuations in the mean and variance. The second step was calculation the mean and variance over time and then plotting. In all three cases the mean and variance appeared to stabilize over time. The last step was using the ADF test to confirm observations. Based on the ADF test Sales and GDP were stationary but AdBudget was non-stationary. AdBudget's p-value was 6.4% which is to high to reject the null hypothesis that is non-stationary.

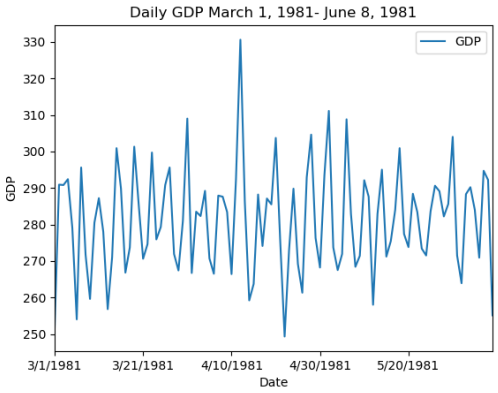
Answers to Asked Questions

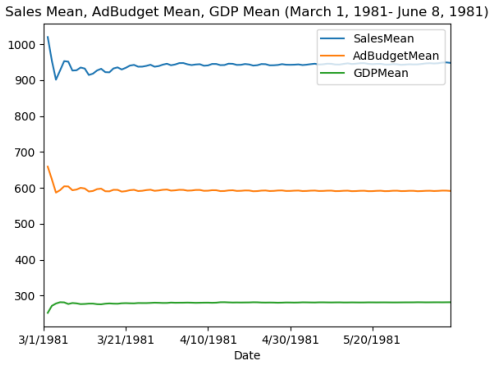
Q1: What is your observation of the time series data is it stationary or not? Why?

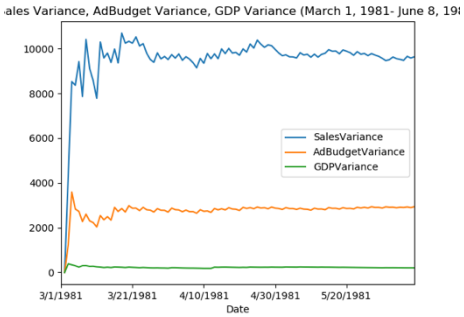
A1: Based on the graphs below, Sales, AdBudget, and GDP all appear to be stationary because there is not a significant amount of fluctuation in the mean and variances over time.











Q2: Perform ADF-test to check if Sales, AdBudget, and GDP are stationary or not. Does your answer for this question reinforce your observations in the previous step?

A2: The ADF tests below confirm by observation that Sales and GDP are in fact stationary. However, it did not confirm my hypothesis that ADF is stationary because the p-value is above the .05 threshold.

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

AdBudget
ADF Statistic: -2.758605
p-value: 0.064434
Critical Values:
1%: -3.504
5%: -2.894
10%: -2.584

GDP
ADF Statistic: -3.227577
p-value: 0.018443
Critical Values:
1%: -3.504
5%: -2.894
10%: -2.584

Conclusion

The objective of this lab was to test the whether the 3 variables Sales, AdBudget, and GDP from the given dataset were stationary or non-stationary. After performing graphical test and the statistical ADF test it was determined that Sales and GDP were stationary and AdBudget were non-stationary.

Appendix

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

# 1 – Load the time series data called 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())

# 4 – Plot Sales, AdBudget and GPD versus timestep.
#Sales

ax = df["Sales"].plot(legend=True, title= "Daily Sales March 1, 1981– June 8, 1981")
ax.set_ylabel("Sales")
plt.show()
#AdBudget

ax=df['AdBudget'].plot(legend=True, title="Daily AdBudget March 1, 1981– June 8,
1981")
ax.set_ylabel("AdBudget")
plt.show()
#GDP

ax=df['GDP'].plot(legend=True, title="Daily GDP March 1, 1981– June 8, 1981")
```

```
ax.set_ylabel('GDP')
plt.show()

# 5 & 6 - Find the time series statistics (average, variance and standard deviation)
of Sales, AdBudget and GDP
print("The Sales mean is:",df['Sales'].mean(), "and the variance
is:",df['Sales'].var(), "with standard deviation:",
df['Sales'].std())
print("The AdBudget mean is:",df['AdBudget'].mean(), "and the variance
is:",df['AdBudget'].var(), "with standard deviation:",
df['AdBudget'].std())
print("The GDP mean is:",df['GDP'].mean(), "and the variance is:",df['GDP'].var(),
"with standard deviation:",
df['GDP'].std())


# 7 - Prove that the Sales, AdBudgetand GDP in the this time series datasetis
stationary
salesmean=[]
salesvariance=[]
admean=[]
advariance=[]
gdpmean=[]
gdpvariance=[]
salesmean=[]
for i in range(len(df.Sales)):
    #Sales
    smean=np.mean(df.Sales[:i])
    salesmean.append(smean)
    svar = np.var(df.Sales[:i])
    salesvariance.append(svar)
    #AdBudget
    amean = np.mean(df.AdBudget[:i])
    admean.append(amean)
    avar = np.var(df.AdBudget[:i])
    advariance.append(avar)
    #GDP
    gmean = np.mean(df.GDP[:i])
    gdpmean.append(gmean)
    gvar = np.var(df.GDP[:i])
    gdpvariance.append(gvar)
df["SalesMean"] = salesmean
df['SalesVariance'] = salesvariance
df["AdBudgetMean"] = admean
df['AdBudgetVariance'] = advariance
df["GDPMean"] = gdpmean
df['GDPVariance'] = gdpvariance
print(df.head())

# 8 - Plot all the means and variances.
#Means
ax = plt.gca()
df["SalesMean"].plot(y='Sales',ax=ax, legend=True, title= "Sales Mean, AdBudget Mean,
GDP Mean (March 1, 1981- June 8, 1981)")
df["AdBudgetMean"].plot(y='AdBudget',ax=ax, legend=True)
df['GDPMean'].plot(y='GDP', ax=ax, legend=True)
plt.show()
#Variances
ax = plt.gca()
df["SalesVariance"].plot(y='Sales',ax=ax, legend=True, title= "Sales Variance,
AdBudget Variance, GDP Variance (March 1, 1981- June 8, 1981)")
```

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

# 9 - Perform an ADF-test to check if the Sales, AdBudget and GDP stationary or not.
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))
ADF_Cal(df['Sales'])
ADF_Cal(df['AdBudget'])
ADF_Cal(df['GDP'])
```