PYTHON I FINAL PROJECT

Project Guidelines

In this project, our program will use the exponential smoothing to help us predict a value in the future. In addition, our program should come up with the linear regression equation to predict the same value as was done in exponential smoothing.

In part 1, we obtain the EUR-USD currency exchange rate. In part 2, we write a Python program that asks the user for the information from part 1 and performs exponential smoothing based on it. The algorithm of our program allows the user to input alpha, display the graph of the original data and the "smoothed data" and have the user verify if this model is appropriate. If it is not then it should loop asking for new entries for alpha until the user indicates the model is appropriate. In part 3, the same Python program is used to develop a linear regression model that is used to predict time period selected. It shows the correlation coefficient to indicate the strength of the model. No other tool is necessary for this project to test the appropriateness of using a linear regression model.

Alternatively the same model was designed by using Excel spreadsheet to check if the algorithm and output of Python-3 program are accurate.

The Python-3 program and its outputs as well as Excel spreadsheets used for the implementation of regression analysis, CSV worksheets constitute a fully workable version of this project.

Research Questions

This regression analysis and evaluations by using the environments of Python-3 and Excel spreadsheet should consist of answer on the following questions:

1. Problems with Data Collection.

2. Logic of Exponential Smoothing.

3. Visual representation of smoothed data

4. Regression Logic

Project Activities

Foreign exchange currency data for the last three years have been downloaded from [Quandl](). Daily currency exchange rates of EUR-USD selected for the period under study have been used for the regression analysis. The best line fit was constructed and visualized using Python graphical capabilities.

Activity Progress

Python-3 code/output

Part 1.

```
# -*- coding: utf-8 -*-
"""
Created on Thu Nov 30 14:35:00 2017

@author: zabdullazade
"""

import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
import pandas as pd
# Load CSV and columns
df = pd.read_csv("C:/WQU/WQU 605 Programming in Python I/Currency.csv")
```

Part 2.

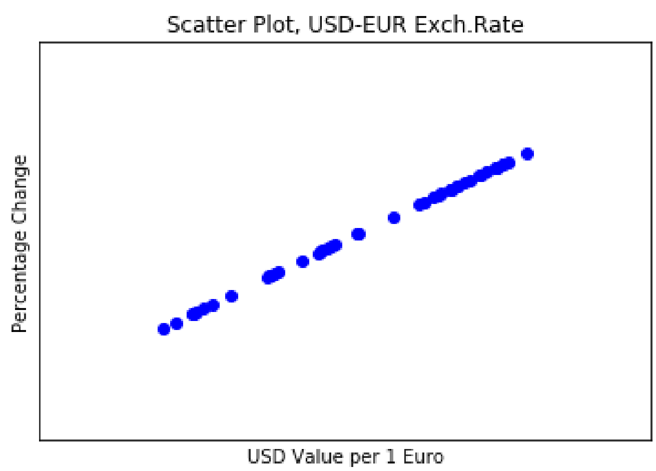
```
alpha = eval(input('Enter α(alpha): '))
```

Enter α(alpha): 0.25

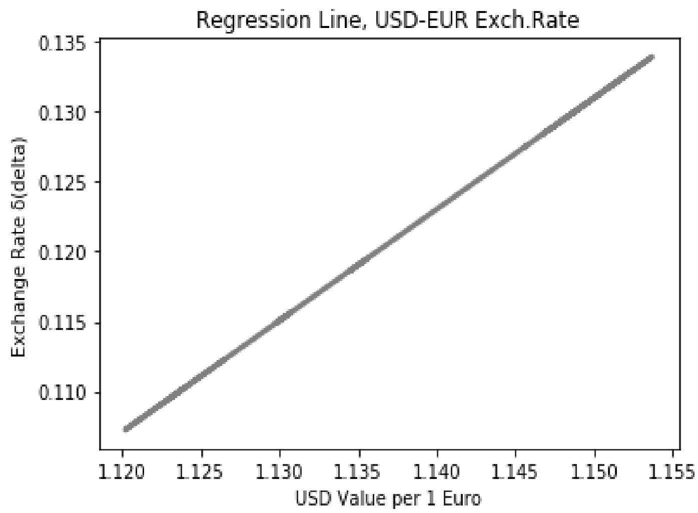
```
X = alpha*df['USD']+(1-alpha)*(1.1203539-0.0000093)
Y = (X-1)/X
X=X.reshape(len(X),1)
Y=Y.reshape(len(Y),1)
# Split the data into training/testing sets
X_train = X[:-45]
X_test = X[-45:]
# Split the targets into training/testing sets
Y_train = Y[:-45]
Y_test = Y[-45:]
```

Part 3.

```
# Plot outputs
plt.scatter(X_test, Y_test, color='blue')
plt.title('Scatter Plot, USD-EUR Exch.Rate')
plt.xlabel('USD Value per 1 Euro')
plt.ylabel('Percentage Change')
plt.xticks(())
plt.yticks(())

plt.show()


# Create linear regression object
regr = linear_model.LinearRegression()
# Train the model using the training sets
regr.fit(X_train, Y_train)

print('_______________')
print('α (alpha) =',alpha)

# Plot outputs
plt.plot(X_test, regr.predict(X_test), color='grey',linewidth=3)
plt.title('Regression Line, USD-EUR Exch.Rate')
plt.xlabel('USD Value per 1 Euro')
plt.ylabel('Exchange Rate δ(delta)')
```



α (alpha) = 0.25



R-Cran (Studio) code/output

Part 1.

```
> # begin
> setwd("C:/WQU/WQU 605 Programming in Python I/")
> mydata = read.csv("Currency.csv")
> mydata
     Period    USD
1   11/29/2017 1.1827
2   11/28/2017 1.1888
3   11/27/2017 1.1952
4   11/24/2017 1.1877

...
```

Part 2.

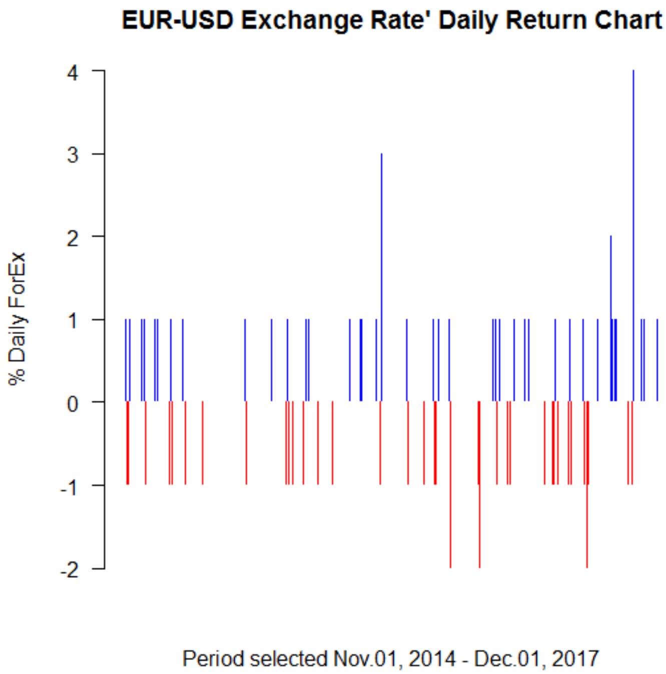
```
> ForEx = data.frame(
+   lnLength = c(mydata1$USD),
+   lnWeight = c(mydata1$USD)
+ )
> color = c("red", "blue")
> plot(lnWeight ~ lnLength, data = ForEx,
+   xlab = "USD Value per 1 Euro",
+   ylab = "Exchange Rate ?(delta)",
+   main = "Daily ForEx Rate Scatter Plot", col=color
+ )
> ForEx.mod = lm(lnWeight ~ lnLength, data = ForEx)
> summary(ForEx.mod)
```

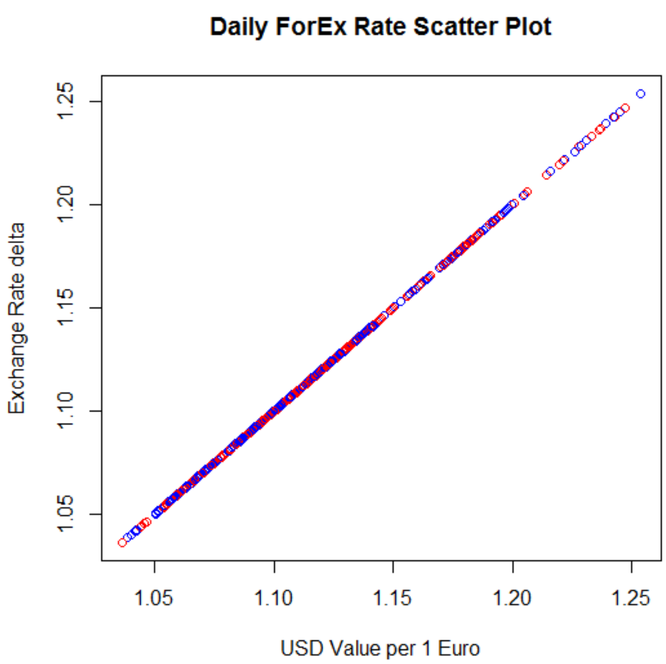
Part 3.

Statistical summary and graphics are given below

```
Call:
lm(formula = lnWeight ~ lnLength, data = ForEx)


Residuals:
       Min         1Q     Median         3Q        Max
-3.313e-17 -2.090e-18 -1.120e-18  2.300e-19  8.830e-16

Coefficients:
             Estimate Std. Error    t value Pr(>|t|)
(Intercept) -1.025e-15  3.129e-17 -3.275e+01   <2e-16 ***
```

```
lnLength     1.000e+00  2.800e-17  3.571e+16   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.211e-17 on 767 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 1.275e+33 on 1 and 767 DF,  p-value: < 2.2e-16

Warning message:
In summary.lm(ForEx.mod) :
  essentially perfect fit: summary may be unreliable
> print(ForEx.mod)

Call:
lm(formula = lnWeight ~ lnLength, data = ForEx)

Coefficients:
(Intercept)     lnLength
 -1.025e-15    1.000e+00
> x <- 2:5
> y <- 1.1203539*3-0.0000093
> plot(x,y,type="o",pch=20,lty=3, xlab = "EUR Value per 1 USD",ylab = "USD Value per 1 EUR",
+ main = "Foreign Exchange Rate, USD-EUR",col="blue")
```


EUR-USD Exchange Rate' Daily Return Chart
% Daily ForEx
4
3
2
1
0
-1
-2
Period selected Nov.01, 2014 - Dec.01, 2017


Daily ForEx Rate Scatter Plot
Exchange Rate delta
1.05
1.10
1.15
1.20
1.25
USD Value per 1 Euro

Excel spreasheet code/output

Part 1.

Date	Period (days)	USD Rate
2017-11-29	1	1.1827
2017-11-28	2	1.1888
2017-11-27	3	1.1952
2017-11-24	4	1.1877

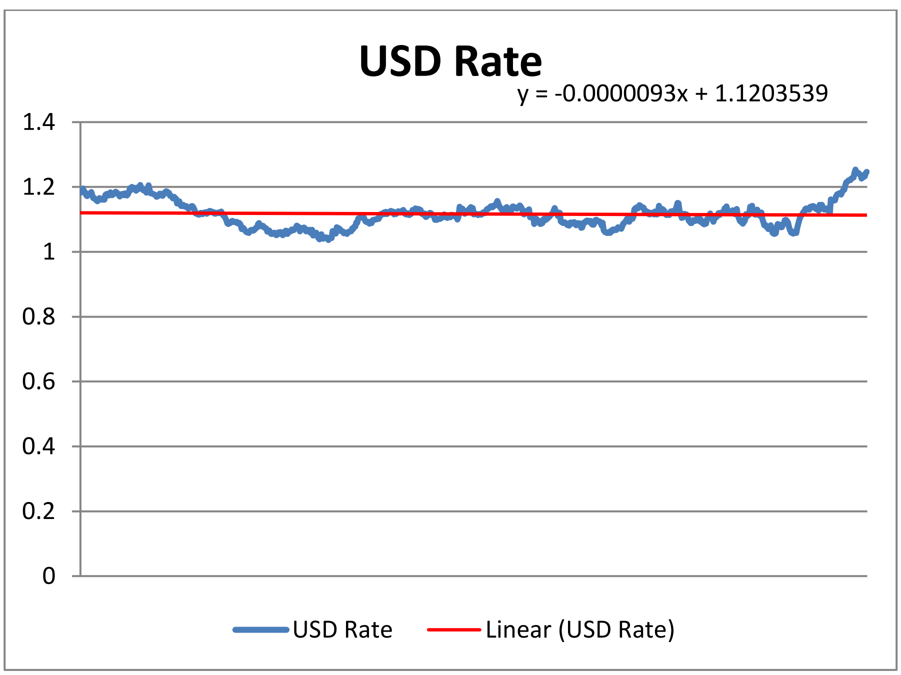
…

Part 2.

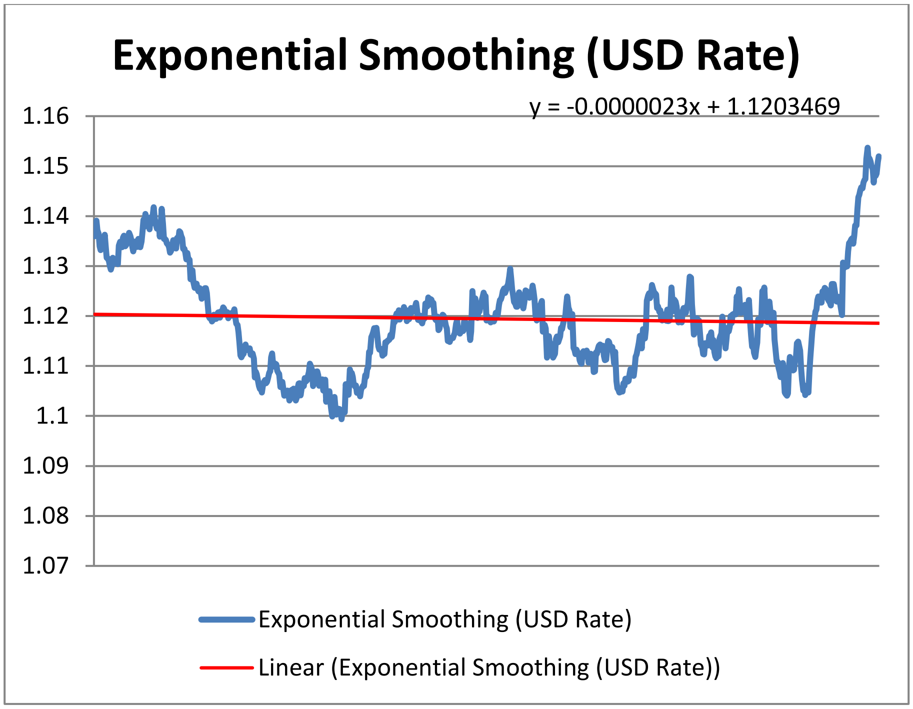
EUR/USD exchange rate statistics	
Average price	1.12
Minimum price	1.04
Maximum price	1.25

Monte Carlo simulation

Exp(Exchange) 1.145379

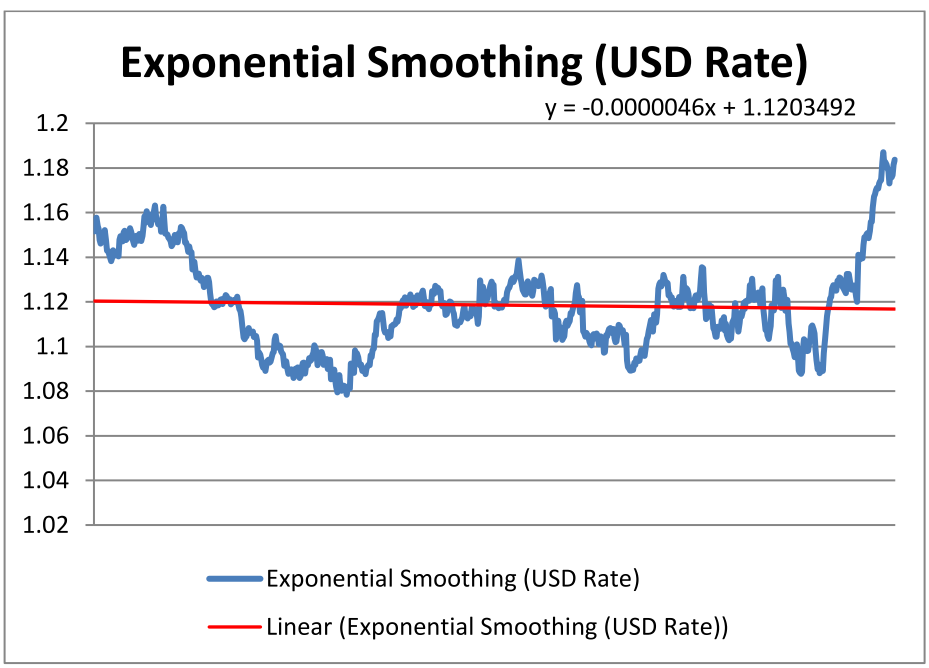


α = 0.25



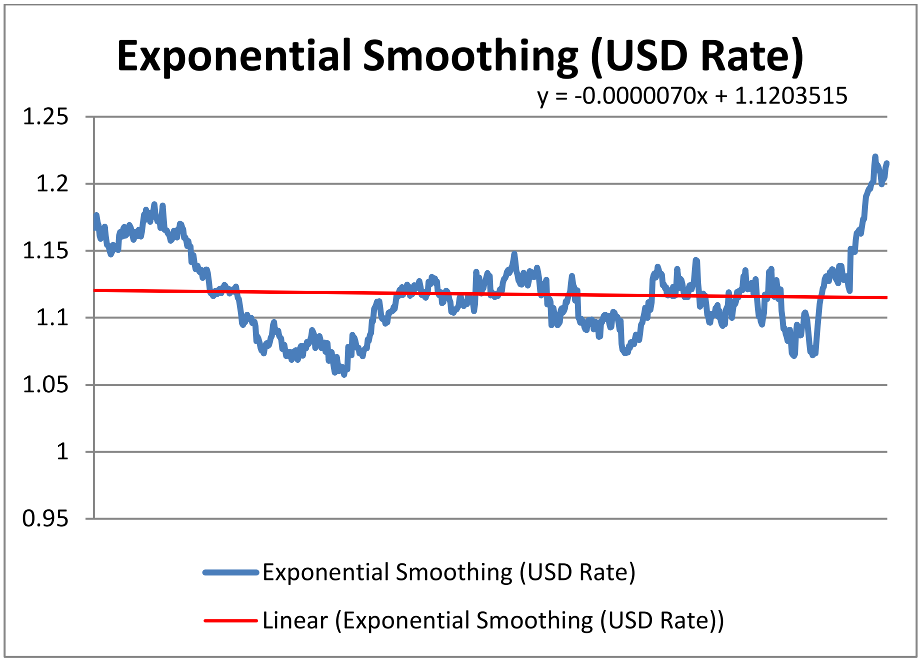
Exponential Smoothing (USD Rate)
y = -0.0000023x + 1.1203469
1.16
1.15
1.14
1.13
1.12
1.11
1.1
1.09
1.08
1.07
Exponential Smoothing (USD Rate)
Linear (Exponential Smoothing (USD Rate))

α = 0.50



Exponential Smoothing (USD Rate)
y = -0.0000046x + 1.1203492
1.2
1.18
1.16
1.14
1.12
1.1
1.08
1.06
1.04
1.02
Exponential Smoothing (USD Rate)
Linear (Exponential Smoothing (USD Rate))

α = 0.75



Conclusion and Regression Logic

By analyzing both regression and exponential smoothing method deliverables, we can notice that simple regression indicates different magnitudes between the related currency exchange rates and time line. The cross-sectional and time-series analyses combined as panel data and plotted on graph with regression and exponential smoothing has less volatility and better prediction when α, exponential smoothing factor is equal to 0.25. That is the more weight is assigned to old data patterns, the better prediction will be introduced in our model. Of course, in the ideal world we would use linear regression by itself, but sometimes not all past data may be available for us to look into.

Hence, exponential smoothing performed by selecting the lowest weight for factor assigned to the recent data and highest weight for factor defining the old data should be logical and result in less volatile conditions in terms or modeling and/or forecasting the future trends in finance.