Econometrics 322 Lab #7

Self-identified Problem

Prof. Pavlyukevich

Enter your Name in the Next Cell

Weiman Yanker

Grading Rubric

Grade: ___ / Max 20 - Total Deductions:

Deduction Type	Deduction	Times Occurred	Total	Comments
Missing Section				
Missing Section Header	-1		0	
Missing Text	-1		0	
Bad Grammar, Misspelling	-1		0	
Poor Formatting	-1		0	
Missing Plots	-1		0	
Illegible Plot	-1		0	
Illegible	-1		0	
Missing Code	-1		0	
Illegible Code	-1		0	
Missing Pkg Lib/Dl	-1		0	
Missing Pkg Lib/Dl	-1		0	
Poor Code	-1		0	
Poor Code	-1		0	
Inefficient Pkg Install	-1	0 Max	0	
Exceptions	-1		0	
Missing Pkg Lib/Dl	-1		0	
Missing Pkg Lib/Dl	-1		0	
Missing Pkg Lib/Dl	-1		0	
Missing	-1		0	
Not in Interval (Absolute/Interpretation)	-2		0	
Not in Interval (Absolute/Interpretation)	-2	0 Max	0	
Missing Conclusion	-2		0	
F Statistic	-2		0	
Incorrect Hypothesis H0	-2		0	
Incorrect Hypothesis H1	-2		0	
Incorrect Hypothesis Test	-2		0	
Incorrect Test	-2		0	
Missing Conclusion	-2		0	
Missing Conclusion	-2		0	
Missing Conclusion

```
# Plot CPI data
plot_cpi <- ggplot(cpi, aligned = TRUE, xlab = "Year", ylab = "Consumer Price Index (CPI)",
  main = "Consumer Price Index (CPI) ")
plot_cpi
```

```
# Plot GDP data
plot_gdp <- ggplot(gdp, aligned = TRUE, xlab = "Year", ylab = "Gross GDP (Billions)",
  main = "US Gross Domestic Product (Billions of Dollars)")
plot_gdp
```

```
# Plot Consumer Price Index (CPI)
plot_cpi <- ggplot(cpi, aligned = TRUE, xlab = "Year", ylab = "Consumer Price Index (CPI)",
  main = "Consumer Price Index (CPI) ")
plot_cpi
```

```
# Plot Gross Domestic Product (GDP)
plot_gdp <- ggplot(gdp, aligned = TRUE, xlab = "Year", ylab = "Gross GDP (Billions)",
  main = "US Gross Domestic Product (Billions of Dollars)")
plot_gdp
```

```
# Plot Consumer Price Index (CPI) in Chained Dollars
plot_cpi <- ggplot(cpi, aligned = TRUE, xlab = "Year", ylab = "Consumer Price Index (CPI) in Chained Dollars",
  main = "Consumer Price Index (CPI) in Chained Dollars")
plot_cpi
```

```
# Plot Gross Domestic Product (GDP) in Chained Dollars
plot_gdp <- ggplot(gdp, aligned = TRUE, xlab = "Year", ylab = "Gross GDP (Billions) in Chained Dollars",
  main = "Gross Domestic Product (GDP) in Chained Dollars")
plot_gdp
```

```
# Plot Gross Domestic Product (GDP) Growth Rate (%)
plot_gdp <- ggplot(gdp, aligned = TRUE, xlab = "Year", ylab = "Gross GDP Growth Rate (%)",
  main = "Gross Domestic Product (GDP) Growth Rate (%)")
plot_gdp
```

```
# Plot Gross Domestic Product (GDP) Growth Rate (%) in Chained Dollars
plot_gdp <- ggplot(gdp, aligned = TRUE, xlab = "Year", ylab = "Gross GDP Growth Rate (%) in Chained Dollars",
  main = "Gross Domestic Product (GDP) Growth Rate (%) in Chained Dollars")
plot_gdp
```

```
# Plot Gross Domestic Product (GDP) Growth Rate (%) in Chained Dollars
plot_gdp <- ggplot(gdp, aligned = TRUE, xlab = "Year", ylab = "Gross GDP Growth Rate (%) in Chained Dollars",
  main = "US Gross Domestic Product (GDP) Growth Rate (%) in Chained Dollars")
plot_gdp
```


Merge Data

```
# Merge data
cpi_1 = cpi[cpi$year >= 1970, c("year", "cpi")]
cpi_1 = cpi[order(cpi_1$year), ]
cpi_1$cpi_per_month = cpi_1$cpi/12
cpi_1$cpi_per_month[1:5]
gdp_1 = gdp[gdp$year >= 1970, c("year", "gdp")]
gdp_1 = gdp[order(gdp_1$year), ]
gdp_1$gdp_per_month = gdp_1$gdp/12
gdp_1$gdp_per_month[1:5]
```

```
>>> OLS Regression Results
==============================================================================
Dep. Variable:                  SPX   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                  0.000
Method:                 Least Squares   F-statistic:                     0.000
Date:                Wed, 13 Jun 2018   Prob (F-statistic):              0.994
Time:                        15:14:41   Log-Likelihood:                -1.3e+05
No. Observations:               10000   AIC:                             2.600e+05
Df Residuals:                   9998   BIC:                             2.600e+05
Df Model:                          1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1412.9346     0.000    3.1e+06      0.000    1412.934    1412.935
SPX            0.0000     0.000      0.007      0.994      -0.000       0.000
==============================================================================
Omnibus:                     1000.000   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1000.000
Skew:                           0.000   Prob(JB):                       0.00
Kurtosis:                       3.000   Cond. No.                         1.00
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.e+16. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
>>> DEF GRAPH(DATA):
>>> DATA.plot(figsize=(15,5))
>>> DATA.columns.values[-1]='SPX'
>>> ax = DATA.plot(figsize=(15,5))
>>> ax.set_xlabel('Date')
>>> ax.set_ylabel('Price')
>>> ax.set_title('S&P 500 Index')
>>> ax.legend(loc=1)
>>> ax.grid(True)
>>> ax.axhline(0, color='k')
>>> ax.axhline(DATA.iloc[-1].values[0], color='r')
>>> ax.axhline(DATA.iloc[-1].values[1], color='r')
>>> ax.axhline(DATA.iloc[-1].values[2], color='r')
>>> ax.axhline(DATA.iloc[-1].values[3], color='r')
>>> ax.axhline(DATA.iloc[-1].values[4], color='r')
>>> ax.axhline(DATA.iloc[-1].values[5], color='r')
>>> ax.axhline(DATA.iloc[-1].values[6], color='r')
>>> ax.axhline(DATA.iloc[-1].values[7], color='r')
>>> ax.axhline(DATA.iloc[-1].values[8], color='r')
>>> ax.axhline(DATA.iloc[-1].values[9], color='r')
>>> ax.axhline(DATA.iloc[-1].values[10], color='r')
>>> ax.axhline(DATA.iloc[-1].values[11], color='r')
>>> ax.axhline(DATA.iloc[-1].values[12], color='r')
>>>
```

```
In [1]: %run -i -e -n -p -q -t -v -w -x -y -z -b -o -s -t -r -d -m -f -c -p -h -h -h
```