stocksandoil

Daniel Posmik

11/23/2021

1. Set-Up

This replication refers to the following article: https://www.brookings.edu/blog/ben-bernanke/2016/02/19/the-relationship-between-stocks-and-oil-prices/

First, I set up the environment, load packages, and load the data set.

```
rm(list=ls())
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tibble' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'purrr' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

## Warning: package 'stringr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.5
```

```
setwd("C:/Users/dposm/Downloads")
df <- read.csv("stocksandoil.csv")
```

Environment is set up.

2. EDA

Now, I convert the date column into date format. This is achieved by first parsing the mdy structured data, and complementing that with the dmy structured data. Moreover, I create an index in the data set.

```
mdy <- mdy(df$date) #Parsing through mdy structured data
dmy <- dmy(df$date) #Parsing through dmy structured data
```

```
## Warning: 713 failed to parse.
```

```
mdy[is.na(mdy)] <- dmy[is.na(mdy)] #mdy precedence over dmy (corresponds to left join)
df$date <- mdy #Updated date column (in correct date format)

df$index <- 1:nrow(df)
```

Additionally, I take a look at the variables.

```
dim(df)
```

```
## [1] 1183    7
```

```
str(df)
```

```
## 'data.frame':    1183 obs. of  7 variables:
##  $ date   : Date, format: "2011-05-31" "2011-06-01" ...
##  $ tenyear: num  3.05 2.96 3.04 2.99 3.01 3.01 2.98 3.01 2.99 3 ...
##  $ sp500  : num  1345 1315 1313 1300 1286 ...
##  $ wti    : num  102.6 100.1 100.5 100.5 98.8 ...
##  $ copper : num  419 410 410 412 412 ...
##  $ dollar : num  930 932 928 923 926 ...
##  $ index  : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
names(df)
```

```
## [1] "date"    "tenyear" "sp500"   "wti"     "copper"  "dollar"  "index"
```

```
summary(df)
```

```
##       date               tenyear          sp500           wti        
##  Min.   :2011-05-31   Min.   :1.430   Min.   :1099   Min.   : 26.55  
##  1st Qu.:2012-07-31   1st Qu.:1.930   1st Qu.:1399   1st Qu.: 64.77  
##  Median :2013-10-04   Median :2.180   Median :1705   Median : 93.05  
##  Mean   :2013-10-04   Mean   :2.222   Mean   :1692   Mean   : 83.40  
##  3rd Qu.:2014-12-06   3rd Qu.:2.540   3rd Qu.:1985   3rd Qu.: 99.17  
##  Max.   :2016-02-10   Max.   :3.220   Max.   :2131   Max.   :110.24  
##      copper          dollar            index       
##  Min.   :194.3   Min.   : 912.6   Min.   :   1.0  
##  1st Qu.:293.9   1st Qu.: 990.4   1st Qu.: 296.5  
##  Median :325.1   Median :1017.1   Median : 592.0  
##  Mean   :319.7   Mean   :1051.0   Mean   : 592.0  
##  3rd Qu.:351.3   3rd Qu.:1113.3   3rd Qu.: 887.5  
##  Max.   :448.3   Max.   :1253.1   Max.   :1183.0  
```

```r
sum(is.na(df))
```

```
## [1] 0
```

Nothing out of the ordinary.

3. Test and Validation Data

Analogously to Bernanke (2016), my cutoff for the training data is 2014-06-01. Moreover, we start the training data set at 2011-06-01.

```r
start <- as.numeric(nrow(data.frame(which(df$date <= "2011-06-01", arr.ind=TRUE))))
cutoff <- as.numeric(nrow(data.frame(which(df$date <= "2014-06-01", arr.ind=TRUE))))
end <- as.numeric(nrow(df))

df.test <- df[start:cutoff,] #Selecting training data set from 6/1/2011 through 6/1/2014.
df.valid <- df[(cutoff+1):end, ] #Selecting remainder for validation set.
```

Thanks to the indices, subsetting is straightforward.

4. Fitting the regression model

Bernanke's regression model is estimated without intercept. It regresses the change in the log oil price on the change in log copper price, the change in log dollar, and the change in the 10Y Treasury rate. The data sources are:

S&P 500: Yahoo Finance WTI (oil): Bloomberg Copper: Bloomberg/CME Dollar spot index: Bloomberg Ten-year Treasury rates: Board of Governors via Fred

```r
reg1 <- lm(diff(log(wti)) ~ diff(log(copper)) + diff(log(dollar)) +
             diff(tenyear) +0, data = df.test)
summary(reg1) #We can see that the results are consistent with Bernanke's 2016 findings.
```

```
##
## Call:
## lm(formula = diff(log(wti)) ~ diff(log(copper)) + diff(log(dollar)) +
##     diff(tenyear) + 0, data = df.test)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.049374 -0.006028  0.000157  0.007568  0.045458
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## diff(log(copper))  0.39600    0.03685  10.746  < 2e-16 ***
## diff(log(dollar)) -0.74269    0.13078  -5.679 1.94e-08 ***
## diff(tenyear)      0.06294    0.00880   7.153 2.02e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.01201 on 750 degrees of freedom
## Multiple R-squared:  0.3661, Adjusted R-squared:  0.3636
## F-statistic: 144.4 on 3 and 750 DF,  p-value: < 2.2e-16
```

With the regression model having run successfully, I now rearrange to obtain the prediction values. On the right hand side of the initial equation (prediction_raw), I multiply the independent variables with their coefficients. Then, I call the diffinv() and exp() function to account for the diff() and log() functions on the right-hand side respectively.

The result is multiplied by 100 to account for the percentage.

```
### Creating the prediction vector, where
### reg1$coefficients are 0.396 (copper), -0.74296 (dollar), and 0.06294 (tenyear)
prediction_raw <- exp(diffinv(0.396*(diff(log(df.valid$copper))) + (-0.74296)*(diff(log(df.valid$dollar
                                0.06294*(diff(df.valid$tenyear))))
prediction <- prediction_raw*100 #Multiply from 100 because the log changes are in percentage terms
```

The prediction values are created successfully.

5. Indexing and Merging

Now, in preparation for the visualization, I merge the prediction values with the original data set (df). Calling the merge function and utilizing the index, I merge both data frames into the df_final data frame.

```
prediction_df <- data.frame(prediction)
prediction_df$index <- (cutoff+1):end

df_final <- merge(df, prediction_df, by=c("index"), all = TRUE)
df_final <- data.frame(df_final)
```

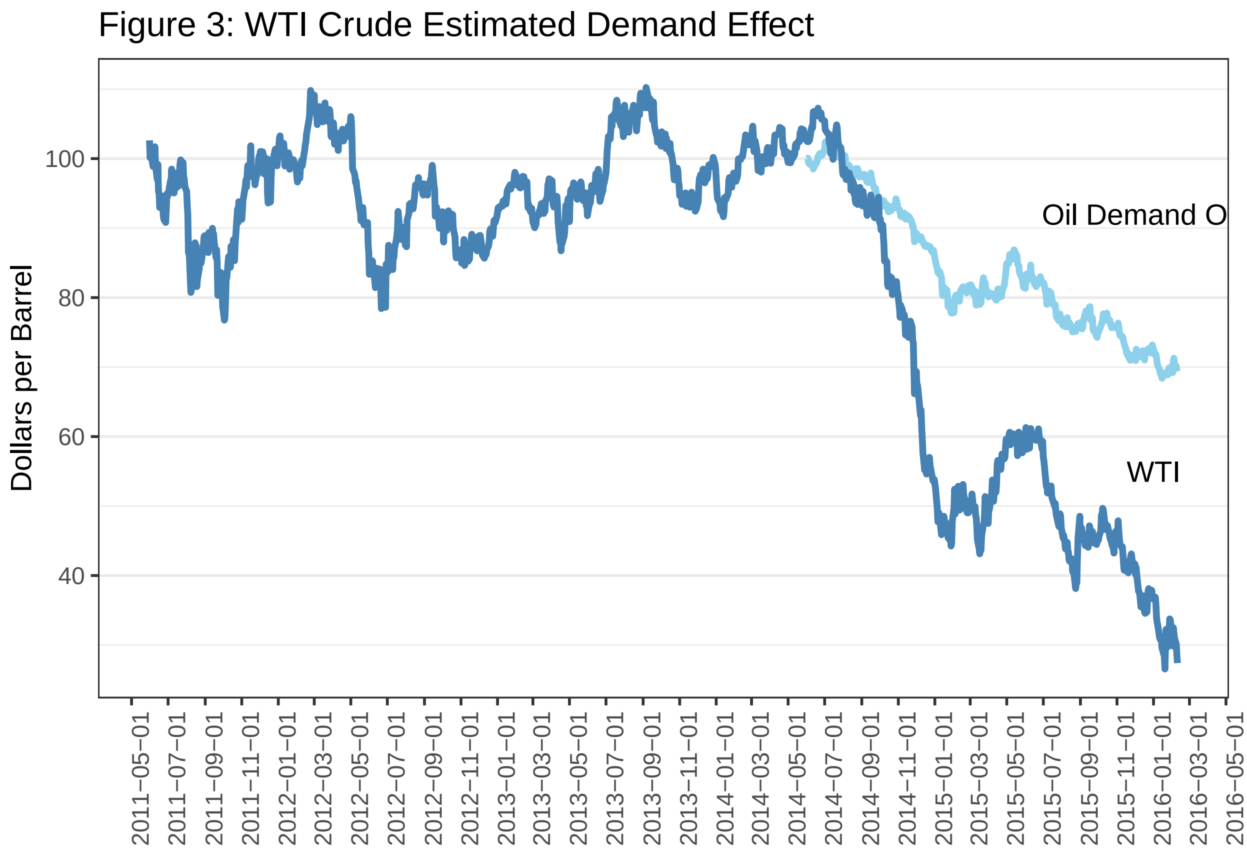
The result is a combined data frame. The prediction values for for the training data set are filled with NA's.

6. Data Visualization

Calling the ggplot function, I try to replicate Dr. Bernanke's visualization as closely as possible.

```
ggplot(df_final, aes(x=date)) + #ggplot call
  geom_line(aes(y = prediction), color = "skyblue", alpha = 0.95, size = 1.2) + #WTI
  geom_line(aes(y = wti), color = "steelblue", size = 1.2) + #Oil Demand
  theme_bw() + #Theme (theme_bw closely resembles to chart)
  labs(color = "black", #Title color
       title = "Figure 3: WTI Crude Estimated Demand Effect") + #Title
  theme(panel.grid.major.x = element_blank(), #Remove major horizontal orientation lines
        panel.grid.minor.x = element_blank(), #Remove minor horizontal orientation lines
        axis.title.x=element_blank(), #Remove x-axis title
        axis.text.x = element_text(angle = 90), #Rotate x-axis labels
        axis.title.y=element_text(), #Add y-axis title
        ) +
  scale_x_date(date_breaks = "2 months") + #Adjust x-axis steps
  annotate("text", x=as.Date("2016-01-01"), y=55, label= "WTI") + #2% Target Annotation
  annotate("text", x = as.Date("2016-01-01"), y=92, label = "Oil Demand Only") + #Core PCE Inflation An
  ylab("Dollars per Barrel")
```

```
## Warning: Removed 755 row(s) containing missing values (geom_path).
```



The result is a close replication of the original analysis.