In-Class Lab 15

ECON 4223 (Prof. Tyler Ransom, U of Oklahoma)

April 4, 2019

The purpose of this in-class lab is to use R to practice with time series forecasting. The lab should be completed in your group. To get credit, upload your .R script to the appropriate place on Canvas.

For starters

Open up a new R script (named `ICL15_XYZ.R`, where `XYZ` are your initials) and add the usual "preamble" to the top:

```
# Add names of group members HERE
library(tidyverse)
library(wooldridge)
library(broom)
library(magrittr)
library(stargazer)
library(zoo)
library(dynlm)
library(pdfetch)
library(tseries)   # You may need to install this package
library(lubridate) # You may need to install this package
library(forecast)  # You will likely have to install this one
```

Load the data

First we'll look at the return on a 3-month treasury bill over the period of 1960q1–1990q4. Second, we'll read in Google's and Apple's stock price data from January 3, 2005 until October 31, 2018.

```
df1 <- as_tibble(intqrt)
df1 %<>% mutate(quarter = seq(yq('1960:Q1'),yq('1990:Q4'), by = 'quarters')) # create quarter
df1 %<>% select(r3,quarter)
df2 <- pdfetch_YAHOO(c("goog","aapl"), fields = c("adjclose"),
                     from = as.Date("2005-01-01"),
                     to = as.Date("2018-11-01"),
                     interval = "1d") %>% as_tibble
df2 %<>% mutate(date=rownames(df2), date=ymd(date)) # create date variable
```

Declare as time series objects

```
df1.ts <- df1 %>% select(r3) %>% zoo(order.by=df1$quarter)
df2.ts <- df2 %>% select(goog,aapl) %>% zoo(order.by=df2$date)
```

Plot time series data

Let's have a look at the 3-month T-bill return for the US over the period 1960–1990:

```r
autoplot(df1.ts) + xlab("Year") + ylab("T-bill return")
```

And now the Google adjusted closing price:

```r
autoplot(df2.ts) + xlab("Year") + ylab("Price")
```

Testing for a unit root

Let's test for a unit root in each of the time series. The way to do this is the Augmented Dickey-Fuller (ADF) test, which is available as `adf.test()` in the `tseries` package.

The function tests H_0 : Unit Root, H_a : Stationary.

```r
adf.test(df1.ts$r3, k=1)
adf.test(df2.ts$goog, k=1)
adf.test(df2.ts$aapl, k=1)
```

1. Which of these time series has a unit root, according to the ADF test? Explain what the consequences are of analyzing a time series that contains a unit root.

Estimating AR(1) models

To alternatively examine the unit root, we can estimate AR(1) models for each series:

```r
est.tbill <- dynlm(r3 ~ L(r3,1), data=df1.ts)
stargazer(est.tbill,type="text")

est.goog  <- dynlm(goog ~ L(goog,1), data=df2.ts)
stargazer(est.goog,type="text")

est.aapl  <- dynlm(aapl ~ L(aapl,1), data=df2.ts)
stargazer(est.aapl,type="text")
```

2. Are the R^2 values from these estimates meaningful?

Forecasting

Now let's use our time series data to forecast future stock prices. First, we should create a shortened version of the time series so we can compare our forecast to actual data:

```r
df2.short    <- df2 %>% filter(date<as.Date("2018-10-01"))
df2.ts.short <- df2.short %>% select(goog,aapl) %>%
                zoo(order.by=df2.short$date)
```

Estimating simple AR models

We can use the `Arima` function to estimate basic AR(1) models on the differenced stock prices.

```r
simple.goog <- Arima(df2.ts.short$goog,order=c(1,1,0))
simple.aapl <- Arima(df2.ts.short$aapl,order=c(1,1,0))
```

This is the same thing as estimating

$$\Delta goog_t = \rho \Delta goog_{t-1} + u_t$$

Estimating ARIMA models

We can also use the `auto.arima` function to allow the computer to choose the best ARIMA model:

```
auto.goog <- auto.arima(df2.ts.short$goog)
auto.aapl <- auto.arima(df2.ts.short$aapl)
```

Plotting forecasts

We can compare the 90-day-ahead forecasts of each model by looking at their plots:

```
autoplot(forecast(simple.goog, h=90))
autoplot(forecast(  auto.goog, h=90))
autoplot(forecast(simple.aapl, h=90))
autoplot(forecast(  auto.aapl, h=90))
```