# ETW3420

# Principles of Forecasting and Applications

Topic 7 Pre-tutorial Activity

In this pre-tutorial activity, you will:

(i) Replicate the figures and results in the Section 7.1 of your lecture notes.

(ii) In doing so, you will learn how to plot graphs using the `ggplot()` function and perform time series linear regression using the `tslm()` function.



## Question 1



The data we will be using is `uschange` - the percentage changes in quarterly personal consumption expenditure, personal disposable income, production, savings and the unemployment rate for the US, 1960 to 2016. (Execute the function `help(uschange)` to see the information).



(a) Print the dataset to see how the data is arranged. Note the heading labels - we will be making reference to these headings later on.

```
uschange
```

(b) Check the structure of the data set.

```
str(uschange)
```

Note that it is a time series object, and NOT a data frame object.

(c) Plot the line charts of Consumption and Income within the same graph.

```
#First, execute the following command and see what you obtain.
uschange[, c("Consumption", "Income")]

#Plot the line charts
autoplot(uschange[, c("Consumption", "Income")]) +
  ylab("% change") +
  xlab("Year")
```

(d) Plot a scatter plot of Consumption vs Income using the `ggplot()` function. You should read about how this function works: `help(ggplot)`

- Notice that the first argument that enters the `ggplot()` function is the data that must be a data.frame object. From Part (b), we see that `uschange` is a time series object, and not a data.frame object. Therefore we need to convert it to a data frame using the `as.data.frame()` function, and label the new output as `uschange.df`:



```
uschange.df <- as.data.frame(uschange)
```



- The second argument is the `mapping` argument which requires us to specify arguments in the `aes()` argument. `aes` stands for 'aesthetics' and for the most basic use, this is where we specify our x and y variables. In this case, our x variable is Income, and y variable is Consumption. Execute the following command and see what is produced.

```
ggplot(data = uschange.df, mapping = aes(x = Income, y = Consumption))
```

- You only get a blank canvas! You get a canvas with only the Y and X axis labelled. No points are shown.
- The `gg` in `ggplot()` refers to the "grammar of graphics", which describes how should plots really be generated. It is a way of thinking of how graphs should be generated. In essence, this grammar is about adding layers.
- So the above code has just given us the first layer - a canvas with just the x- and y-axes.

- Now we need to add the data points to get the scatter plot. We do this by adding (i.e. `+`) another layer of points on this canvas. Specifically, we add a *geometric* layer called `geom_point`. So the code extends to become:

```
ggplot(data = uschange.df, mapping = aes(x = Income, y = Consumption)) +
  geom_point()
```

- Great! So we now have a scatter plot. But how do we also include the line of best fit? Well, by adding another layer! This layer is called 'geom_smooth'.

```
ggplot(data = uschange.df, mapping = aes(x = Income, y = Consumption)) +
  geom_point() +
  geom_smooth(method = 'lm', se = F)
```

- In the `geom_smooth()` function, we specified 'lm' to be the method, meaning a 'linear model' (i.e. OLS). And `se=F` means that we do not want to plot the standard errors.



(e) Regress Consumption against Income and print the results.

- Since this is time series data, we shall use the `tslm()` function. If dealing with cross-sectional data, a linear regression model is fitted using the `lm()` function.
- The `summary()` function then prints the result of the fitted model.
- As `tslm()` works with time series object, we use `uschange` as the data set rather than `uschange.df`.

```
tslm(Consumption ~ Income, data = uschange) %>% summary()
```

(f) Estimate a multiple linear regression of Consumption against the other 4 variables. Save the output in the label `fit`. Obtain the predicted (i.e. fitted) values of Consumption by the model.

```r
#Estimate regression
fit <- tslm(Consumption ~ Income + Production + Unemployment + Savings, data=uschange)

#Print results
summary(fit)

#Obtain fitted values

fitted(fit)
```

(g) Plot the actual and fitted values of Consumption - as line graphs and as a scatter plot.



```r
#Line chart
autoplot(uschange[,"Consumption"], series = "Data") +
  autolayer(fitted(fit), series = "Fitted")
```



- To produce a scatter plot, we need to use the `ggplot()` function. Recall from earlier on, the data argument to enter the `ggplot()` function must be a data.frame object.



- We also only have 2 variables here: the actual and fitted values of Consumption.
- So what we need to do is to combine these 2 variables to become a data frame (lets call it `df`) using the `data.frame()` function:

```r
#Combine Actual and Predicted consumption values into a dataframe, labeled as `df`

df <- data.frame(Data = uschange[,"Consumption"], Prediction = fitted(fit))

#print to see what is produced; notice the heading labels

df
```

- Now we can go ahead to produce the scatter plot:

```r
#Scatter plot


ggplot(data = df, mapping = aes(x = Prediction, y = Data)) +
  geom_point() +
  ylab("Actual % change in consumption") +
  xlab("Predicted % change in consumption")
```