Tutorial 1: Time Series Graphics

The topics included in this tutorial are:

- Introduction to R for Time Series;
- Trend, Seasonal component, Stationary component:
- Holt-Winters Filtering.

Getting started with R Firstly, install *R*, with a suitable editor. For the editor, I recommend RStudio.

Loading Data The data is available on the course home page in the directory:

`http://www.mimuw.edu.pl/~noble/courses/TimeSeries/Data/`

With older versions of RStudio, it was possible to go to the top right hand window, click on 'Workspace' and then 'Import Dataset'. It gave you two options; either a text file or a web URL. By following the instructions, the data set could be loaded quite easily.

There was a problem with this and the option no longer works well.

It is therefore safer to import data using command-line instructions. One way to do it is the following:

```
>www<-"https://www.mimuw.edu.pl/~noble/courses/TimeSeries/Data/atmospheric-carbo
n-dioxide-recor.csv"
> atmos <- read.csv(www,header=T)
```

The first two lines should be all one line; the split in carbon-dioxide was due to typesetting in LaTeX and problems with line breaks.

`ts` Objects

We have already come across the `ts` command in the lecture: The Mauna Loa carbon dioxide data is loaded as follows:

```
> www =
"https://www.mimuw.edu.pl/~noble/courses/TimeSeries/data/atmospheric-
carbon-dioxide-recor.csv"
> carbon = read.csv(www)
```

and we delete observation 611 which is 'na' by:

```
> carbon = carbon[-611,]
```

and then select the column of carbon dioxide values by:

```
> y = carbon$MaunaLoaCO2
```

`y` is now a list of carbon dioxide counts. If we want to express it as a Time Series with period 12 (the monthly values are given), we can do this using the `ts` command as follows:

```
> MaunLoaCo2 = ts(data = y, frequency = 12)
```

This gets it into an appropriate format, where each row represents a year; rows are of length 12.

Now check out the `stl` command by typing

```
?stl
```

This denotes 'Seasonal Decomposition of Time Series by Loess', where Loess means *locally estimated scatterplot smoothing.*

```
> output.stl = stl(MaunLoaCo2, s.window = "periodic")
> plot(output.stl)
```

Time Plots

For the labs in this course, we'll use data sets from the package **fpp2**, which may be installed along with all its dependencies by:

```
install.packages("fpp2")
```

Now activate it using:

```
> library(fpp2)
```

You see that various dependencies have been activated; as with most of the Mathematical Statistics courses, we find that **ggplot2** is an extremely useful package for graphs and visualations of data.

Try the following commands:

```
> autoplot(melsyd[,"Economy.Class"]) +
+     ggtitle("Economy class passengers: Melbourne-Sydney") +
+     xlab("Year") +
+     ylab("Thousands")
```

The output is given in the bottom right hand window. The data set `melsyd` gives numbers of passengers travelling on the Melbourne-Sydney route each year. `autoplot` is a useful command; the syntax here is self explanatory.

The command `ggseasonplot` is useful provided there are not too many years; each year is in a different colour. For example:

```
> ggseasonplot(a10, year.labels=TRUE, year.labels.left=TRUE) +
+     ylab("$ million") +
+     ggtitle("Seasonal plot: antidiabetic drug sales")
```

This gives a plot for antidiabetic drug sales in Australia. There is a dip at the beginning of each year due to stockpiling; we can also see an increasing trend. The years (from 1991 to 2007) are given different colours.

You can do this in polar co-ordinates by setting `polar = TRUE`

```
> ggseasonplot(a10, polar=TRUE) +
+     ylab("$ million") +
+     ggtitle("Polar seasonal plot: antidiabetic drug sales")
```

Alternatively, you can collect the data for each season (here a season is a month) into separate time series plots. Horizontal lines indicate the means for each month.

```
> ggsubseriesplot(a10) +
+     ylab("$ million") +
+     ggtitle("Seasonal subseries plot: antidiabetic drug sales")
```

Now consider a data set giving (a) the half-hour electricity demand and (b) temperatures for Victoria, Australia.

```
autoplot(elecdemand[,c("Demand","Temperature")], facets=TRUE) +
  xlab("Year: 2014") + ylab("") +
  ggtitle("Half-hourly electricity demand: Victoria, Australia")
```

The `qplot` command is useful for scatterplots:

```
qplot(Temperature, Demand, data=as.data.frame(elecdemand)) +
  ylab("Demand (GW)") + xlab("Temperature (Celsius)")
```

Scatterplot Matrices

When there are several potential predictor variables, it is useful to plot each variable against each other variable. The **GGally** package is useful for this. Consider the five time series showing quarterly visitor numbers for five regions of New South Wales, Australia. It appears in the bottom right hand corner of RStudio by typing:

```
autoplot(visnights[,1:5], facets=TRUE) +
  ylab("Number of visitor nights each quarter (millions)")
```

Now activate **GGally** (installing it first if you have not already done so)

```
 library(GGally)
```

Now try

```
GGally::ggpairs(as.data.frame(visnights[,1:5]))
```

to get scatterplots between each pair of variables in the data set.

In this example, the second column of plots shows there is a strong positive relationship between visitors to the NSW north coast and visitors to the NSW south coast, but no detectable relationship between visitors to the NSW north coast and visitors to the NSW south inland. Outliers can also be seen. There is one unusually high quarter for the NSW Metropolitan region, corresponding to the 2000 Sydney Olympics. This is most easily seen in the first two plots in the left column, where the largest value for NSW Metro is separate from the main cloud of observations.

Lag Plots

We now consider quarterly Australian beer production and make scatterplots, where the horizontal axis shows the *lagged* values of the time series. Each graph shows y_t plotted against y_{t-k} for different values of k.

```
beer2 <- window(ausbeer, start=1992)
gglagplot(beer2)
```

Here the colours indicate the quarter of the variable on the vertical axis. The lines connect points in chronological order. The relationship is strongly positive at lags 4 and 8, reflecting the strong seasonality in the data. The negative relationship seen for lags 2 and 6 occurs because peaks (in Q4) are plotted against troughs (in Q2).

The autocorrelation may be plotted as follows:

```
ggAcf(beer2)
```

When data have a trend, the autocorrelations for small lags tend to be large and positive because observations nearby in time are also nearby in size. So the ACF of trended time series tend to have positive values that slowly decrease as the lags increase.

When data are seasonal, the autocorrelations will be larger for the seasonal lags (at multiples of the seasonal frequency) than for other lags.

When data are both trended and seasonal, you see a combination of these effects. The monthly Australian electricity demand series is plotted using

```
aelec <- window(elec, start=1980)
autoplot(aelec) + xlab("Year") + ylab("GWh")
```

and shows both trend and seasonality. Its ACF is plotted using:

```
ggAcf(aelec, lag=48)
```

The slow decrease in the ACF as the lags increase is due to the trend, while the "scalloped" shape is due the seasonality.

Exercises

1. Apply Holt-Winters filtering to the Mauna Loa data. Plot the original data and the one-step fitted values. Plot the Holt-Winters predictors for the next four years. Try this with both the additive and multiplicative seasonal models. Which is best (i.e. which gives best prediction)?

2. Use the help function to explore what the series `gold`, `woolyrnq` and `gas` represent from the **fpp2** package.

   (a) Use `autoplot()` to plot each of these in separate plots.

   (b) What is the frequency of each series?

   Hint: apply the frequency() function.

   (c) Use `which.max()` to spot the outlier in the gold series. Which observation is it?

3. The file `tute1.csv` is in the course data directory. Open it in Excel (or some other spreadsheet application), and review its contents. You should find four columns of information. Columns B through D each contain a quarterly series, labelled Sales, AdBudget and GDP. Sales contains the quarterly sales for a small company over the period 1981-2005. AdBudget is the advertising budget and GDP is the gross domestic product. All series have been adjusted for inflation.

   (a) Read the data into R:

   ```
   tute1 <- read.csv("tute1.csv", header=TRUE)
   View(tute1)
   ```

   (b) Convert the data to time series

   ```
   mytimeseries <- ts(tute1[,-1], start=1981, frequency=4)
   ```

   (The [,-1] removes the first column which contains the quarters as we don't need them now.)

   (c) Construct time series plots of each of the three series

   ```
   autoplot(mytimeseries, facets=TRUE)
   ```

   Find out what 'facets' does and check what happens when you don't include `facets=TRUE`.

4. The file `retail.xlsx` in the course data directory contains some monthly Australian retail data. These represent retail sales in various categories for different Australian states, and are stored in a MS-Excel file. Data of this format can be read into R with the following (suitably modified to take into account where the data is stored):

```
retaildata <- readxl::read_excel("retail.xlsx", skip=1)
```

The second argument (skip=1) is required because the Excel sheet has two header rows. Select one of the time series as follows (but replace the column name with your own chosen column):

```
myts <- ts(retaildata[,"A3349873A"],
frequency=12, start=c(1982,4))
```

(a) Explore your chosen retail time series using the following functions:
`autoplot()`, `ggseasonplot()`, `ggsubseriesplot()`, `gglagplot()`, `ggAcf()`
Can you spot any seasonality, cyclicity and trend? What do you learn about the series?

5. You will find the unemployment rate for the US state of Maine from January 1996 - August 2006 in:

```
> ww2<-"https://www.mimuw.edu.pl/~noble/courses/TimeSeries/Data/Maine.dat"
> maine<-read.table(ww2,header=T)
```

Now try:

```
 > attach(maine)
 > class(maine)
[1] "data.frame"
```

It is not a time series object; it is a *data frame.* Try:

```
> maine.ts <- ts(unemploy, start = c(1996, 1), freq = 12)
```

to create a Time Series object. The average over the 12 months of each year is obtained by:

```
> Maine.annual.ts <- aggregate(maine.ts)/12
```

Now try plotting the series:

```
> layout (1:2)
> plot(maine.ts, ylab = "unemployed (%)")
> plot(Maine.annual.ts, ylab = "unemployed (%)")
```

The February figures seem to be 20% higher than average, while the August figures 20% lower.

```
> Maine.Feb <- window(maine.ts, start = c(1996,2), freq = TRUE)
> Maine.Aug <- window(maine.ts, start = c(1996,8), freq = TRUE)
> Feb.ratio <- mean(Maine.Feb) / mean(maine.ts)
> Aug.ratio <- mean(Maine.Aug) / mean(Maine.month.ts)
> Feb.ratio
[1] 1.223
> Aug.ratio
[1] 0.8164
```

6. Following from the previous exercise, you will find the general US unemployment data from January 1996 to October 2006 in the file `USunemp.dat` in the course data directory. Does Maine seem to be sharing the same trends as the US generally?