UNIVERSITY of LIMERICK

OLLSCOIL LUIMNIGH

Time Series Analysis

MS 4218

Dr. Joe Lynch

joseph.lynch@ul.ie

Re-designated Course timetable

New schedule agreed in first class!

Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
0900:1000 C2062	1300:1500 C2062			

Course Information

- 3 hours per week
- Mixture of lecture, tutorial and laboratory work using ***R*** software
- Week 6: Mid-term - 15 marks
- Week 10: Lab exam - 15 marks
- End of Week 13: Assignment completion - 30 marks
- Final exam: 40 marks

Resources

- Time Series Analysis with Applications in R:
  Jonathan D. Cryer, Kung-Sik Chan
  - Chapters on https://sulis.ul.ie
- Supplementary reading material:
  - Introductory Time Series with R: Cowpertwait Metcalfe
  - The Analysis of Time Series: Chatfield
  - Practical Time Series: Janacek
- Library textbooks: first floor, Section 519.55

Final Exam

Grades*	A1: 90+	A2: 80-89	
	B1: 70-79	B2: 60-69	B3: 55-59
	C1: 50-54	C2: 45-49	C3: 40-44
	D1: 35-39	D2: 30-34	
	F: $<$ 30		

Syllabus

- Components of a time series;
  Smoothing methods;
  Trend projection;
  Deseasonalising a time series;

- Autocorrelation;
  Autoregressive models;
  Integrated models;
  Estimation in the time domain;
  Box-Jenkins approach;

Syllabus cont.

- Spectral analysis;
  Spectral distribution/density functions;
  Fourier analysis and Fast Fourier transform;
  Periodogram analysis;
- Forecasting methods;
- Applied time series analysis using *R* software.

Aims

On successful completion of this module, you should be able to:

- Calculate theoretical acf, pacf, spectrum and spectral density of stationary ARIMA models
- Use sample acf, sample pacf, cumulative periodogram and time series diagnostic plots for a set of data to identify a suitable ARIMA model

Aims cont.

- Calculate and use model fitting and parameter estimation methods and criteria, e.g., maximum likelihood, Yule-Walker equations, AICs, BICs
- Calculate forecasts and appropriate forecast prediction error for seasonal ARIMA models
- Compute and interpret *R* output for time series analysis of real data

Time Series

A collection of observations made sequentially through time.

- Finance: inflation rate, average income, ISEQ index
- Business: CPI, sales
- Weather: T°, humidity, wind
- Agriculture: crop yield, livestock, soil erosion
- Biological: heart rate, sugar level, %body fat

Discrete v Continuous

Discrete: observations usually taken at equal intervals.

Time series on this course are discrete.

Continuous: observations made continuously through time e.g., machine monitoring various indices.

Can convert from continuous to discrete by sampling at regular intervals.

Dependence

Successive observations are not usually independent.

Analysis must take account of the time order of the observations.

When successive observations are dependent, future values can be predicted from past observations.

If a time series can be predicted exactly, it is **deterministic**.

If not, it is **stochastic**.

Stochastic Time Series

Future only partly determined by past values.

Exact predictions are impossible.

Future values have a probability distribution conditioned on past values.

Purpose of Time Series Analysis

- to understand or model the stochastic mechanism that gives rise to an observed series
- to predict or forecast the future values of a series based on the history of that series and, possibly, other related series or factors.

Setting up R

Click on *R* icon located on desktop.

If not present, download from "http://cran.r-project.org/"
Download *R* for Windows
Install *R* for the first time
Download R 3.0.2 for Windows
Save File

Refer to *R* manual on sulis.

Start up

Create folder on USB key named TimeSeries.
On R menu, click on File; New script

Set up the working directory by typing into script file:

```
setwd("C:/Documents and Settings/User/TimeSeries")
```

The precise location of your folder is found by Right-clicking on the folder, click Properties, and then copy and paste the location into setwd()

Change the backslash to forward slash when entering the address.
Highlight the command and then press Control R to run the command.

library(TSA)

This library and 6 other libraries listed below for the Cryer Chan book must be first downloaded and then loaded into *R*.
Click on Packages, install packages

Choose Ireland in Cran Mirror window

Choose Packages: Highlight TSA
$\pm$ Hold down control key and highlight (leaps, locfit, MASS, mgcv, TSA, tseries, uroot)
Click OK

Click Packages: Load Packages (the seven just downloaded).

Library TSA

Type: library(help=TSA)

The two basic time series objects found in library(TSA) are functions and data.

Typing the name of the function gives the code for that function.

Typing ? before the function name gives details of the arguments it takes and its output.

Nature of Data

For time series analysis, data must be of time-series class.

Data from library(TSA) are all of time series class.

Other data may not be.

Data can be coerced to time series objects and status confirmed with:

```
dataname<-ts(dataname,start=c(x,y),frequency=z)
class(dataname)
```

?ts gives details about arguments input and the output of the ts() function.

Reading in other data into R

Once your working directory has been set with setwd(), extraneous data can be read in with:

```
1.read.table("Filename", header=T)
2.scan("Filename", header=T)
3.read.csv("Filename", header=T)
```

in the case of 1.multivariate and 2.univariate data respectively.

Check the class of the data with: class(data) and use ts() if required.

If object is a dataframe: attach(data) creates objects from the column headings.

Datasets in library(TSA)

5 different datasets are used repeatedly throughout the course:

- 100 year record of annual LA rainfall
- Colour property from an industrial chemical process
- Annual Canadian hare numbers over 31 years
- Monthly average T°F in Dubruque, Iowa for 12 years
- Monthly sales of oil filters for construction equipment

Patterns

In a plot, we are looking for:

- Patterns:
  - Trend
  - Seasonal - regular variation
  - Cyclical - possibly irregular variation
- Noise
- Discontinuities
- Outliers

Plotting the data

Firstly, read in the data. In the case of the annual LA rainfall record, we type: data(larain).

The first analytical step is to create a time plot by plotting the observations over time.

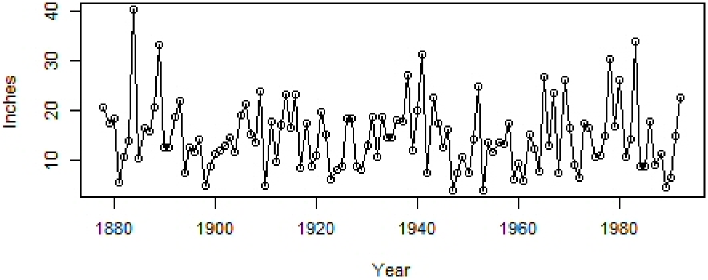
```
win.graph(width=4.875, height=2.5,pointsize=8)
plot(larain,ylab="Inches",xlab="Year",type="o")
```

Type ?win.graph, and check out pointsize

Type ?plot and see the array of arguments that can be taken.

Exhibit1.1

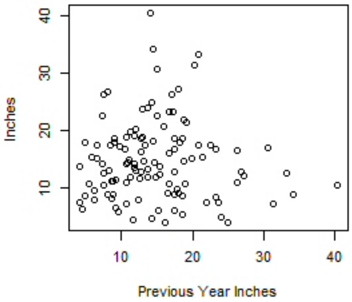
Time series plot of LA annual rainfall



Is there dependence from year to year?

```
plot(y=larain,x=zlag(larain),ylab="Inches",
xlab="Previous Year Inches")
```

Scatter plot of LA annual rainfall v Last year's rainfall



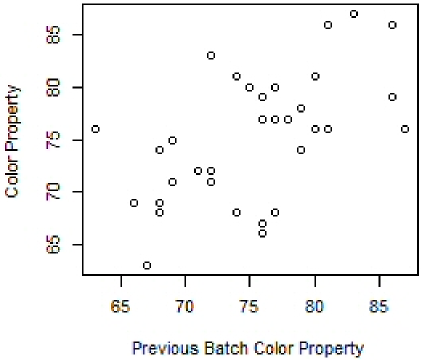
Type ?zlag. Prediction not possible.

Exhibit1.3

Here the neighbours seem to be quite similar, so prediction might be possible.

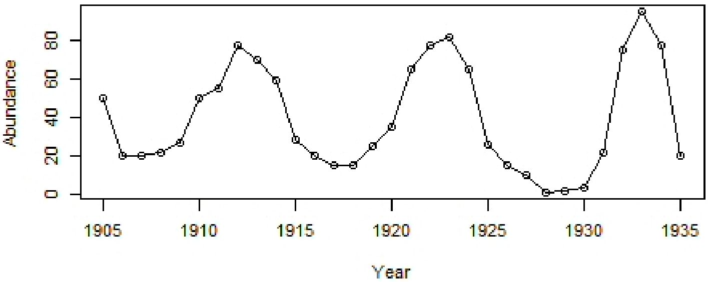
Exhibit1.4

Colour v previous colour value; $r = 0.5549$.




```
cor(color[-1],color[-(length(color))])
```

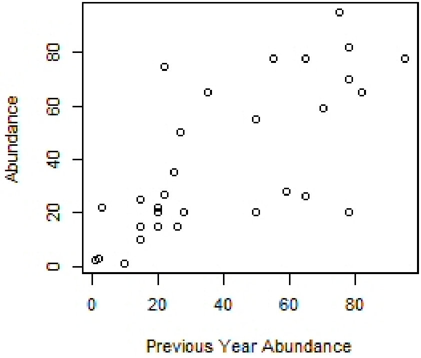
Exhibit1.5



Cyclical behaviour and neighbours are more closely related than before.

Exhibit1.6

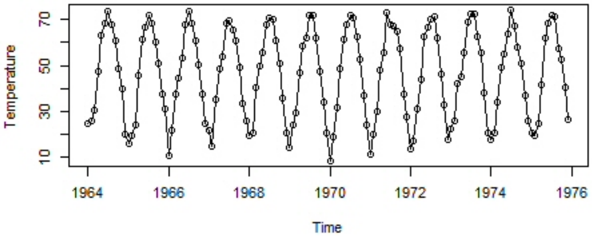
Abundance of Canadian Hare v previous year's hare: $r = 0.703$




```
cor(hare[-1],hare[-(length(hare))])
```

Exhibit1.7

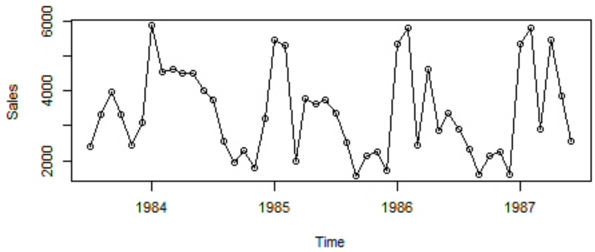
```
data(tempdub)
plot(tempdub,xlab="Time",ylab="Temperature",type="o
```



Seasonality on show here. All January months are cold, and all July months are hot. Models for such series must accommodate this variation while keeping the similarities.

Exhibit1.8

```
data(oilfilters)
plot(oilfilters,xlab="Time",ylab="Sales",type="o")
```



Seasonality is again on show here.

Display plotting symbols

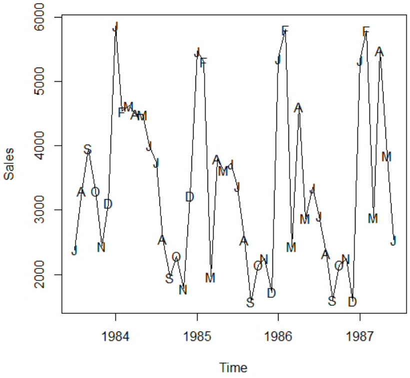
```
plot(oilfilters,type="l",ylab="Sales")
points(y=oilfilters,x=time(oilfilters),
pch=as.vector(season(oilfilters)))
```

time() extracts the vector of times

as.vector(season(oilfilters)) gives a character string.

Exhibit1.9

Oil filter sales with plotting symbols



Chapter 2

Chapter 2 looks at fundamental statistical concepts required to analyse time series data.

In particular, we look at

- Expectation
- Variance
- Auto-covariance
- Auto-correlation
- Stationarity

for various time series models.

Chapter 3

Chapter 3 discusses trends and seasonality.

Trend is the underlying direction in the time series.

Seasonality is a regular variation repeated annually.

Trend and seasonality may or may not be of specific interest.

If removed, the residual data will still show some correlation.

Chapter 4

Chapter 4 looks at main basic models used to analyse stationary data:
data that have time-independent means and variances.

Specifically we look at:

- moving average(MA) models
- auto-regressive(AR) models
- combined ARMA models

Chapter 4

Chapter 5 looks at techniques to enforce data to be stationary. Specifically we look at:

- Differencing and the resultant ARIMA models
- Other transformations:
  - taking logs
  - Box-Cox

Box-Jenkins approach to Model Building

- Model specification
- Model fitting
- Model diagnostics

Model specification: covered in Chapter 6

Plot the data.

Compute summary statistics specific for time series.

Apply background knowledge.

Tentatively choose a model.

Choose model with smallest number of parameters to be estimated.

Model fitting: covered in Chapter 7

Find the best possible estimates of the unknown parameters:

- Method of moments
- Least squares
- Maximum likelihood

Model diagnostics: covered in Chapter 8

Does the model fit the data well?

Are the model assumptions well satisfied?

If yes, well and good.

If no, another model needs to be specified.

Remaining chapters

Chapter 9 deals with forecasting future time series and the errors therein.

As well as modelling a dataset, the project will deal with forecasting.

Chapter 10 looks at seasonal arima models.

Chapter 13 deals with time series, not in the time domain, but in the frequency domain.

Next Lecture

Basic concepts that serve as models for time series.

- Mean, variances, covariances
- Random walk
- Moving average
- Stationarity
- White noise
- Random cosine wave