Part 1-a

Time series analysis is all about, doing predictive analytics, predicting future values of variable or forecasting certain amount along a particular direction

We will see :-

What is Time series

Regression vs Timeseries

Look for trends & seasonality

Smoothing

ACf

ARIMA

Lagging

Box pierce test.
Filters & Convolutions
frequency analysis
Multiplicative time series analysis
forecasting analysis

In Time series, Time as a dimension we are trying to forecast.
We are dealing with two variables where x axis is time & y axis can be any other thing like temp, stock price, etc.

We can't apply regression everywhere.
For example Bengaluru Petrol Price, we want to predict price of petrol.
We have to see whether we take whole previous data or just a slice of it for prediction.
, Sensex daily closing → 1984 → 1994 no increase, the data can be misleading, but lastly it increased 10x.
We are only predicting data on past events, why it is happening is not our concern

Sensex Small cap → The statistical properties of this data is not same as Sensex.
Since here is greater change, it is
Lesser volatility more predictable is the data.

DATE / /
PAGE

Time series Part 1b

Take relevant slice of data in time series analysis for accurate predictions.
Time series is all about trying to separate out trend, seasonal variance from the data as in Airline passengers.

Remove yearly avg, ~~we get~~ or lets say components of the data are removed one by one from the original data and therefore what is left is called Residue. And what we get is quite different from original data. Hence Time series analysis all about revealing those pattern. For example Airline & Rainfall dataset

Time series Part 2a

Valid time series data must have.
- Min of 50 data points
- Equally spaced time intervals
- No missing data points.
- Stationary: no trend, no seasonality and no change in frequency or amplitude of noise (i.e. its statistical distribution properties must not change over time)

Frequency and Period.

Frequency: how many times does an event occur in given unit of time
- Eg: 4 times in a year.

Period: Inverse of frequency.
- → E.g. once in 3 months (annual frequency = 4)

Seasonality = periodicity (Period is 1 year → seasonality otherwise periodicity)
→ Cyclic variations in repeated patterns.

Time domains: at what points in time did the event occur and at what amplitude.

Eg: SW monsoon occurs in June - Sept.

Frequency domain: at what frequencies does the event repeat.

Eg. SW monsoon as a periodicity of 1 per year

If we remove seasonal variance in Atmospheric CO_2 data, we get a trend.

for non linear trend, we can use log of data for analyzing & fit regression line rather than doing Time series analysis

Time series Part 2b

Statistical distribution property of stock data is not same for 1970 - 1990 as in 1990 to 2010, First there is an increase & then a decrease after peak. So there is non stationality in data. Therefore we need to split data

For Time-Series analysis

We decompose the data into:

- Trend : increase or decrease over long term
- Seasonality : periodic changes
- Noise or irregularity
- ~~other~~
- Cyclicality (beyond 1 year)
- Other.

Time - Series operations.

1. Create Time-Series data
   a) Vector of numeric data and vector of index (date and/or time)

2. Plotting
3. Extracting, subsetting, merging, filling padding and lagging

Main operations:

- Computing successive difference : diff
- Apply daily / weekly / monthly / quarterly / yearly
- Apply a rolling function : rollapply
- Moving average (a type of rolling mean)
- Smoothening a time-series
- acf.
- Test for autocorrelation : Box.test.
- Partial auto correlation function : pacf
- Cross-correlation function : ccf
- Detrending
- Fitting an arima model : auto.arima
- Running diagnostics on an ARIMA model : tsdiag.
- Making forecasting using ARIMA : predict
- Testing for mean reversion : adf.test

Part 3a

```
data(Airpassengers)
library(timeseries)
library(forecast)

class(Airpassagers) → ts
```

This dataset has 12 data point per year.

actual trend can be find out by

```
d ← stl(Airpassengers, s.window = 12)
```

Take avg. of data points

```
plot(aggregate(Airpassengers, FUN = mean))

boxplot(Airpassengers)
```

similarly for cycle

Part 3b

In this CO_2 had linear trend. Seasonal variance is perfect. —

Next we used rainfall dataset & create ts object

```
ft ← seq(from = as.Date("1871-01-01"), by = "month",
         length.out = 1692)

ts ← as.timeSeries(rain$rainfall, ft)
```

Rainfall data is too complex to see seasonality and trends.

Part 3c

head(ts) → oldest values

tail(ts) → newest values

- xts.first(ts, "month")
- xts.last(ts, "2 weeks")

ts[i] ts[,i]

Complex

ts[as.Date("1917-06-15")]

data on particular date

dates ← seq(startdate, enddate, increement)

window(ts, start = startdate, end = enddate)

merge(ts1, ts2)

union of two time series with NA inserts.

merge(ts1, ts2, all = False)

Gives intersection of two time series

Pad a time series by merging with an empty ts object with desired ones.

lagging → shift ts points by k units lags(ts, k)

k positive, k negative.

Successive differences → diff(x)

year on year diff = diff(x, lag=12)

rollapply(ts, width, f, align = "right")

To apply at every n^th point, by=n

Detrending

for linear trend use regression to fit line.

Subtract trend from data values

use decomposition function like stl

Part 1a

Smoothing

avg. of local values is called moving avg. or Rolling mean.

disadvantage: Creates mountain for every spike. Distorts shape of curve

Exponential Smoothing

$S_i = \alpha x_i + (1-\alpha) S_{i-1}$ prev ↳ data points.

$= \alpha x_i + (1-\alpha)[\alpha x_{i-1} + (1-\alpha) S_{i-2}]$

$= \alpha x_i + (1-\alpha)[\alpha x_{i-1} + (1-\alpha)[\alpha x_{i-2} + (1-\alpha) S_{i-3}]]$

$= \alpha [x_i + (1-\alpha) x_{i-1} + (1-\alpha)^2 x_{i-2}] + (1-\alpha)^3 S_{i-3}$

$= \alpha \sum_{j=0}^{i} (1-\alpha)^j x_{i-j}$

Single exponential smoothing works only if there is no trend, if it is so values lag behind

Double Exponential Smoothing

$S_i = \alpha x_i + (1-\alpha)(S_{i-1} + t_{i-1})$ $t_i \rightarrow$ trend.

$t_i = \beta (S_i - S_{i-1}) + (1-\beta) t_{i-1}$

↳ diff b/w smooth curr val and smooth prev val

Triple smoothing

when trend & seasonality both present

for additive case.

$S_i = \alpha (x_i - p_{i-L}) + (1-\alpha)(S_{i-1} + t_{i-1})$

$t_i = \beta (S_i - S_{i-1}) + (1-\beta) t_{i-1}$

$p_i = \gamma (x_i - S_i) + (1-\gamma) p_{i-L}$ → curr values - smooth curr val

$x_{i+h} = S_i + h t_i + p_{i-L+h}$ ↳ periodicity value & power before

Multiplicative

$$s_i = \alpha \frac{x_i}{p_{i-K}} + (1-\alpha)(s_{i-1} + t_{i-1})$$

$$t_i = \beta (s_i - s_{i-1}) + (1-\beta) t_{i-1}$$

$$p_i = \gamma \frac{x_i}{s_i} + (1-\gamma) p_{i-K}$$

$$x_{i+h} = (s_i + h t_i) p_{i-K+h}$$

p is periodic component and K is length of period.

~~alfa~~

all of this packaged into Holt winters Model.

Method

↳ Exponential weighting

↳ Model level: single

↳ Trend level: double

↳ Seasonality level: triple

Holt Winters (ts, alpha = 0.2, blta = 0.2, gamma = 0.2, seasonal = c("additive", "multiplicative"))

Single exponential: beta = NULL, gamma = NULL

Double exponential: gamma = NULL

Specify above values to good fit b/w smoothen curve & actual ~~da~~ curve

Time Series Modelling

↳ A classic Time Series Model

$Y = T + S + C + I$

Y = data

T = trend.

S = seasonal (regular path occur with time)

C → longer term patterns like business cycle

I → irregular.

DATE / /
PAGE

→ → additive model
Multiplicative can be used if seasonality is multiplicative

Part 4b

data ("Airpassengers")
since in seasonal plot, seasonality was amplifying over time we choose multiplicative
h1 ← HoltWinters (Airpassengers, alpha=0.5, beta=0.3 gamma=0.2, seasonal="multiplicative")
Try for various values of alpha, beta & gamma

got good fit incase of single ~~value~~ exponential Smoothing
for prediction
predict (h1, n.ahead=12)

if you have data points say 100, then you shouldn't predict more than 50 for future cases.

Part 5a

Auto correlation

$$c(K) = \frac{\sum_{i=1}^{N-k} (x_i - \mu)(x_{i+k} - \mu)}{\sum_{i=1}^{N} (x_i - \mu)^2}$$

with $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$

diff with μ & correlating with K values ahead

repeat for K times, and its actually a K then autocorrelation high.

Box.tes → Box.test (x) → return p value

Time Series Regression Model
Linear Regression: $Y_i = BX_i + \varepsilon_i$
linear trend + white noise

Autoregression (AR): $X_t = \Phi X_{t-1} + \varepsilon_t$
Moving Average (MA): $\varepsilon_t = w_t + \Theta w_{t-1}$
Combine two - ARIMA $X_t = \Phi X_{t-1} + w_t + \Theta w_{t-1}$
Differencing: $X_t - X_{t-1}$
Integrated → ARIMA: initial diff. steps non-stationary

Assumptions
Noise terms are
↳ Independent
↳ Identically distributed
↳ Normally distributed
↳ Having zero mean

ARIMA
ARIMA → Auto-Regressive Moving Average
ARIMA → Generalization of ARIMA

auto regressive Integrated Moving Average.
Arima if non stationary; initial diff steps are integrated to reduce non-stationarity
arima (p, d, q)
p = order of AR (# of time lags)
q = order of MA (# moving avg. / coeff)

d = degree of differencing (# times past values are subtracted

for ex)

p → 2, find two values of k two different periodic components

q → How I do moving avg, exponential avg etc.

arima is popular, because it can be automated

Part 5b

Non seasonal ARIMA

ARIMA(p, d, q)

p → order of auto-regression

d → degree of diff.

q → order of moving avg.

Seasonal arima

↳ ARIMA (p, d, q) (P, D, Q)m

↳ m → no. of periods in each season

↳ P, D, Q → for seasonal component

ARIMA (0, 0, 0) X_t = ept - White Noise

AR(1) = ARIMA (1, 0, 0)

Random Walk : I(1) = ARIMA (0, 1, 0)

$X_t = X_{t-1} + ept$

MA(1) = ARIMA (0, 0, 1)

ARIMA (0, 1, 1) → basic exponential smoothing

ARIMA (0, 2, 2) → double exponential smoothing

ARIMA
Method

Box Jenkins approach
To find right values of p, d, q for ARIMA modelling?
↳ Model identification (i.e, its order)
↳ Parameter estimation → fit model to data.
↳ Diagnostic checking to validate model

Pacf → partial auto correlation function. at lag k is corr. between all data points that are exactly k steps apart.
→ after accounting for their correlation with the data b/w those k steps.

Pacf helps to find degree of AR in ARIMA.

Rcode
library (forecast)
auto.arima (x)
m ← arima (x, order = c(p,d,q))
fsdiag (m)
predict (m, n.ahead = 10)

Testing for Mean Reversion
↳ If a time series data is moving away from its mean, then it has a tendency to come back to its mean.

Augmented Dicky-Fuller test
adf.test (x)

Part 6a

Box.test (CO2) give pvalue

acf (Airpassengers) shows auto correlation.

acf (CO2)

acf (fit) both true & -ve auto correlation

m1 ← auto.arima (Air Passengers)

since ma going from -ve to +ve so we should not consider.

Here we checked whether p, q, d values given by auto.arima, really fit the model or not.?

Part 6b

m2 ← auto.arima (CO2)

ARIMA (1,1,1) (1,1,2) [12]

m3 → auto.arima (rainfall)

ARIMA (0,0,0) (0,0,0) [1,2]

since -ve to +ve so its not a good model.

Taking your own model

m4 ← arima (fit, c(2,0,1))

DATE / /
PAGE

Part 7

Periodgram
Lomb-Scargle Periodogram
↳ Converting from time domain to freq. domain
↳ ~~Yearly~~ Seeing various long term cycles.
↳ Doing moving average
↳ plotting periodicity

Heteroskedasticity
Non constant residual variance
Volatility clusters in financial data
ARCH and GARCH methods
ARCH : Auto regressive conditional Heteroskedasticity
Garch : Generalized ARCH
Applications in Financial Modelling & Economics

Graph Analytics
Graphs & Graph Theory.
$G = (V, E)$
V = {vertex or node}
E
Directed & undirected graphs
Paths & Cycles
Connected components & cliques
Graph algos shortest path etc
Diameter of graph
- Graph centrality
  - Degree centrality
  - closeness centrality
  - Eigenvector centrality
  - Between centrality

Centre & Centrality

Centrality : denotes importance of a node

Centre of graph is point that is at least distance from all nodes.

Degree centrality : Number of connections that a node has

$$cd(v_i) = d_i$$

Betweenness Centrality

↳ Bridges or nodes that span boundaries

↳ proportion of all pairs shortest paths passing through node

Higher network importance = Higher betweenness

It helps to find key participants in network those who connect people.

Closeness Centrality : measure of shortest paths in a graph.

↳ Inversely proportional to distances in graph

$$C_c(v_i) = \frac{1}{I_{v_i}}$$

Diameter of graph : Length of longest shortest path

Eigen vector Centality

↳ finding influential nodes in a network

Ramayana

↳ Characters are nodes

↳ Relations b/w characters as edges

from text to Social graph

Input text of Ramayan

Step 1 → program to identify characters

Step 2 → co-occurence of potential relations

Step 3 → build Social graph of Ramayana characters

Step 4 → apply graph analytics to graph

Importance of Relation

no. of time of their co occurence, is strength or Importance of their relationship

Centality Findings

Centre of graph: Ram & Laxman

Centality Scores.

Rama = 87.5% 56 relation

Sita = 46.88% 30 relation

Lakshma = 43.75% 28 relation

DATE / /
PAGE

Role of hanuman

- Hanuman as highest Betweenness centrality
- Third highest Eigen vector Centrality
- Fourth in Degree Centrality
- Fourth in closeness centrality

Identifying sub-stories

- Subgraph or clique detection algorithm to find sub stories