Week 1

R software setup

* Download and Install R for windows and Mac

Basic Descriptive Statistics with R

1. Using Packages in R

* data(package=’faraway’) – this is the package
* data(coagulation, package='faraway') – this is a set of data within

faraway package

* ls() – shows the list
* coagulation – this is the name of the dataset
* plot( coag ~ diet, data=coagulation) – plotting a plot depending on the data you provide. If it’s a time series, you will most likely get a line plot. But you may get a boxplot.
* summary(coagulation)
* mean(worldcup$Time) – finding the mean Time played in worldcup dataset

Descriptive Statistics

1. Numerical descriptions: 5 number summary, standard deviation
2. Graphical Descriptions: histograms, time plots

Inferential Statistics

* Straight Line regression
* Regression models and diagnostics
* T-tests
* Correlation

Stationarity

1. Strong
2. Weak

Autocovariance and autocorrelation

Random Walks

Moving Average Processes • Introduction to Simulation

R code

data.1 = c(10, 20, 30, 40 ,50) Entering a dataset using c()

data.1

print(data.1)

help()

class() Tells you the class (example: ts = “time series)

summary(data.1) Gives you 5 number summary (min, 1qrt, median,3qrt,max)

mean(data.1) Gives you average sample

sum(data.1)/5 Gives you sum (then divided by 5)

sd(data.1) Gives you sample standard deviation σ (sigma). Amount of

variation

Standard deviation is a measure of spread.

Low = closely clustered around the mean/average

High = dispersed

Distribution should/must be normal (normal bell curve)

* Is it standard/expected OR unusual/unexpected

66 (1 sigma) -95 (2 sigma) - 99.7 (3 sigma) rule

hist(dataset\_var\_name, hist() function. Add variable name that holds the dataset

xlab='My data points', xlab - x axis (to change the x axis title)

main='Histogram of my data', main – change main title

freq=F, Frequency or count. Boolean value, by default, it is True. If

you do this, it changes to PROBABILITY.

col='green', Changes the color of bars

breaks=10) Changes the number of bins

density(small.size.dataset) Smooth density function

lines(density(small.size.dataset),

col='red', Changes the color of the density line

lwd=5) Changes the line width

Scatterplot (bivariate data)

set.seed=2016

Test\_1\_scores=round(rnorm(50, 78, 10)) Create random numbers. Rnorm = normal dist

1st – 50 numbers

2nd – Avg around 78

3rd – Std dev 10

plot(Test\_2\_scores~Test\_1\_scores) Creates a scatterplot (y\_axis, x\_axis)

plot(Test\_2\_scores~Test\_1\_scores,

main='Test scores for two exams (50 students)',

xlab='Test\_1\_scores',

ylab='Test 2 scores',

col='blue')

Plotting time series data in R

help(co2) Describes the dataset in package

co2.linear.model = lm(co2 ~ time(co2) ) lm() function to create linear model

co2 – the dataset

time(co2) – pulls out the time part

**(** co2.linear.model = lm(co2 ~ time(co2)) **)** Notice blue EXTRA brackets around the whole

code. This will give you the output. It’s the

same as not using the brackets and calling

the variable co2.linear.model to show results

plot(co2, main='Atmospheric CO2 Concentration') co2 is the dataset. main sets the title

abline(co2.linear.model) Generates the intercept & slope line on the

model that you develop

residuals – the difference between predicted and actual. Residuals should be NORMALLY distributed

This is good when you have a large dataset. What do you do if you have small

datasets?? Use qq plot

co2.residuals = resid(co2.linear.model) resid() function. Pass in the model.

qqnorm(co2.residuals) qqnorm() function. Pass in the model.

qq stands for quantile quantile. This is a

quantile plot on normal dist.

qqline(co2.residuals)

plot(co2.residuals~time(co2), Doing a time plot of our residuals

xlim=c(1960, 1963), selecting 1960 to 1963

main="Zoom In") setting title

Reviewing Basic Inferential Statistics

plot(extra~group, Boxplot of extra sleep by group

data=sleep, sleep data

main="Extra sleep by group") new title

> attach(sleep) ?? not sure what this does

> extra.1=extra[group==1] attaching all EXTRA column values to

Variable extra.1

> extra.2=extra[group==2]

> extra.1 Notice the output of extra.1

[1] 0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0.0 2.0

Test the hypothesis

t.test(extra.1, extra.2,

paired=TRUE,

alternative="two.sided" )

Output:

Paired t-test

data: extra.1 and extra.2

t = -4.0621, df = 9, p-value = 0.002833 t – measure of variability

p-value – highly significant (<0.05 or 0.01)

**alternative hypothesis**: true difference in means is not equal to 0 (R believes there is a difference between 2

drugs). Remember, NULL hypothesis (Ho)

means that there is no difference. If your p

value is small, reject the NULL (p<α). That’s

why R says there is a difference between the

2 drugs.

95 percent confidence interval: Confidence Interval – Where we believe the

-2.4598858 -0.7001142 mean would be.

sample estimates:

mean of the differences

-1.58

Correlation Function – Association between 2 variables (using trees dataset)

pairs(trees, Creates a pairs plot using trees data

pch=21,

bg=c(“red”)) Dots are red

cov(trees) Calculates the COVARIANCE

Output:

Girth Height Volume

Girth 9.847914 10.38333 49.88812

Height 10.383333 40.60000 62.66000

Volume 49.888118 62.66000 270.20280

Cor(trees) Calculates the CORRELATION!

Output:

Girth Height Volume

Girth 1.0000000 0.5192801 0.9671194

Height 0.5192801 1.0000000 0.5982497

Volume 0.9671194 0.5982497 1.0000000

Time plots (from “astsa”) package

install.packages("astsa") Instructions to install a package in R called ‘astsa’.

data(package='astsa') ‘jj’ is a johnson johnson dataset

data(jj, package='astsa')

jj

require(astsa) Get the package

help(astsa) or help(jj) Documentation on astsa package or jj dataset

ts() Function to make the dataset a time series

plot.ts() If your dataset isn’t a time series data

plot(jj, type='o', marks are o

main='Quarterly Earning JJ', title

ylab="Earnings”, y axis label

xlab='Quarters', x axis label

col=”green) color

Looking at the graph:

* + - * 1. There is an upward trend
        2. There is SEASONALITY (there’s ups/downs within a year)
        3. In the beginning, the VARIATION is NOT significant as

compared to the 1980s. The spikes are much greater!

VIOLATES STATIONARY PRINCIPLE

plot(flu, type='o', flu dataset. marks are o

main=”Influenza deaths per 10K”, title

ylab="Per 10K”, y axis label

xlab='Months', x axis label

col=”green)

Stationarity (weak stationary time series)

* No systematic change in mean (i.e No trend)
* No systematic change in variation
* No periodic fluctuations

Stochastic process

* Collection of random variables indexed by time

Autocovariance coefficients

* Covariance – measures the LINEAR dependence between 2 random variables (x,y)
* cov() function in R
* acf(time\_series, type=’covariance’) auto correlation function

prp=ts(rnorm(100)) Generating 100 random numbers

print(prp)

(acf(prp, type='covariance')) acf() function. Gives us a plot of auto covariance

Output: coefficient at lag 0, lag 1 and so on

Autocovariances of series ‘prp’, by lag

0 1 2 3 4 5

1.01138 0.05349 -0.00609 0.01031 -0.01106 -0.10813

6 7 8 9 10 11

0.03182 -0.09362 0.02525 0.16769 0.02629 0.02974

12 13 14 15 16 17

-0.14650 -0.06805 0.02515 0.04188 -0.01688 0.07221

18 19 20

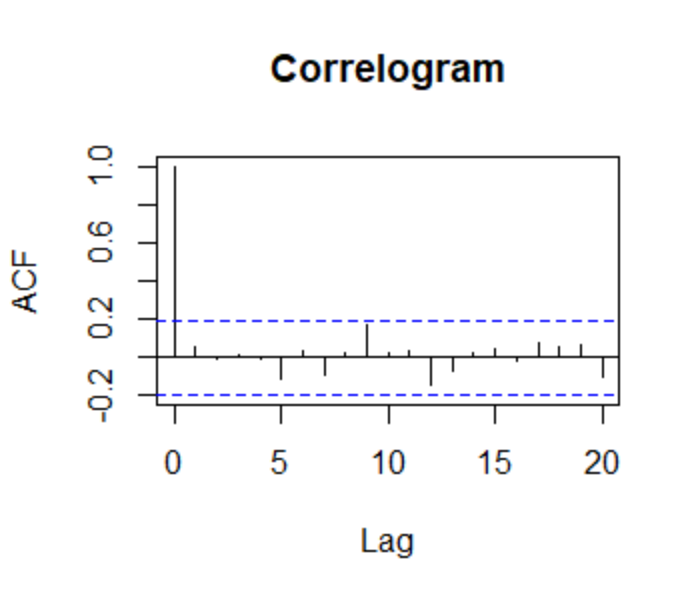
0.05344 0.06307 -0.10508

Autocorrelation function

* Assume weak stationarity (no variation, flucutations)

acf(prp, main=”Correlogram”) Generates a plot

Output:



* It will always start at 1
* After that, there is not much correlation at different lags.
* Blue dash lines shows SIGNIFICANCE LEVEL

(acf(prp, main=”Correlogram”)) Will get plot and autocorrelation coefficients by

Output: surrounding it with parenthesis

Autocorrelations of series ‘prp’, by lag

0 1 2 3 4 5 6

1.000 0.053 -0.006 0.010 -0.011 -0.107 0.031

7 8 9 10 11 12 13

-0.093 0.025 0.166 0.026 0.029 -0.145 -0.067

14 15 16 17 18 19 20

0.025 0.041 -0.017 0.071 0.053 0.062 -0.104

Random Walk model

* Random noise/White noise/Residual
* As you go on a Random Walk, you ACUMMULATE noise

x=NULL

x[1] = 0

for(i in 2:1000) {

x[i]=x[i-1]+rnorm(1)

}

print(x)

random\_walk = ts(x)

plot(random\_walk,

main=”A random walk”,

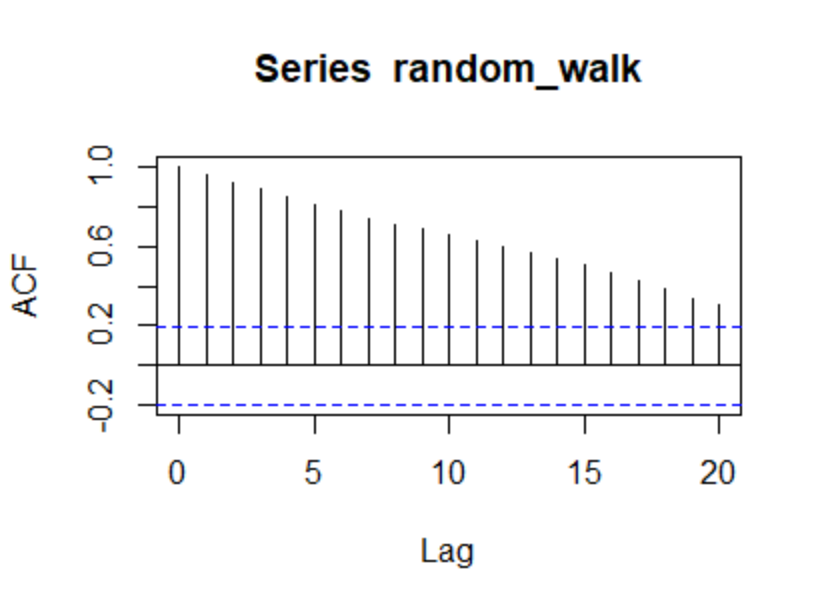
ylab= “ “,

xlab=”Days”,

col=”blue”,

lwd=2)

Output:



* High high correlation 20 steps back, there is NO stationarity

Remove the trend?

diff(random\_walk) Use diff() to remove the trend. Do this to get a stationary time

series from a random walk

plot(diff(random\_walk))

acf(diff(random\_walk))

Moving Average Process (MAq) – code below is to simulate moving average process in R

noise=rnorm(10000) Generate noise

ma\_2=NULL Introduce a variable

for(i in 3:10000){ Loop for generating MA(2) process

ma\_2[i]=noise[i]+0.7\*noise[i-1]+0.2\*noise[i-2]

}

moving\_average\_process=ma\_2[3:10000] Shift data to left by 2 units

moving\_average\_process=ts(moving\_average\_process) Put time series structure usingts()

par(mfrow=c(2,1)) Partition output graphics as a multi frame of 2 rows and 1 column. This creates a frame so you can see multiple plots

# plot the process and plot its ACF

plot(moving\_average\_process, main='A moving average process of order 2', ylab=' ', col='blue')

acf(moving\_average\_process, main='Correlogram of a moving average process of order 2')

US GDP data

install.packages("WDI") install WDI package

library("WDI") start the WDI package

gdp<-WDI(country-c("US"), set a gdp variable name. pull US data

indicator = c("NY.GDP.PCAP.CD"), pull this info using indicator

start=1960, start year

end=2016) end year

print(gdp)

head(gdp)

names(gdp)<-c(“iso2c”, use name() function to change column names

“country”,

“GDPperCap”,

“year”)

gdp<-gdp[order(gdp$year),] user order() function to sort in ascending

order using year column. Don’t forget COMMA

Next we need to check for stationarity. How? Plot it!! Should not see trend

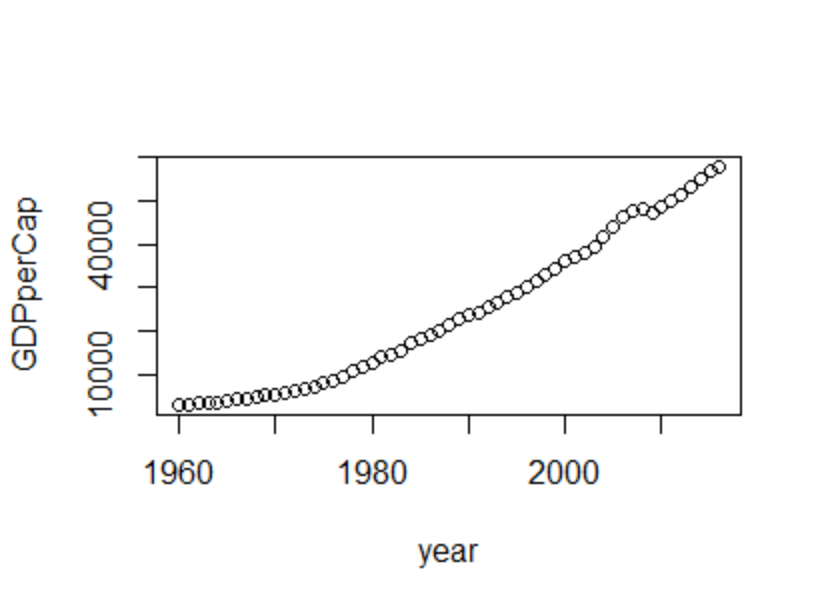
plot(GDPperCap~year, data=gdp) Plot GDPperCap column against year column

Note the TREND. It is NOT STATIONARY!!

We need to transform the dataset into

Timeseries format using ts() function.

Then, we need to do DIFFING!!



us<-ts(gdp$GDPperCap, Use ts() to transform GDPperCap column

start=min(gdp$year), start using min() function on year column

end=max(gdp$year)) end using max() function on year column

us Note: No longer df format!!

To implement DIFFING, you need the forecast package

install.packages("forecast") install forecast package

library(“forecast”) start the package

usOPT<-auto.arima(us) use auto.arima to find OPTIMAL model for “us”

variable (that holds the time series data)

usOPT To see the results

result:

Series: us

ARIMA(2,2,2) 1st # AR, 2nd # No of diffing, 3rd # MA

Coefficients:

ar1 ar2 ma1 ma2

1.1605 -0.5761 -1.5658 0.6772

s.e. 0.1925 0.1528 0.1888 0.1739

sigma^2 estimated as 260606: log likelihood=-419.58

AIC=849.15 AICc=850.38 BIC=859.19

To find Alpha 2 and Beta 2 (note that you need to go 2 years back based on optimal model),

Use COEF() function

coef(usOPT)

result:

ar1 ar2 ma1 ma2

1.1604934 -0.5761457 -1.5658090 0.6771980

1.1604 and -0.5761 are the values for beta 1 and 2 in our AR model

-1.5658 and 0.677 are the values for alpha 1 and 2 in our MA model

To PREDICT the GDPperCap in the USA for the next 5 years. Use predict() function

predict(usOPT, Enter 3 parameters! 1st, the model name

n.ahead=5, how many years you want to predict

se.fit=T) I’m not sure what this is

$pred

Time Series:

Start = 2017

End = 2021

Frequency = 1

[1] 58850.47 59818.93 60945.07 62228.00 63602.02 This is your next 5yr

prediction

$se

Time Series:

Start = 2017

End = 2021

Frequency = 1

[1] 510.4956 960.9012 1336.6428 1636.7300 1894.0292

GDPUSAForecast<- Can also use forecast() function

forecast(object = usOPT, Enter variable that holds results

h=5) Number of years you want to forecast

GDPUSAForecast

Result:

Point Forecast Lo 80 Hi 80 Lo 95

2017 58850.47 58196.24 59504.70 57849.92

2018 59818.93 58587.48 61050.37 57935.60

2019 60945.07 59232.10 62658.05 58325.30

2020 62228.00 60130.45 64325.55 59020.07

2021 63602.02 61174.72 66029.31 59889.79

Hi 95

2017 59851.02

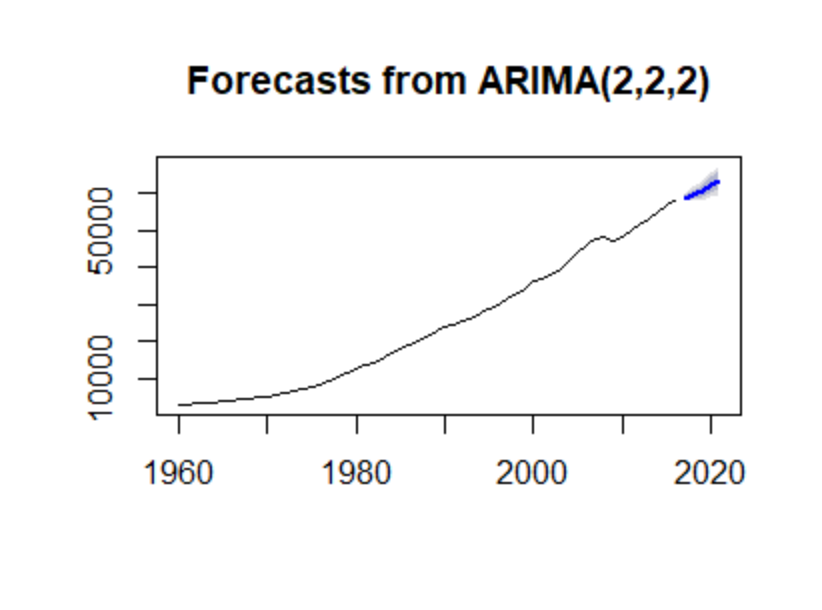
2018 61702.26

2019 63564.84

2020 65435.93

2021 67314.25

plot(GDPUSAForecast) To visualize final result, plot it



NOTES

* Use ACF plot to determine MA model
* Use PACF plot to determine the AR model
* Select the chart that has the LEAST number of vertical lines that exceed the dotted dash (going

Dotted dash means its significant). The more independent variables, the more complicated

the analysis. If there are 2 lines above significance in the PACF chart versus 5 lines in the ACF

chart, then we will choose PACF plot which means, AR(2) – 2 denoting the number of significant

lines