|  |  |
| --- | --- |
|  | |
| Data Advanced Data Analytics  Advanced Data Aanlyics Exam | |
| Module code : B8IT109 | |
| Ciaran Finnegan  Student No : 10524150  15/06/2020 |  |
|  |  |

Table of Contents

1 ADA – Exam Paper Submission 3

1.1 Course Details 3

1.2 Exam Declaration 3

2 Question One 4

2.1 Question 1 – from PDF 4

2.2 Output From RStudio Cloud Console 5

3 Question Two 6

3.1 Question 2 – from PDF 6

3.2 Output from RStudio Cloud Console 7

4 Question Three 8

4.1 Question 3 – from PDF 8

4.2 Output from RStudio Cloud Console 9

5 Question Four 10

5.1 Question 4 – from PDF 10

5.2 Output from RStudio Cloud Console 11

# ADA – Exam Paper Submission

## Course Details

|  |  |
| --- | --- |
| Module Code | B8IT109 (B8IT109\_1920\_SME3\_BHD08DNW) |
| Module Name | Advanced Data Analytics – Fri/Sat Part-time March 2019 Intake |
| Date | 15th June 2020 |
| Student number | 10524150 |
| Student name | Ciaran Finnegan |

## Exam Declaration

*By uploading this exam from my Moodle account I Ciaran Finnegan am confirming that this document is all my own work.*

*I understand that DBS will carry out checks such as text-matching (via Urkund), bench-marking and viva voce exams in order to verify the authenticity of submissions.*

## Notes on Exam Submission Format

* The exam answers are in two parts;
  + Part 1 is a reproduction of the exam question.
  + Part 2 is the answer to the exam question and is a copy/paste of the contents of my RStudio console.
* I have attempted to keep the following colour scheme with the answer output from the RStudio console;
  + **BLUE** represents my comments in the R code.
  + **GREEN** represents actual R code that has been executed.
  + **BLACK** represents the output of the executed code.

# Question Four – Time Series

## Question 1 – from PDF

Use dataset available on

http://www.stat.ufl.edu/~winner/data/clotthes\_expend.csv , apply time

series analysis, consider **sales.b** as your time series variable:

1. Validate the assumptions using graphical visualization.

**(5 Marks)**

b) Fit the optimized model for **sales.b** and provide the

coefficient estimates for the fitted model. **(7.5 Marks)**

c) What is the estimated order for AR and MA?

**(5 Marks)**

d) Forecast h=10 step ahead prediction of **sales.b** on the plot of

the original time series.

**(7.5 Marks)**

**(Total: 25 Marks)**

## Output From RStudio Cloud Console – Question Four

## Exam Advanced Data Analytics : Module Code B8IT109

## Advanced Data Analytics : Module Code B8IT109

> ## Student Name : Ciaran Finnegan 10524150

>

> ## Exam Submission

>

> ## June 15th 2020

>

> ## Question 4 : Time Series Analysis

>

> ## Use dataset available on

> ## http://www.stat.ufl.edu/~winner/data/clotthes\_expend.csv ,

> ## apply time series analysis,

> ## consider sales.b as your time series variable:

>

>

>

> ## Read in the CEP dataset

> datasetCEP <- read.csv("clotthes\_expend.csv")

>

> ## Brief Review of number of rows, head and tail of dataset records

> ## the and structure of dataset

> nrow(datasetCEP)

[1] 85

> head(datasetCEP)

year sales.b price.index sales.index pop.m realgdp.b ad.gdppct

1 1929 9.0 27.859 32.30554 121.77 1,056.60 2.8

2 1930 7.8 26.591 29.33323 123.08 966.7 2.7

3 1931 6.7 23.231 28.84077 124.04 904.8 2.7

4 1932 4.9 19.081 25.68000 124.84 788.2 2.8

5 1933 4.5 19.446 23.14101 125.58 778.3 2.3

6 1934 5.5 22.545 24.39565 126.37 862.2 2.5

> tail(datasetCEP)

year sales.b price.index sales.index pop.m realgdp.b ad.gdppct

80 2008 319.5 99.130 322.3040 304.09 14,830.40 2.7

81 2009 306.5 100.000 306.5000 306.77 14,418.70 2.6

82 2010 320.6 99.347 322.7073 309.33 14,783.80 2.6

83 2011 338.9 101.089 335.2491 311.59 15,020.60 2.6

84 2012 353.7 104.744 337.6804 313.91 15,369.20 2.6

85 2013 360.7 105.732 341.1455 316.16 15,710.30 2.6

> str(datasetCEP)

'data.frame': 85 obs. of 7 variables:

$ year : int 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 ...

$ sales.b : num 9 7.8 6.7 4.9 4.5 5.5 5.8 6.3 6.6 6.5 ...

$ price.index: num 27.9 26.6 23.2 19.1 19.4 ...

$ sales.index: num 32.3 29.3 28.8 25.7 23.1 ...

$ pop.m : num 122 123 124 125 126 ...

$ realgdp.b : chr "1,056.60" "966.7" "904.8" "788.2" ...

$ ad.gdppct : num 2.8 2.7 2.7 2.8 2.3 2.5 2.3 2.3 2.3 2.2 ...

>

>

> ## Minor Clean up of CEP dataset

> sum(is.na(datasetCEP))

[1] 0

> datasetCEP <- na.omit(datasetCEP)

> sum(is.na(datasetCEP))

[1] 0

> nrow(datasetCEP) # Confirm rows after missing data removed = 0

[1] 85

> ## No missing rows in CEP dataset

>

> ## Review 'sales.b' attribute

> table(datasetCEP$sales.b)

4.5 4.9 5.5 5.8 6.3 6.5 6.6 6.7 6.9 7.2 7.8 8.5 9 10.6 12.9 14.1 15.9 17.5 18 18.6 18.9 19.3 20.5 21.2 21.3 21.4

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

22.4 23.4 23.6 23.9 25.4 25.9 26.6 27.9 28.6 31.1 32.7 35.8 37.5 41.3 44.3 45.5 49 53.5 59.2 62.4 66.9 72.2 79.3 89.3 96.4 103

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

113.2 116.7 126.4 137.6 146.8 157.2 167.7 178.2 190.4 195.2 199.1 211.2 219.1 227.4 231.2 239.5 247.5 257.8 271.1 277.9 278.8 280.8 285.3 297.5 306.5 310.7

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

319.5 320.2 320.6 323.7 338.9 353.7 360.7

1 1 1 1 1 1 1

>

> sales\_price <- datasetCEP$sales.b

>

>

> #################################################################

> ## Q.4 (Part a)

> >

> ## (a) Validate the assumptions using graphical visualization.

>

> ## Run functions to look at the structure of the closing price

> ## dataset for our chosen stock

> View(sales\_price)

>

> ## <Screen shot of 'View' output...here>



>

>

> ## Invoke the 'ts' function on the 'sales\_price' time series.

> T <- ts(sales\_price, frequency = 1)

> >

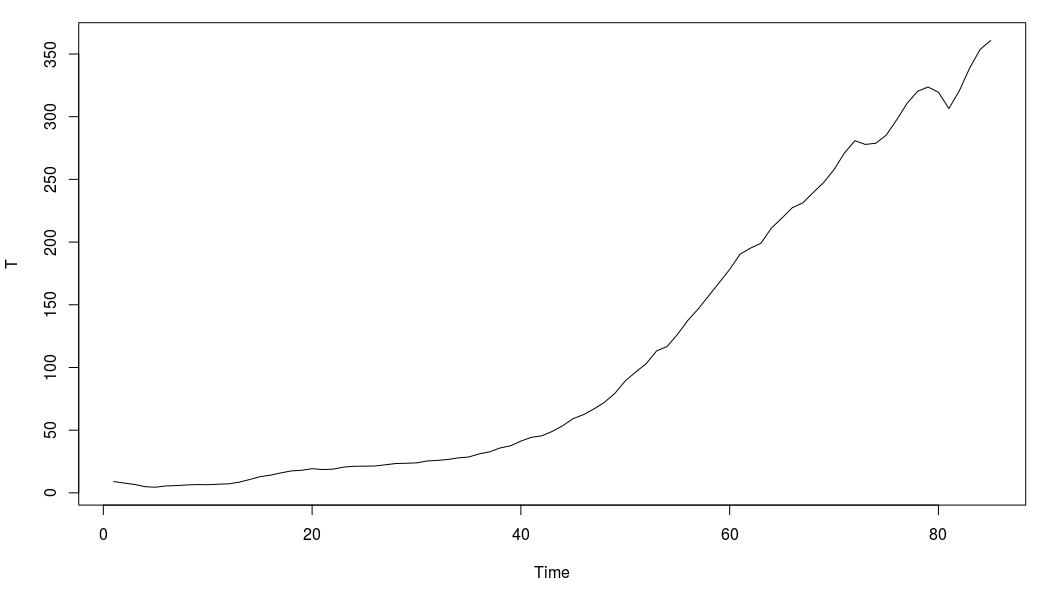
> ## Generate the plot of the time series variable- the range

> ## represents the sales prices extracted from the time range

> ## of data (frequency = 1 so every daily sales price is plotted).

> plot(T)

>



>

> ## I can see that the time series is not particularly stationary

> ## in terms of mean or variance

>

>

> ## I apply 'diff' and 'log' functions to smooth out the graph plot

>

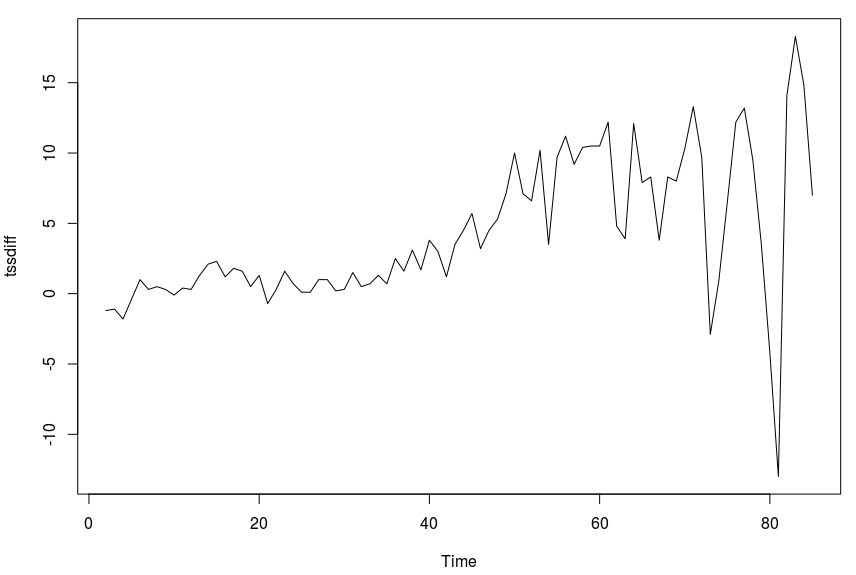
> ## Apply 'diff' function

> tssdiff=diff(T) # Stationary in mean

> plot(tssdiff)

>

> ## <Put diff plot graph here..>



>

> ## The plot of 'diff' is more stationary in mean, with an average

> ## somewhat around zero.

>

>

> ## Apply log function, then applying 'diff', to achieve a

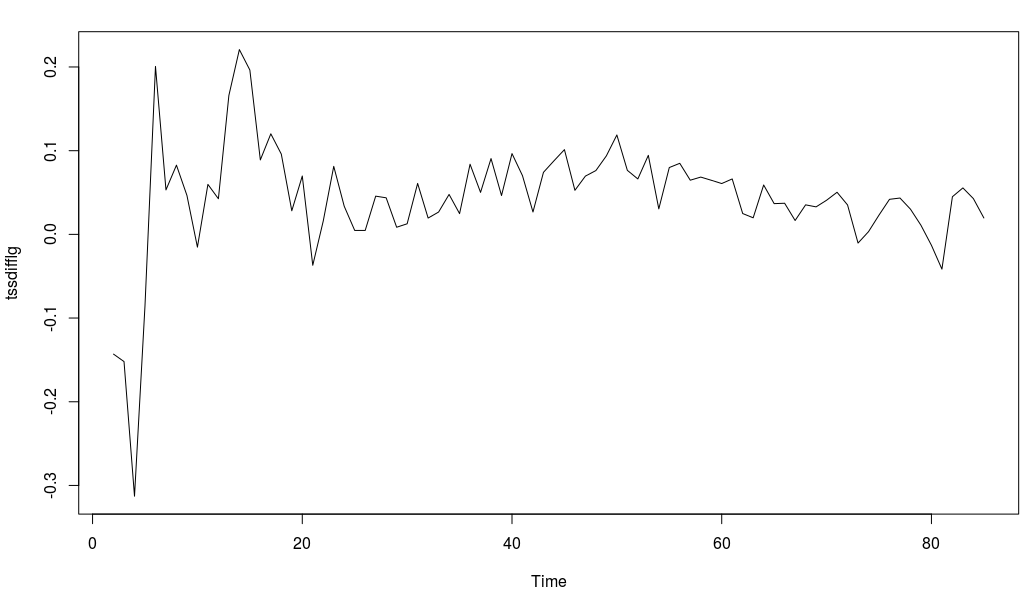
> ## stationary visualisation in mean and variance

> tssdifflg = diff(log(T))

> plot(tssdifflg)

>

> ## <Put log plot graph here..>



>

>

> ## This graph shows mean as approximately stationary and the

> ## variance also stationary between -0.1 and 0.02,

> ## apart from a few outliers

>

>

> ## --------------------------------------------------------------------------------

> ## Also - run ggqqplot to graphing the data and show level of normality in the data set

> ## --------------------------------------------------------------------------------

>

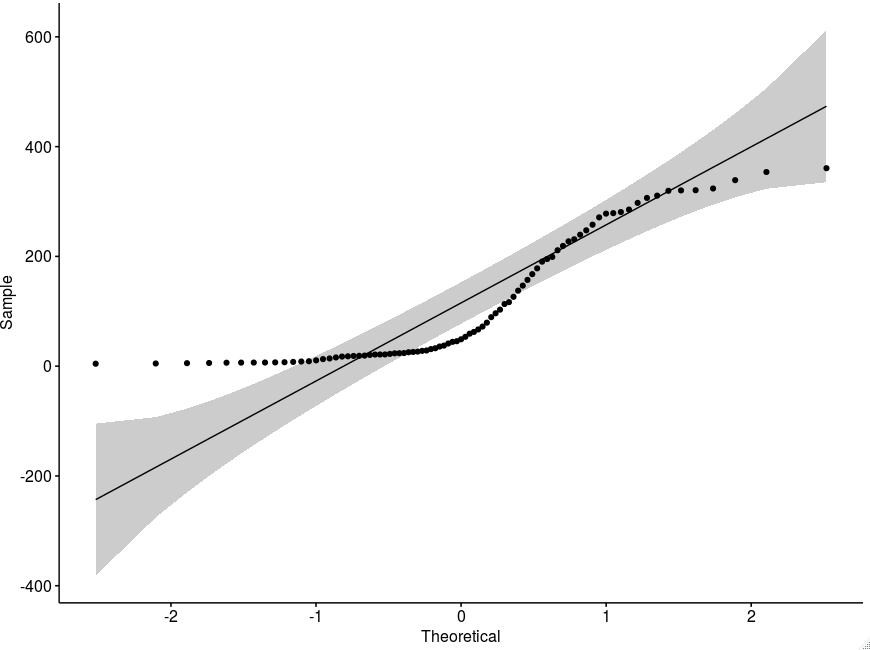
>

> ## ggqqlog plot graph

> ggqqplot(T)

>

> ## <Put ggqqlog plot graph here..>



>

>

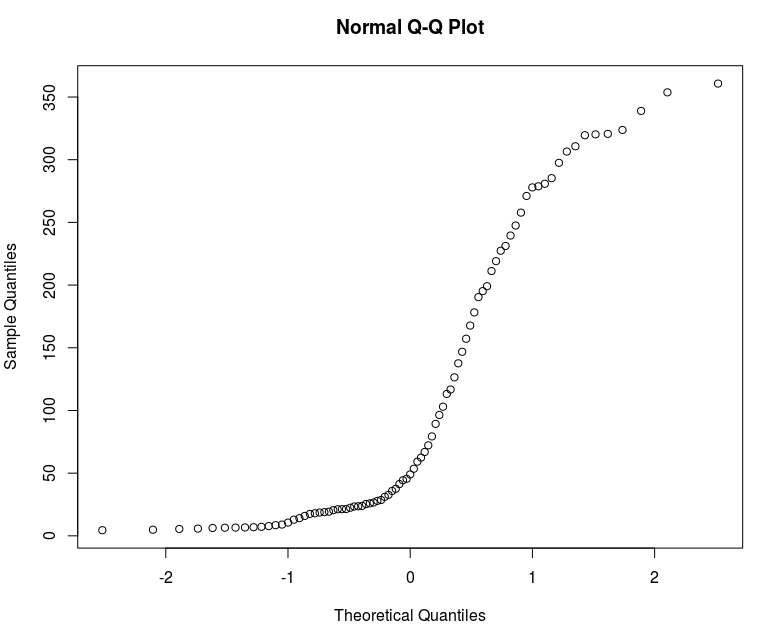
> ## qqnorm plot graph

> qqnorm(T)

>

> ## <Put qqnorm plot graph here..>

>



> #################################################################

> ## Q.4 (Part b)

>

> ## Fit the optimized model for ‘sales.b’ and provide the coefficient

> ## estimates for the fitted model.

>

> ## To compute optimised coefficient estimates for fitted model we have

> ## two approaches :-

> ## 1:- Apply 'acf' and 'pacf' to get estimation of order, and also estimate parameters.

> ## 2:- Apply ARIMA manually

> ## 3:- Apply Auto ARIMA

> ## 4:- Select the model with the lowest AiC (Akaike Information Criterion)

> ## value and use those coefficient values

>

>

> ## It is necessary to apply both methods (manual and automatic) and see which

> ## one has a lower AIC, then determine that method is optimised.

> ## Try and find as low a value of AIC as possible

>

>

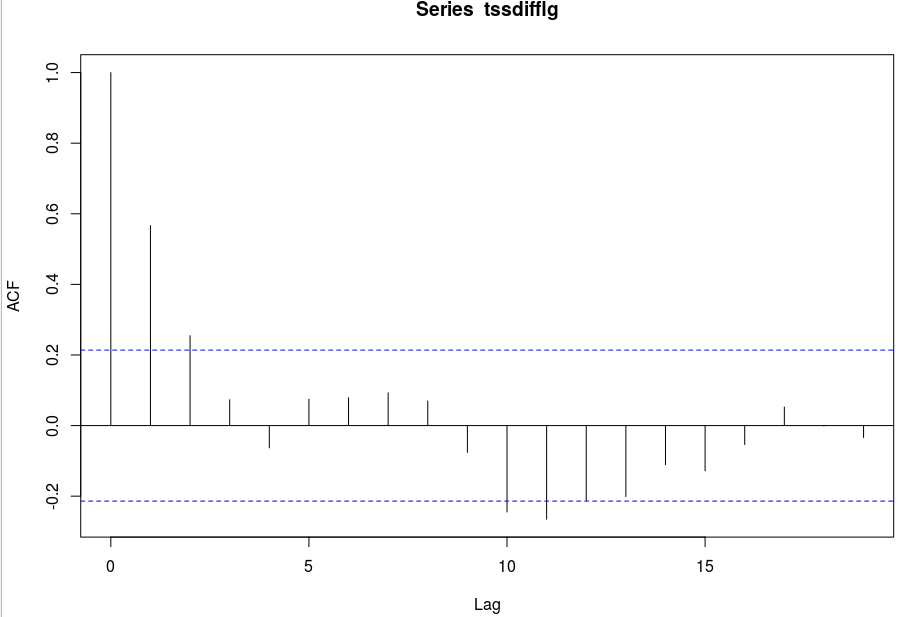
> ## 1 - Apply 'acf' and 'pacf' to get estimations of 'q' and 'p'

> ## acf = autocorrelation function. Gives us the estimation for 'q'

> acf(tssdifflg)

>

> ## <Put acf plot graph here..>



>

> ## There are two initial lags outside the boundary, therefore q = 3.

> ## (Above or below boundary line is not important).

>

>

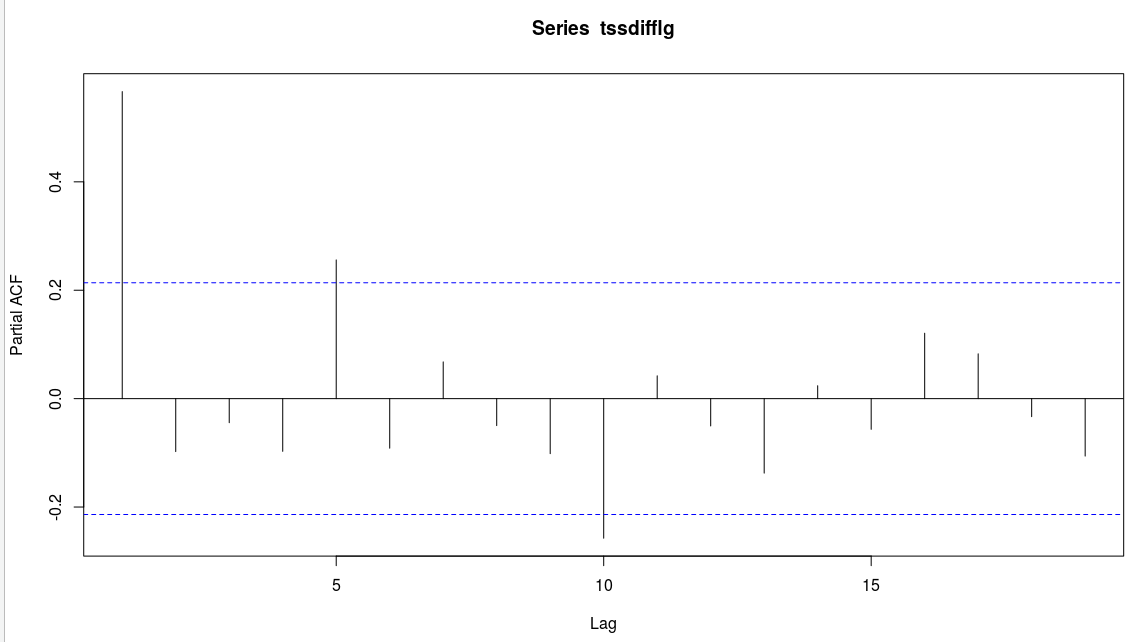
>

> ## pacf = partial autocorrelation function. Gives is the estimation for 'p'

> pacf(tssdifflg)

>

> ## <Put pacf plot graph here..>



>

> ## One initial lag is outside are outside the bounds, therefore p = 1

>

>

> ## Now use 'arima' function to fit a manual ARIMA; p = 1, (1 diff used), q = 3.

> ## The original time series with sales.b is used.

> ## ARIMA (p,d,q) Model : Using original time series 'T'

> ## Parameter Estimation

> manual.fit <- arima(T, c(1,1,3), method="ML") # Fitted Model

> ## Display value of 'manual.fit'

> manual.fit

Call:

arima(x = T, order = c(1, 1, 3), method = "ML")

Coefficients:

ar1 ma1 ma2 ma3

0.9918 -0.3247 -0.3709 -0.1285

s.e. 0.0114 0.1150 0.1056 0.1215

sigma^2 estimated as 14.19: log likelihood = -231.46, aic = 472.93

>

> ## With p = 1, we see one value for the 'ar1' coefficient

> ## With q = 3, we see two values for the 'ma' (moving average) coefficients

> ## The values just under the 'ar1', 'ma1', and 'ma2' headings are the estimation

> ## of parameters

>

> ## Coefficients:

> ## ar1 ma1 ma2 ma3

> ## 0.9918 -0.3247 -0.3709 -0.1285

>

>

> ## We can see the aic (Akaike Information Criterion) value = 472.93

>

>

>

>

>

> ## Next we need to apply 'auto.arima' to generate a fitted model

> ## 'seasonal' = F - time series does not have a seasonality trend

> auto.fit <- auto.arima(T, seasonal = FALSE)

> auto.fit

Series: T

ARIMA(3,2,2)

Coefficients:

ar1 ar2 ar3 ma1 ma2

0.8457 -0.5780 -0.2651 -1.4855 0.7370

s.e. 0.1409 0.1398 0.1353 0.1086 0.0905

sigma^2 estimated as 10.27: log likelihood=-213.96

AIC=439.92 AICc=441.03 BIC=454.43

>

> ## AIC = 439.92

>

> ## 'seasonal' = F - time series does not have a seasonality trend

> auto.fit.T <- auto.arima(T, seasonal = TRUE)

> auto.fit.T

Series: T

ARIMA(3,2,2)

Coefficients:

ar1 ar2 ar3 ma1 ma2

0.8457 -0.5780 -0.2651 -1.4855 0.7370

s.e. 0.1409 0.1398 0.1353 0.1086 0.0905

sigma^2 estimated as 10.27: log likelihood=-213.96

AIC=439.92 AICc=441.03 BIC=454.43

> # 'seasonal' flag makes no difference to result

>

> ## AIC = 439.92

>

>

> ## Automated coefficient are lower as 472.93 (Manual) > 439.92 (Auto).

> ## Therefore Auto ARIMA is better than manual fitting.

>

>

>

> #################################################################

> ## Q.4 (Part c)

>

> ## What is the estimated order for AR and MA?

> auto.fit

Series: T

ARIMA(3,2,2)

Coefficients:

ar1 ar2 ar3 ma1 ma2

0.8457 -0.5780 -0.2651 -1.4855 0.7370

s.e. 0.1409 0.1398 0.1353 0.1086 0.0905

sigma^2 estimated as 10.27: log likelihood=-213.96

AIC=439.92 AICc=441.03 BIC=454.43

> ## The best model shows Series: T ARIMA(3,2,2)

> ## Therefore the estimated order for AR is p = 3, and MA is q = 2

>

>

>

> #################################################################

> ## Q.4 (Part d)

>

> ## Forecast h=10 step ahead prediction of wage on the plot of the

> ## original time series.

>

>

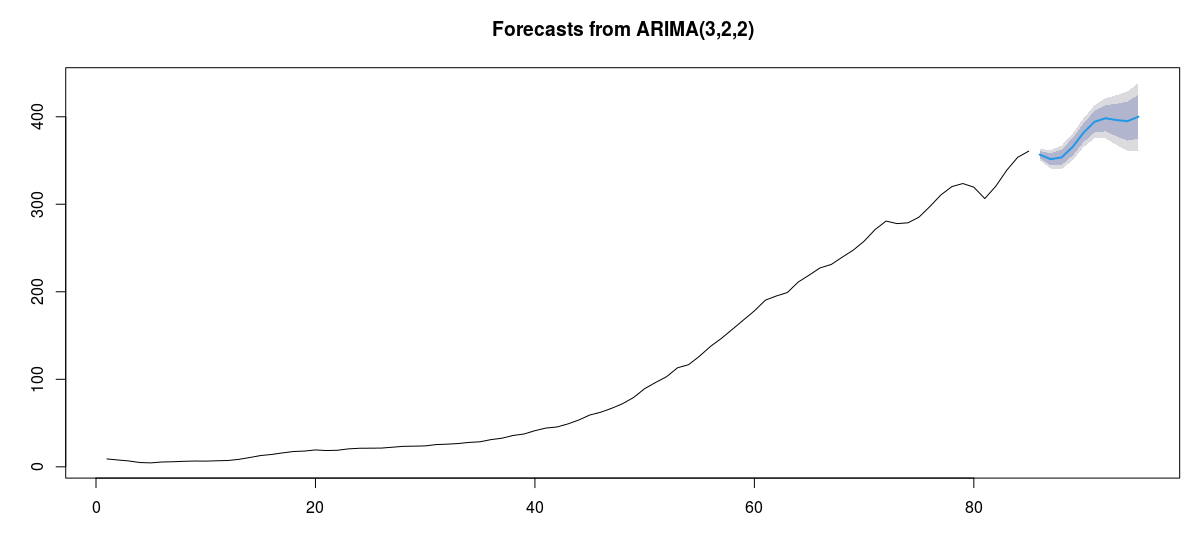
> # The best model to use is the auto fitting - as determined in the analysis in the previous

> ## steps in the question.

> auto.fcast <- forecast(auto.fit, h = 10) # Prediction for 10 step ahead

> ## Plot this forecast

> plot(auto.fcast)



|  |
| --- |
|  |
|  |
|  |
|  |
| |  | | --- | |  | |

# Question Two

## Question 2 – from PDF

## Output from RStudio Cloud Console

> ## CA Two Advanced Data Analytics : Module Code B8IT109

# Question Three

## Question 3 – from PDF

## Output from RStudio Cloud Console

## CA Two Advanced Data Analytics : Module Code B8IT109

# Question Four

## Question 4 – from PDF

## Output from RStudio Cloud Console

> ## CA Two Advanced Data Analytics : Module Code B8IT109