PRACTICAL – 09

# AIM: Demonstration of Time-series forecasting (Using Python or R)

# THEORY:

Time series forecasting is a method in the statistics field to analyze historical data with a time component and

create a prediction based on it.

Some classic examples of time series forecasting methods are Moving Average, ARIMA, and Exponential

Smoothing. These methods have been used for a long time and are still useful now because of how easy it is

for users to explain the result — although with less accurate prediction.

Whether you need a classical approach or a machine learning-driven model, many have developed Python

packages to access all these methods. Some of the famous packages are Statsmodel, pmdarima, and sktime.

However, the forecasting model is not limited to only the ones that are listed above because many great

packages are worthy of consideration.

1. StatsForecast – It is a Python package that provides a collection of univariate time-series Forecasting

models. What is unique about StatsForecast is the model provides fast training and is optimized for

high-accuracy models. Also, the package offers several benchmarks we could use when training

various models.

2. PyAF - PyAF or Python Automatic Forecasting is an open-source Python package to automatically

develop time-series forecasting models (either univariate or with exogenous data). The model was

built on top of Scikit-Learn and Pandas, so expect familiar APIs. The package also offers various

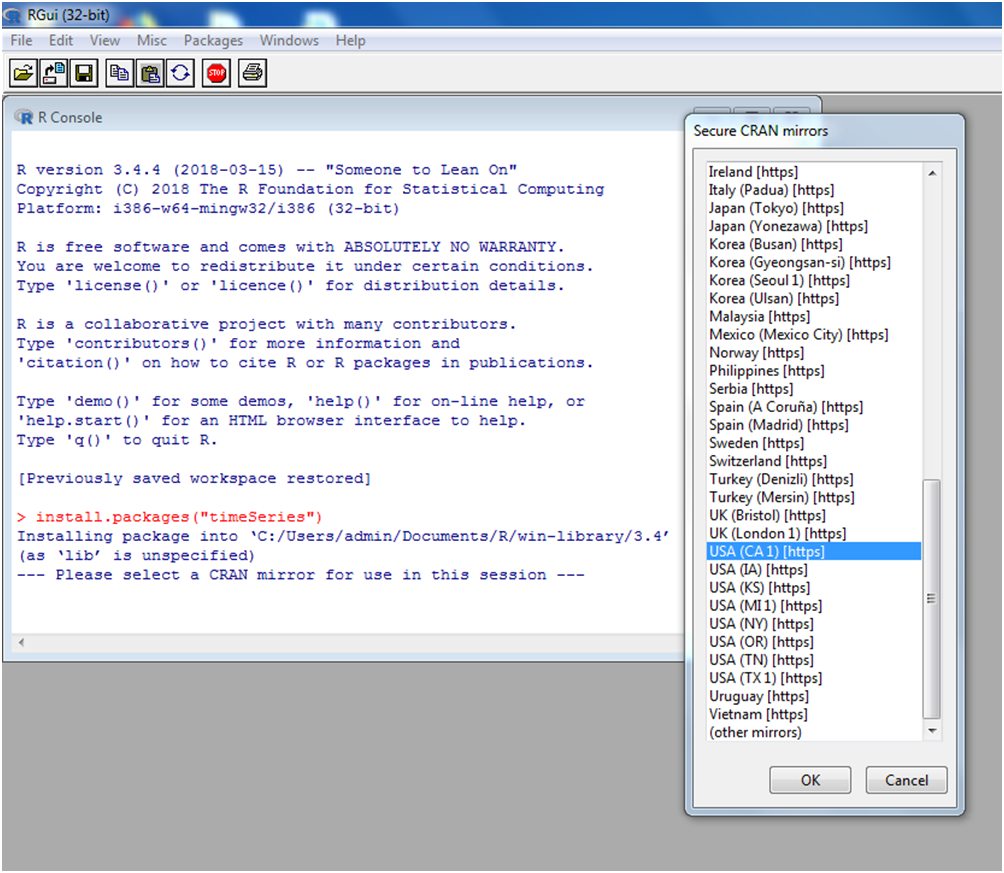
models to use in a few lines as much as possible.

3. NeuralProphet - It is a Python Package for developing a time-series model based on Facebook Prophet

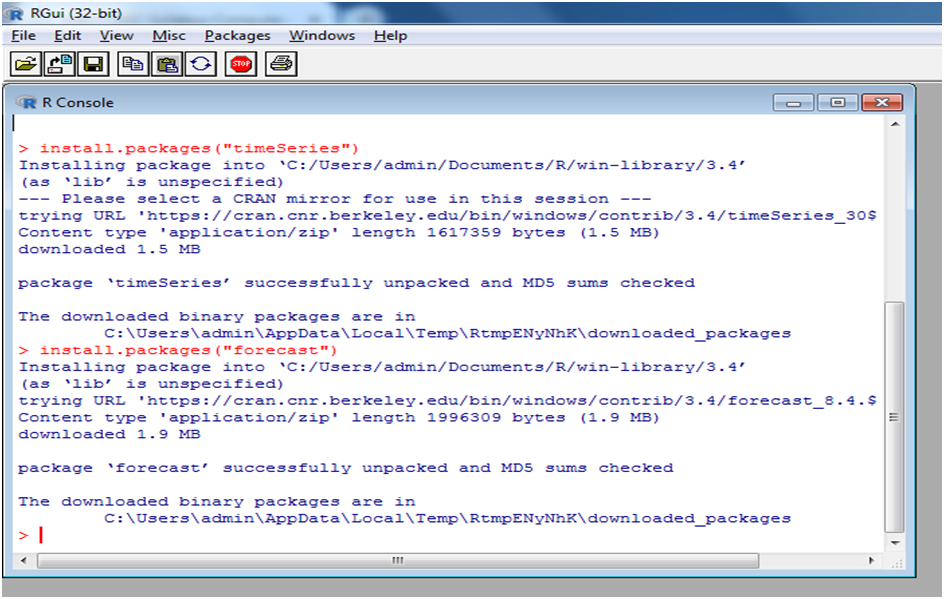
but with Neural Network architecture. The package is based on PyTorch and could easily be used with

as few lines as possible.

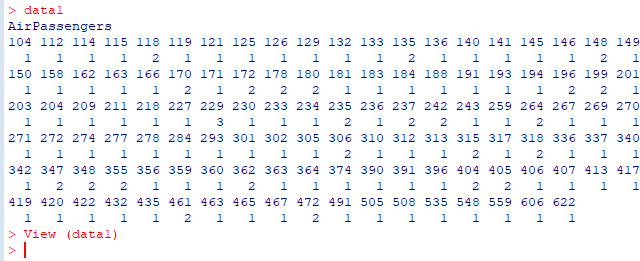
IMPLEMENTATION AND OUTPUT:

STEP 1: Install time series install.packages(“timeSeries”)

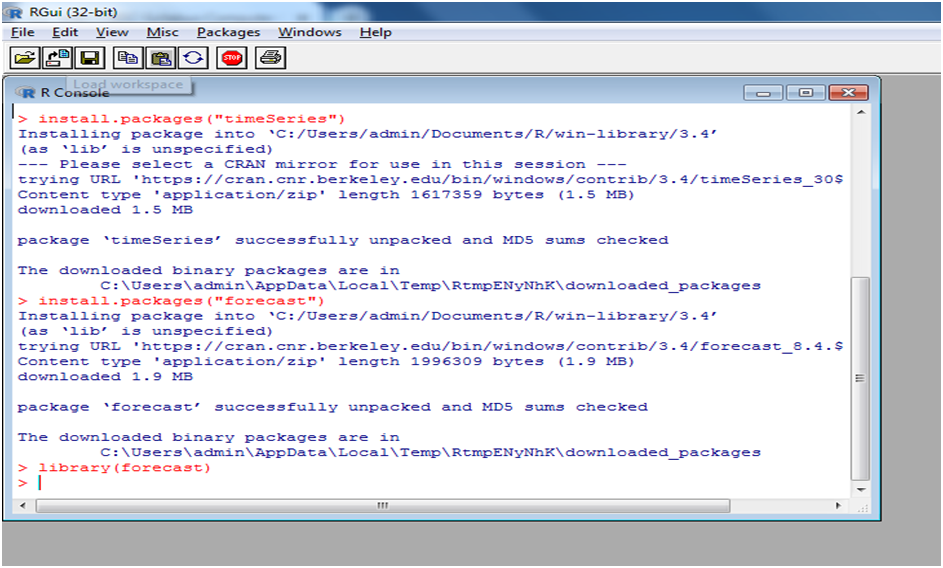
Step 2: Install package forecast install.packages(“forecast”)



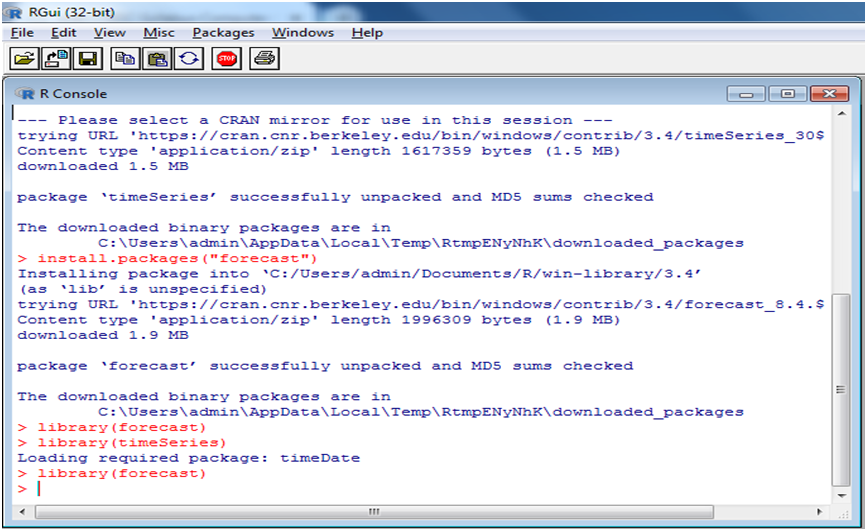
data1=table(AirPassengers)

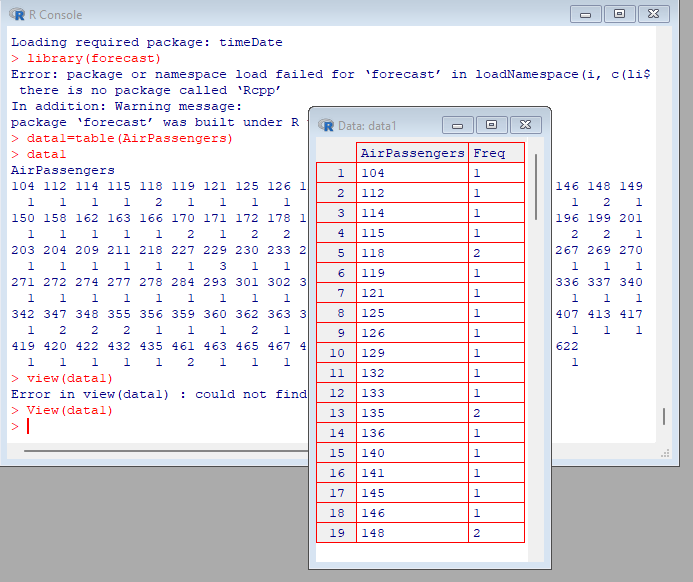
Data1 view(data1)

**Step 3:** library (timeSeries) #library(forecast)

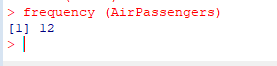


library(forecast)



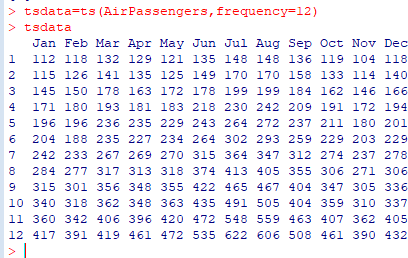
Step 5: Air Passengers data data1=table(AirPassengers) data1

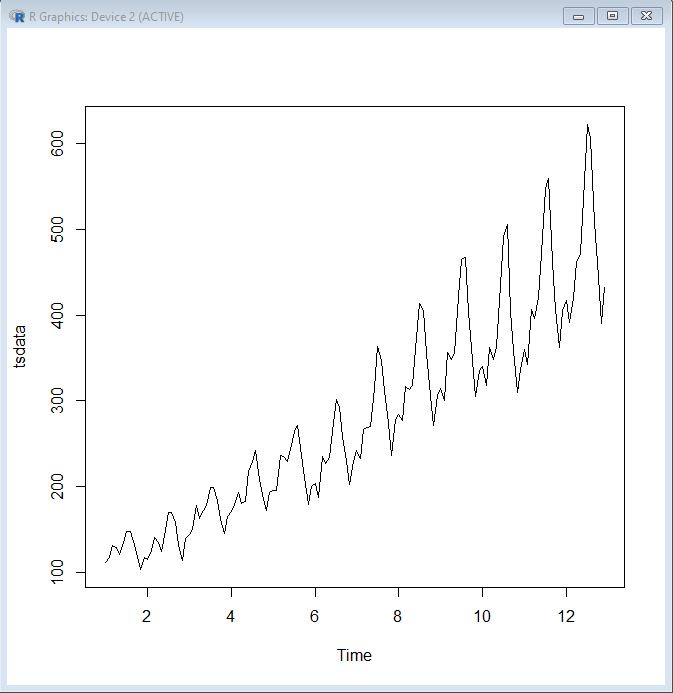
frequency (AirPassengers)



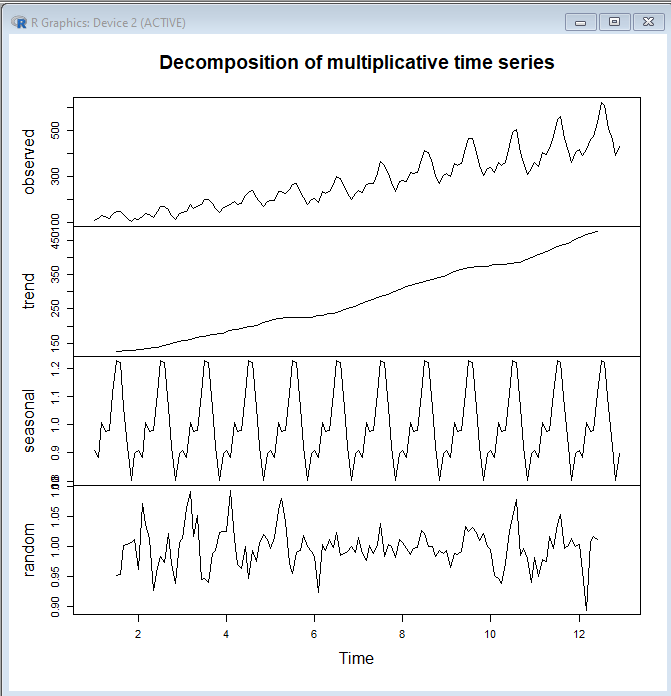
tsdata=ts(AirPassengers,frequency=12)

* tsdata

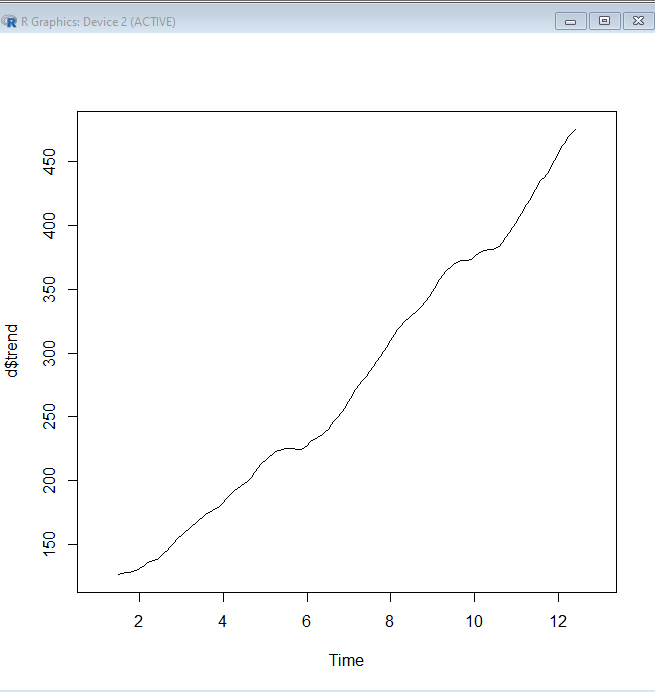


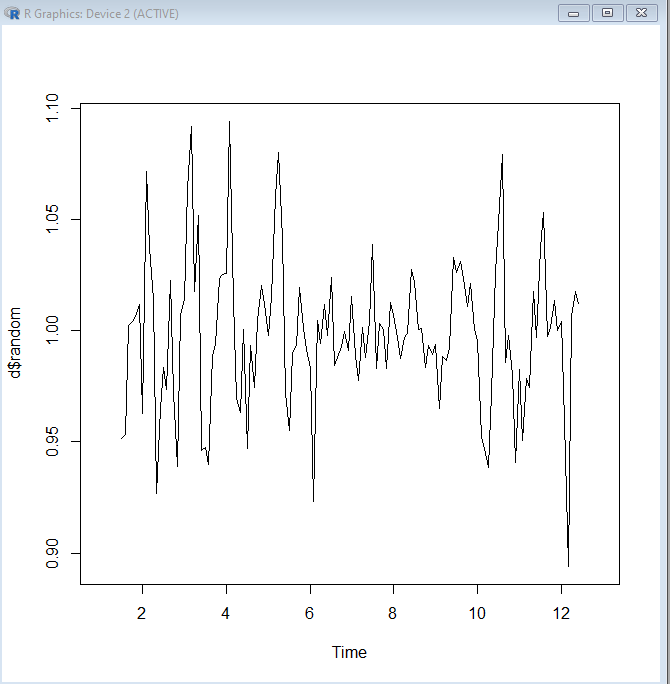
plot(tsdata)

* d=decompose(tsdata,"multiplicative")
* plot(d)

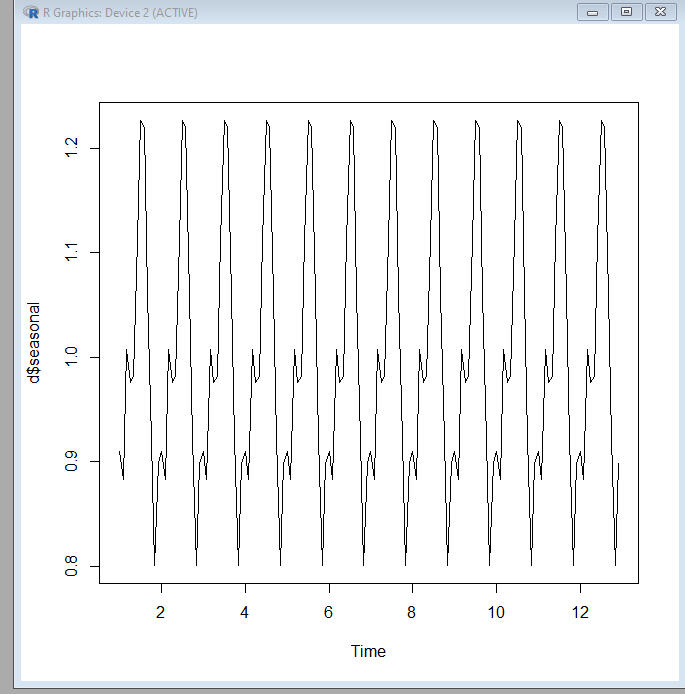


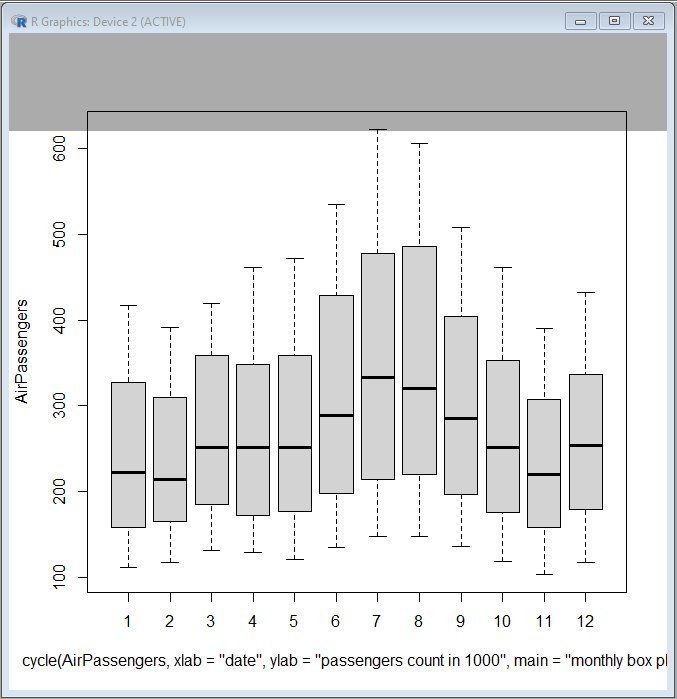
plot(d$trend)



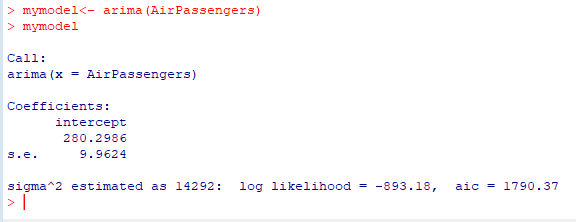
plot(d$random)

plot(d$seasonal)



boxplot(AirPassengers~cycle(AirPassengers,xlab="date",ylab="passengers count in 1000",main="monthly box plot"))

mymodel<- arima(AirPassengers) mymodel



CONCLUSION: Hence we have successfully implemented Demonstration of Time-series forecasting (Using Python or R)