Practical 8

-Shivam Tawari A-58

Aim: Write a program in R for implementing a time series analysis on given dataset.

Theory:

- Time series is a series of data points in which each data point is associated with a time stamp.
- The data for the time series is stored in an R object called time-series object.
- It is also a R data object like a vector or dataframes.
- Time series object is created by using the ts() function.

timeseries.object.name ← ts(data, start, end, frequency)

Here,
→ data is a vector or matrix containing the value used in time series.

→ start specifies the start time for the first

→ end specifies the end time for the last observation in a time series.

→ Frequency specifies the number of observations per unit time.

The value of the frequency parameter in the ts() function decides the time intervals at which the data points are measured.

frequency = 12 (pegs the data points for every month of a year)
frequency = 4 (pegs the data points for every quarter of a year)
frequency = 6 (pegs the datapoints for every 10 mins of an hour)
frequency = 24*6 (pegs the data points for every 10 minutes of a day)

Example 1

Code:

```
rain ← c(799, 1174.8, 865.1, 1334.6, 635.4, 918.5,
         685.5, 998.6, 784.2, 985, 882.8, 1071)
rain.timeseries ← ts(rain, start = c(2012, 1), frequency
                     = 12)
```

```
print (rain. timeseries)
Png ( file = "rainfall. png")
plot (rain. timeseries)
dev. off ()
```

#Example 2

Code:

```
data (AirPassengers)
Class (AirPassengers)
start (AirPassengers)
end ( AirPassengers)
frequency (AirPassengers)
summary (AirPassengers)
plot (AirPassengers)
abline( reg = lm(AirPassengers ~ time (AirPassengers ))')
cycle ( AirPassengers)
plot (aggregate ( AirPassengers , fun = mean))

tsdata & ← ts (AirPassengers , frequency = 12 )
ddata ← decompose (tsdata , "multiplicative")
plot (ddata)
boxplot (AirPassengers ~ cycle (AirPassengers))
plot ( ddata $ trend )
plot (ddata $ seasonal)
```

plot (ddata $random)
plot (AirPassengers)
abline (reg = lm (AirPassengers ~ time (AirPassengers))

Example 3

Code :

```
kings ← scan ("http://robjhyndman.com/tsdldata/
          misc/kings.dat", skip = 3)
kings
kingstimeseries ← ts(kings)
kingstimeseries
plot.ts(kingstimeseries)
install.packages ("TTR")
library ("TTR")
kingstimeseries SMA8 ← SMA(kingstimeseries, n=8)
plot.ts (kingstimeseries SMA8)
```

SMA () function :

Simple Moving Average is a method of time series smoothing and is actually a very basic forecasting technique. It does not need estimation of parameters, but rather is based on order selection.

The SMA() function in the TTR package can be used to smooth time series data using a moving average. The SMA function takes a span argument as n order. To calculate the moving average of order 8, we set n = 8.

Conclusion: Hence, we have successfully implemented time series analysis on the given dataset.

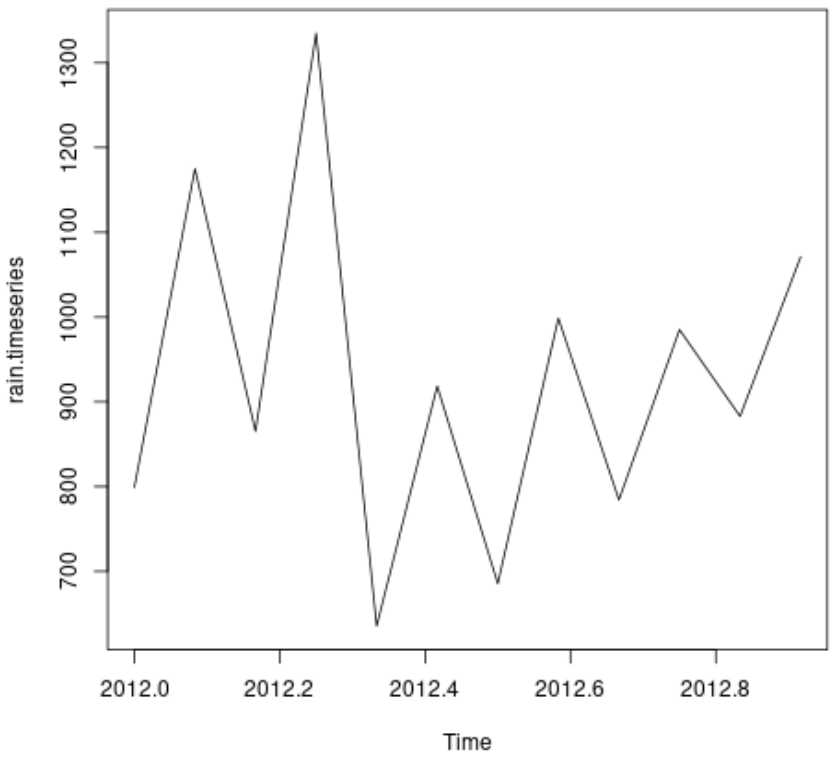
Example 1:

Code:

```
1  # Shivam Tawari (A-58)
2  rain <- c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)
3  rain.timeseries <- ts(rain,start = c(2012,1),frequency = 12)
4  print(rain.timeseries)
5  png(file = "rainfall.png")
6  plot(rain.timeseries)
7  dev.off()
8
```

Output:

```
> # Shivam Tawari (A-58)
> rain <- c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)
> rain.timeseries <- ts(rain,start = c(2012,1),frequency = 12)
> print(rain.timeseries)
        Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct    Nov
2012  799.0 1174.8  865.1 1334.6  635.4  918.5  685.5  998.6  784.2  985.0  882.8
        Dec
2012 1071.0
> png(file = "rainfall.png")
> plot(rain.timeseries)
> dev.off()
null device
          1
```



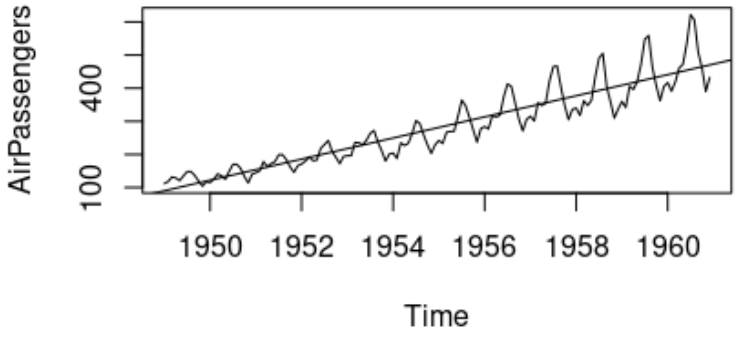
Example 2:

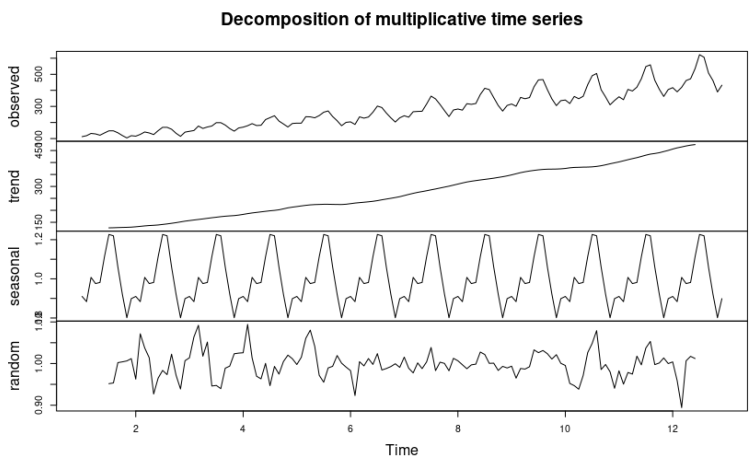
Code:

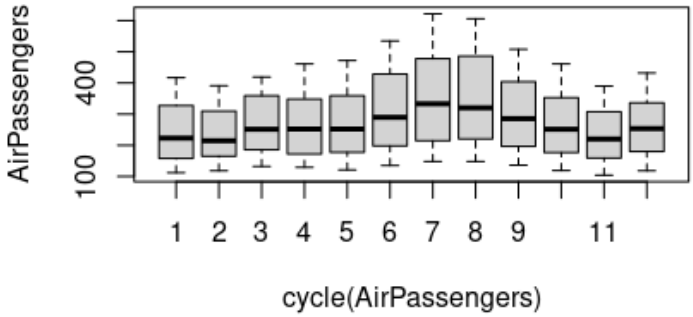
```r
# Shivam Tawari (A-58)
data(AirPassengers)
class(AirPassengers)
start(AirPassengers)
end(AirPassengers)
frequency(AirPassengers)
summary(AirPassengers)
plot(AirPassengers)
abline(reg=lm(AirPassengers~time(AirPassengers)))
cycle(AirPassengers)
plot(aggregate(AirPassengers,FUN=mean))

tsdata<- ts(AirPassengers,frequency=12)
ddata<- decompose(tsdata,"multiplicative")
plot(ddata)
boxplot(AirPassengers~cycle(AirPassengers))
plot(ddata$trend)
plot(ddata$seasonal)
plot(ddata$random)
plot(AirPassengers)
abline(reg=lm(AirPassengers~time(AirPassengers))
```

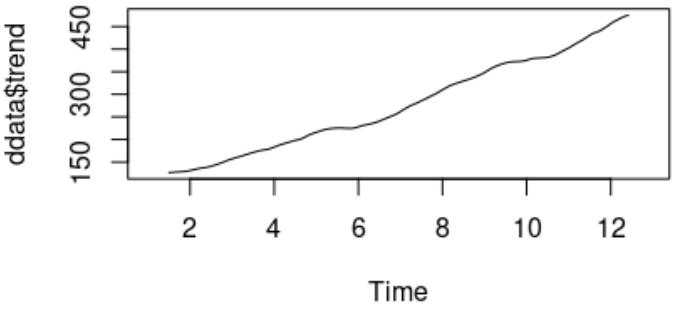
Output:

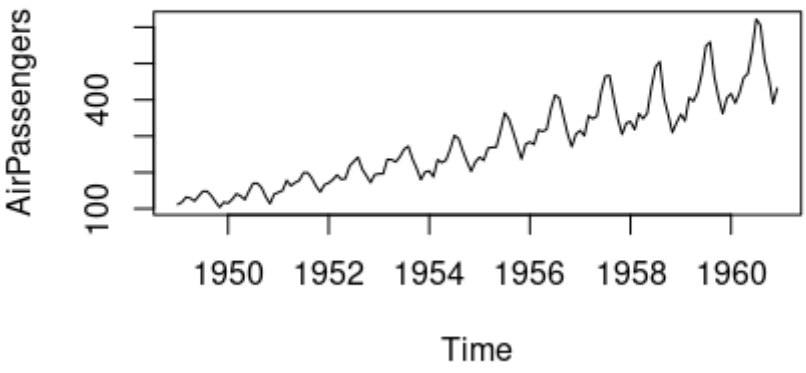
```
Console  Terminal  Jobs
/cloud/project/
> # Shivam Tawari (A-58)
> data(AirPassengers)
> class(AirPassengers)
[1] "ts"
> start(AirPassengers)
[1] 1949    1
> end(AirPassengers)
[1] 1960   12
> frequency(AirPassengers)
[1] 12
> summary(AirPassengers)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  104.0   180.0   265.5   280.3   360.5   622.0
> plot(AirPassengers)
> abline(reg=lm(AirPassengers~time(AirPassengers)))
> cycle(AirPassengers)
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949   1   2   3   4   5   6   7   8   9  10  11  12
1950   1   2   3   4   5   6   7   8   9  10  11  12
1951   1   2   3   4   5   6   7   8   9  10  11  12
1952   1   2   3   4   5   6   7   8   9  10  11  12
1953   1   2   3   4   5   6   7   8   9  10  11  12
1954   1   2   3   4   5   6   7   8   9  10  11  12
1955   1   2   3   4   5   6   7   8   9  10  11  12
1956   1   2   3   4   5   6   7   8   9  10  11  12
1957   1   2   3   4   5   6   7   8   9  10  11  12
1958   1   2   3   4   5   6   7   8   9  10  11  12
1959   1   2   3   4   5   6   7   8   9  10  11  12
1960   1   2   3   4   5   6   7   8   9  10  11  12
> plot(aggregate(AirPassengers,FUN=mean))
```

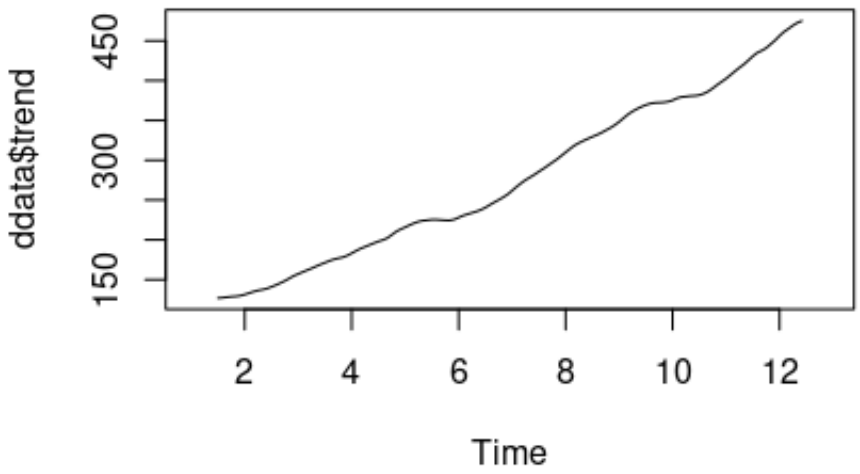

AirPassengers
100
400
1950
1952
1954
1956
1958
1960
Time

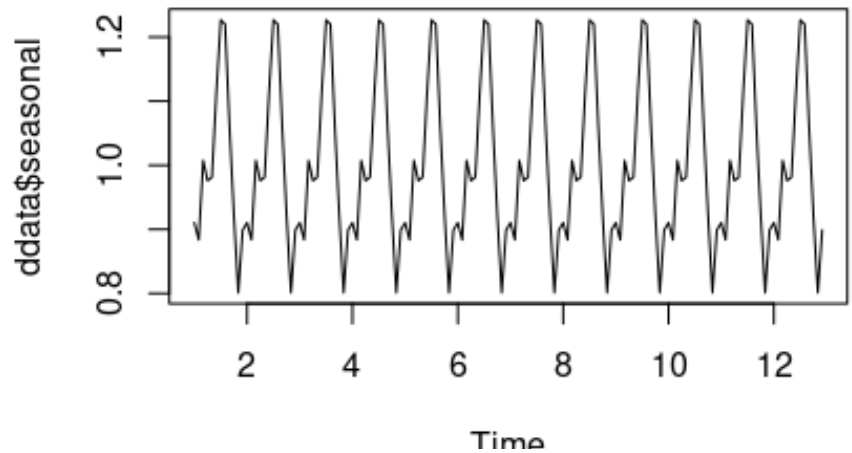

Decomposition of multiplicative time series
observed
trend
seasonal
random
Time

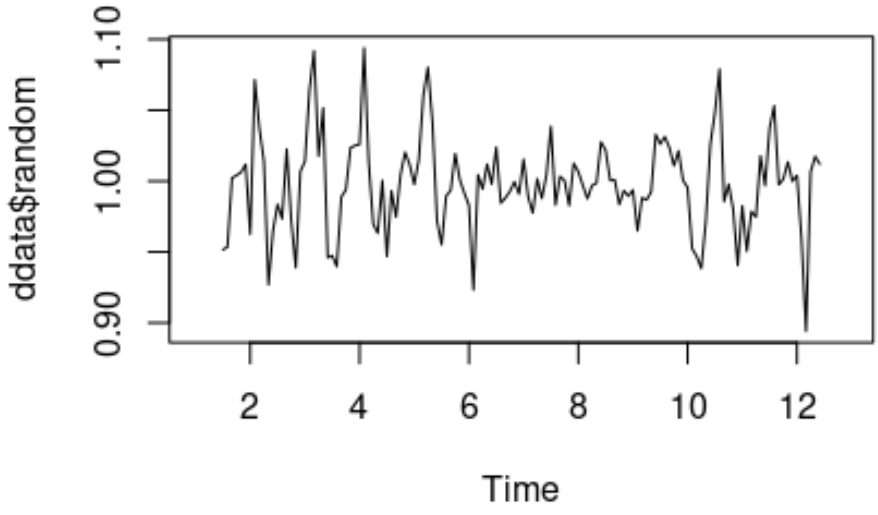

AirPassengers
100
400
1
2
3
4
5
6
7
8
9
11
cycle(AirPassengers)


ddata$trend
150
300
450
2
4
6
8
10
12
Time


AirPassengers
100
400
1950
1952
1954
1956
1958
1960
Time


ddata$trend
150
300
450
2
4
6
8
10
12
Time


ddata$seasonal
0.8
1.0
1.2
2
4
6
8
10
12
Time


ddata$random
0.90
1.00
1.10
2
4
6
8
10
12
Time

Example 3:

Code:

```r
# Shivam Tawari (A-58)
kings <- scan("http://robjhyndman.com/tsdldata/misc/kings.dat",skip=3)
kings
kingstimeseries<- ts(kings)
kingstimeseries
plot.ts(kingstimeseries)

install.packages("TTR")
library("TTR")
kingstimeseriesSMA8<- SMA(kingstimeseries,n=8)
plot.ts(kingstimeseriesSMA8)
```

Output:

```
> kingstimeseries
Time Series:
Start = 1
End = 42
Frequency = 1
 [1] 60 43 67 50 56 42 50 65 68 43 65 34 47 34 49 41 13 35 53 56 16 43 69 59 48 59
[27] 86 55 68 51 33 49 67 77 81 67 71 81 68 70 77 56
> plot.ts(kingstimeseries)
> install.packages("TTR")
Error in install.packages : Updating loaded packages
> install.packages("TTR")
Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
(as 'lib' is unspecified)
trying URL 'http://package-proxy/focal/src/contrib/TTR_0.24.2.tar.gz'
Content type 'application/x-tar' length 533735 bytes (521 KB)
==================================================
downloaded 521 KB

* installing *binary* package 'TTR' ...
* DONE (TTR)

The downloaded source packages are in
        '/tmp/Rtmpv6lLn4/downloaded_packages'
> library("TTR")
> kingstimeseriesSMA8<- SMA(kingstimeseries,n=8)
> plot.ts(kingstimeseriesSMA8)
```