Time Series

Ques1 – Product Demand

We would re-arrange a data little bit to get it from xls file. Attached file contains the rearranged data for both the question.

TimeSeriesAssignment.xlsx

Read the data into R

```
library(xlsx)
mydf <- read.xlsx("TimeSeriesAssignment.xlsx",header=TRUE, sheetIndex = 1)
head(mydf)
```

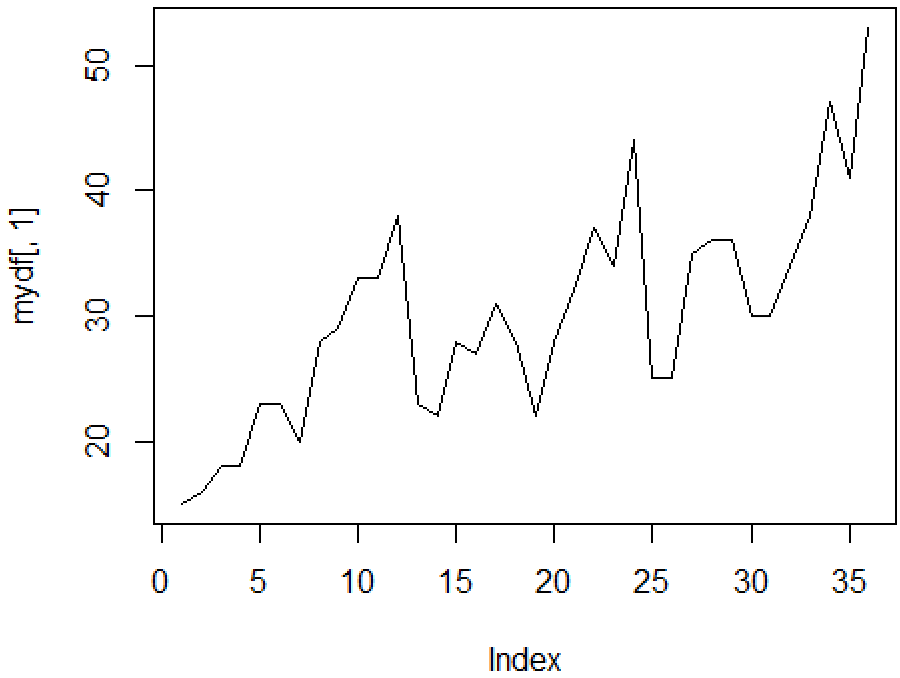
Visualize and understand the data

```
summary(mydf)
```

```
Min.   :15.00
1st Qu.:23.00
Median :29.50
Mean   :30.00
3rd Qu.:35.25
Max.   :53.00
```

We could see mean and median are quite close. With minimum of 15.00 and maximum of 53.
Let us visualize the data

```
plot(mydf[,1], type ="l")
```

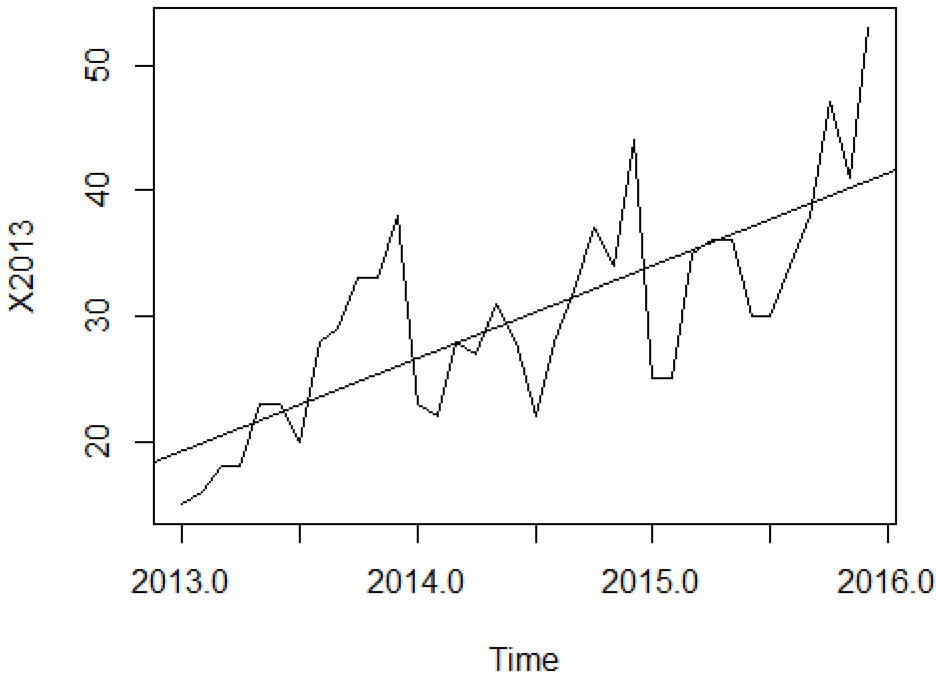


We could observe that there is a mid-year drop and then a spike at year end. Otherwise data has a general increasing pattern

Convert the data into time series object for further analysis
tmydf <- ts(mydf, start=c(2013, 1), end=c(2015, 12), frequency=12)
tmydf

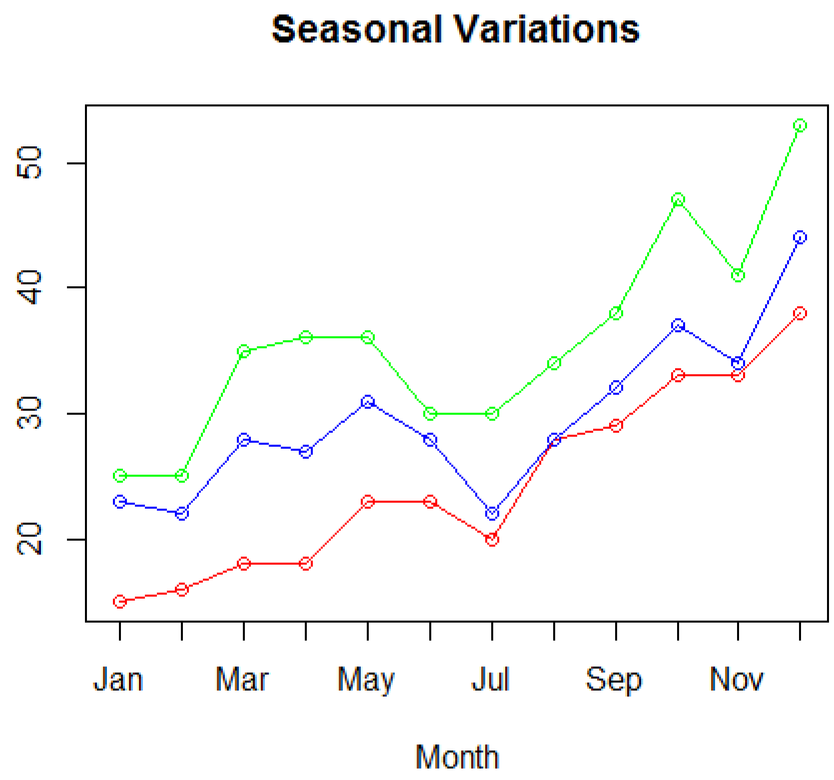
```
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2013  15  16  18  18  23  23  20  28  29  33  33  38
2014  23  22  28  27  31  28  22  28  32  37  34  44
2015  25  25  35  36  36  30  30  34  38  47  41  53
```

plot(tmydf)
m = lm(tmydf~time(tmydf))
abline(m)



We could see the trend, there is increasing mean over time
We could also see the variations across the seasons

#seasonal plots
seasonplot(tmydf, col = c("red","blue","green"), main="Seasonal Variations")

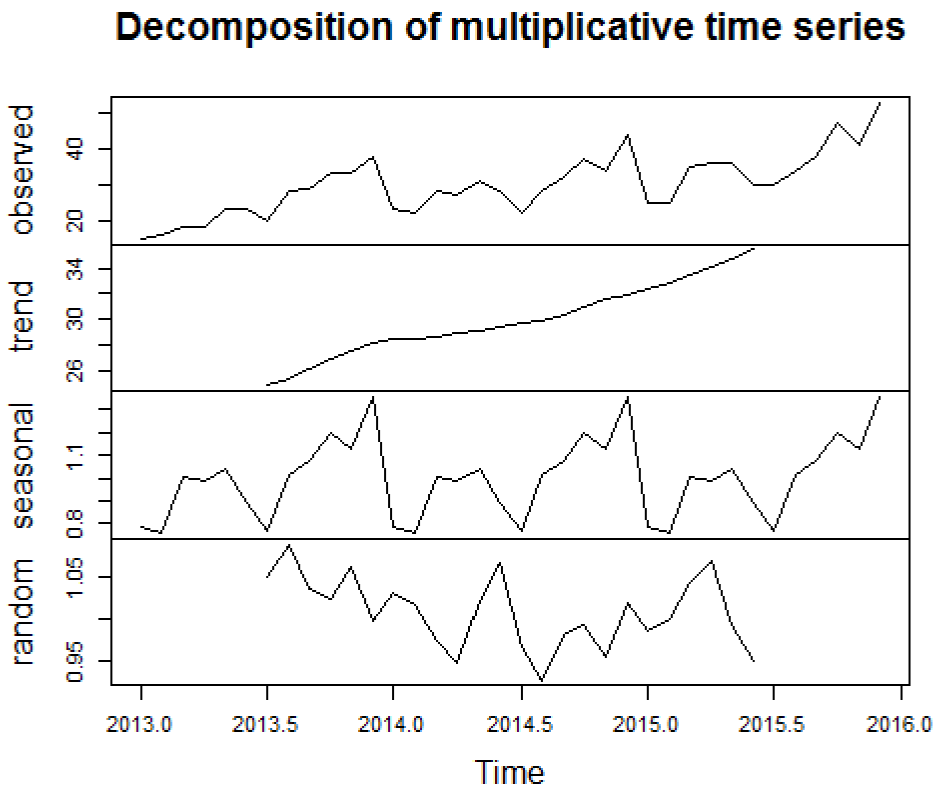


This confirms the dip in July from May, and then it increases again.
On continuation from year end we could see a sharp dip in months of Jan as well which is considerably larger then mid-year dip.

a. Calculate the seasonality index using methods of averages

(Finding the index by decomposing first)
Decomposing time series means separating out the seasonality, trend and randomness from the given series
It can be done by stl as well as decompose commands

```
dmydf = decompose(tmydf, type="multiplicative")
dmydf$seasonal
plot(dmydf)
```



Seasonal Index

```
dmydf$figure
 [1] 0.7849451 0.7595719 1.0046528 0.9869980 1.0411383 0.8917442 0.7662387
1.0099715 1.0739782 1.2002249 1.1266298 1.3539066
```

Let us know reconfirm this by using method of moving averages

mydfma= ma(tmydf,order = 12, centre = T)
mademand=tmydf/mydfma
ma_demand=t(matrix(data=mademand, nrow=12))
seasonal_demand=colMeans(ma_demand,na.rm = T)
seasonal_demand

```
0.7912900 0.7657117 1.0127737 0.9949761 1.0495541 0.8989524 0.7724325 1.01
81354 1.0826595 1.2099266 1.1357367 1.3648506
```

Hence we could see both are roughly same

De-seasonalize the data assuming that Y_t is product of trend and seasonality

For multiplicative time series detrended data = demand/Seasonilty, therefore detrended demand can be calculated as below

ts_calculations <- data.frame(DetrendedData =dmydf$x/dmydf$seasonal, SeasonalIndex = dmydf$seasonal)
ts_calculations

	Detrended data	Seasonality
1	19.10962	0.7849451
2	21.06450	0.7595719
3	17.91664	1.0046528
4	18.23712	0.9869980
5	22.09121	1.0411383
6	25.79215	0.8917442
7	26.10153	0.7662387
8	27.72355	1.0099715
9	27.00241	1.0739782
10	27.49485	1.2002249
11	29.29090	1.1266298
12	28.06693	1.3539066
13	29.30141	0.7849451
14	28.96368	0.7595719
15	27.87032	1.0046528
16	27.35568	0.9869980
17	29.77510	1.0411383
18	31.39914	0.8917442
19	28.71168	0.7662387
20	27.72355	1.0099715
21	29.79576	1.0739782
22	30.82756	1.2002249
23	30.17850	1.1266298
24	32.49855	1.3539066
25	31.84936	0.7849451
26	32.91328	0.7595719
27	34.83790	1.0046528
28	36.47424	0.9869980
29	34.57754	1.0411383
30	33.64194	0.8917442
31	39.15229	0.7662387
32	33.66432	1.0099715
33	35.38247	1.0739782
34	39.15933	1.2002249
35	36.39172	1.1266298
36	39.14598	1.3539066

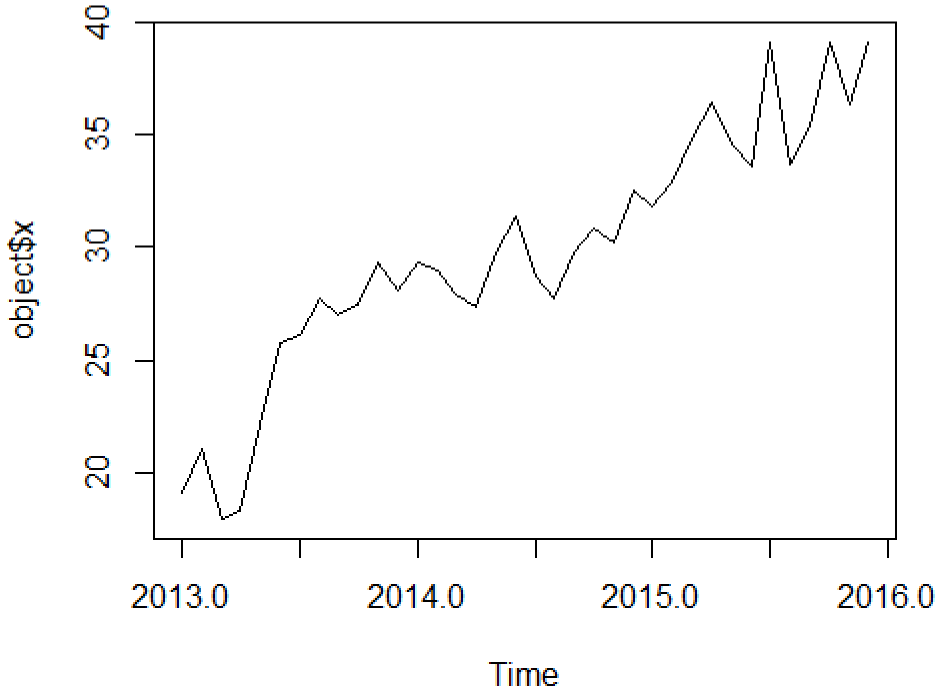
We could also use inbuilt function seasadj for same

This returns seasonally adjusted data constructed by removing the seasonal component.

```
seasadj(dmydf)
          Jan      Feb      Mar      Apr      May      Jun      Jul      A
ug      Sep      Oct
2013 19.10962 21.06450 17.91664 18.23712 22.09121 25.79215 26.10153 27.723
55 27.00241 27.49485
2014 29.30141 28.96368 27.87032 27.35568 29.77510 31.39914 28.71168 27.723
55 29.79576 30.82756
2015 31.84936 32.91328 34.83790 36.47424 34.57754 33.64194 39.15229 33.664
32 35.38247 39.15933
          Nov      Dec
2013 29.29090 28.06693
2014 30.17850 32.49855
2015 36.39172 39.14598
```

Both returns the same value

```
plot(seasadj(dmydf))
```



Develop the best forecasting model by comparing MAPE of MA, ES (exponential smoothing) and ARMA models. Compare the models using MAPE and Theil's coefficient.

#library(tseries)

Before we actually develop the models it is good to get some intuition about the lags.
Auto arima will anyways find this and adjust any autocorrelation

H0 (Null Hypothesis) – Series is non stationary
And Ha (Alternate Hypothesis) – Series is stationary

adf.test(na.omit(dmydf$random))

```
        Augmented Dickey-Fuller Test

data:  na.omit(dmydf$random)
Dickey-Fuller = -2.0431, Lag order = 2, p-value = 0.556
alternative hypothesis: stationary
```

Since p value is greater than .05 so we fail to reject the null hypothesis, indicating series to be non stationary

Let us try again but with lag of 1 to see if series becomes stationary

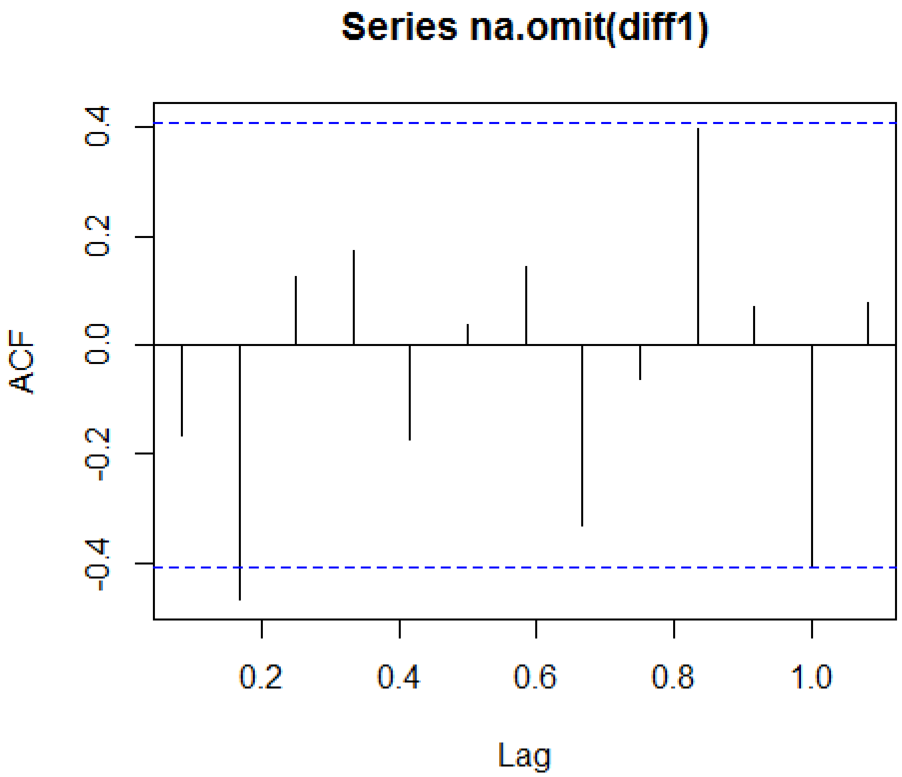
#differencing since above was not stationary

diff1 = diff(dmydf$random, differences = 1)
adf.test(na.omit(diff1))

```
        Augmented Dickey-Fuller Test

data:  na.omit(diff1)
Dickey-Fuller = -3.8435, Lag order = 2, p-value = 0.03261
alternative hypothesis: stationary
```

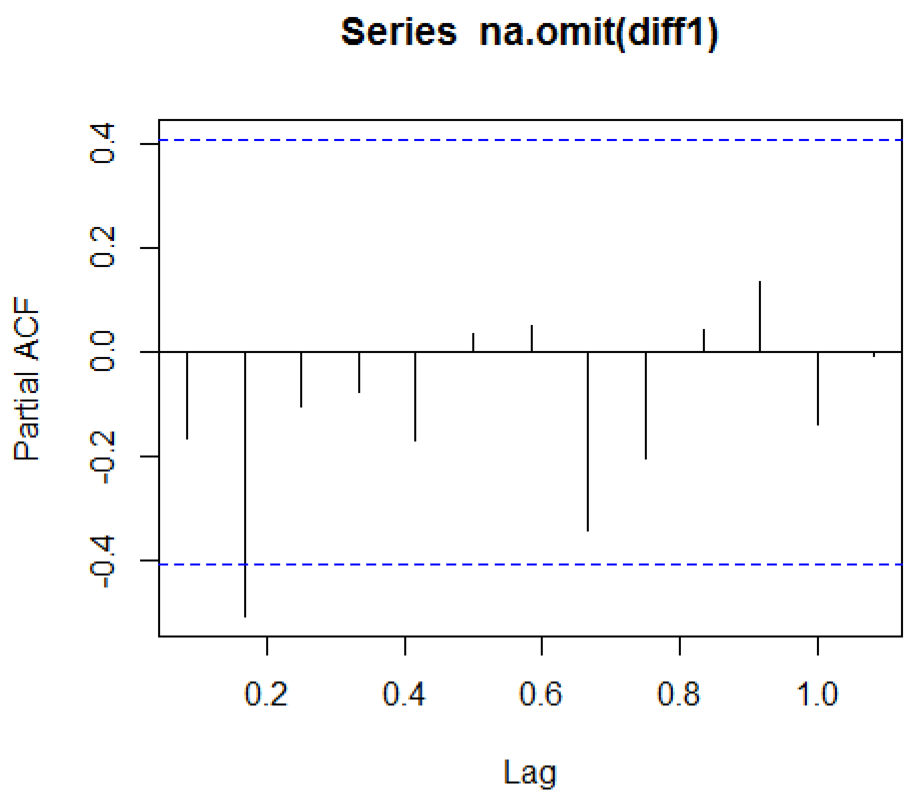
p value is now less than .05 so we can reject the null hypothesis and assume series is stationary. This gives us the intuition that lag of 1 period exists in data.

acf(na.omit(diff1))



Thus now no lag is seen in residue

pacf(na.omit(diff1))



QUESTION 3: Develop the best forecasting model by comparing MAPE of MA, ES (exponential smoothing) and ARMA models. Compare the models using MAPE and Theil's coefficient.

```
library(forecast)
library(AnalyzeTS)

# We can store actual sales data in variable, we can compare with predicted value to find accuracy of the model.
actual = data.frame(mydf)

# moving average model is same as built earlier
mydfma<- ma(tmydf, order=12)
plot(mydfma)
result = abs((mydfma-actual[,1])/actual[,1])*100
colMeans(result,na.rm = TRUE) # MAPE value
resid = data.frame(mydfma)
av.res(actual,resid)
```

We get MAPE value as 15.34151
And theil's coefficient as .785

We can similarly compare the two values for sma and cma

```
          sma       cma    mydfma min.model
ME    2.97000 -0.10763  0.18402       cma
MAE   5.21000  4.12847  4.44444       cma
MPE   5.53498 -3.46390 -2.41424       cma
MAPE 14.71552 14.57483 15.34150       cma
```

```
MSE   47.18972 26.92693 28.81929       cma
RMSE   6.86947  5.18911  5.36836       cma
U      0.94323  0.77105  0.78479       cma
```

Thus out of these 3 we get best model from cma (Cumulative moving average)

Let us know see models such as arima and ets

Arima

auar = auto.arima(tmydf[,1])
resid = data.frame(auar$residuals)
av.res(Y=actual,E=resid)

```
                   ME   MAE    MPE  MAPE   MSE  RMSE         U
auar.residuals -0.025 1.492 -0.532 4.848 4.554 2.134 0.3367331
```

Exponential State Smoothing

Please refer to the following link on ets vs Holt winters
https://robjhyndman.com/hyndsight/estimation2/

s_ets = ets(tmydf)
resid = data.frame(s_ets$residuals)
av.res(Y=actual,E = resid)

```
                   ME  MAE   MPE  MAPE   MSE  RMSE          U
s_ets.residuals 0.023 0.06 0.064 0.214 0.005 0.075 0.01195802
```

We could see that MAPE and theil's coefficient turns out better for ets model among all the models

Below are two good links to know more on time series models.
https://www.datascience.com/blog/introduction-to-forecasting-with-arima-in-r-learn-data-science-tutorials
https://www.r-bloggers.com/time-series-analysis-using-r-forecast-package/

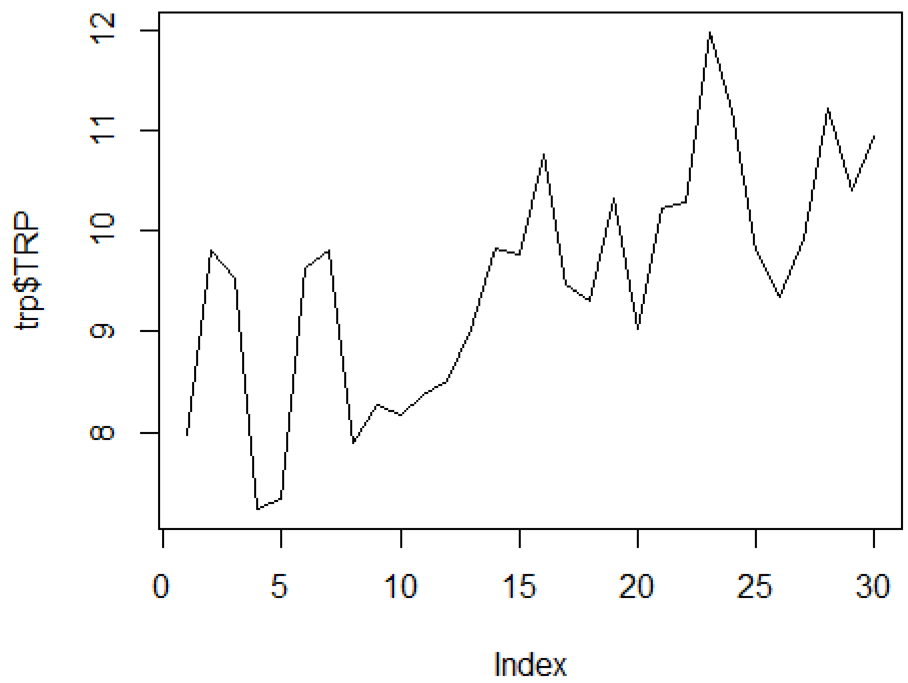
Ques2 – Television program

Before we start with solution let us again have a brief idea of data

```
summary(trp)
    Episode           TRP        
 Min.   : 1.00   Min.   : 7.230  
 1st Qu.: 8.25   1st Qu.: 8.633  
 Median :15.50   Median : 9.695  
 Mean   :15.50   Mean   : 9.512  
 3rd Qu.:22.75   3rd Qu.:10.265  
 Max.   :30.00   Max.   :11.990  
```

Mean and median again are quite close and above 9.5

plot(trp$TRP, type ="l")



Develop a forecasting model using regression $Y_t = \beta_0 + \beta_1 t$, where Y_t is the TRP at time t. Is there any trend in the data? Use the regression model developed to answer?

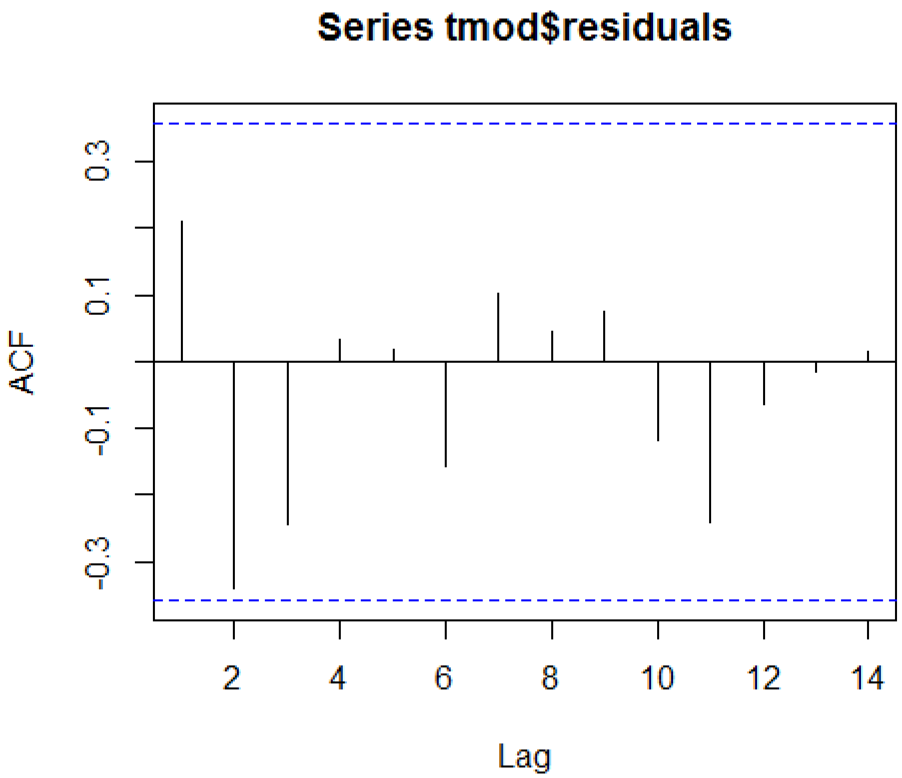
```
tstrp = as.ts(trp)
tmod = tslm(tstrp[,2]~trend)
summary(tmod)
```

```
Call:
tslm(formula = tstrp[, 2] ~ trend)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.2419 -0.6874 -0.2039  0.5565  1.8002 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  8.11018    0.32775  24.745  < 2e-16 ***
trend        0.09042    0.01846   4.898 3.67e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8752 on 28 degrees of freedom
Multiple R-squared:  0.4614,	Adjusted R-squared:  0.4422 
F-statistic: 23.99 on 1 and 28 DF,  p-value: 3.669e-05
```

Equation could be written as $Y_t = \mathbf{8.11 + .0904}\boldsymbol{t}$

Is there an auto-correlation in the data? Conduct an appropriate hypothesis test to justify your answer.

Lag in residual could be seen from acf plot
acf(tmod$residuals)



Since vertical lines does not cross the confidence interval line (blue lines), hence we can conclude that no lag is there.

Let us also test this with statistical methods

Durbin-Watson test
H0 = autocorrelation of the disturbances is 0
Ha= Autocorrelation is not 0

```
dwtest(tmod, alt="two.sided")
        Durbin-Watson test

data:  tmod
DW = 1.5778, p-value = 0.1653
alternative hypothesis: true autocorrelation is not 0
```

Since p value is > 0.05, thus we fail to reject the null hypothesis, which means we can reject the hypothesis that autocorrelation is 0

The television channel would like to replace the program with a new program, the average TRP of new program will be 8 points. Based on the model developed, comment whether they should replace the program with a new program.

We could see that our regression equation is $\underline{\boldsymbol{Y_t = 8.11 + .0904t}}$
Also, there is an increasing trend overall for episode.

Considering these two points it would not be a good idea to replace it with series having TRP of 8
Calculate the probability that the TRP for episode 31 will be more than 10.

Prediction for next 5 episode shows the following data:

Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
31	10.91315	9.686491	12.13981	8.998754	12.82754
32	11.00357	9.769275	12.23786	9.077257	12.92988
33	11.09399	9.851632	12.33634	9.155095	13.03288
34	11.18440	9.933572	12.43524	9.232282	13.13653
35	11.27482	10.015103	12.53454	9.308830	13.24081

Mean value for TRP = 9.51
Sd can be calculated with help of confidence interval

We know that for any values in 95% confidence interval lies in 1.96 times standard deviation
Thus 12.82754 = 10.91315 + 1.96*sd = .9767

Now assuming normal distribution we can find the probability as

```
pnorm(10,10.91315,.9767, lower.tail = FALSE) = 82.5089
```

~83% of chance that probability is greater than 10