STAT - 535 : Forecasting Methods for Management
Assignment 1
(Name : Renuka Chintalapati)

We have the file RestaurantSales.txt that consists of monthly US retail sales for restaurants and other places for period 1992(1) to 2022(10). The values are mentioned in millions of dollars.

The dataframe is read using the following command:

rsales<-read.csv("/Users/renukachintalapati/Downloads/RestaurantSales.txt")

We assign rsales name to the dataframe. These commands are given next:
attach(rsales)
head(rsales)

Time<-as.numeric(Time)
The above command, converts the variable Time to numeric class.

fMonth<-as.factor(Month)
The variable month has been converted to factor in the above line.

#Augmenting fMonth to rsales:
rsales<-data.frame(rsales,fMonth)

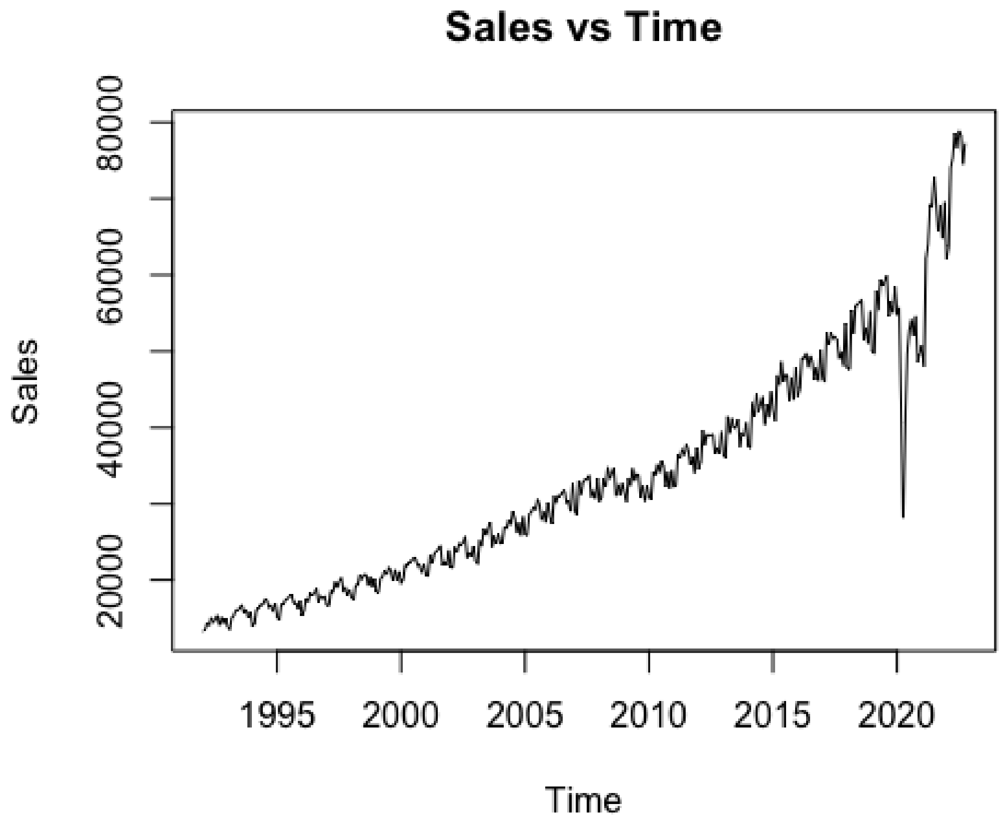
#Checking a sample of rsales dataframe:
head(rsales)

Output:

```
> head(rsales)
        Date Year Month Time Sales logSales        c348       s348       c432       s432 fMonth
1 1992-01-01 1992     1    1 13325 9.497397 -0.57757270  0.8163393 -0.9101060  0.4143756      1
2 1992-02-01 1992     2    2 13474 9.508517 -0.33281954 -0.9429905  0.6565858 -0.7542514      2
3 1992-03-01 1992     3    3 14346 9.571226  0.96202767  0.2729519 -0.2850193  0.9585218      3
4 1992-04-01 1992     4    4 14065 9.551445 -0.77846230  0.6276914 -0.1377903 -0.9904614      4
5 1992-05-01 1992     5    5 15077 9.620926 -0.06279052 -0.9980267  0.5358268  0.8443279      5
6 1992-06-01 1992     6    6 14384 9.573872  0.85099448  0.5251746 -0.8375280 -0.5463943      6
```

Q1. A). We need to make separate time-series plots for
i). Sales -

#Time series plot of Sales:
plot(ts(Sales,start=c(1992,1),freq=12),xlab="Time",ylab="Sales",main="Sales vs Time")

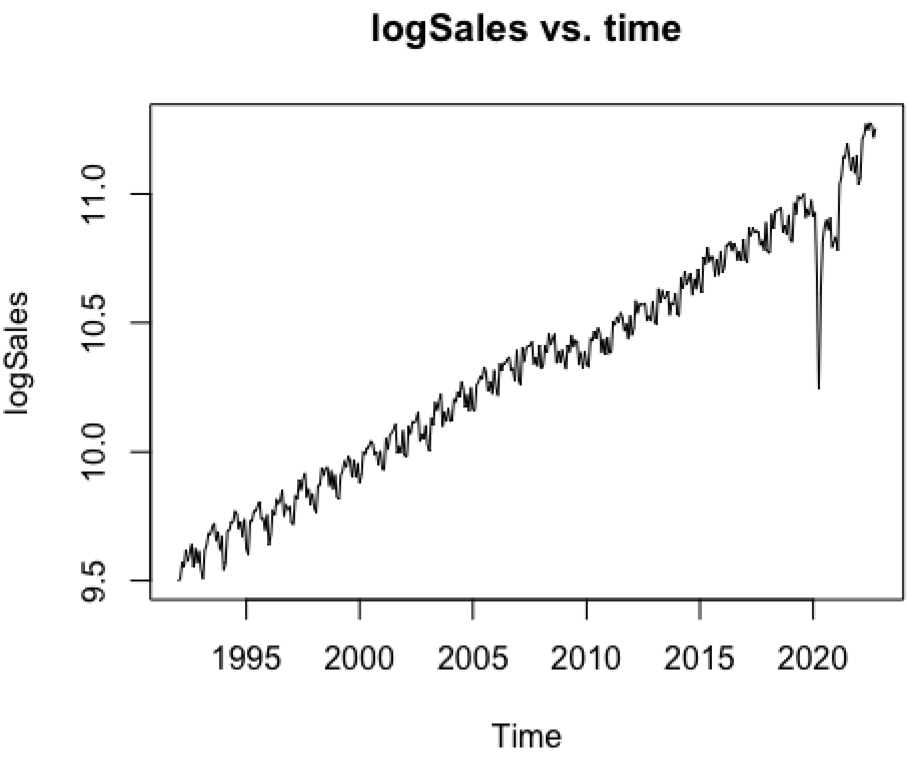


ii). logSales -

 # Time series plot of logSales:

We have the logSales time series plot. The start date is 1992(1) with a frequency of 12.

plot(ts(logSales,start=c(1992,1),freq=12),xlab="Time",ylab="logSales",main="logSales vs. time")



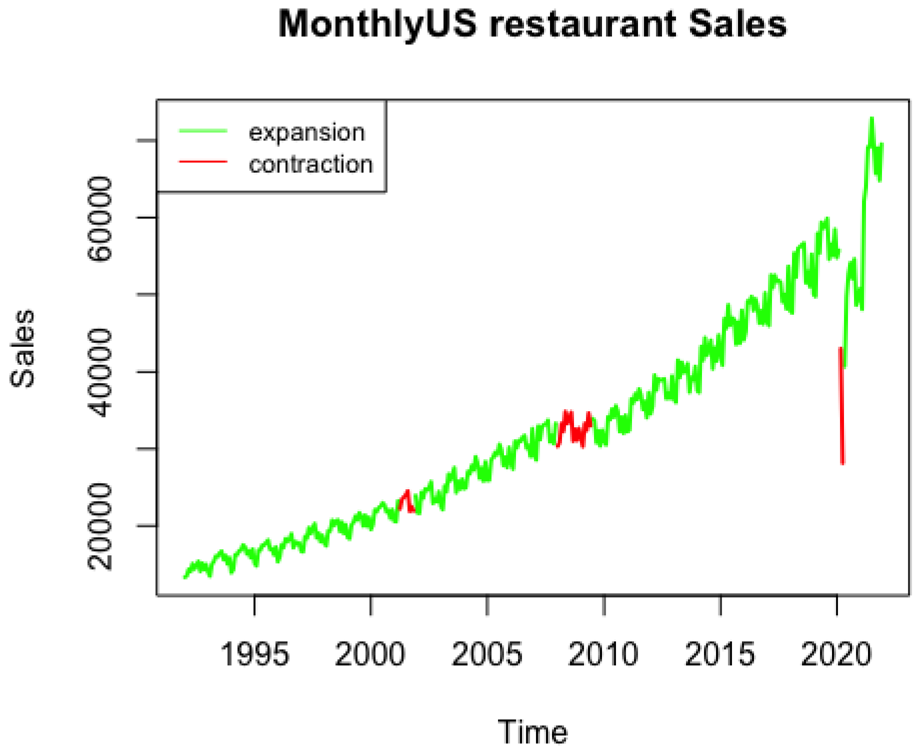
List the economic downturns:

A). Economic downturns can be observed in in the plots. These regions will be marked now. So, the economic downturns have been obtained from the NBER website.

#For sales plot:

```
salesexpansion<-c(Sales[1:111],rep(NA,8),Sales[120:192],rep(NA,18),Sales[211:338],rep(NA,2),Sales[341:360])
salescontraction<-c(rep(NA,111),Sales[112:119],rep(NA,73),Sales[193:210],rep(NA,128),Sales[339:340],rep(NA,20))
plot(ts(salesexpansion,start=c(1992,1),freq=12),ylab="Sales",main="MonthlyUS restaurant Sales",col="green",lwd=2)
lines(ts(salescontraction,start=c(1992,1),freq=12),col="red",lwd=2)>legend("topleft",legend=c("expansion","contraction"),col=c("green","red"),lty=1,cex=0.8)
```

Output:



We can see that there is an increase in the value of trend from 1995 to 2019. The trend can be rising due to better standard of living conditions of people. Due to better technology and increase in the number of jobs, we can say that the household income must have risen. People have more money to spend and thus started spending on restaurants as well.

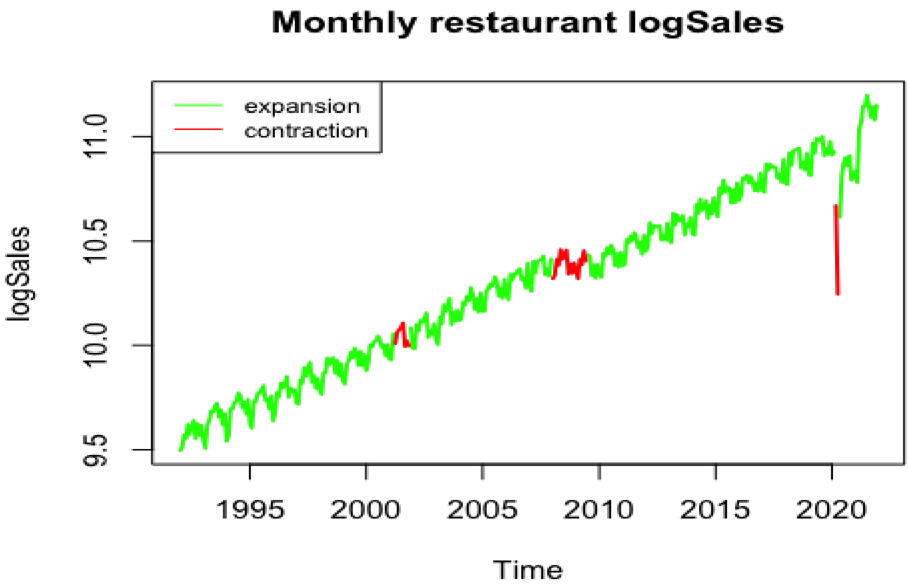
Now let us talk about seasonality of the plot. We observe that there is seasonality in the plot. Initially seasonality decreases as the year begins and later on increases. People might have spent a lot during the winter vacation in the previous year because of which they weren't able to spend a lot of money eating outside during the beginning of the year. We can see that there are a few changes in the seasonal part in 2001 (mostly due to dot com bubble), 2008-2009 recession and in 2020 due to Covid-19 outbreak.

About volatility, we can say that the volatility in the case of Sales plot is increasing as the number of years pass.

#For logSales plot:

```
logsalesexpansion<-c(logSales[1:111],rep(NA,8),logSales[120:192],rep(NA,18),logSales[211:338],rep(NA,2),logSales[341:360])
logsalescontraction<-c(rep(NA,111),logSales[112:119],rep(NA,73),logSales[193:210],rep(NA,128),logSales[339:340],rep(NA,20))
plot(ts(logsalesexpansion,start=c(1992,1),freq=12),ylab="logSales",main="Monthly restaurant logSales",col="green",lwd=2)
lines(ts(logsalescontraction,start=c(1992,1),freq=12),col="red",lwd=2)
```

```
legend("topleft",legend=c("expansion","contraction"),col=c("green","red"),lty=1,cex=0.8)
```



In the case of logSales part, we can see that there is an increasing trend in the curve. Also, the volatility is higher in the early years as compared to the later ones. We can also see that the seasonal component is similar to that of the Sales part. We can mark economic downturns in the years 2001, 2008-2009 and 2020. The reasons could be Dot-com bubble, recession in the years of 2008-2009 and the famous Covid-19 outbreak.

B). There are time-series fluctuations from 2020(1) to 2022(10). These are due to the outbreak of COVID-19. We observe that there is a huge dip in the curve in 2020 when the outbreak occurred. People avoided restaurants as there were lockdowns and they also wished to follow social distancing. Sales gradually started improving due to awareness (like people wearing masks, taking preventive measures and all) and people visiting restaurants again. We can again see a dip again probably in the beginning of 2021. This can be possible due to various strong waves of Covid and people avoiding restaurants again. We can see that there is a rise again in the sales in 2022 because of vaccinations and medicines. People again started visiting restaurants. We thus can say that there were many fluctuations in the period between 2020(1) and 2022(10).

C). As the variance in the plots increases with the rise in level, we observe that multiplicative model would be more useful. We can also see that there is a seasonal pattern in the time series plot and it increases with the increase in time. Here, seasonality is proportional to the time series' level. Hence, a multiplicative model is best suited for such cases.

Q2. Let us now fit a multiplicative decomposition model to the variable Sales. We just need to include polynomial trend and a seasonal component using fMonth. We can use poly(Time,4) for this purpose.

#Fitting a multiplicative decomposition model with polynomial trend and seasonal component:
model1 <- lm(logSales ~ poly(Time, 4) + fMonth, data = rsales[1:336,]);summary(model1)

```
Call:
lm(formula = logSales ~ poly(Time, 4) + fMonth, data = rsales[1:336,
    ])

Residuals:
      Min        1Q    Median        3Q       Max
-0.064103 -0.018447 -0.002466  0.019289  0.073813

Coefficients:
                Estimate Std. Error  t value Pr(>|t|)
(Intercept)    10.179945   0.005011 2031.673  < 2e-16 ***
poly(Time, 4)1  7.252109   0.026519  273.473  < 2e-16 ***
poly(Time, 4)2  0.005593   0.026502    0.211  0.83300
poly(Time, 4)3  0.188088   0.026541    7.087 8.81e-12 ***
poly(Time, 4)4  0.193603   0.026502    7.305 2.22e-12 ***
fMonth2        -0.008088   0.007083   -1.142  0.25436
fMonth3         0.101958   0.007083   14.394  < 2e-16 ***
fMonth4         0.075938   0.007083   10.721  < 2e-16 ***
fMonth5         0.122273   0.007084   17.261  < 2e-16 ***
fMonth6         0.097606   0.007084   13.778  < 2e-16 ***
fMonth7         0.117871   0.007085   16.637  < 2e-16 ***
fMonth8         0.118941   0.007085   16.787  < 2e-16 ***
fMonth9         0.035159   0.007086    4.962 1.14e-06 ***
fMonth10        0.069756   0.007087    9.843  < 2e-16 ***
fMonth11        0.019859   0.007088    2.802  0.00539 **
fMonth12        0.084125   0.007089   11.867  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0265 on 320 degrees of freedom
Multiple R-squared:  0.9958,    Adjusted R-squared:  0.9956
F-statistic:  5072 on 15 and 320 DF,  p-value: < 2.2e-16
```

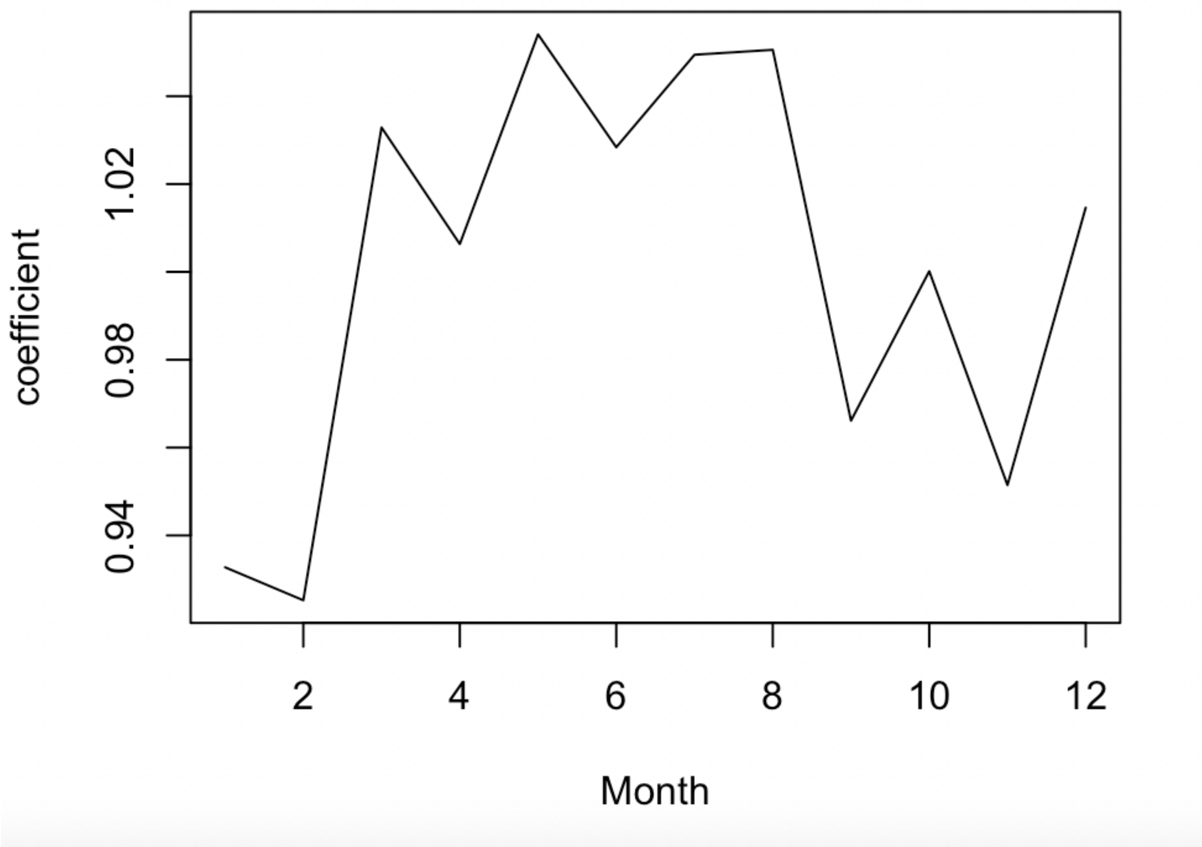
Observation: We notice that the deviation of 2nd month from 1st month is not significant.

A). Let us tabulate and plot the static seasonal indices:

#Static seasonal indices of the above model are as follows:
b10<-coef(model1)[1]
b20<-coef(model1)[6:16]+b10
b30<-c(b10,b20)
seas2<-exp(b30-mean(b30))
seas2

```
(Intercept)     fMonth2     fMonth3     fMonth4     fMonth5     fMonth6     fMonth7     fMonth8
  0.9327514   0.9252380   1.0328697   1.0063416   1.0540672   1.0283843   1.0494380   1.0505610
    fMonth9    fMonth10    fMonth11    fMonth12
  0.9661296   1.0001400   0.9514606   1.0146141
```

#Plotting the seasonal indices:
plot(seas2, type = "l", xlab = "Month", ylab = "coefficient")



Observations: 1. We observe that the months of March, May, and October have peaks.
2. Febuary, April, June, September, and November have minimum points in the plot.
3. The curve is constant from July to August.
4. Sales were highest in the month of May and lowest in the month of Feb.
Conclusion: We can say that the sales were highest in May and lowest in Febuary. The sales could be highest in the month of may as it is summer and people can travel from one place to another. We can also see that they can eat outside when staying outside. Also, kids stay at homes during the summer and parents might be ordering more food from outside to satisfy them. The sales are lowest in the month of Febuary. This is a winter month and people must have travelled to different places in the winter break. After some expenditure

during their vacations, they may wish not to spend money on restaurants in Feb and this is the reason why the sales were lowest.

2 B). Let us now create residuals from the fit above.
They are as follows-
model1 <- lm(logSales ~ poly(Time, 4) + fMonth, data = rsales[1:336,]); summary(model1)

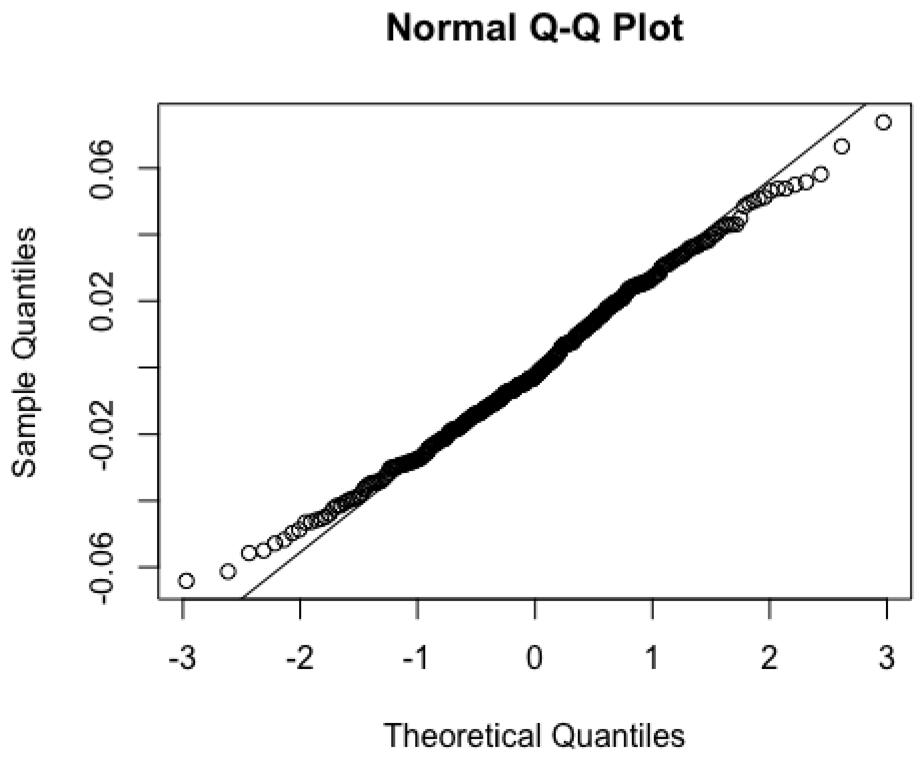
```
Call:
lm(formula = logSales ~ poly(Time, 4) + fMonth, data = rsales[1:336,
    ])

Residuals:
      Min        1Q    Median        3Q       Max
-0.064103 -0.018447 -0.002466  0.019289  0.073813

Coefficients:
                Estimate Std. Error  t value Pr(>|t|)
(Intercept)    10.179945   0.005011 2031.673  < 2e-16 ***
poly(Time, 4)1  7.252109   0.026519  273.473  < 2e-16 ***
poly(Time, 4)2  0.005593   0.026502    0.211  0.83300
poly(Time, 4)3  0.188088   0.026541    7.087 8.81e-12 ***
poly(Time, 4)4  0.193603   0.026502    7.305 2.22e-12 ***
fMonth2        -0.008088   0.007083   -1.142  0.25436
fMonth3         0.101958   0.007083   14.394  < 2e-16 ***
fMonth4         0.075938   0.007083   10.721  < 2e-16 ***
fMonth5         0.122273   0.007084   17.261  < 2e-16 ***
fMonth6         0.097606   0.007084   13.778  < 2e-16 ***
fMonth7         0.117871   0.007085   16.637  < 2e-16 ***
fMonth8         0.118941   0.007085   16.787  < 2e-16 ***
fMonth9         0.035159   0.007086    4.962 1.14e-06 ***
fMonth10        0.069756   0.007087    9.843  < 2e-16 ***
fMonth11        0.019859   0.007088    2.802  0.00539 **
fMonth12        0.084125   0.007089   11.867  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0265 on 320 degrees of freedom
Multiple R-squared:  0.9958,    Adjusted R-squared:  0.9956
F-statistic:  5072 on 15 and 320 DF,  p-value: < 2.2e-16
```

Observations: 1. We clearly observe that poly(Time, 4)2 and fMonth2 are not significant according to summary given.

Let us now form a normal quantile plot of these residuals:

#Normal quantile plot of residuals:

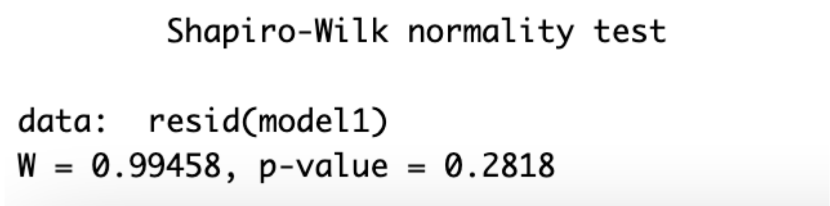
qqnorm(resid(model1))
qqline(resid(model1))



Observations: We observe that the line normal Q-Q plot hasn't fitted all the points well. There are several outliers present at the 2 ends of the line.

#Shapiro-Wilk test for normality:
shapiro.test(resid(model1))

Output:

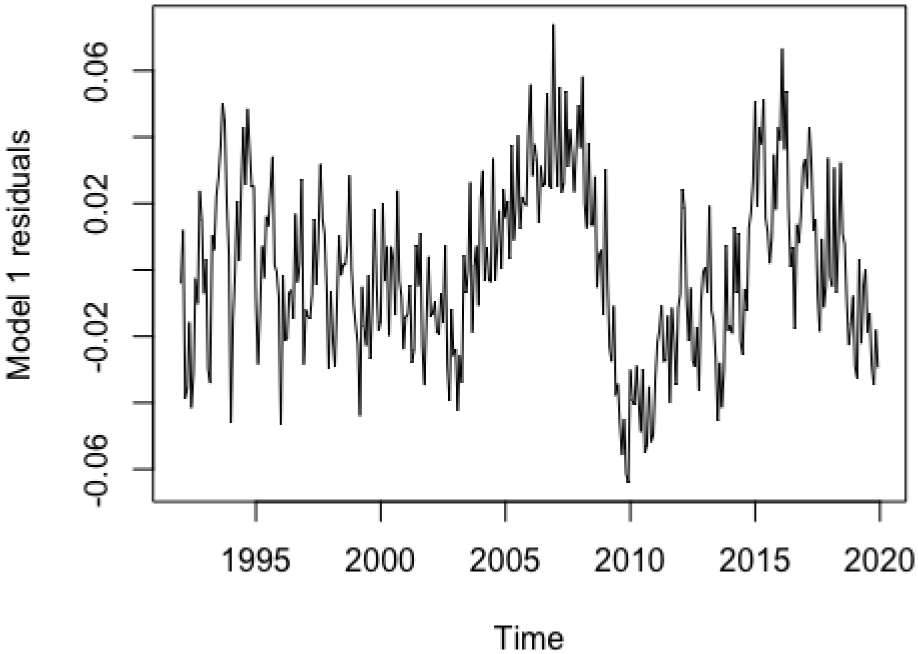


```
        Shapiro-Wilk normality test

data:  resid(model1)
W = 0.99458, p-value = 0.2818
```

Observation: We notice that p-value > 0.05 and thus this implies normal distribution.

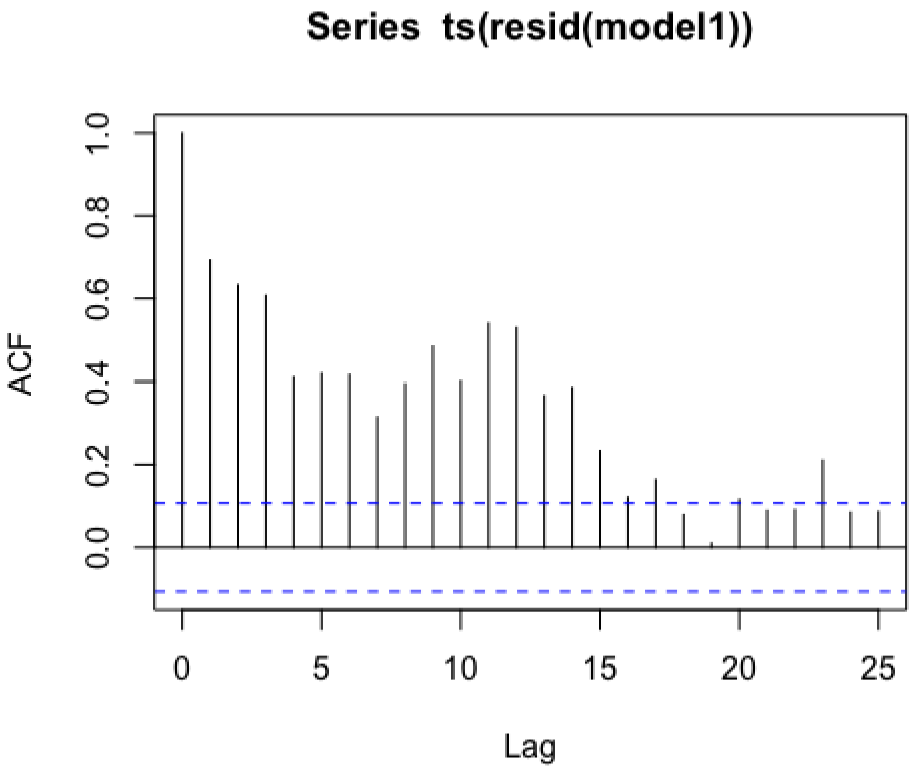
#Plot of the residuals versus time
plot(ts(resid(model1),start=c(1992,1),freq=12),ylab="Model 1 residuals")

Output:



Observations: We observe that the above model 1 residual plot failed to capture the trend. We can see that this has happened beginning from the year 2003 ending in about 2017. This may be due to the dot cum bubble burst and the 2008- 2009 recession. These events had a profound effect on the trend for many years ahead. Thus, the model failed to capture the trend part of sales.

#Plotting ACF:



Observation: We notice that the spikes are above the first blue-dashed line in the plot above. The autocorrelations are significantly not similar to zero. It is also to be noted that for white noise, the autocorrelation spikes should be close to zero. As the spikes are outside the bounds, we can say that the series is not white noise.

The series has failed to capture the seasonality component.

Q3. Let us now add the calendar trigonometric pairs to the model and refit.

#Adding all the calendar trigonometric pairs to the model:
model2 <- lm(logSales ~ poly(Time, 4) + c348 + s348 + c432 + s432 + fMonth, data = rsales[1:336,]); summary(model2)

We have added all the trigonometric calendar values here. The residual standard error below is equal to 0.02559 on 316 degrees of freedom. Multiple R-squared being equal to 0.9961 and the value of adjusted r-squared is 0.9959. The summary also gives the values of F-statistic and p-value.

```
Call:
lm(formula = logSales ~ poly(Time, 4) + c348 + s348 + c432 +
    s432 + fMonth, data = rsales[1:336, ])

Residuals:
      Min        1Q    Median        3Q       Max
-0.059589 -0.018375 -0.001342  0.018177  0.068285

Coefficients:
                Estimate Std. Error  t value Pr(>|t|)
(Intercept)    10.179734   0.004839 2103.630  < 2e-16 ***
poly(Time, 4)1  7.252430   0.025610  283.188  < 2e-16 ***
poly(Time, 4)2  0.005092   0.025593    0.199 0.842426
poly(Time, 4)3  0.188580   0.025633    7.357 1.64e-12 ***
poly(Time, 4)4  0.192948   0.025593    7.539 5.05e-13 ***
c348           -0.007529   0.001975   -3.813 0.000165 ***
s348           -0.005771   0.001975   -2.922 0.003733 **
c432           -0.002271   0.001980   -1.147 0.252283
s432            0.003373   0.001971    1.712 0.087947 .
fMonth2        -0.007609   0.006841   -1.112 0.266859
fMonth3         0.102001   0.006840   14.912  < 2e-16 ***
fMonth4         0.076159   0.006841   11.133  < 2e-16 ***
fMonth5         0.122592   0.006841   17.919  < 2e-16 ***
fMonth6         0.097711   0.006842   14.282  < 2e-16 ***
fMonth7         0.118108   0.006843   17.261  < 2e-16 ***
fMonth8         0.119187   0.006843   17.418  < 2e-16 ***
fMonth9         0.035371   0.006844    5.168 4.20e-07 ***
fMonth10        0.069850   0.006844   10.206  < 2e-16 ***
fMonth11        0.020281   0.006846    2.962 0.003283 **
fMonth12        0.084142   0.006847   12.290  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02559 on 316 degrees of freedom
Multiple R-squared:  0.9961,    Adjusted R-squared:  0.9959
F-statistic:  4295 on 19 and 316 DF,  p-value: < 2.2e-16
```

Observation: We notice that the cosine pair - c432 and s432 are not significant according to the summary table. We can now test for their significance.

#Refitting the model as follows:

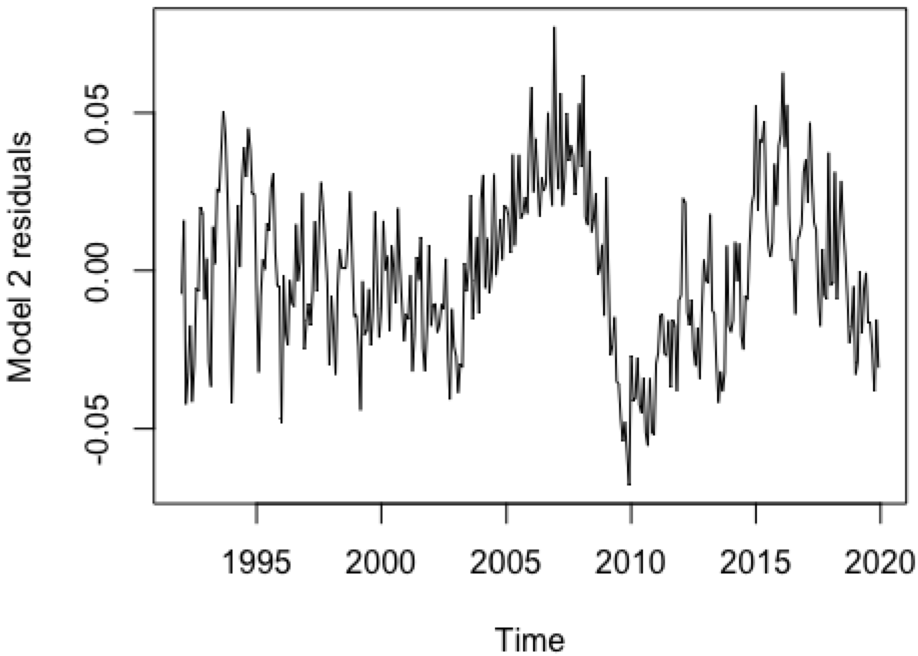
model2 <- lm(logSales ~ poly(Time, 4) + c432 + s432 + fMonth, data = rsales[1:336,]); summary(model2)

```
Call:
lm(formula = logSales ~ poly(Time, 4) + c432 + s432 + fMonth,
    data = rsales[1:336, ])

Residuals:
      Min        1Q    Median        3Q       Max
-0.067954 -0.017485 -0.001455  0.019242  0.077301

Coefficients:
                Estimate Std. Error  t value Pr(>|t|)
(Intercept)    10.179838   0.004997 2037.381  < 2e-16 ***
poly(Time, 4)1  7.252220   0.026443  274.262  < 2e-16 ***
poly(Time, 4)2  0.005376   0.026426    0.203  0.83893
poly(Time, 4)3  0.188257   0.026466    7.113 7.54e-12 ***
poly(Time, 4)4  0.193310   0.026426    7.315 2.11e-12 ***
c432           -0.002262   0.002045   -1.106  0.26941
s432            0.003293   0.002035    1.618  0.10657
fMonth2        -0.007860   0.007064   -1.113  0.26668
fMonth3         0.101956   0.007063   14.435  < 2e-16 ***
fMonth4         0.076126   0.007064   10.777  < 2e-16 ***
fMonth5         0.122344   0.007064   17.320  < 2e-16 ***
fMonth6         0.097698   0.007064   13.830  < 2e-16 ***
fMonth7         0.118041   0.007065   16.707  < 2e-16 ***
fMonth8         0.118948   0.007065   16.836  < 2e-16 ***
fMonth9         0.035384   0.007067    5.007 9.17e-07 ***
fMonth10        0.069747   0.007067    9.870  < 2e-16 ***
fMonth11        0.020058   0.007068    2.838  0.00484 **
fMonth12        0.084179   0.007069   11.908  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02643 on 318 degrees of freedom
Multiple R-squared:  0.9959,    Adjusted R-squared:  0.9956
F-statistic:  4501 on 17 and 318 DF,  p-value: < 2.2e-16
```

#Plot of residual - model 2:

plot(ts(resid(model2),start=c(1992,1),freq=12),ylab="Model 2 residuals")

Output:



Observation: We notice that the model couldn't capture the entire trend of the sales. The explanation can Be similar to the previous explanation in second question. We can see that there were incidents of Dot com bubble and 2008-2009 recession because of which, the sales were greatly effected. The periods of 2003 to 2017 have a different trend.

#Testing the significance of the cosine pair:
shapiro.test(resid(model2))

Output:

```
        Shapiro-Wilk normality test

data:  resid(model2)
W = 0.99534, p-value = 0.4119
```

Observation: We notice that p-value > 0.05 and as mentioned in the question, the cosine pair : c432 and s432 will be discarded and refit.

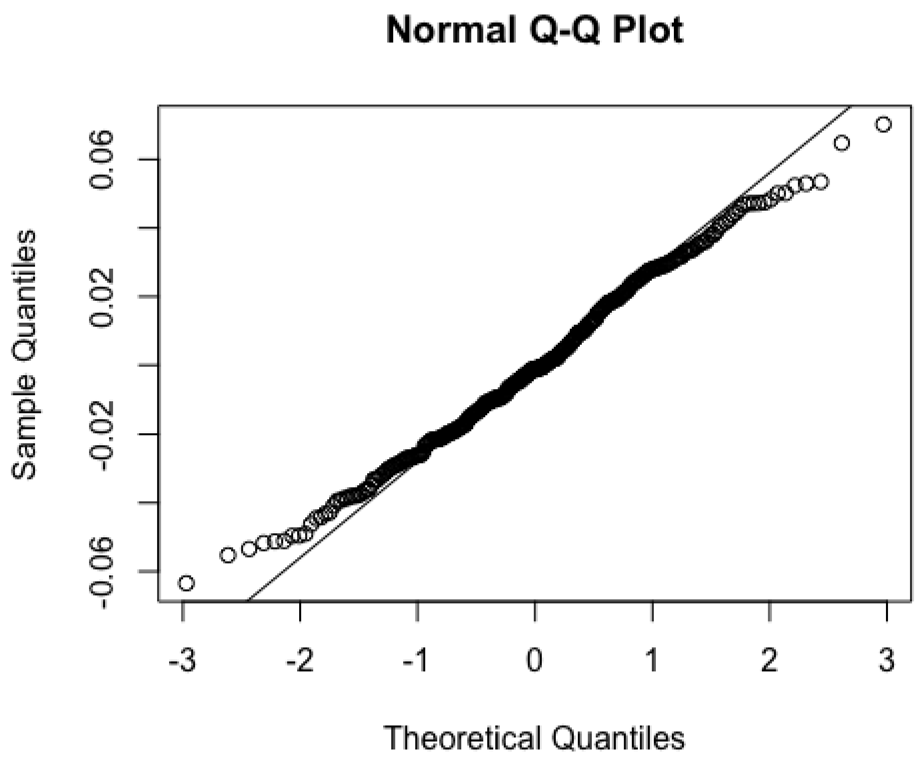
#Refitting the model with c348 and s348 values:

model3 <- lm(logSales ~ poly(Time, 4) + c348 + s348 + fMonth, data = rsales[1:336,]); summary(model3)

```
Call:
lm(formula = logSales ~ poly(Time, 4) + c348 + s348 + fMonth,
    data = rsales[1:336, ])

Residuals:
     Min       1Q   Median       3Q      Max 
-0.06344 -0.01884 -0.00110  0.01897  0.07021 

Coefficients:
                Estimate Std. Error  t value Pr(>|t|)    
(Intercept)    10.179843   0.004856 2096.380  < 2e-16 ***
poly(Time, 4)1  7.252321   0.025700  282.190  < 2e-16 ***
poly(Time, 4)2  0.005314   0.025683    0.207 0.836212    
poly(Time, 4)3  0.188413   0.025723    7.325 1.98e-12 ***
poly(Time, 4)4  0.193248   0.025683    7.524 5.49e-13 ***
c348           -0.007513   0.001982   -3.791 0.000179 ***
s348           -0.005740   0.001982   -2.896 0.004041 ** 
fMonth2        -0.007842   0.006864   -1.142 0.254137    
fMonth3         0.102003   0.006864   14.860  < 2e-16 ***
fMonth4         0.075968   0.006865   11.067  < 2e-16 ***
fMonth5         0.122520   0.006865   17.847  < 2e-16 ***
fMonth6         0.097616   0.006866   14.218  < 2e-16 ***
fMonth7         0.117936   0.006866   17.177  < 2e-16 ***
fMonth8         0.119178   0.006867   17.356  < 2e-16 ***
fMonth9         0.035142   0.006867    5.117 5.38e-07 ***
fMonth10        0.069860   0.006868   10.171  < 2e-16 ***
fMonth11        0.020077   0.006869    2.923 0.003719 ** 
fMonth12        0.084089   0.006870   12.239  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02568 on 318 degrees of freedom
Multiple R-squared:  0.9961,	Adjusted R-squared:  0.9959 
F-statistic:  4766 on 17 and 318 DF,  p-value: < 2.2e-16
```

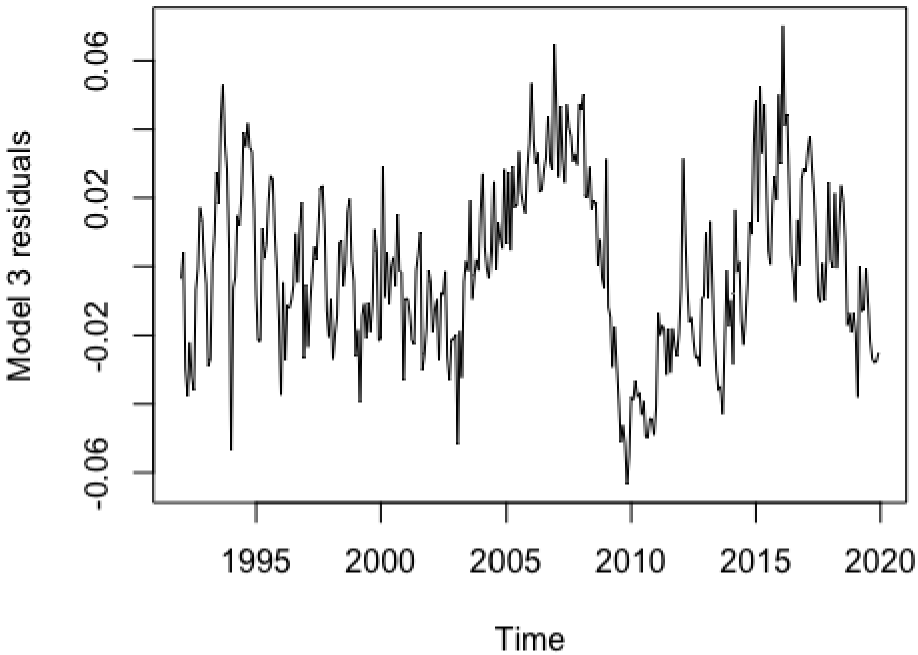
#Residual analysis of the above model:

```
#Normal QQ-plot:
qqnorm(resid(model3))
qqline(resid(model3))
```



Observation: We observe that there are outliers in the above plot. These outliers are clearly visible. The model isn’t that great.

```
# Plot of residual - model3:


plot(ts(resid(model3),start=c(1992,1),freq=12),ylab ="Model 3 residuals")
```



Observation: We notice that our model couldn’t fit the trend part perfectly. There are changes in the trend from the years 2003 to 2017. We can use the similar reasoning used in the above part of the question.

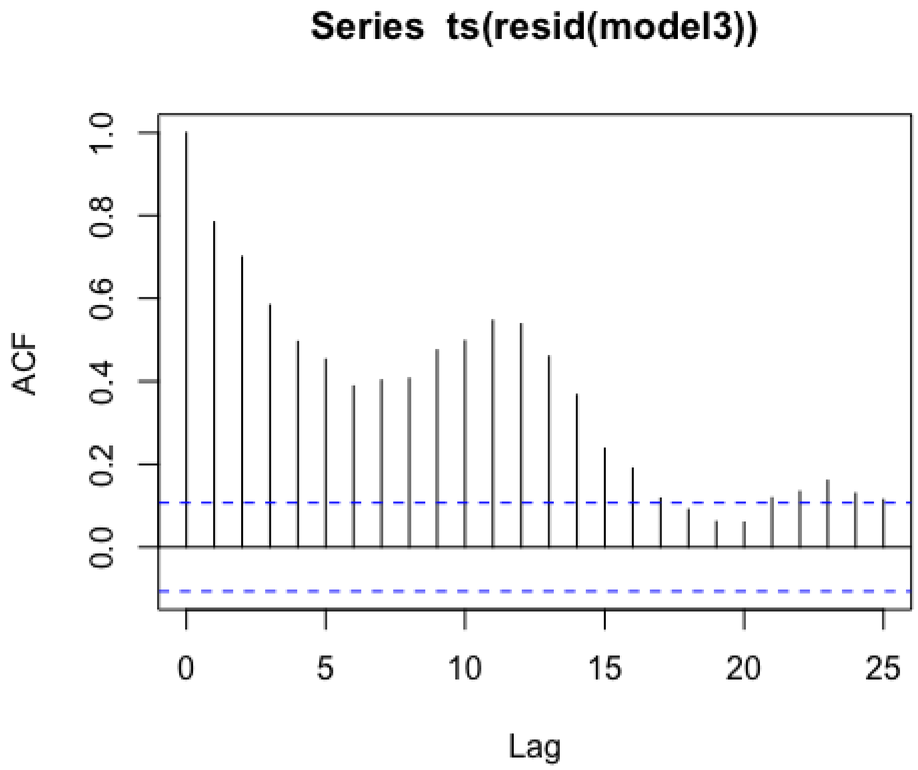
#Testing the significance of the cosine pair:
shapiro.test(resid(model3))

```
        Shapiro-Wilk normality test

data:  resid(model3)
W = 0.9936, p-value = 0.166
```

Observation: P-value has reduced here. The p-value is equal to 0.166 and we can see that there is no significance for the case of non- normality by this value.

#ACF plot:
acf(ts(resid(model3)))

Output:



Observation: We observe that the plot of autocorrelation is not close to zero. Hence, the plot is not close to white noise. The plot has not captured seasonal component. We are not able to reduce the spikes to white noise here.

Q4.

Creating a lag residual variable for model3:

lag1resid<-c(NA,resid(model3)[1:335],rep(0,34))

Add the lag 1 residuals to the original data frame

rsales <- cbind(rsales, lag1resid)

Refit the model with the added lag 1 residual value:

model4 <- lm(logSales ~ poly(Time, 4) + fMonth + c348 + s348 + c432 + s432 + lag1resid, data = rsales[1:336,]); summary(model4)

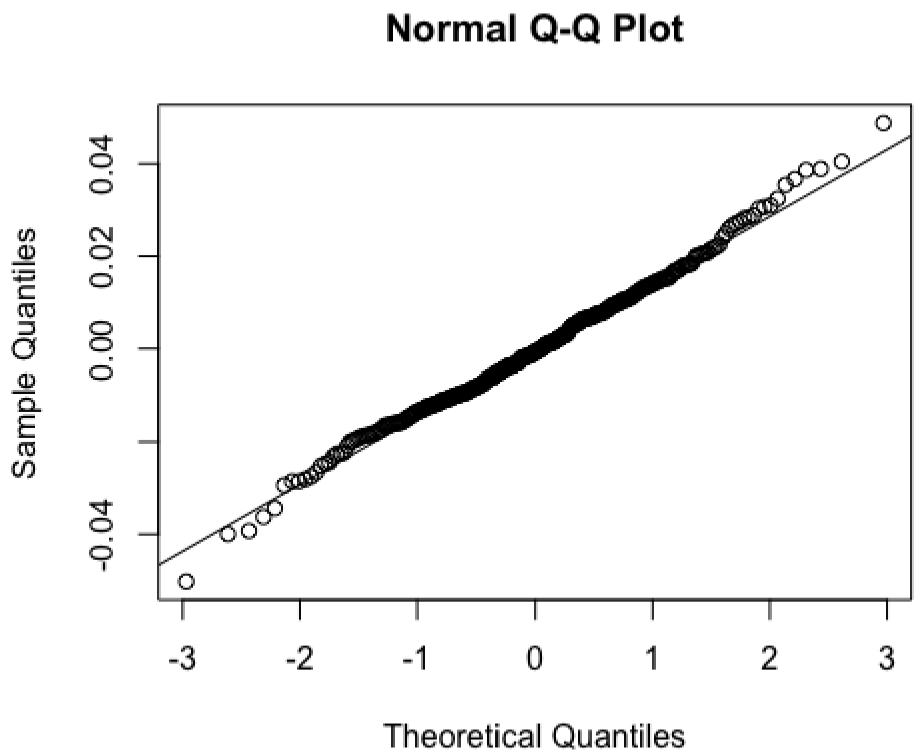
Output:

```
Call:
lm(formula = logSales ~ poly(Time, 4) + fMonth + c348 + s348 +
    c432 + s432 + lag1resid, data = rsales[1:336, ])

Residuals:
      Min        1Q    Median        3Q       Max
-0.050237 -0.010054 -0.000622  0.009457  0.048804

Coefficients:
                 Estimate Std. Error  t value Pr(>|t|)
(Intercept)    10.179212   0.002910 3497.584  < 2e-16 ***
poly(Time, 4)1  7.249647   0.015189  477.309  < 2e-16 ***
poly(Time, 4)2  0.003397   0.015232    0.223   0.8237
poly(Time, 4)3  0.184301   0.015290   12.054  < 2e-16 ***
poly(Time, 4)4  0.190580   0.015321   12.439  < 2e-16 ***
fMonth2        -0.007019   0.004078   -1.721   0.0862 .
fMonth3         0.102404   0.004078   25.112  < 2e-16 ***
fMonth4         0.076760   0.004077   18.826  < 2e-16 ***
fMonth5         0.123036   0.004077   30.176  < 2e-16 ***
fMonth6         0.098263   0.004077   24.100  < 2e-16 ***
fMonth7         0.118639   0.004077   29.101  < 2e-16 ***
fMonth8         0.119667   0.004077   29.349  < 2e-16 ***
fMonth9         0.035986   0.004077    8.826  < 2e-16 ***
fMonth10        0.070290   0.004077   17.240  < 2e-16 ***
fMonth11        0.020927   0.004078    5.132 5.04e-07 ***
fMonth12        0.084613   0.004078   20.748  < 2e-16 ***
c348           -0.007554   0.001167   -6.474 3.67e-10 ***
s348           -0.005875   0.001169   -5.028 8.36e-07 ***
c432           -0.002736   0.001172   -2.334   0.0202 *
s432            0.006706   0.001172    5.722 2.46e-08 ***
lag1resid       0.809900   0.033273   24.341  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01511 on 314 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.9986,    Adjusted R-squared:  0.9986
F-statistic: 1.161e+04 on 20 and 314 DF,  p-value: < 2.2e-16
```

#Performing residual analysis of the above model:

#Normal quantile plot of the residual model:

qqnorm(resid(model4))
qqline(resid(model4))



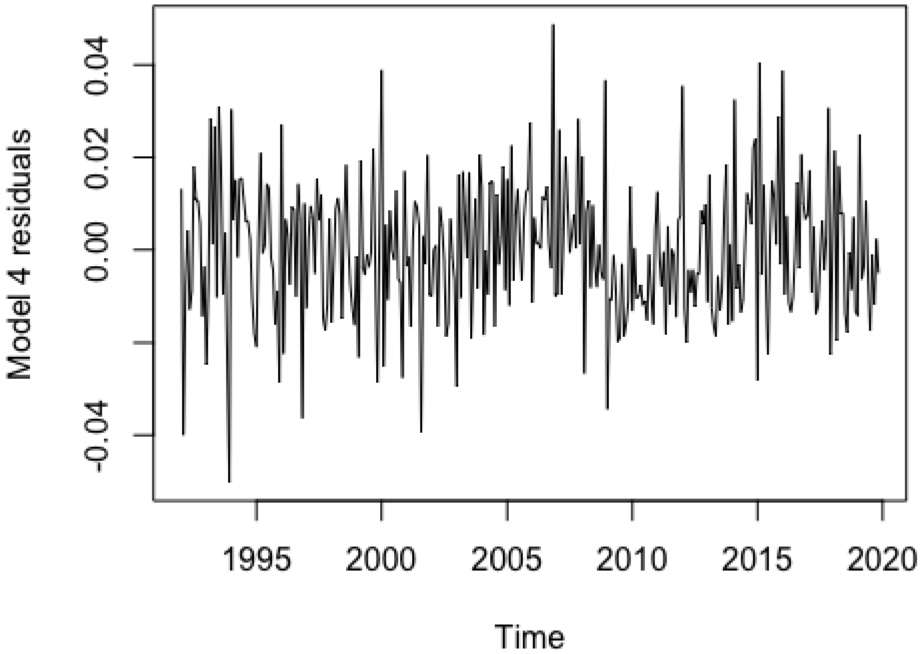
Observation: There are very few outliers in the above case. This means that the model has become better and is heading towards normal distribution case.
#Testing the significance of the cosine pair:
shapiro.test(resid(model4))

```
        Shapiro-Wilk normality test

data:  resid(model4)
W = 0.99404, p-value = 0.2136
```

Observation: We observe that p-value is again greater than 0.2136. This ensures that we have a normal distribution.

#Plot of residuals:
plot(ts(resid(model4),start=c(1992,1),freq=12),ylab ="Model 4 residuals")

Output:

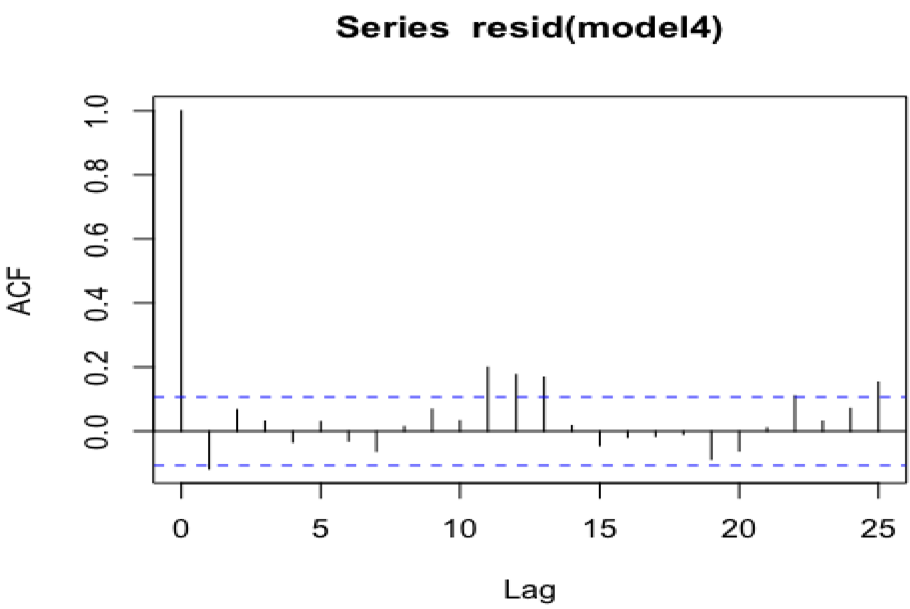


Observation: We can infer that the model has captured the trend part well here. We can also see some volatility throughout the years. The trend component has considerably improved here in this case.

#ACF plot of the residuals of model 4:

acf(resid(model4))

Observation: We notice that the spikes in the autocorrelation plot are now closer to zero. There are only a few spikes that are lying outside the bounds of the



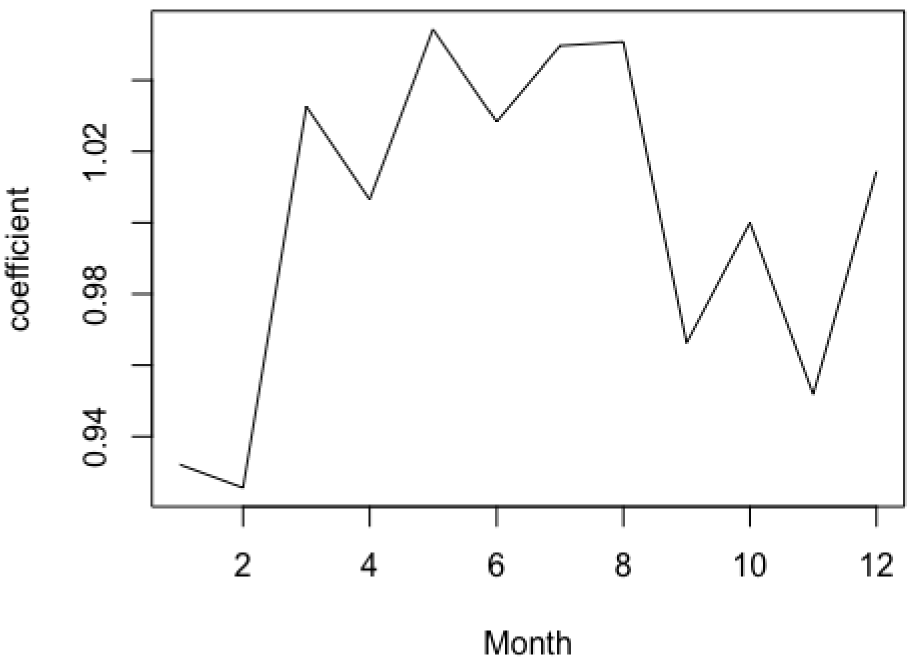
plot. Thus, we can infer that the distribution is closer to white noise.
Also, in the above plot there are far fewer lags and the residual autocorrelations are also quite significant here.

b). To calculate the static seasonal estimates of the above model:

b11<-coef(model4)[1]
b21<-coef(model4)[6:16]+b11
b31<-c(b11,b21)
seas4<-exp(b31-mean(b31))
Seas4

```
(Intercept)    fMonth2    fMonth3    fMonth4    fMonth5    fMonth6    fMonth7    fMonth8
  0.9321168  0.9255967  1.0326278  1.0064836  1.0541547  1.0283605  1.0495293  1.0506091
    fMonth9   fMonth10   fMonth11   fMonth12
  0.9662711  0.9999926  0.9518287  1.0144190
```

#Plotting static seasonal estimates:
plot(seas4, type = "l", xlab = "Month", ylab = "coefficient")



Observation: We can notice that there are peaks in the months of March, May, October and December.
There is a minimum in the months of Febuary, April, June, September, and November.
The curve is constant from the months of July to August. We can provide the same reasoning as we did in the previous question. We might conclude that the sales were at their best in May and their lowest in February. Given that it is summer and people may travel, it is possible that May will see the highest sales. Additionally, we can observe that while they are outside, they can eat outside. In addition, since children are at home over the summer, parents may order more takeout to feed them. The month of February had the lowest sales. Since it is winter, it stands to reason that during the break people would have traveled. The sales were at their lowest in February because people might not want to spend money at restaurants after making some purchases during their vacations. Also, people may not wish to move outside during cold winters and thus the sales must have plummeted.

#To tabulate the static seasonal indices:

cbind(seas2,seas4)

Output:

```
                seas2     seas4
(Intercept) 0.9327514 0.9321168
fMonth2     0.9252380 0.9255967
fMonth3     1.0328697 1.0326278
fMonth4     1.0063416 1.0064836
fMonth5     1.0540672 1.0541547
fMonth6     1.0283843 1.0283605
fMonth7     1.0494380 1.0495293
fMonth8     1.0505610 1.0506091
fMonth9     0.9661296 0.9662711
fMonth10    1.0001400 0.9999926
fMonth11    0.9514606 0.9518287
fMonth12    1.0146141 1.0144190
```

Observation: We have tabulated the values of static seasonal indices of question 2 and question 4 above. The intercept is the base and the seasonal indices have been applied to that base.

Q5. Including cosine and sine dummies instead of fMonth :

```
cosm<-matrix(nrow=length(Time),ncol=6)
sinm<-matrix(nrow=length(Time),ncol=5)
for(i in 1:5){
  cosm[,i]<-cos(2*pi*i*Time/12)
  sinm[,i]<-sin(2*pi*i*Time/12)
}
cosm[,6]<-cos(pi*Time)
c1<-cosm[,1];c2<-cosm[,2];c3<-cosm[,3];c4<-cosm[,4];c5<-cosm[,5];c6<-cosm[,6]
s1<-sinm[,1];s2<-sinm[,2];s3<-sinm[,3];s4<-sinm[,4];s5<-sinm[,5]
rsales<-data.frame(rsales,c1,s1,c2,s2,c3,s3,c4,s4,c5,s5,c6)
```

#Refitting the model again:

```
model5 <- lm(logSales ~ poly(Time, 4) + c1 +s1 +c2+ s2+ c3+ s3+ c4+ s4+ c5+ s5+ c6+
c348 + s348 +c432 + s432+ lag1resid, data = rsales[1:336,]); summary(model5)
```

```
Call:
lm(formula = log(Sales) ~ poly(Time, 4) + c1 + s1 + c2 + s2 +
    c3 + s3 + c4 + s4 + c5 + s5 + c6 + c348 + s348 + c432 + s432 +
    lag1resid, data = rsales[1:336, ])

Residuals:
      Min        1Q    Median        3Q       Max
-0.050237 -0.010054 -0.000622  0.009457  0.048804

Coefficients:
                 Estimate Std. Error   t value Pr(>|t|)
(Intercept)    10.2495090  0.0008256 12414.487  < 2e-16 ***
poly(Time, 4)1  7.2496466  0.0151886   477.309  < 2e-16 ***
poly(Time, 4)2  0.0033972  0.0152323     0.223   0.8237
poly(Time, 4)3  0.1843008  0.0152901    12.054  < 2e-16 ***
poly(Time, 4)4  0.1905802  0.0153211    12.439  < 2e-16 ***
c1             -0.0452337  0.0011688   -38.701  < 2e-16 ***
s1             -0.0076596  0.0011677    -6.560 2.22e-10 ***
c2              0.0076564  0.0011668     6.562 2.19e-10 ***
s2             -0.0086207  0.0011685    -7.377 1.45e-12 ***
c3              0.0199178  0.0011657    17.086  < 2e-16 ***
s3             -0.0138245  0.0011694   -11.822  < 2e-16 ***
c4              0.0100195  0.0011668     8.587 4.21e-16 ***
s4              0.0013102  0.0011687     1.121   0.2631
c5              0.0184909  0.0011688    15.821  < 2e-16 ***
s5              0.0270439  0.0011671    23.172  < 2e-16 ***
c6              0.0034651  0.0008256     4.197 3.53e-05 ***
c348           -0.0075543  0.0011669    -6.474 3.67e-10 ***
s348           -0.0058754  0.0011686    -5.028 8.36e-07 ***
c432           -0.0027359  0.0011721    -2.334   0.0202 *
s432            0.0067057  0.0011719     5.722 2.46e-08 ***
lag1resid       0.8099004  0.0332726    24.341  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Observation: We can clearly see that the value of s4 is not significant as shown in the summary table. Hence, we test for its significance by discarding it first.

#Refitting the model: (by removing the c4 and s4 components)

model6 <- lm(log(Sales) ~ poly(Time, 4) + c1 +s1 +c2+ s2+ c3+ s3+ c5+ s5+ c6+ c348 + s348 +c432 + s432+ lag1resid, data = rsales[1:336,]); summary(model6)

#Performing Anova test:
anova(model5, model6)

Output:

```
> anova(model5, model6)
Analysis of Variance Table

Model 1: log(Sales) ~ poly(Time, 4) + c1 + s1 + c2 + s2 + c3 + s3 + c4 +
    s4 + c5 + s5 + c6 + c348 + s348 + c432 + s432 + lag1resid
Model 2: log(Sales) ~ poly(Time, 4) + c1 + s1 + c2 + s2 + c3 + s3 + c5 +
    s5 + c6 + c348 + s348 + c432 + s432 + lag1resid
  Res.Df      RSS Df Sum of Sq      F    Pr(>F)
1    314 0.071674
2    316 0.088806 -2 -0.017132 37.527 2.434e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Observation: So, According to the data in the ANOVA table, Model 5 fits the data more closely than Model 6 does. This can be established because Model 5 has a smaller p-value and a lower residual sum of squares (RSS), which show that Model 5 offers a better fit.

Conclusion: Hence, we will retain the values c4 and s4 as they are significant. We will consider model 5 only for our data analysis further.

#Amplitude calculations:

```
ampltd<-c(rep(0,times=6))
b2<-coef(model5)[6:16]
for(i in 1:5){
  i1<-2*i-1
  i2<-i1+1
  ampltd[i]<-sqrt(b2[i1]^2+b2[i2]^2)
}
ampltd[6]<-abs(b2[11])
ampltd
```

Output:

```
> ampltd
[1] 0.045877595 0.011529855 0.024245352 0.010104752 0.032761070 0.003465079
```

#Phase calculations:

```
phase<-c(rep(0,times=6))
for(i in 1:5){
  i1<-2*i-1
  i2<-i1+1
  phase[i]<-atan(-b2[i2]/b2[i1])
  if(b2[i1]<0)phase[i]<-phase[i]+pi
  if((b2[i1]>0)&(b2[i2]>0))phase[i]<-phase[i]+2*pi
}
if(b2[11]<0)phase[6]<-pi
Phase
```

Output:

```
> phase
[1] 2.9738487 0.8445749 0.6067400 6.1531574 5.3121161 0.0000000
```

#Peak calculations:

```
peak<-c(rep(0,times=6))
for(i in 1:5){
  peak[i]<-(12/i)-6*phase[i]/(pi*i)
}
if(phase[6]>0)peak[6]<-1
peak
```

Output:

```
> peak
[1] 6.32036729 5.19349039 3.61373732 0.06208373 0.37092111 0.00000000
```

Observation: We observe from the above values of amplitude, phase, and peak that the fundamental, second, and the fifth harmonics have the highest amplitudes.

Q6. We will use the decompose command in R with multiplicative formulation for the estimation of the static seasonal indices. The code is as follows:

```
logSales <- (rsales$logSales)
logSales.ts<-ts(logSales[1:336],freq=12)
logSales.decmps<-decompose(logSales.ts)
seasd<-logSales.decmps$seasonal

Sales <- (rsales$Sales)
Sales.ts<-ts(Sales[1:336],freq=12)
Sales.decmpsm<-decompose(Sales.ts,type="mult")
seasdmult<-Sales.decmpsm$seasonal
```

seasdmult<-seasdmult[1:12]/prod(seasdmult[1:12])^(1/12)

#Tabulation for the above cases:

cbind(seas2,seas4,exp(seasd)[1:12],seasdmult)

	seas2	seas4		seasdmult
(Intercept)	0.9327514	0.9321168	0.9319649	0.9319097
fMonth2	0.9252380	0.9255967	0.9239054	0.9236661
fMonth3	1.0328697	1.0326278	1.0333608	1.0330546
fMonth4	1.0063416	1.0064836	1.0068277	1.0064879
fMonth5	1.0540672	1.0541547	1.0539489	1.0537000
fMonth6	1.0283843	1.0283605	1.0293359	1.0291676
fMonth7	1.0494380	1.0495293	1.0493083	1.0493266
fMonth8	1.0505610	1.0506091	1.0503230	1.0507351
fMonth9	0.9661296	0.9662711	0.9664737	0.9668892
fMonth10	1.0001400	0.9999926	1.0006273	1.0009637
fMonth11	0.9514606	0.9518287	0.9513096	0.9514540
fMonth12	1.0146141	1.0144190	1.0147964	1.0148211

The values have been tabulated above.

Observation: We can see that the values above are very similar. We have tabulated the static seasonal indices of model2, model4 and other models as described in question 6. We can see that the seasonal component has been maintained throughout. Thus same seasonal pattern was maintained in the above table for the models.

Q7. Analysis in this assignment about restaurants and other eating places:

Analysis: We have a file of RestaurantSales.txt that contains US retail sales for restaurants and other eating locations from 1992(1) to 2022(10). The values of sales are mentioned in millions of dollars. As we plotted the Sales and logSales plots, we could observe that there was an increasing trend in both the cases. We could also notice the strong seasonal component in the plots. The seasonal component first reduced and then started rising gradually. In the winter months, the value of seasonal component was low and in the summer months it reached it's peak. This shows that people were eating out more in the months of summer and less in winters. This could be due to several factors as discussed previously. We can also see that volatility increased in Sales plot in the later years whereas in the case of logSales plot, it was more in the initial years. Also, we could see that there were fluctuations in the periods of 2020(1) to 2022(10). This was mainly due to the Covid-19 outbreak. The initial months of 2020 saw a dip in the sales due to lockdown and social distancing matters. As 2021 aproached, there was a rise in the sales due to awareness, masks and other preventive measures being taken by the people. We can also see that there was again a dip possibly due to the strong waves of Covid-19. Well this period disrupted everyone a lot!

We fitted 4th degree polynomial trend for our model. We used multiplicative decomposition model to the variable sales. Initially we saw a lot of outliers when the calendar trigonometry pairs were included in the model. We also saw that as we started using cosine and sine dummies without fMonth variable, there was a clear improvement in the model. Also, we used all possible ways to evaluate a residual model.