Homework 3

Yuanjian Zhou

Nov 22nd 2019

1 Introduction

India is one of the biggest developing countries throughout the world, and there are a lot of places of interest in India like Taj Mahal, which attract millions of tourists from a large variety of countries. Tourism in India is important for the country's economy and is growing rapidly. The World Travel and Tourism Council calculated that tourism generated 240 billion dollars, 9.2% of India's GDP in 2018 and supported 42.673 million jobs, 8.1% of its total employment. The data used in this report is the monthly tourism income in India from Jan 2001 to Sep 2019 (calculated in million dollars).

This data clearly captures the rapid growth of tourism in India during the past two decades, and since tourism relates closely to seasonality, it is a very suitable dataset for this homework.

2 Results

2.1 Modeling and Forecasting Trend

Before modelling the trend, first a plot showing what the data looks like is essential (question 1a).

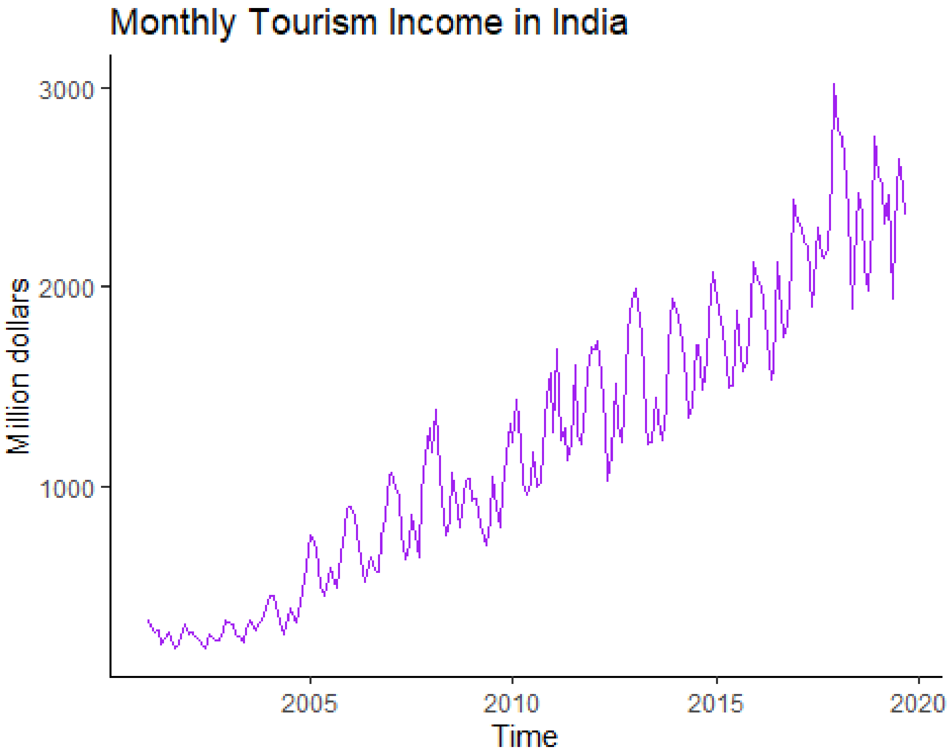


Fig 1.1

Recalling the lecture, a covariance stationary time series should satisfy the following three conditions:

$$\mu_{Y1} = \mu_{Y2} = \cdots = \mu_{Yn}$$

$$\sigma_{Y1} = \sigma_{Y2} = \cdots = \sigma_{Yn}$$

$$\rho_{(Y_t, Y_{t-k})} = \rho_{Y_{|k|}}$$

As can be seen from the plot, this data is far from covariance stationary. The mean increases with time, and there exists heteroskedasticity even in a single year (question 1b).

Next, ACF and PACF plots should be checked to further see whether it is covariance stationary. (question 1c)

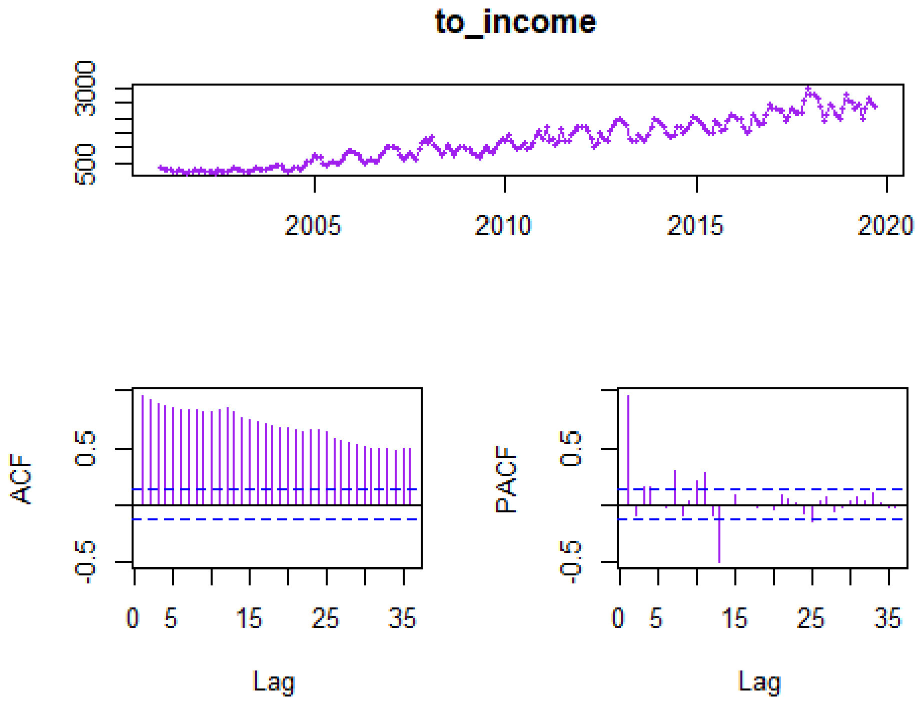


Fig 1.2

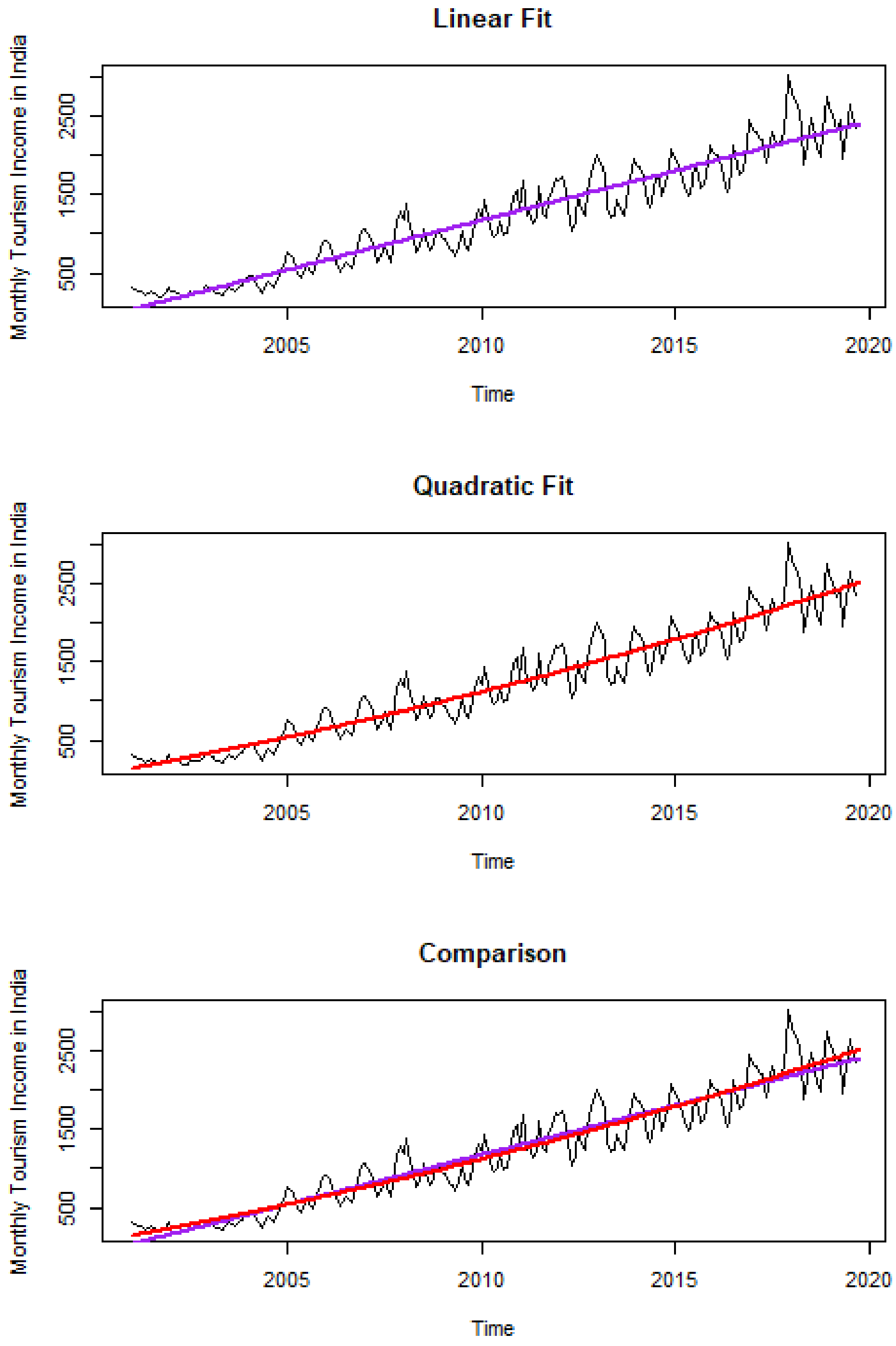
From the ACF and PACF plots, conclusion can be achieved as expected, the ACF all exceeds 2 standard error limits, showing a strong serial interdependence; the PACF plot shows a strong serial correlation between y_t and y_{t-13}.

After knowing the characteristics of the data, we start to model the trend. In this report, a linear model and a non-linear model are experimented. Below are the models.(question 1d)

$$Model1: \ \widehat{INCOME_t} = \beta_0 + \beta_1 t$$

$$Model2: \ \widehat{INCOME_t} = \beta_0 + \beta_1 t + \beta_2 t^2$$

Initially, the original time series plot with the respective fit are drawn below. However, these models look quite similar so a picture putting them together are drawn for comparison.


Linear Fit

Quadratic Fit

Comparison

Fig 1.3

Then, several diagnostic process are performed, including residual plots, residual histograms and statistic summary.

- Residuals vs. fitted values (question 1e)

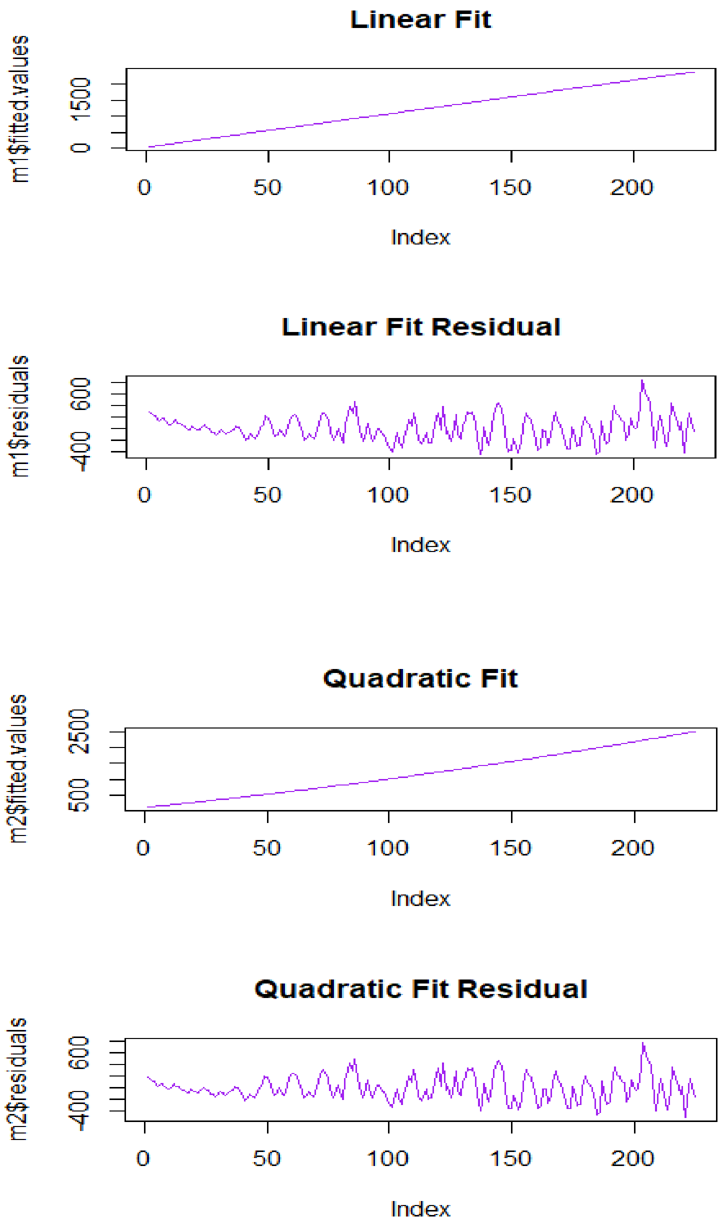


Fig 1.4

The upper plots show that the linear fit residuals does not have trend any more, but it seems to have seasonality. For stationarity, it seems to have the same mean while the variance increases with time. The lower plots share the same pattern with the upper ones..

- Histograms of residuals (question 1f)

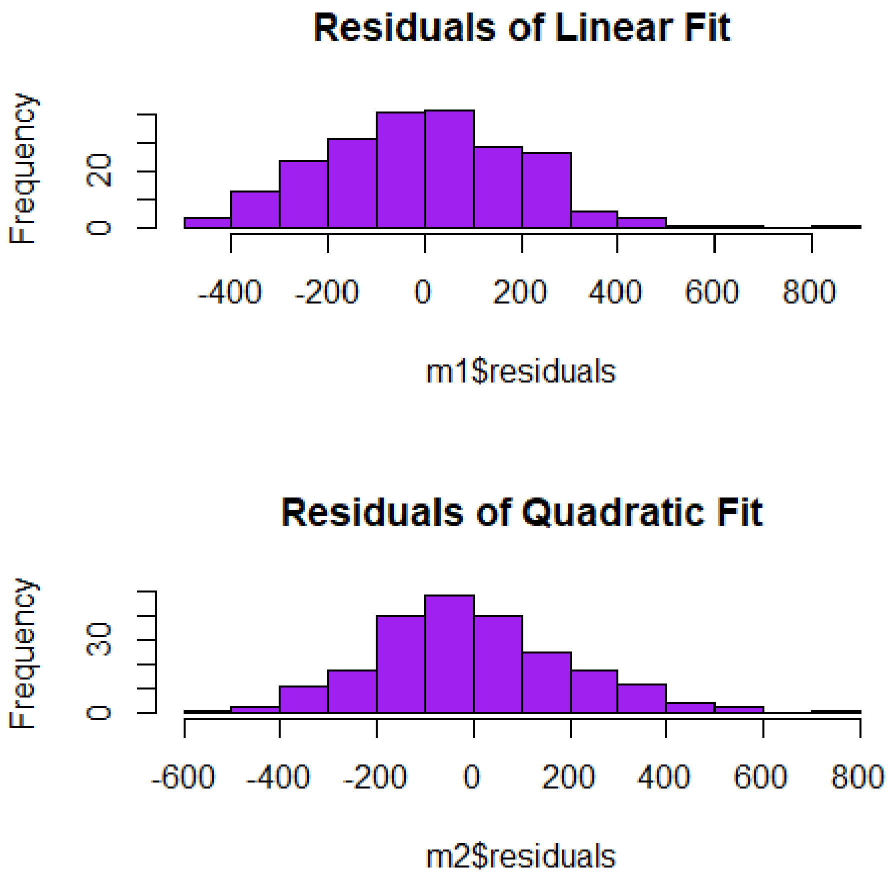


Fig 1.5

The two histograms show similar characeristics: Close to normal distribution, but have a a little too long tails on both sides, especially the right one. Compared with the linear model, the quadratic model looks a little more symmetric.

- Diagnostic statistics (1g)

```
## Call:
## lm(formula = to_income ~ t)
##
## Coefficients:
##                    Estimate       Std. Error          t value        Pr(>|t|)
## (Intercept) -2.523e+05      5.270e+03  -      47.87         <2e-16 ***
## t                  1.261e+02      2.621e+00         48.10          <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Multiple R-squared:  0.9121, Adjusted R-squared:  0.9117
## F-statistic:  2313 on 1 and 223 DF,  p-value: < 2.2e-16

##
## Call:
## lm(formula = to_income ~ t + t2)
```

```
##
## Coefficients:
##                    Estimate    Std. Error     t value    Pr(>|t|)
## (Intercept)   7.276e+06    2.124e+06      3.425     0.000733 ***
## t            -7.363e+03    2.113e+03     -3.484     0.000595 ***
## t2            1.863e+00    5.256e-01      3.544     0.000481 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Multiple R-squared:  0.9168, Adjusted R-squared:  0.916
## F-statistic:  1223 on 2 and 222 DF,  p-value: < 2.2e-16
```

In the linear model, t statistics for each parameter is significant at 0.1% level, adjusted R-square reaches 0.9117 and the F- statistic also shows significance. When it comes to the quadratic model, each parameter is also significant at 0.1% level, adjusted R-square reaches 0.916, a little larger than the linear model, and the F- statistic is statistically significant.

After the diagnostic process, a model should be chosen with AIC and BIC criterion separately. (question 1h).

```
##    df      AIC
## m1  3 3056.744
## m2  4 3046.364

##    df      BIC
## m1  3 3066.992
## m2  4 3060.028
```

From the values of AIC and BIC, apparently the quadratic model should be selected with both AIC and BIC.

Next, this model is be used to forecast h steps ahead, here it is selected to be 3 years head, since the original one is a monthly data, the forecast steps h should be 36.

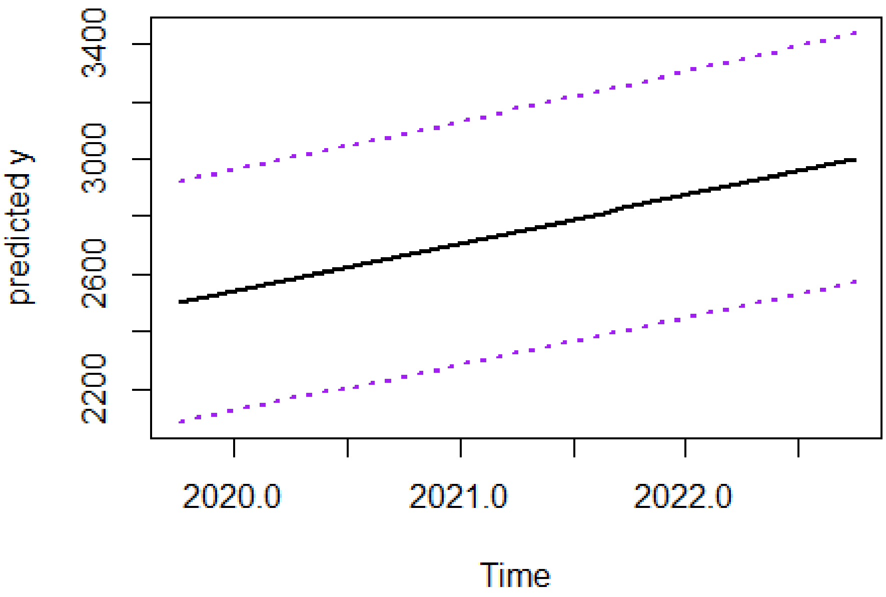


Fig 1.6

As can be shown above, the black line is forecast value, and the purple lines are prediction intervals.

2.2 Modeling and Forecasting Seasonality

Different of the above section, here the "tslm" function in "forecast" package can be used to conveniently build the model with fully season dummies. (question 2a)

```
## 
## Call:
## tslm(formula = to_income ~ season + 0)
## 
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)    
## season1    1360.5      165.6   8.217 2.02e-14 ***
## season2    1382.9      165.6   8.352 8.57e-15 ***
## season3    1254.6      165.6   7.577 1.07e-12 ***
## season4    1132.4      165.6   6.839 8.31e-11 ***
## season5     964.0      165.6   5.822 2.12e-08 ***
## season6    1049.5      165.6   6.339 1.36e-09 ***
## season7    1278.0      165.6   7.719 4.50e-13 ***
## season8    1160.5      165.6   7.009 3.12e-11 ***
## season9    1072.8      165.6   6.479 6.29e-10 ***
## season10   1118.7      170.1   6.576 3.67e-10 ***
## season11   1310.6      170.1   7.704 4.91e-13 ***
## season12   1483.6      170.1   8.721 7.93e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 721.7 on 213 degrees of freedom
## Multiple R-squared:  0.7517, Adjusted R-squared:  0.7377
## F-statistic: 53.74 on 12 and 213 DF,  p-value: < 2.2e-16
```

From the summary above, the F-statistic is still statistically significant while the adjusted R-squared 0.7377 is smaller than the linear model. Besides, all the t-statistics are significant.

Then, the seasonal factors are plotted to detect seasonal effects. (question 2b)

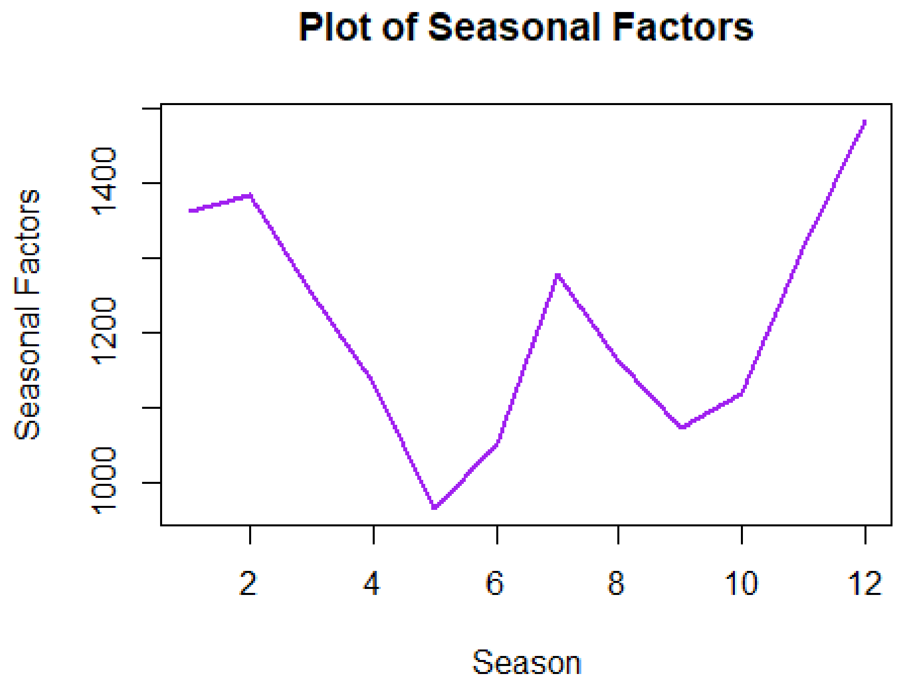


Fig 2.1

From seasonal factors plot, the fluctuation in tourism can be seen. In spring, tourism income is be smaller since there is no long-time vocations; income increases from June and reaches a local peak in Angust, which is the summer vacation period; income continuously decreases from Angust to October and starts to go up rapidly from October to February, which is the period covering two biggest holidays: Spring Festival and Christmas.

Afterwards, the trend model from question 1 are added to the seasonal model. Again, the respective residuals vs. fitted values are plotted. (question 2c)

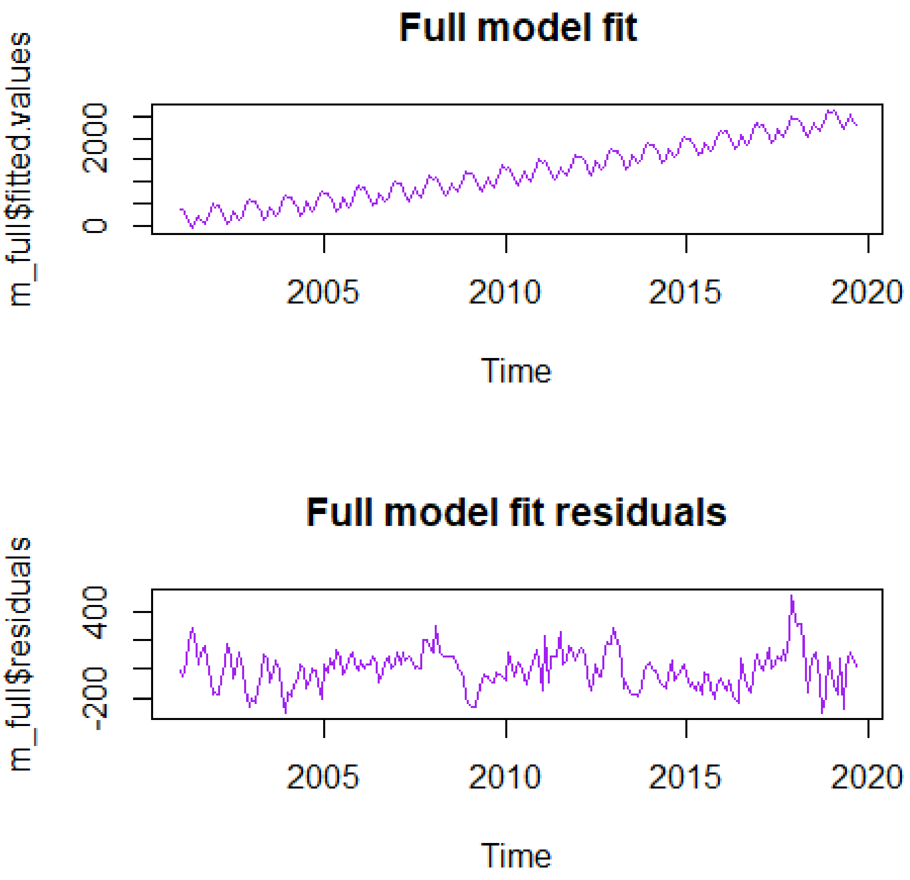


Fig 2.2

The plot shows the residual series looks more random since 2005, compared to the plots in question 1. And it still seems to have the constant mean, but this time it has a more constant variance except some outliers.

Then, the diagnostic statistics of the full model are also be checked, together with the error metrics for this time.(question 2d)

```
## 
## Call:
## tslm(formula = to_income ~ t + season + t2)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -307.13  -93.78    4.11   87.93  518.84 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  7.685e+06  1.385e+06   5.547 8.64e-08 ***
## t           -7.771e+03  1.378e+03  -5.638 5.48e-08 ***
## season2      1.184e+01  4.405e+01   0.269  0.78843    
## season3     -1.269e+02  4.405e+01  -2.881  0.00437 ** 
## season4     -2.597e+02  4.405e+01  -5.896 1.46e-08 ***
## season5     -4.387e+02  4.405e+01  -9.959  < 2e-16 ***
## season6     -3.638e+02  4.405e+01  -8.257 1.62e-14 ***
## season7     -1.459e+02  4.406e+01  -3.312  0.00109 **
## season8     -2.741e+02  4.406e+01  -6.221 2.62e-09 ***
## season9     -3.725e+02  4.406e+01  -8.454 4.64e-15 ***
## season10    -2.673e+02  4.467e+01  -5.985 9.20e-09 ***
## season11    -8.596e+01  4.467e+01  -1.924  0.05568 .
## season12     7.639e+01  4.468e+01   1.710  0.08875 .
## t2           1.964e+00  3.428e-01   5.730 3.44e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 135.8 on 211 degrees of freedom
## Multiple R-squared:  0.9664, Adjusted R-squared:  0.9644
## F-statistic: 467.5 on 13 and 211 DF,  p-value: < 2.2e-16
```

```
##                        ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -2.152553e-15 131.4753 104.5138 -0.3680711 14.48379 0.6522249
##                   ACF1
## Training set 0.6399676
```

From the summary statistic and error metrics above, the conclusion can be achieved: From t-statistic, F-statistic and adjusted R-squared, this is a quite good model with a 0.9644 adjusted R-square, while the interpretation of error metrics becomes a little harder, thus some common ones are interpreted below..

- ME means Mean Error, and it simply sum all errors, so it makes no much sense.
- RMSE means root mean square error, MAE means mean absolute error, these two metrics are similar, they mean that we will be about -100 to 100 away from the real data.
- MPE means mean percentage error, as ME, it also makes no much sense.
- MASE means mean absolute percentage error, it means we will have about 14% measure error, which is a little large.

Afterwards, this model is used to get a 36 steps ahead forecast. (question 2e)

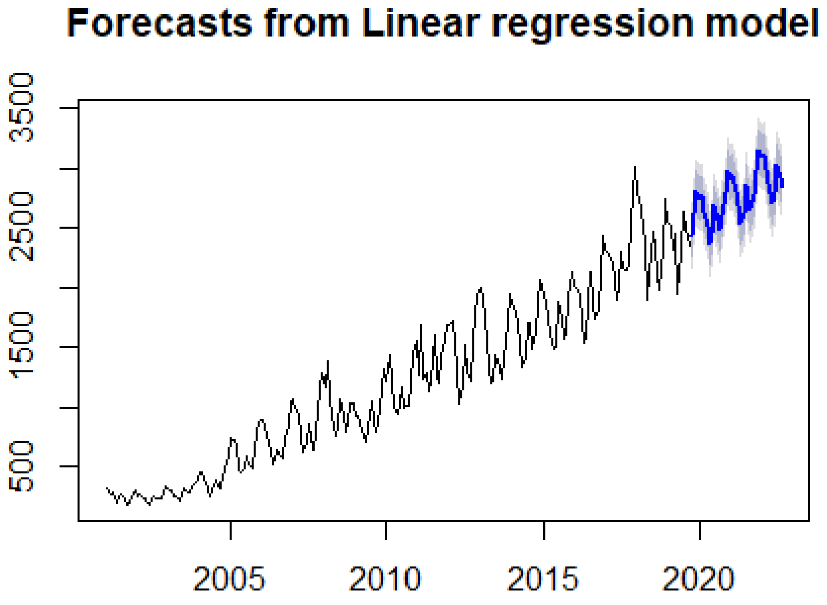


Fig 2.3

Above is the forecast with prediction intervals.

Finally, the seasonal adjustment is performed. And when talking about seasonal adjustment, the "stl" plot show be drawn first to see whether it is an additive or multiplicative adjustment. (question 1f)

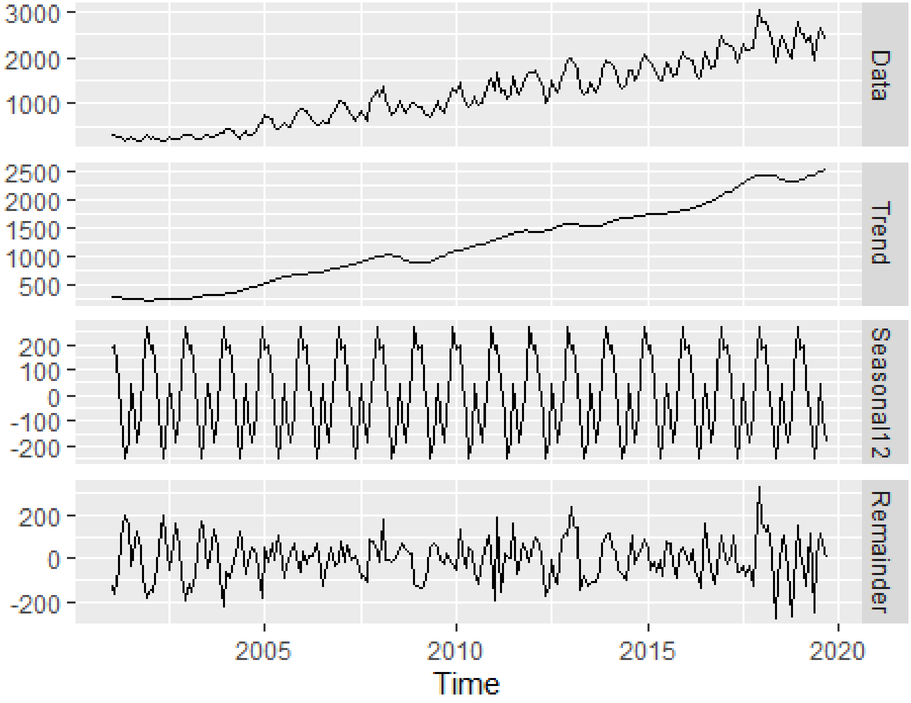


Fig 2.4

Based on the stl plot, an additive model is more suitable, so it is used for the data.

Then the seasonal adjusted data with the quadratic fit is plotted.

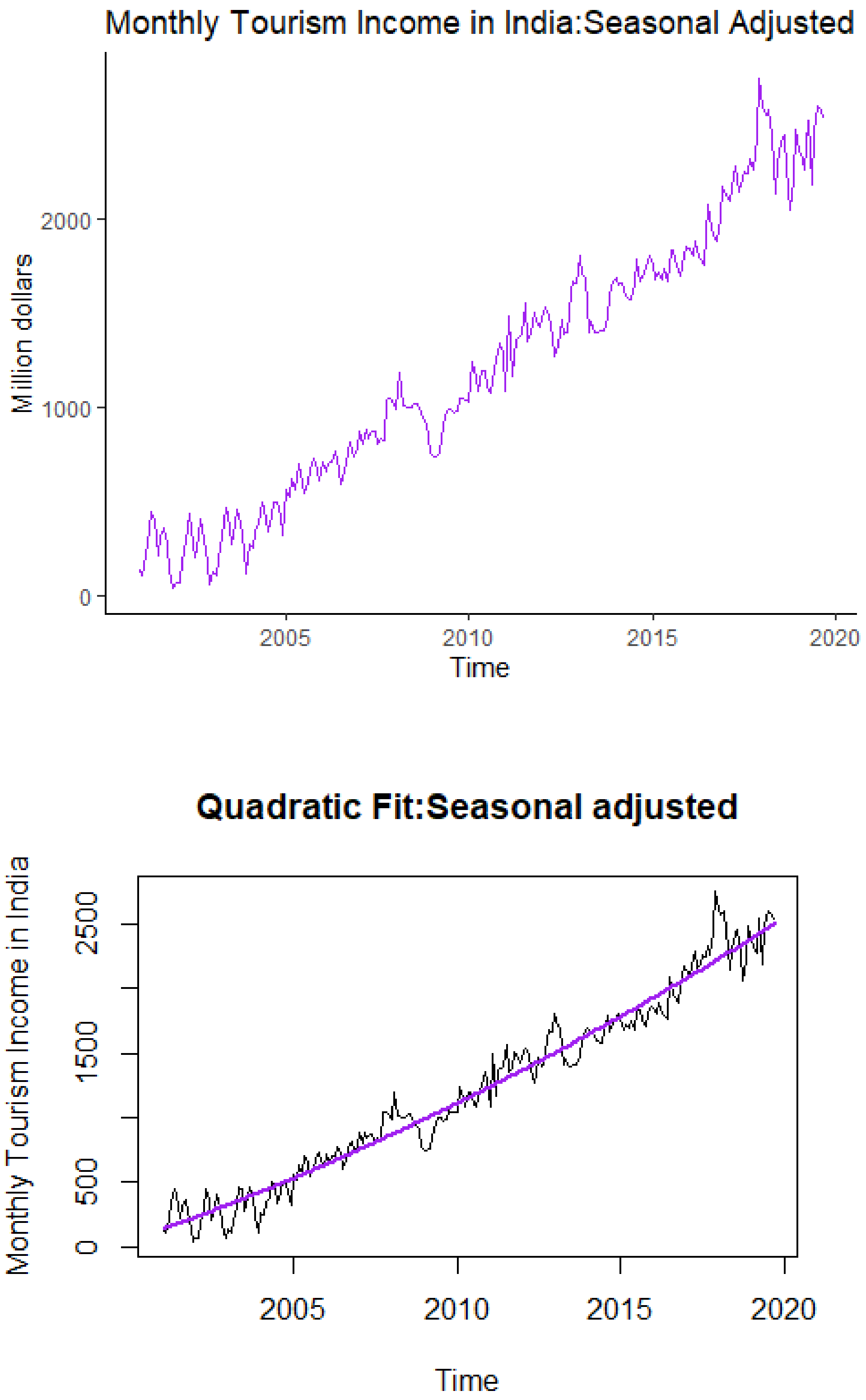


Fig 2.5

The trend model from question 1 is still appropriate, actually it fits even better. It is because the seasonal component is removed, the fluctuation will be smaller than before.

3 Conclusion and Future Work

From question 2, the final model combines the quadratic trend model with the seasonal dummies. As can be seen from the forecast plot, it looks quite good. When the seasonal adjustment is removed, the trend model looks better than in question 1, which shows that our trend model fits this data quite well.

While, it can be seen that some error metrics of the final model are not satisfying, in the future, two directions of improving the forecast can be considered.: one is to use "ets" function which is a better way of performing decomposition, another is to use ARIMA model which will be taught in the following lectures.

4 Reference

1. Ministry of Tourism(India) (2019), Retrieved from https://insights.ceicdata.com/series/21448401_SR522329
2. Tourism in India, Retrieved August 22, 2008, https://en.wikipedia.org/wiki/Tourism_in_India.
3. Rob Hyndman (2019). Forecast. R package version 8.9 https://www.rdocumentation.org/packages/forecast
4. Hadley Wickham (2019). Tidyverse. R package version 1.3.0 https://rdocumentation.org/packages/tidyverse

5 R Source Code

```
===============================Preparation===============================

# Load necessary packages
library(tidyverse)
library(forecast)

# Import data into R:
setwd("C:/Users/zyj37/Desktop/MAE/ECON 430/Homework/3")
to = read.csv("tourism_income.csv",header=F)

# Convert data to time series format:
to_income  = ts(to[,2],freq=12,start=2001)

# Generate the time dummy variables
t = seq(2001,2019.75,length=length(to_income))

===============================Question 1================================

##########################################1a##########################################

# Plot the original data
autoplot(to_income,col="purple")+
  ylab("Million dollars")+
  ggtitle("Monthly Tourism Income in India")+
  theme_classic()

############################################1c##############################################

# Draw ACF and PACF plot
tsdisplay(to_income,col="purple")

############################################1d##############################################

par(mfrow=c(3,1))
# Linear Fit
m1 = lm(to_income~t)
# Plot the original time series plot with linear fit
plot(to_income,xlab="Time",ylab="Monthly Tourism Income in India",
     main = "Linear Fit")
lines(t,m1$fitted.values,col="purple",lwd=2)

# Quadratic Periodic Fit
t2 = t^2
m2 = lm(to_income~t+t2)
# Plot the original time series plot with Quadratic periodic fit
plot(to_income,xlab="Time",ylab="Monthly Tourism Income in India",
     main = "Quadratic Fit")
lines(t,m2$fitted.values,col="purple",lwd=2)

# Draw them in the same plot
plot(to_income,xlab="Time",ylab="Monthly Tourism Income in India",
     main = "Comparison")
lines(t,m1$fitted.values,col="purple",lwd=2)
lines(t,m2$fitted.values,col="red",lwd=2)

############################################1e##############################################

par(mfrow=c(2,1))
# plot the residuals vs. fitted values for model 1.
plot(m1$fitted.values,main = "Linear Fit",col="purple",type ="l")
plot(m1$residuals,main = "Linear Fit Residual",col="purple",type="l")


# plot the residuals vs. fitted values for model 2.
plot(m2$fitted.values,main = "Quadratic Fit",col="purple",type ="l")
plot(m2$residuals,main = "Quadratic Fit Residual",col="purple",type="l")

############################################1f##############################################
# plot the histograms of residuals
hist(m1$residuals,breaks="FD",col="purple",main="Residuals of Linear Fit")
hist(m2$residuals,breaks="FD",col="purple",main="Residuals of Quadratic Fit")

############################################1g##############################################

# Summary statistics of m1
s_m1 = summary(m1)
s_m1
```

```
# Summary statistics of m2
s_m2 = summary(m2)
s_m2

############################################1h#############################################

# use AIC and BIC to select models.
AIC(m1,m2)
BIC(m1,m2)

############################################1i############################################

# Perform forecast and draw the plot

# generate our data frame for forecast
tn=data.frame(t=seq(2019.75,2022.75,length.out=36))
tn = tn %>%
  mutate(t2 = t^2)

# perform forecast and draw the plot
pred_m2 = predict(m2,tn,interval="prediction")
matplot(tn$t,pred_m2,lty=c(1,3,3),col=c("black","purple","purple"), type="l", lwd=2, ylab="predicted y",xlab="Time")

############################################2a#############################################

# run the regression with fully season dummies.
m3 = tslm(to_income ~ season+0)
summary(m3)

############################################2b#############################################

# plot estimated seasonal factors
plot(m3$coefficients,type='l',ylab='Seasonal Factors', xlab="Season",lwd=2, main="Plot of Seasonal Factors",col="purple")

############################################2c#############################################

# Run the full model
t2=t^2
m_full = tslm(to_income~t+season+t2)

# Plot the fitted values vs. residuals of full model
par(mfrow=c(2,1))
plot(m_full$fitted.values,
     main = "Full model fit",col="purple")
plot(m_full$residuals,
     main = "Full model fit residuals",col="purple")

############################################2d#############################################

# Get summary statistic and error metrics of the model.
summary(m_full)
accuracy(m_full)

############################################2e#############################################
```

```
# Generate the prediction data frame
df = data.frame(seasonaldummy(to_income,h=36))
df = df %>%
  mutate(t =seq(2019.75,2022.75,length.out=36)) %>%
  mutate(t2=t^2)

# Plot the forecast
plot(forecast(m_full,df))

########################################2f##########################################

# perfrom the "mstl" function which is an improved version of stl
mstl(to_income,s.window = "periodic") %>%
  autoplot()
to_income_stl = stl(to_income,s.window = "periodic")

# function for seasonal adjustment
to_income_seasadj = seasadj(to_income_stl)

# draw the seasonal adjusted plot and the fitted line
autoplot(to_income_seasadj,col="purple")+
  ylab("Million dollars")+
  ggtitle("Monthly Tourism Income in India:Seasonal Adjusted")+
  theme_classic()

plot(to_income_seasadj,xlab="Time",ylab="Monthly Tourism Income in India",
     main = "Quadratic Fit:Seasonal adjusted")
lines(t,m2$fitted.values,col="purple",lwd=2)
```