Project 1 - U.S. Flights Passenger Data

Ian Cranfield et al.

4/28/2020

# I. Introduction

Airports and Airlines are faced with a multitude of decisions that could be greatly improved with accurate forecasting. Airlines must plan out flights and flight patterns months in advance, must be able to accomodate periods of increased travel and periods of less travel, and must price accordingly for these periods. A good understanding of how travel patterns will behave in the future and the long term is extremely valuable to those making these decisions. In this project, we will analyze passenger data retrieved from the U.S. Bureau of Transportation Statistics on April 22, 2020. Our data spans from October 2002 to January 2020. For the purposes of this project we will focus on analyzing domestic flights passenger data. As travel tends to occur during holidays and vacation times, we can expect to see some degree of seasonality in our data. Additionally since air travel has become more accessible and a more efficient method of travel we can expect to see an upward trend. We will analyze the data to see if these are correct characterizations. Using this data we will create a model and forecast travel trends and seasonality for 2 years from January 2020.

# Clear previous sessions, and load necessary libraries

# remove and clear previous sessions  
rm(list = ls(all = TRUE))  
  
# load necessary libraries   
  
library(tseries)  
library(zoo)  
library(quantmod)  
library(XML)  
library(readxl)  
library(dplyr)  
library(ggplot2)  
library(ggfortify)  
library(fpp2)  
library(psych)  
library(dynlm)  
library(tis)  
library(lubridate)  
require("datasets")  
library(timeSeries)  
library(gridExtra)  
library(stats)  
library(seasonal)  
library(dplyr)  
library(scales)  
library(readxl)  
  
# Read the excel file into a data frame (remove the first row and use the next one as headers)  
passengers\_df <- read\_excel("Airline Passengers.xlsx", skip = 1, col\_names = TRUE, col\_types = "numeric")  
  
# Filter and clean up the data so we only see the relevant information  
passengers\_df <- passengers\_df %>%  
 dplyr::filter(Month > 0)  
  
# Inspect the data  
head(passengers\_df, 12) # first data point 10/2002, monthly frequency

## # A tibble: 12 x 5  
## Year Month DOMESTIC INTERNATIONAL TOTAL  
## <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2002 10 48054917 9578435 57633352  
## 2 2002 11 44850246 9016535 53866781  
## 3 2002 12 49684353 10038794 59723147  
## 4 2003 1 43032450 9726436 52758886  
## 5 2003 2 41166780 8283372 49450152  
## 6 2003 3 49992700 9538653 59531353  
## 7 2003 4 47033260 8309305 55342565  
## 8 2003 5 49152352 8801873 57954225  
## 9 2003 6 52209516 10347900 62557416  
## 10 2003 7 55810773 11705206 67515979  
## 11 2003 8 53920973 11799672 65720645  
## 12 2003 9 44213408 9454647 53668055

# Create time series variable for domestic flights  
domestic\_ts <- ts(passengers\_df$DOMESTIC, start = c(2002,10), freq=12)  
  
str(domestic\_ts)

## Time-Series [1:208] from 2003 to 2020: 48054917 44850246 49684353 43032450 41166780 ...

#We will focus on the domestic flights data for this project

# II. Results

## 1. Modeling and Forecasting Trend

### a. Show the time-series plot of your data

# Plotting the data - only focusing on domestic flights for the entire time frame (2002-2020)  
autoplot(domestic\_ts) +   
 ggtitle("Monthly Domestic U.S. Air Passengers Data") +   
 ylab("Passengers") + xlab("Year") +   
 theme(plot.title = element\_text(hjust = 0.5))

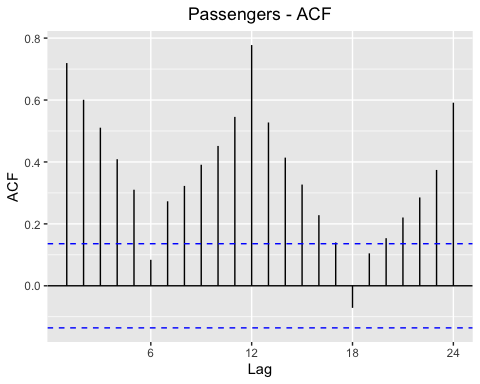
Show the time-series plot of your data-1.png)

### b. Does your plot in (a) suggest that the data are covariance stationary? Explain your answer.

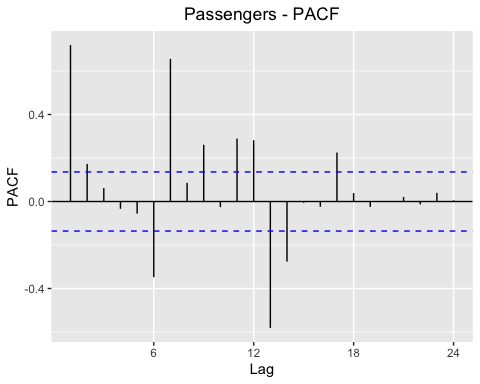
Looking at the plot we can conclude that the data is non-stationary, as it is clearly drifting upwards and not reverting consistently around a mean therefore it is not covariance stationary.

### c. Plot and discuss the ACF and PACF of your data.

# Side by side view of the ACF and PACF   
par(mfrow=c(2,1))  
ggAcf(domestic\_ts) + ggtitle("Passengers - ACF") + theme(plot.title = element\_text(hjust = 0.5))



ggPacf(domestic\_ts) + ggtitle("Passengers - PACF") + theme(plot.title = element\_text(hjust = 0.5))



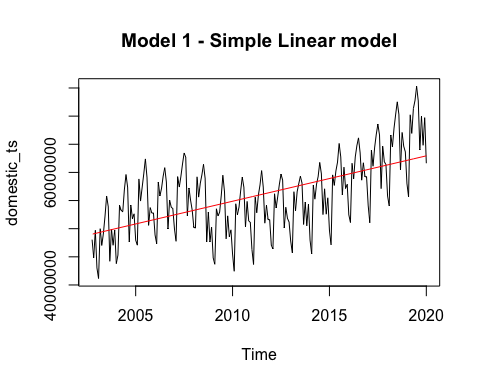
Viewing the ACF and PACF plots confirms our belief that the data is non-stationary. The ACF shows significant correlations on almost all lags exceeding the confidence interval suggesting that the data is non-stationary. Since the lags do not phase out towards zero we can conclude the data is non-stationary. The peaks are statistically significant, meaning the data exhibits time dependence. Particularly, we see the strongest dependence in the ACF plot for lags of multiples of 12, making the claim for seasonality in the data significant. Examining the PACF of the data it is clear that there is time dependency as there are significant lags up to seventeen months prior.

### d. Fit a Linear and Nonlinear model.

options(scipen=999)  
# Create t variable  
t<-seq(2002.783, 2020 ,length=length(domestic\_ts))  
  
#Model 1: Simple Linear model  
m1 <- lm(domestic\_ts ~ t)   
summary(m1)

##   
## Call:  
## lm(formula = domestic\_ts ~ t)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12666090 -3725379 97277 3844000 12740314   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1566740363 151562150 -10.34 <0.0000000000000002 \*\*\*  
## t 806774 75352 10.71 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5427000 on 206 degrees of freedom  
## Multiple R-squared: 0.3575, Adjusted R-squared: 0.3544   
## F-statistic: 114.6 on 1 and 206 DF, p-value: < 0.00000000000000022

plot(domestic\_ts, main="Model 1 - Simple Linear model") # plot the main  
lines(t,m1$fitted.values, col="red")+ theme(plot.title = element\_text(hjust = 0.5)) # lines plots a graph over the main

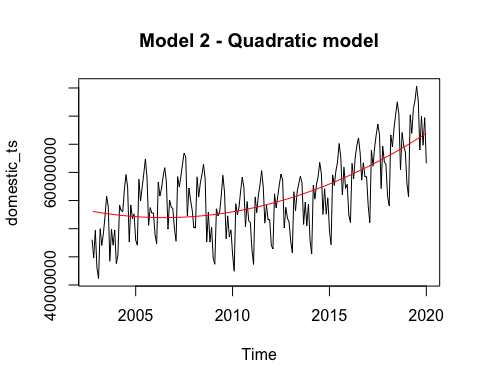


## NULL

#Model 2: Quadratic  
m2 <- lm(domestic\_ts ~ t + I(t^2))  
summary(m2)

##   
## Call:  
## lm(formula = domestic\_ts ~ t + I(t^2))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11711091 -3237646 668390 3476980 11394403   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 328801989546 64407377197 5.105 0.000000753 \*\*\*  
## t -327692940 64042883 -5.117 0.000000713 \*\*\*  
## I(t^2) 81660 15920 5.129 0.000000672 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5122000 on 205 degrees of freedom  
## Multiple R-squared: 0.4306, Adjusted R-squared: 0.425   
## F-statistic: 77.52 on 2 and 205 DF, p-value: < 0.00000000000000022

plot(domestic\_ts, main = "Model 2 - Quadratic model")  
lines(t,m2$fitted.values, col="red")+ theme(plot.title = element\_text(hjust = 0.5))

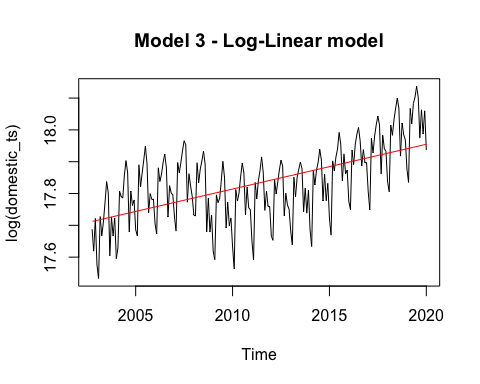


## NULL

#Model 3: Log-Linear  
m3 <- lm(log(domestic\_ts) ~ t)  
summary(m3)

##   
## Call:  
## lm(formula = log(domestic\_ts) ~ t)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.251888 -0.060825 0.006664 0.073705 0.197895   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -10.473724 2.723835 -3.845 0.000161 \*\*\*  
## t 0.014074 0.001354 10.393 < 0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09754 on 206 degrees of freedom  
## Multiple R-squared: 0.344, Adjusted R-squared: 0.3408   
## F-statistic: 108 on 1 and 206 DF, p-value: < 0.00000000000000022

plot(log(domestic\_ts), main = "Model 3 - Log-Linear model")  
lines(t,m3$fitted.values, col="red")+ theme(plot.title = element\_text(hjust = 0.5))

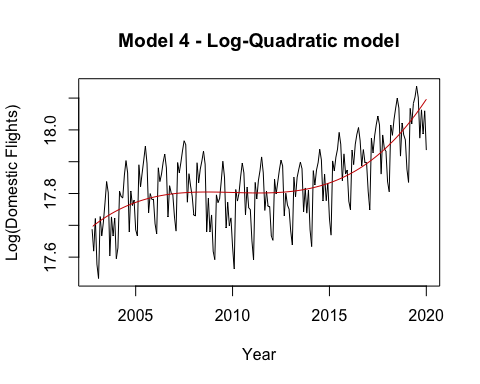


## NULL

#Model4: Log-Quadratic  
ldomestic <- log(domestic\_ts)  
  
m4 = lm(ldomestic ~ t + I(t^2) + I(t^5))  
summary(m4)

##   
## Call:  
## lm(formula = ldomestic ~ t + I(t^2) + I(t^5))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.239499 -0.068956 0.007813 0.069138 0.166611   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) -1518755.899647626560181 306065.561306091840379 -4.962  
## t 1888.998078652666663 380.417513540456468 4.966  
## I(t^2) -0.626525141132495 0.126088040455307 -4.969  
## I(t^5) 0.000000000007715 0.000000000001549 4.979  
## Pr(>|t|)   
## (Intercept) 0.00000147 \*\*\*  
## t 0.00000144 \*\*\*  
## I(t^2) 0.00000142 \*\*\*  
## I(t^5) 0.00000136 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.08845 on 204 degrees of freedom  
## Multiple R-squared: 0.4657, Adjusted R-squared: 0.4579   
## F-statistic: 59.27 on 3 and 204 DF, p-value: < 0.00000000000000022

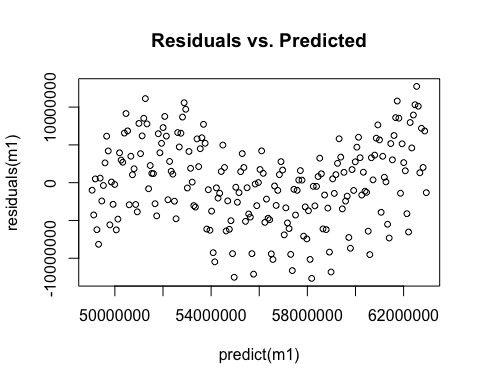
plot(ldomestic,xlab="Year", ylab="Log(Domestic Flights)", main="Model 4 - Log-Quadratic model")  
lines(t, m4$fit,col="red3")+ theme(plot.title = element\_text(hjust = 0.5))



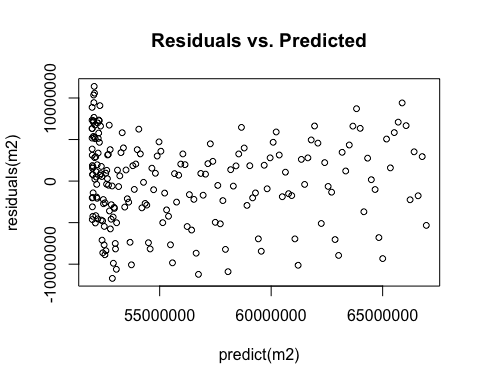
## NULL

### e. Residuals vs. Fitted plots for All 4 Models.

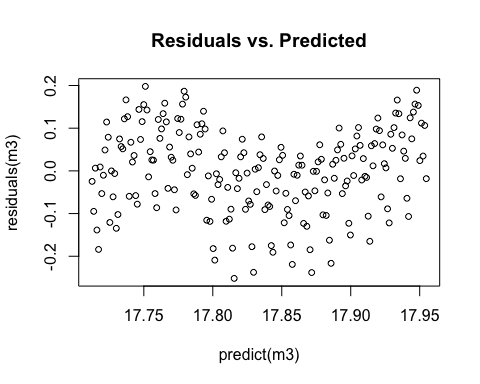
options(scipen=999)  
plot(predict(m1), residuals(m1), cex=0.8, main="Residuals vs. Predicted")



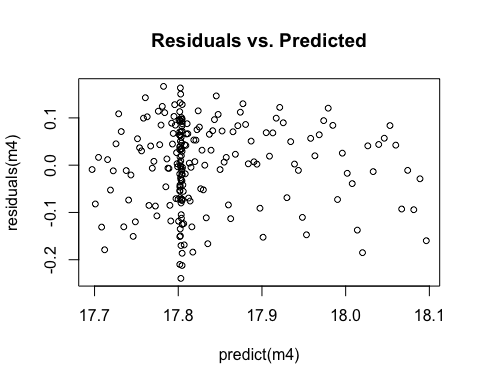
plot(predict(m2), residuals(m2), cex=0.8, main="Residuals vs. Predicted")



plot(predict(m3), residuals(m3), cex=0.8, main="Residuals vs. Predicted")



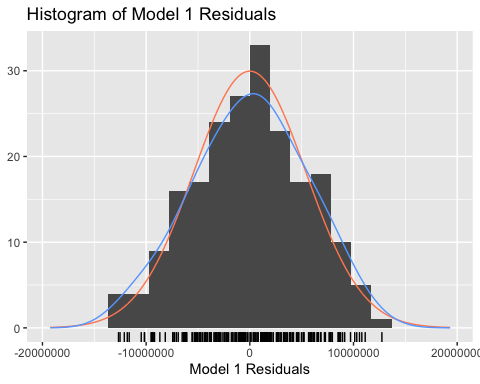
plot(predict(m4), residuals(m4), cex=0.8, main="Residuals vs. Predicted")



Viewing the residual plots reveals that there was still some seasonality that was unaccounted for in our first model. It should be noted the the x-axis is different on for Model 4 due to the addition of a log into the model. Viewing model 4 it is clear that much more of the variation has been accounted for in the model as the residuals demonstrate a more random distribution. There is however still some clustering towards the left half of the model at 17.8. From this we can conclude that our model is a decent fit to the data.

### f. Histograms for All 4 Models

gghistogram(m1$residuals, add.normal = TRUE, add.kde = TRUE , bins = 13) +   
 ggtitle("Histogram of Model 1 Residuals") + xlab("Model 1 Residuals") + ylab(NULL)

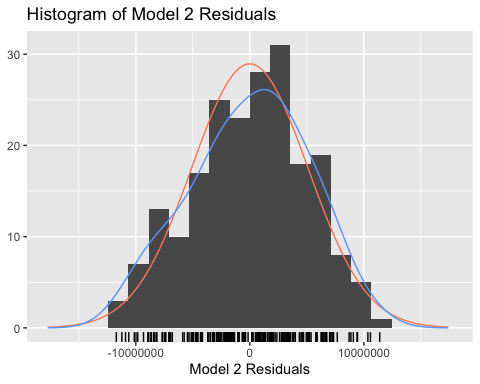


shapiro.test(m1$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: m1$residuals  
## W = 0.99231, p-value = 0.346

The Histogram of Model 1 Residuals from a *Linear* model show a normally distributed plot (red) and a Kernal Density Estimation (blue) which indicate that the variance is normally distributed. The best way to truly test normality are through tests, so we also ran a Shapiro-Wilk normality test. For model 1, the p-value is greater than 0.05 so we fail to reject that model 1 is Normally Distributed.

gghistogram(m2$residuals, add.normal = TRUE, add.kde = TRUE, bins = 13) +   
 ggtitle("Histogram of Model 2 Residuals") + xlab("Model 2 Residuals") + ylab(NULL)

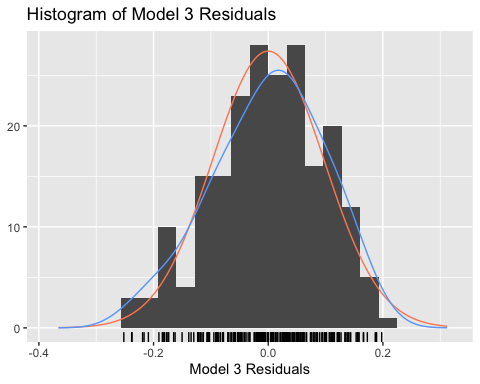


shapiro.test(m2$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: m2$residuals  
## W = 0.98626, p-value = 0.04148

The Histogram of Model 2 Residuals shows a slightly skewed to the right. Indicating that that more values have positive errors. The Shapiro-Wilk normality test has a p-value of 0.04 which is less than 0.05, so we reject that model 2 is Normally Distributed.

gghistogram(m3$residuals, add.normal = TRUE, add.kde = TRUE, bins = 14) +   
 ggtitle("Histogram of Model 3 Residuals") + xlab("Model 3 Residuals") + ylab(NULL)

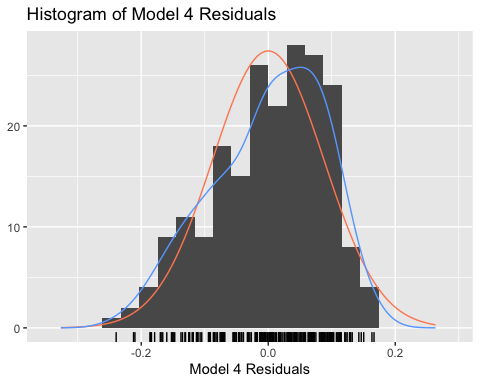


shapiro.test(m3$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: m3$residuals  
## W = 0.9856, p-value = 0.03274

The Histogram of Model 3 Residuals from a *Log-Linear* model that shows a slightly more normal distribution than Model 2. The Shapiro-Wilk normality test has a p-value of 0.04 so we reject that model 3 is Normally Distributed.

gghistogram(m4$residuals, add.normal = TRUE, add.kde = TRUE, bins = 14) +   
 ggtitle("Histogram of Model 4 Residuals") + xlab("Model 4 Residuals") + ylab(NULL)



shapiro.test(m4$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: m4$residuals  
## W = 0.97045, p-value = 0.0002337

The Histogram of Model 4 Residuals from a *Log-Quadratic* Model show a slightly skewed distribution densely on the right. The error values are on the positive side so it may suggest it is over valued. The Shapiro-Wilk normality test has a low p-value of 0.0002337 which is less than 0.05 so we reject that the histogram is normally distributed.

\*To further analyze if the residuals are normally distributed we should look at Normal Q-Q plots.

### g. Model Diagnostics Statistics (R^2, t-dist, f-dist).

##### Linear Model

summary(m1)

##   
## Call:  
## lm(formula = domestic\_ts ~ t)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12666090 -3725379 97277 3844000 12740314   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1566740363 151562150 -10.34 <0.0000000000000002 \*\*\*  
## t 806774 75352 10.71 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5427000 on 206 degrees of freedom  
## Multiple R-squared: 0.3575, Adjusted R-squared: 0.3544   
## F-statistic: 114.6 on 1 and 206 DF, p-value: < 0.00000000000000022

Looking at the summary it tells us that the intercept and the t variable are statistically significant by looking at the low p-values and high t-values. The adjusted R-Squared is quite low at 0.3544. The high F-statistic and low p-value indicates this model is significant.

##### Quadratic Model

summary(m2)

##   
## Call:  
## lm(formula = domestic\_ts ~ t + I(t^2))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11711091 -3237646 668390 3476980 11394403   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 328801989546 64407377197 5.105 0.000000753 \*\*\*  
## t -327692940 64042883 -5.117 0.000000713 \*\*\*  
## I(t^2) 81660 15920 5.129 0.000000672 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5122000 on 205 degrees of freedom  
## Multiple R-squared: 0.4306, Adjusted R-squared: 0.425   
## F-statistic: 77.52 on 2 and 205 DF, p-value: < 0.00000000000000022

This *Quadratic* model has a statistically significant Intercept and Explanatories, with a slightly higher Adjusted R-Squared than the linear model. It has a lower F-stat, and lower residual standard error.

##### Log - Linear Model

summary(m3)

##   
## Call:  
## lm(formula = log(domestic\_ts) ~ t)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.251888 -0.060825 0.006664 0.073705 0.197895   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -10.473724 2.723835 -3.845 0.000161 \*\*\*  
## t 0.014074 0.001354 10.393 < 0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09754 on 206 degrees of freedom  
## Multiple R-squared: 0.344, Adjusted R-squared: 0.3408   
## F-statistic: 108 on 1 and 206 DF, p-value: < 0.00000000000000022

The Intercept (log variable) and the estimator are statistically significant with low p-values and high t-values. The R-squared and Adjusted R-Squared is less than the linear, and quadratic model at 0.344 and 0.3408, respectively. The F-stat has a very low p-value so this model has overall significance. This model also has a really low Residual standard error compared to the Linear and Quadratic models.

##### Log Quadratic Model

summary(m4)

##   
## Call:  
## lm(formula = ldomestic ~ t + I(t^2) + I(t^5))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.239499 -0.068956 0.007813 0.069138 0.166611   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) -1518755.899647626560181 306065.561306091840379 -4.962  
## t 1888.998078652666663 380.417513540456468 4.966  
## I(t^2) -0.626525141132495 0.126088040455307 -4.969  
## I(t^5) 0.000000000007715 0.000000000001549 4.979  
## Pr(>|t|)   
## (Intercept) 0.00000147 \*\*\*  
## t 0.00000144 \*\*\*  
## I(t^2) 0.00000142 \*\*\*  
## I(t^5) 0.00000136 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.08845 on 204 degrees of freedom  
## Multiple R-squared: 0.4657, Adjusted R-squared: 0.4579   
## F-statistic: 59.27 on 3 and 204 DF, p-value: < 0.00000000000000022

The (log) intercept as well as the t, t^2, and t^5 have low p-values and high t-values so it is statistically significant. The F-stat is lower than the other models but by looking at the p-value it is really low which means the model has overall significance. The R-Squared and Adjusted R-Squared is much higher than the other models at 0.4657 and 0.4579, respectively.

##### Data Frame of all of the R-Squared and Adjusted R-Squared

#Gather the R suqared and Adjusted R squared results into a a data frame in order to consturct a table   
  
results <- data.frame(c("Model 1", "Model 2", "Model 3", "Model 4"),c("Linear","Quadratic","Log-Linear","Log-Quadratic") ,c(summary(m1)$r.squared, summary(m2)$r.squared, summary(m3)$r.squared, summary(m4)$r.squared), c(summary(m1)$adj.r.squared, summary(m2)$adj.r.squared, summary(m3)$adj.r.squared, summary(m4)$adj.r.squared))  
  
names(results) <- c( "Model", "Description", "R-Squared", "Adj. R Squared")

Results

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Description | R-Squared | Adj. R Squared |
| Model 1 | Linear | 0.3575255 | 0.3544067 |
| Model 2 | Quadratic | 0.4306040 | 0.4250489 |
| Model 3 | Log-Linear | 0.3439576 | 0.3407729 |
| Model 4 | Log-Quadratic | 0.4657166 | 0.4578595 |

### h. Model selection using AIC and BIC.

#Extract the results of AIC and BIC tests for all models and construct a summary table   
model\_AIC <- AIC(m1,m2,m3,m4)  
  
model\_BIC <- BIC(m1,m2,m3,m4)  
  
model\_select <- data.frame(c("Model 1", "Model 2", "Model 3", "Model 4"), c("Linear","Quadratic","Log-Linear","Log-Quadratic") , model\_AIC$AIC, model\_BIC$BIC )  
  
names(model\_select) <- c("Model","Description","AIC","BIC")

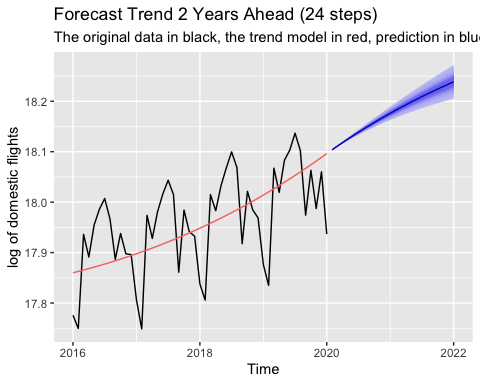
Model Selection With AIC and BIC

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Description | AIC | BIC |
| Model 1 | Linear | 7045.1566 | 7055.1692 |
| Model 2 | Quadratic | 7022.0404 | 7035.3906 |
| Model 3 | Log-Linear | -373.9811 | -363.9685 |
| Model 4 | Log-Quadratic | -412.6833 | -395.9956 |

Using both AIC and BIC and looking at the lowest values we should select the Log-Quadratic Model. (Model 4)

### i. Forecast h-steps ahead

# Create a forecast for Model 4 (The log-quadratic), 24 steps ahead (2 years into the future) and plot it on top of the data and the model  
  
# fit\_m4 is the time series of the moedel we created   
  
fit\_m4 <- ts(m4$fitted.values, start = c(2002,10), frequency = 12)  
  
# Plot it all together  
autoplot(window(ldomestic,2016,2020)) +   
 autolayer(window(fit\_m4,2016,2020)) +   
 autolayer(forecast(window(fit\_m4,2016,2020),h=24, level = c(80,95), fan=TRUE)) +  
 ggtitle("Forecast Trend 2 Years Ahead (24 steps)", "The original data in black, the trend model in red, prediction in blue") +   
 ylab("log of domestic flights") + theme(legend.position = "none")



## 2. Modeling and Forecasting Seasonality

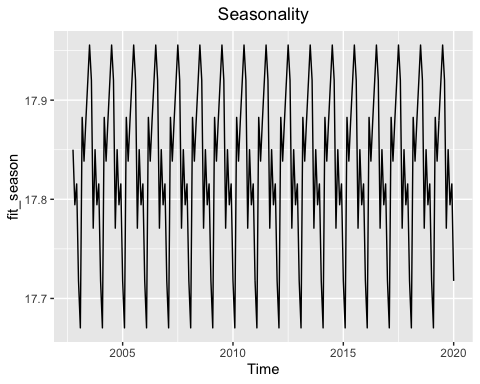
### a. Construct and Test a Model with a Full Set of Seasonal Dummies

# Use TSLM to a model for seasonality using seasonal indicator varaibales   
mseason <- tslm(ldomestic ~ season)  
  
summary(mseason)

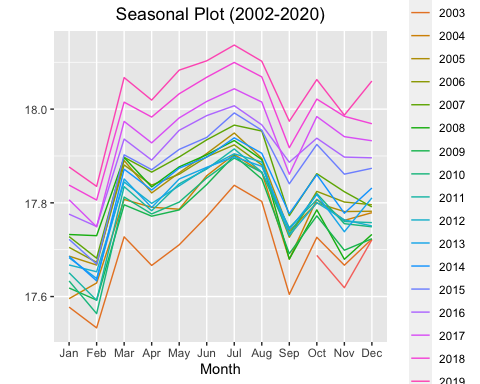
##   
## Call:  
## tslm(formula = ldomestic ~ season)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.17566 -0.05508 -0.01969 0.05192 0.24453   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 17.71769 0.02136 829.515 < 0.0000000000000002 \*\*\*  
## season2 -0.04736 0.03065 -1.545 0.123877   
## season3 0.16491 0.03065 5.381 0.000000210290026 \*\*\*  
## season4 0.12083 0.03065 3.943 0.000112 \*\*\*  
## season5 0.16179 0.03065 5.279 0.000000342953395 \*\*\*  
## season6 0.19960 0.03065 6.513 0.000000000605913 \*\*\*  
## season7 0.23787 0.03065 7.762 0.000000000000453 \*\*\*  
## season8 0.20291 0.03065 6.621 0.000000000334381 \*\*\*  
## season9 0.05321 0.03065 1.736 0.084086 .   
## season10 0.13236 0.03021 4.382 0.000019155044797 \*\*\*  
## season11 0.07682 0.03021 2.543 0.011761 \*   
## season12 0.09780 0.03021 3.238 0.001415 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09062 on 196 degrees of freedom  
## Multiple R-squared: 0.4612, Adjusted R-squared: 0.431   
## F-statistic: 15.25 on 11 and 196 DF, p-value: < 0.00000000000000022

### b. Plot the estimated seasonal factors and interpret your plot

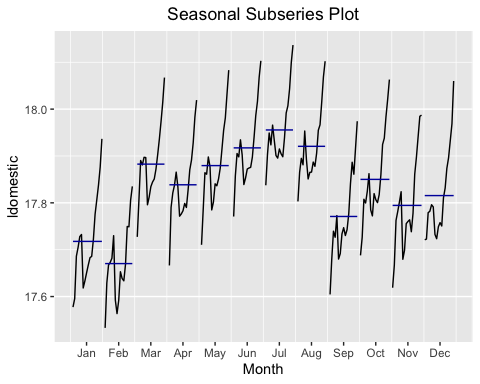
# fit\_season is the seasnaol model fitted values  
fit\_season <- mseason$fitted.values  
  
#Plot the seasonal model  
autoplot(fit\_season) + ggtitle("Seasonality") + theme(plot.title = element\_text(hjust = 0.5))



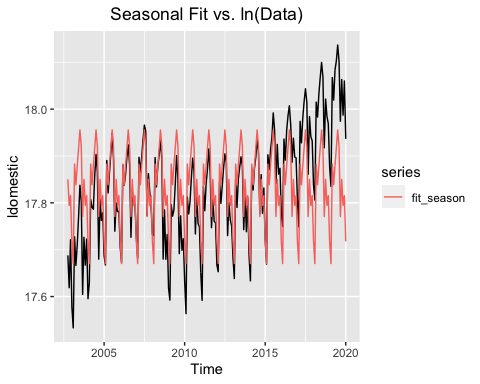
# Seasonality Plot allows us to inspect the seasonality behavior and compare it over time  
ggseasonplot(ldomestic)+ ggtitle("Seasonal Plot (2002-2020)") + theme(plot.title = element\_text(hjust = 0.5))



# Devide the data into subseries to further inspect seasonality behavior



# Plot the data series against the seasonal model we built   
autoplot(ldomestic) +   
 autolayer(fit\_season)+ ggtitle("Seasonal Fit vs. ln(Data)") + theme(plot.title = element\_text(hjust = 0.5))



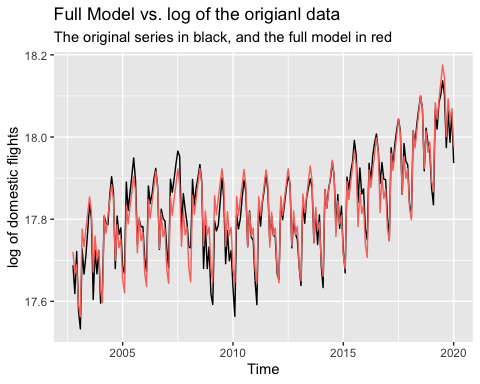
Viewing the Seasonality plot reaveals that there is clearly seasonality occuring from year to year with the peaks occuring in the middle of the year and the troughts occuring at the end/begining of each year.

Viewing the seasonality of a given year and the seasonal plot, we can see this pattern as domestic flight passengers are low at the beginning of the year in January. Then we can see a small spike around March, as peaple travel for spring, and there is a continued increase until July where where there is a peak, this makes sense intuitively as many people make travel plans in the spring and summer months. We then see a dip around september and another small spike around October and December. The spike in October could be due to cooler weather in some areas and lower fares causing an increase in travel. The december spike is likely due to increased travel during the Holidays.

Viewing the seasonal fit vs. the natural log of the actual data, we can see that the seasonl fit closely matches the actaul data.

### c. Add the trend model to improve the plot = Full Model

# Create a full model: log-quadratic + seasonal indicator variables  
mfull <- lm(ldomestic ~ t + I(t^2) + I(t^5) + seasonaldummy(ldomestic))  
  
# fit\_full is the fitted values of the full model as time series object  
fit\_full <- ts(mfull$fitted.values, start = c(2002,10), frequency = 12)  
  
  
# Plot the data against the full model  
  
autoplot(ldomestic) +   
 autolayer(fit\_full) +   
 ggtitle("Full Model vs. log of the origianl data", subtitle = "The original series in black, and the full model in red") + theme(legend.position = "none") +  
 ylab("log of domestic flights")



### d. Interpret the summary statistics (including error metrics) of the full model

summary(mfull) # summary of full model, trend and seasonal models

##   
## Call:  
## lm(formula = ldomestic ~ t + I(t^2) + I(t^5) + seasonaldummy(ldomestic))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.082578 -0.021058 0.000688 0.023243 0.082713   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) -1491894.9803490275517106 116649.2953928006318165  
## t 1855.6989478598677579 144.9866962996907489  
## I(t^2) -0.6155171176413329 0.0480553279222568  
## I(t^5) 0.0000000000075807 0.0000000000005905  
## seasonaldummy(ldomestic)Jan -0.0998867050106245 0.0112250616554621  
## seasonaldummy(ldomestic)Feb -0.1337869877344861 0.0113957398851691  
## seasonaldummy(ldomestic)Mar 0.0766852921766170 0.0113944629960406  
## seasonaldummy(ldomestic)Apr 0.0307977473437594 0.0113935372437285  
## seasonaldummy(ldomestic)May 0.0699277185417606 0.0113929583378981  
## seasonaldummy(ldomestic)Jun 0.1058950495803436 0.0113927240622482  
## seasonaldummy(ldomestic)Jul 0.1422967104876072 0.0113928342705071  
## seasonaldummy(ldomestic)Aug 0.1054530838211304 0.0113932908873417  
## seasonaldummy(ldomestic)Sep -0.0461583993270389 0.0113940979141987  
## seasonaldummy(ldomestic)Oct 0.0386837087248386 0.0112259320467190  
## seasonaldummy(ldomestic)Nov -0.0189118686836011 0.0112250612517121  
## t value Pr(>|t|)   
## (Intercept) -12.790 < 0.0000000000000002 \*\*\*  
## t 12.799 < 0.0000000000000002 \*\*\*  
## I(t^2) -12.809 < 0.0000000000000002 \*\*\*  
## I(t^5) 12.837 < 0.0000000000000002 \*\*\*  
## seasonaldummy(ldomestic)Jan -8.899 0.000000000000000411 \*\*\*  
## seasonaldummy(ldomestic)Feb -11.740 < 0.0000000000000002 \*\*\*  
## seasonaldummy(ldomestic)Mar 6.730 0.000000000188388797 \*\*\*  
## seasonaldummy(ldomestic)Apr 2.703 0.007483 \*\*   
## seasonaldummy(ldomestic)May 6.138 0.000000004661204497 \*\*\*  
## seasonaldummy(ldomestic)Jun 9.295 < 0.0000000000000002 \*\*\*  
## seasonaldummy(ldomestic)Jul 12.490 < 0.0000000000000002 \*\*\*  
## seasonaldummy(ldomestic)Aug 9.256 < 0.0000000000000002 \*\*\*  
## seasonaldummy(ldomestic)Sep -4.051 0.000073824180053341 \*\*\*  
## seasonaldummy(ldomestic)Oct 3.446 0.000698 \*\*\*  
## seasonaldummy(ldomestic)Nov -1.685 0.093645 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.03367 on 193 degrees of freedom  
## Multiple R-squared: 0.9267, Adjusted R-squared: 0.9214   
## F-statistic: 174.4 on 14 and 193 DF, p-value: < 0.00000000000000022

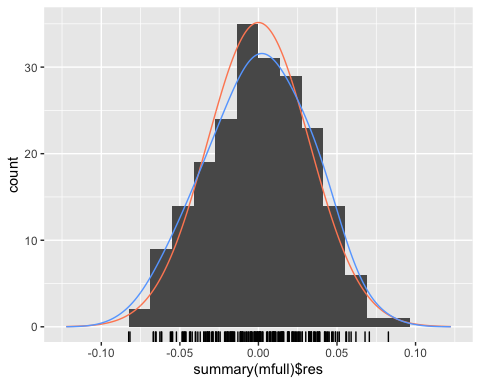
AIC(m4, mseason, mfull) # AIC of full model for model selection

## df AIC  
## m4 5 -412.6833  
## mseason 13 -394.9361  
## mfull 16 -803.9542

BIC(m4, mseason, mfull) # BIC of full model for model selection

## df BIC  
## m4 5 -395.9956  
## mseason 13 -351.5481  
## mfull 16 -750.5536

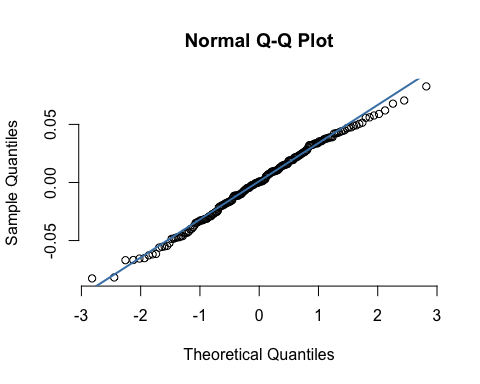
gghistogram(summary(mfull)$res, add.normal = TRUE, add.kde = TRUE) # histogram with normal dist and kernal density



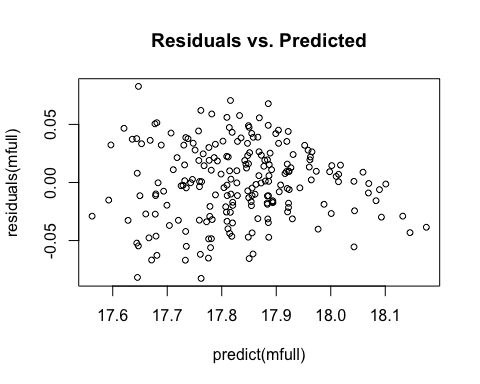
shapiro.test(mfull$residuals) # shapiro test for normality

##   
## Shapiro-Wilk normality test  
##   
## data: mfull$residuals  
## W = 0.9938, p-value = 0.5407

qqnorm(mfull$residuals, pch = 1, frame = FALSE) # normal QQ test for normal distribution  
qqline(mfull$residuals, col = "steelblue", lwd = 2)



plot(predict(mfull), residuals(mfull), cex=0.8, main="Residuals vs. Predicted") # Residuals vs Predicted for full model



Looking at the summary statistics of the full model which is the trend model with the seasonal indicator variables, all of the explanatories have low p-values and high t-values. The F-stat is a large number with a very low p-value indicating that the full model has overall significance. But, a very important statistic seen here is an exceptionally high R-Squared and Adjusted R-Squared which is at 0.9267 and 0.9214, respectively. These values are almost double than without the trend model.

Looking at the values of AIC and BIC , the full model has the lowest values so the full model is the best model for selection.

Analyzing the histogram, we see the residuals have a normal distribution and with a further test, the Shapiro-Test confirms that the residuals are Normally Distributed with a W of 0.99 and a high p-value - we can assume Normality.

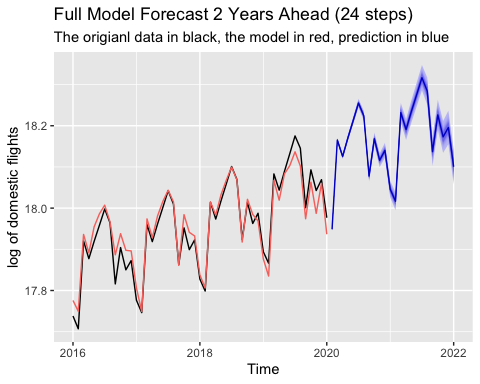
The normal QQ plot confirms are analysis that the residuals are normally distributed.

The Residuals vs Predicted plot shows a more random distribution of the values centering around the middle where residuals equal to 0.

After the analysis of the statistics, we can use the full model for forecasting.

### e. Forecast h-steps ahead using the full model

forecast(window(fit\_full,2016,2020), h=24, fan=TRUE) %>% autoplot() +   
 autolayer(window(ldomestic,2016,2020)) +   
 ggtitle("Full Model Forecast 2 Years Ahead (24 steps)", subtitle = "The origianl data in black, the model in red, prediction in blue") + ylab("log of domestic flights") +  
 theme(legend.position = "none")



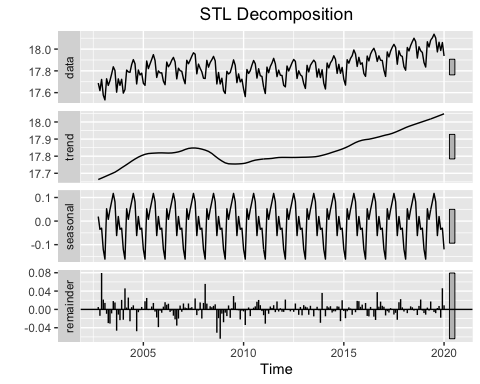
# III. Conclusion:

Our forecast shows a strong upward trend and seasonality using the full model. We believe it is an accurate forecast for the next two years regarding U.S Domestic Flight Air Passengers. What we can do to improve the model is to include data from February 2020 and March 2020 because our data stops at January 2020. The data from February and March would include the decrease of passengers from the current global pandemic from COVID-19 and we would be able to forecast the impact of the pandemic relating to flight passengers.

Additionally, this model does not take into account business cycle behavior and could be improved further by regarding cyclical effects. Looking at the original series, we see a trough around 2007-2008, which would make sense due to the financial crash of that period.

We use STL decomposition to further inspect the behavior of the data:

# STL decomposition  
stl(ldomestic, s.window = "periodic", robust = TRUE) %>% autoplot() + ggtitle("STL Decomposition") + theme(plot.title = element\_text(hjust = 0.5))

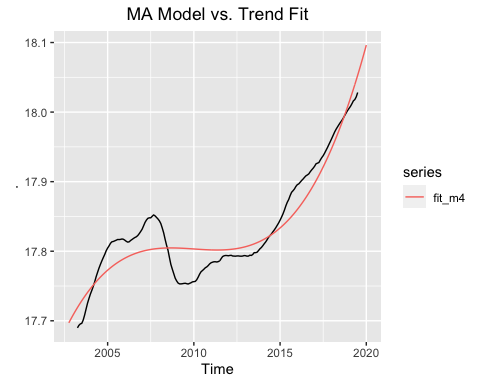


The STL decomposition shows structure in the remainder, indicating that there is more than just the trend activity and seasonal activity. This shows further evidence that there is cylical behavior that is unexplained by either the trend behavior or the seasonal behavior.

We can further inspect behavior of the trend model to fortify this cyclical behavior, by inspecting our trend model against seasonally adjusted data.

Using Moving Average smoothing to remove seasonality, we get this plot:

# Plot MA(12) Smoothed data against our trend model  
ma(ldomestic, 12) %>% autoplot() +   
 autolayer(fit\_m4) + ggtitle("MA Model vs. Trend Fit") + theme(plot.title = element\_text(hjust = 0.5))



This plot shows that even though our trend model is doing a pretty good job in modeling the trend behavior, there is still cyclical behavior that is not completely explained by our models.

Future work should inspect the use of MA and AR models to further the prediction of the overall data. Additionally, there is room to transform the data series into log-difference in order to inspect other behaviors in the percent-change level of the data, where more complex models can be fitted (namely, MA and AR models).

# IV. References

Data Set of Monthly Air Passengers of Domestic Flights in the United States from the Bureau of Transportation Statistics T-100 Market data.

<https://www.transtats.bts.gov/Data_Elements.aspx?Data=1>

# V. R Source Code

# remove and clear previous sessions  
rm(list = ls(all = TRUE))  
  
# load necessary libraries   
  
library(tseries)  
library(zoo)  
library(quantmod)  
library(XML)  
library(readxl)  
library(dplyr)  
library(ggplot2)  
library(ggfortify)  
library(fpp2)  
library(psych)  
library(dynlm)  
library(tis)  
library(lubridate)  
require("datasets")  
library(timeSeries)  
library(gridExtra)  
library(stats)  
library(seasonal)  
library(dplyr)  
library(scales)  
library(readxl)  
  
  
# Read the excel file into a data frame (remove the first row and use the next one as headers)  
passengers\_df <- read\_excel("Airline Passengers.xlsx", skip = 1, col\_names = TRUE, col\_types = "numeric")  
  
# Filter and clean up the data so we only see the relevant information  
passengers\_df <- passengers\_df %>%  
 dplyr::filter(Month > 0)  
  
# Inspect the data  
head(passengers\_df, 12) # first data point 10/2002, monthly frequency  
  
# Create time series variable for domestic flights  
domestic\_ts <- ts(passengers\_df$DOMESTIC, start = c(2002,10), freq=12)  
  
str(domestic\_ts)  
  
#We will focus on the domestic flights data for this project  
  
### a. Show the time-series plot of your data   
  
# Plotting the data - only focusing on domestic flights for the entire time frame (2002-2020)  
autoplot(domestic\_ts) +   
 ggtitle("Monthly Domestic U.S. Air Passengers Data") +   
 ylab("Passengers") + xlab("Year") +   
 theme(plot.title = element\_text(hjust = 0.5))  
  
###c. Plot and discuss the ACF and PACF of your data.   
  
# Side by side view of the ACF and PACF   
par(mfrow=c(2,1))  
ggAcf(domestic\_ts) + ggtitle("Passengers - ACF") + theme(plot.title = element\_text(hjust = 0.5))   
ggPacf(domestic\_ts) + ggtitle("Passengers - PACF") + theme(plot.title = element\_text(hjust = 0.5))   
  
### d. Fit a Linear and Nonlinear model.  
  
options(scipen=999)  
# Create t variable  
t<-seq(2002.783, 2020 ,length=length(domestic\_ts))  
  
#Model 1: Simple Linear model  
m1 <- lm(domestic\_ts ~ t)   
summary(m1)  
plot(domestic\_ts, main="Model 1 - Simple Linear model") # plot the main  
lines(t,m1$fitted.values, col="red")+ theme(plot.title = element\_text(hjust = 0.5)) # lines plots a graph over the main  
  
#Model 2: Quadratic  
m2 <- lm(domestic\_ts ~ t + I(t^2))  
summary(m2)  
plot(domestic\_ts, main = "Model 2 - Quadratic model")  
lines(t,m2$fitted.values, col="red")+ theme(plot.title = element\_text(hjust = 0.5))   
  
#Model 3: Log-Linear  
m3 <- lm(log(domestic\_ts) ~ t)  
summary(m3)  
plot(log(domestic\_ts), main = "Model 3 - Log-Linear model")  
lines(t,m3$fitted.values, col="red")+ theme(plot.title = element\_text(hjust = 0.5))   
  
#Model4: Log-Quadratic  
ldomestic <- log(domestic\_ts)  
  
m4 = lm(ldomestic ~ t + I(t^2) + I(t^5))  
summary(m4)  
  
plot(ldomestic,xlab="Year", ylab="Log(Domestic Flights)", main="Model 4 - Log-Quadratic model")  
lines(t, m4$fit,col="red3")+ theme(plot.title = element\_text(hjust = 0.5))   
  
### e. Residuals vs. Fitted plots for All 4 Models.  
  
#Plot the data series against the seasonal model we built   
  
options(scipen=999)  
plot(predict(m1), residuals(m1), cex=0.8, main="Residuals vs. Predicted")  
plot(predict(m2), residuals(m2), cex=0.8, main="Residuals vs. Predicted")  
plot(predict(m3), residuals(m3), cex=0.8, main="Residuals vs. Predicted")  
plot(predict(m4), residuals(m4), cex=0.8, main="Residuals vs. Predicted")  
  
### f. Histograms for All 4 Models  
  
gghistogram(m1$residuals, add.normal = TRUE, add.kde = TRUE , bins = 13) +   
 ggtitle("Histogram of Model 1 Residuals") + xlab("Model 1 Residuals") + ylab(NULL)  
  
shapiro.test(m1$residuals) ## shapiro wilk tests for normality. if W is 1 and p value is higher than 0.05 than normal distribution  
  
gghistogram(m2$residuals, add.normal = TRUE, add.kde = TRUE, bins = 13) +   
 ggtitle("Histogram of Model 2 Residuals") + xlab("Model 2 Residuals") + ylab(NULL)  
  
shapiro.test(m2$residuals)  
  
gghistogram(m3$residuals, add.normal = TRUE, add.kde = TRUE, bins = 14) +   
 ggtitle("Histogram of Model 3 Residuals") + xlab("Model 3 Residuals") + ylab(NULL)  
  
shapiro.test(m3$residuals)  
  
gghistogram(m4$residuals, add.normal = TRUE, add.kde = TRUE, bins = 14) +   
 ggtitle("Histogram of Model 4 Residuals") + xlab("Model 4 Residuals") + ylab(NULL)  
  
shapiro.test(m4$residuals)  
  
### g. Model Diagnostics Statistics (R^2, t-dist, f-dist).  
  
##### Linear Model  
  
summary(m1)  
  
##### Quadratic Model  
  
summary(m2)  
  
##### Log - Linear Model  
  
summary(m3)  
  
##### Log Quadratic Model  
  
summary(m4)  
  
##### Data Frame of all of the R-Squared and Adjusted R-Squared  
  
#Gather the R suqared and Adjusted R squared results into a a data frame in order to consturct a table   
  
results <- data.frame(c("Model 1", "Model 2", "Model 3", "Model 4"),c("Linear","Quadratic","Log-Linear","Log-Quadratic") ,  
 c(summary(m1)$r.squared, summary(m2)$r.squared, summary(m3)$r.squared, summary(m4)$r.squared),   
 c(summary(m1)$adj.r.squared, summary(m2)$adj.r.squared, summary(m3)$adj.r.squared, summary(m4)$adj.r.squared))  
  
names(results) <- c( "Model", "Description", "R-Squared", "Adj. R Squared") # rename data frame  
  
library(knitr)  
kable(results, caption = "Results") ## creates table for the data frame above  
  
### h. Model selection using AIC and BIC.  
  
#Extract the results of AIC and BIC tests for all models and construct a summary table   
model\_AIC <- AIC(m1,m2,m3,m4)  
  
model\_BIC <- BIC(m1,m2,m3,m4)  
  
model\_select <- data.frame(c("Model 1", "Model 2", "Model 3", "Model 4"), c("Linear","Quadratic","Log-Linear","Log-Quadratic") , model\_AIC$AIC, model\_BIC$BIC )  
  
names(model\_select) <- c("Model","Description","AIC","BIC")  
  
library(knitr)  
kable(model\_select, caption = "Model Selection With AIC and BIC")  
  
### i. Forecast h-steps ahead  
  
# Create a forecast for Model 4 (The log-quadratic), 24 steps ahead (2 years into the future) and plot it on top of the data and the model  
  
# fit\_m4 is the time series of the moedel we created   
  
fit\_m4 <- ts(m4$fitted.values, start = c(2002,10), frequency = 12)  
  
# Plot it all together  
autoplot(window(ldomestic,2016,2020)) +   
 autolayer(window(fit\_m4,2016,2020)) +   
 autolayer(forecast(window(fit\_m4,2016,2020),h=24, level = c(80,95), fan=TRUE)) +  
 ggtitle("Forecast Trend 2 Years Ahead (24 steps)", "The original data in black, the trend model in red, prediction in blue") +   
 ylab("log of domestic flights") + theme(legend.position = "none")   
  
## 2. Modeling and Forecasting Seasonality  
  
### a. Construct and Test a Model with a Full Set of Seasonal Dummies  
  
# Use TSLM to a model for seasonality using seasonal indicator varaibales   
mseason <- tslm(ldomestic ~ season)  
  
summary(mseason)  
  
### b. Plot the estimated and interpret your plot  
  
# fit\_season is the seasnaol model fitted values  
fit\_season <- mseason$fitted.values  
  
#Plot the seasonal model  
autoplot(fit\_season) + ggtitle("Seasonality") + theme(plot.title = element\_text(hjust = 0.5))   
  
# Seasonality Plot allows us to inspect the seasonality behavior and compare it over time  
ggseasonplot(ldomestic)+ ggtitle("Seasonal Plot (2002-2020)") + theme(plot.title = element\_text(hjust = 0.5))   
  
# Divide the data into subseries to further inspect seasonality behavior  
ggsubseriesplot(ldomestic) + ggtitle("Seasonal Subseries Plot") + theme(plot.title = element\_text(hjust = 0.5))   
  
# Plot the data series against the seasonal model we built   
autoplot(ldomestic) +   
 autolayer(fit\_season)+ ggtitle("Seasonal Fit vs. ln(Data)") + theme(plot.title = element\_text(hjust = 0.5))   
  
### c. Add the trend model to improve the plot = Full Model  
  
# Create a full model: log-quadratic + seasonal indicator variables  
mfull <- lm(ldomestic ~ t + I(t^2) + I(t^5) + seasonaldummy(ldomestic))  
  
# fit\_full is the fitted values of the full model as time series object  
fit\_full <- ts(mfull$fitted.values, start = c(2002,10), frequency = 12)  
  
# Plot the data against the full model  
  
autoplot(ldomestic) +   
 autolayer(fit\_full) +   
 ggtitle("Full Model vs. log of the origianl data", subtitle = "The original series in black, and the full model in red")   
+ theme(legend.position = "none") +  
 ylab("log of domestic flights")   
  
### d. Interpret the summary statistics (including error metrics) of the full model  
  
summary(mfull) # summary of full model, trend and seasonal models  
  
AIC(m4, mseason, mfull) # AIC of full model for model selection  
BIC(m4, mseason, mfull) # BIC of full model for model selection  
  
gghistogram(summary(mfull)$res, add.normal = TRUE, add.kde = TRUE) # histogram with normal dist and kernal density   
shapiro.test(mfull$residuals) # shapiro test for normality  
  
qqnorm(mfull$residuals, pch = 1, frame = FALSE) # normal QQ test for normal distribution  
qqline(mfull$residuals, col = "steelblue", lwd = 2)  
  
plot(predict(mfull), residuals(mfull), cex=0.8, main="Residuals vs. Predicted") # Residuals vs Predicted for full model  
  
### e. Forecast h-steps ahead using the full model uaing forecast()  
  
forecast(window(fit\_full,2016,2020), h=24, fan=TRUE) %>% autoplot() +   
 autolayer(window(ldomestic,2016,2020)) +   
 ggtitle("Full Model Forecast 2 Years Ahead (24 steps)", subtitle = "The origianl data in black, the model in red, prediction in blue") + ylab("log of domestic flights") +  
 theme(legend.position = "none")