144_Project_1

Krish Methi, Krithik J, Andrew Brown

1/20/2024

```r
rm(list=ls(all=TRUE))

# Load libraries
library(tseries)
library(ggplot2)
library(forecast)
library(fpp3)
library(tseries)
library(seasonal)
library(fable)
library(stats)
require(graphics)
library(dplyr)
library(tsibble)
library(tsibbledata)
```

Introduction

The time series data we are working with are air traffic passenger statistics from the San Francisco International Airport, reported on a monthly level by airline. Before running analysis on the data, we have combined the data so that by month, the passenger count is totaled to list the total number of passenger arrivals across all airlines. There are two columns, the month, and the total passenger count. Running time series analysis can be impactful in understanding how travel trends have evolved over time, and how travel patterns change seasonally.

1. Modeling and Forecasting Trend

a. Create a Time-series plot of the data

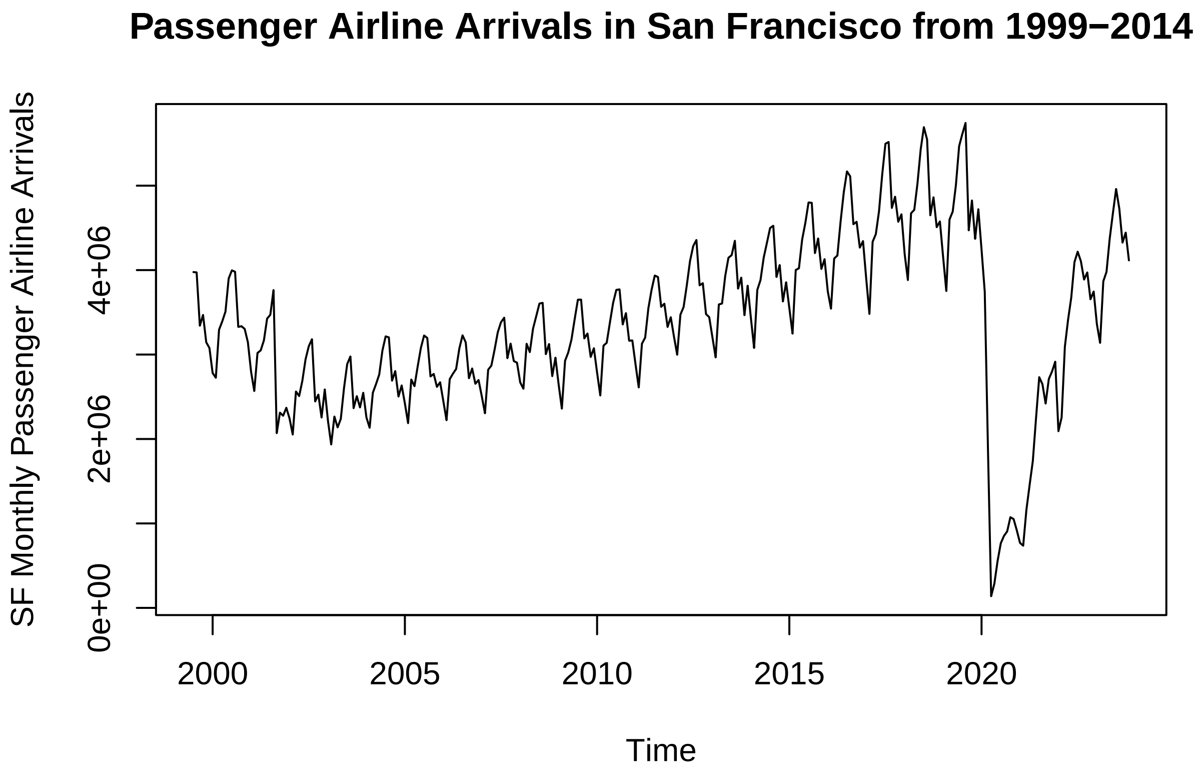
Step 1: Load in the time series and create the tsibble

```r
# Loading in the data and giving it the name 'df'
df <- read.csv("/Users/kmethi2/Downloads/Air_Traffic_Passenger_Statistics_20240120.csv")

# Grouping the data by the month and creating a new column to find total arrivals
df <- df %>%
  mutate(month = format(as.Date(Activity.Period.Start.Date), "%Y-%m")) %>%
  group_by(month) %>%
  summarise(total_passengers = sum(Passenger.Count))
```

```r
# Formatting and arranging the data to fit a monthly timeline chronologically
df <- df %>%
  mutate(month = as.Date(paste0(month, "-01"), format = "%Y-%m-%d")) %>%
  as_tsibble(index = month, key = total_passengers) %>%
  arrange(month)
```

Step 2: Initial time series plot of the data

```r
# Creating a time series of the data from the start point and going by month
ts <- ts(df$total_passengers, start = c(1999,7), frequency = 12)
t<-seq(1999, 2024,length=length(ts))

# Plotting the data accordingly
plot(ts, xlab = "Time", ylab = "SF Monthly Passenger Airline Arrivals",
     main = "Passenger Airline Arrivals in San Francisco from 1999-2014")
```

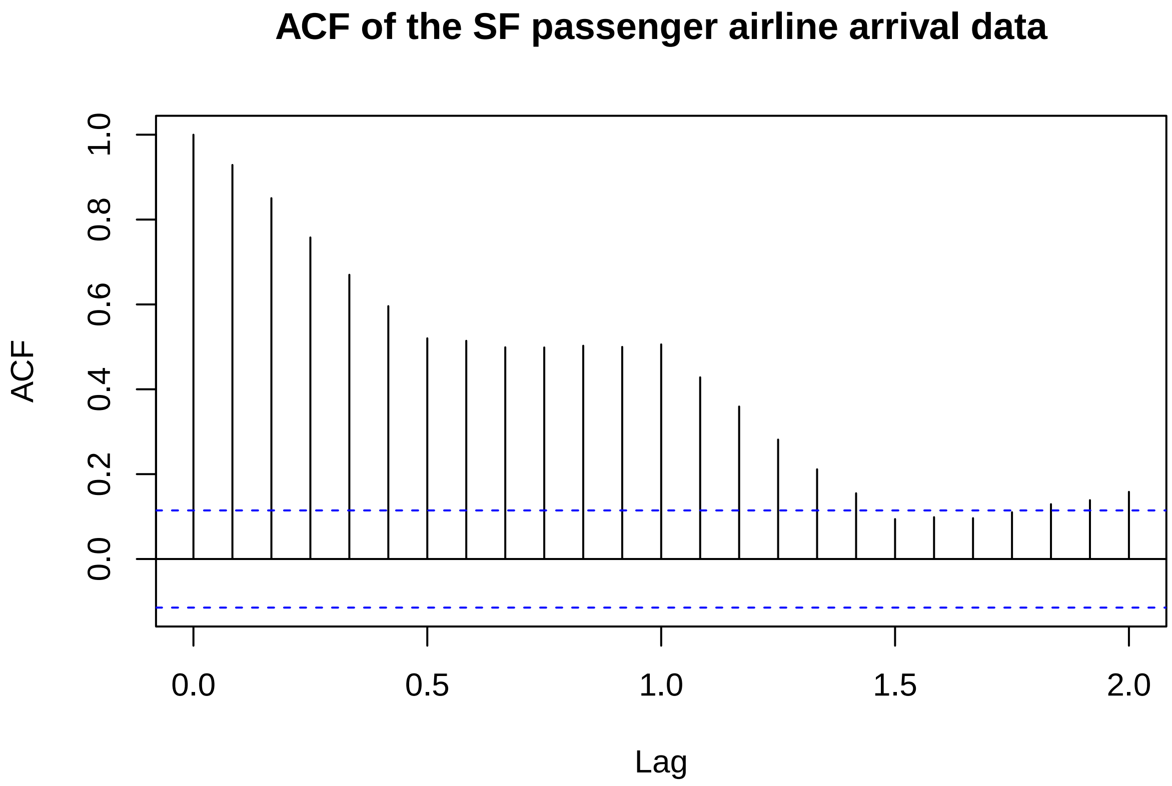


b. Explaining if data is covariance stationary

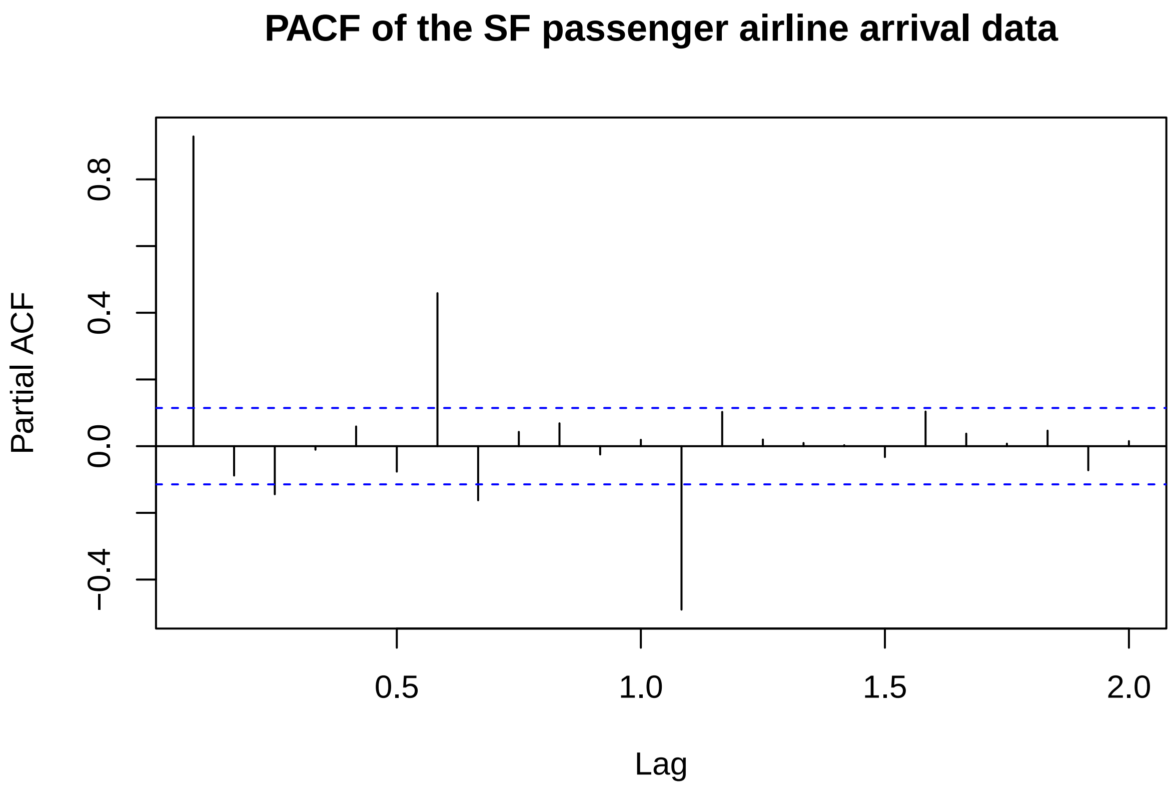
The plot suggests that the data is not covariance stationary as we can see the mean of the data isn't completely constant,there is a slight positive trend, when taking out the COVID pandemic recession and travel ban. The plot is not mean reverting as well.

c. Discussing and plotting the ACF and PACF of the data

```
#Using the ACF and PACF functions to plot the data and discuss
acf(ts, main = "ACF of the SF passenger airline arrival data")
```



```
pacf(ts, main = "PACF of the SF passenger airline arrival data")
```

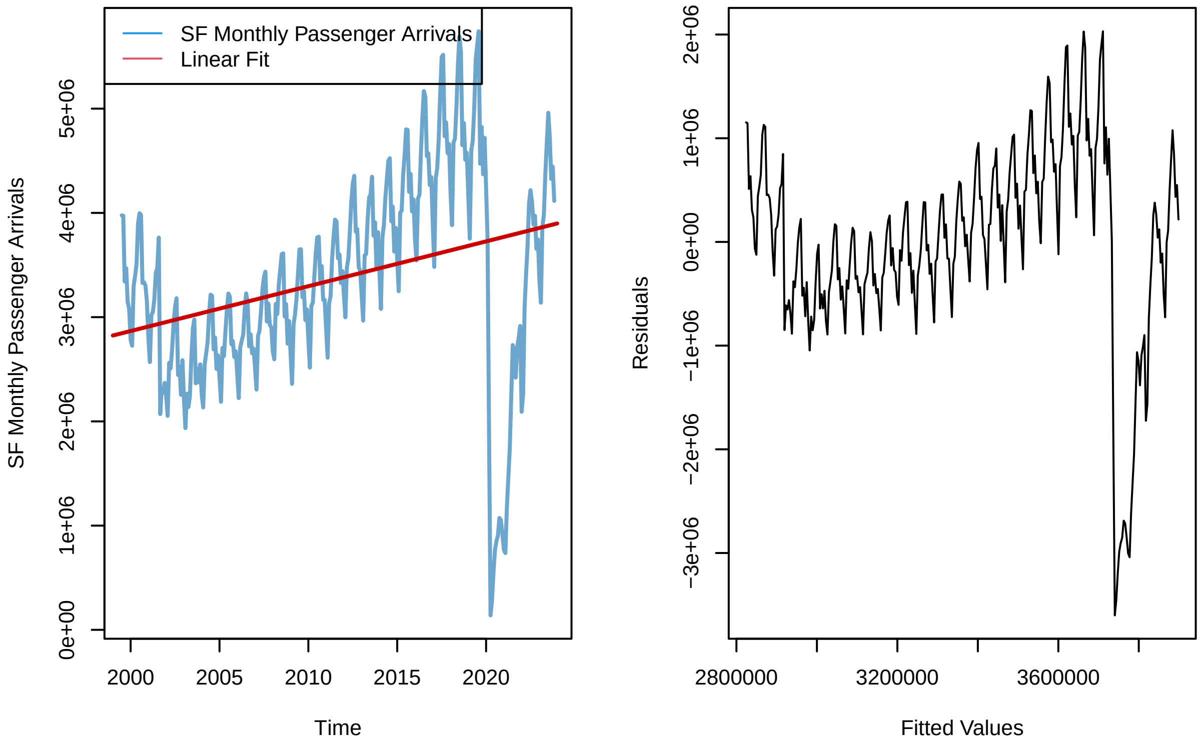


Discussion: First things first, when looking at both the plots, we can see that there are many bars crossing the threshold for both the ACF and the PACF plots. This tells us that there are definitely some dynamics in play in this time series, and that there certainly are plenty of significant lags. Furthermore, we can kind of see a pattern in the plots as well, some indication of cycles in the data. In terms of the difference between the two, we can see that in the PACF, there are lags that don't meet the threshold, however in the ACF, virtually all of the lags clear the threshold. In both plots, there are spikes in the beginning of the plots, and in the PACF, there is a spike just after a 0.5 lag, suggesting an influence of those lags on the total passenger arrivals.

d. and e. Fit a linear and nonlinear model to the series along with residuals and fitted values

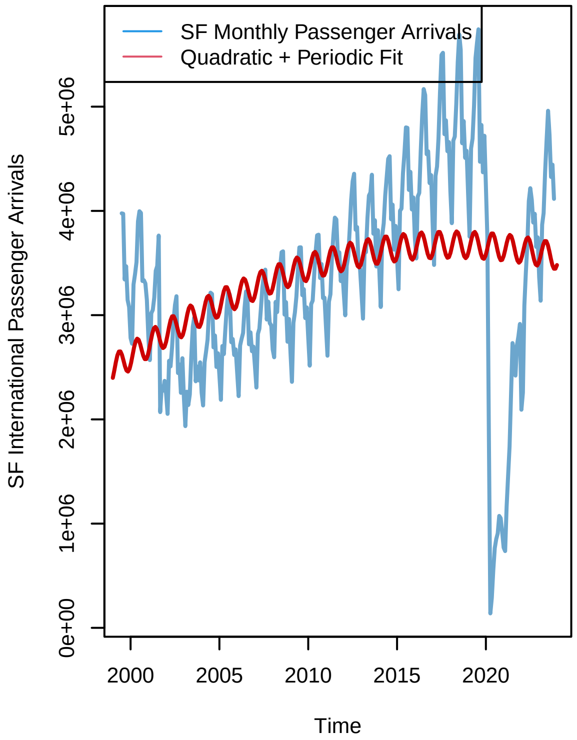
Linear Model

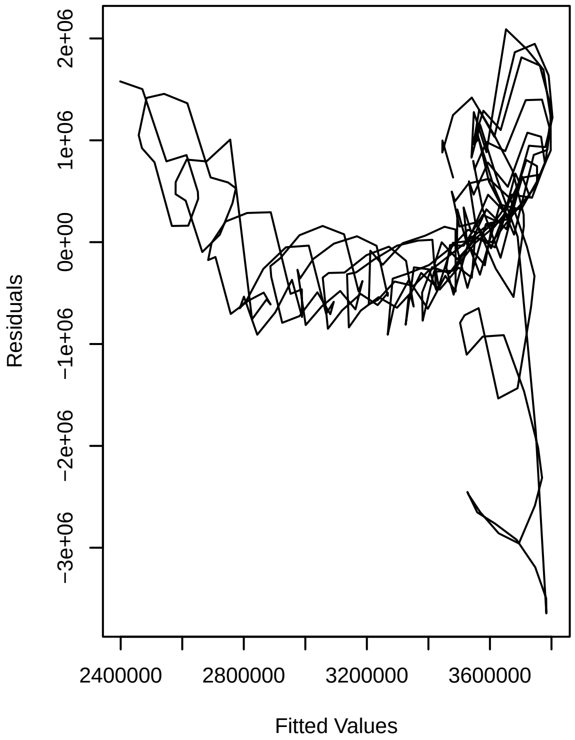
```r
# Using a linear model as stated above with the two variables
m1=lm(ts~t)
layout(matrix(c(1,1,2,2,1,1,2,2,1, 1, 2, 2, 1, 1, 2, 2), nrow = 4, ncol = 4, byrow = TRUE))
plot(ts,ylab="SF Monthly Passenger Arrivals", xlab="Time", lwd=2, col='skyblue3')
lines(t,m1$fit,col="red3",lwd=2)
legend(x = "topleft",
       legend = c("SF Monthly Passenger Arrivals", "Linear Fit"),
       lty = c(1,1),
       col = c(4,2))
# Plotting the residuals of the model on the fitted values
plot(m1$fitted.values,m1$res, ylab="Residuals",type='l',xlab="Fitted Values")
```



Quadratic and Periodic Fit

```
# Similar to the linear model, but instead using a quadratic form for the fit
sin.t<-sin(2*pi*t)
cos.t<-cos(2*pi*t)
t2<-t^2
m2=lm(ts~t+t2+sin.t+cos.t)
layout(matrix(c(1,1,2,2,1,1,2,2,1, 1, 2, 2, 1, 1, 2, 2), nrow = 4, ncol = 4, byrow = TRUE))
plot(ts,ylab="SF International Passenger Arrivals", xlab="Time", lwd=2, col='skyblue3')
lines(t,m2$fit,col="red3",lwd=2)
legend(x = "topleft",
       legend = c("SF Monthly Passenger Arrivals", "Quadratic + Periodic Fit"),
       lty = c(1,1),
       col = c(4,2))
# plotting the residuals of the model on the fitted values
plot(m2$fitted.values,m2$residuals, ylab="Residuals",type='l',xlab="Fitted Values")
```

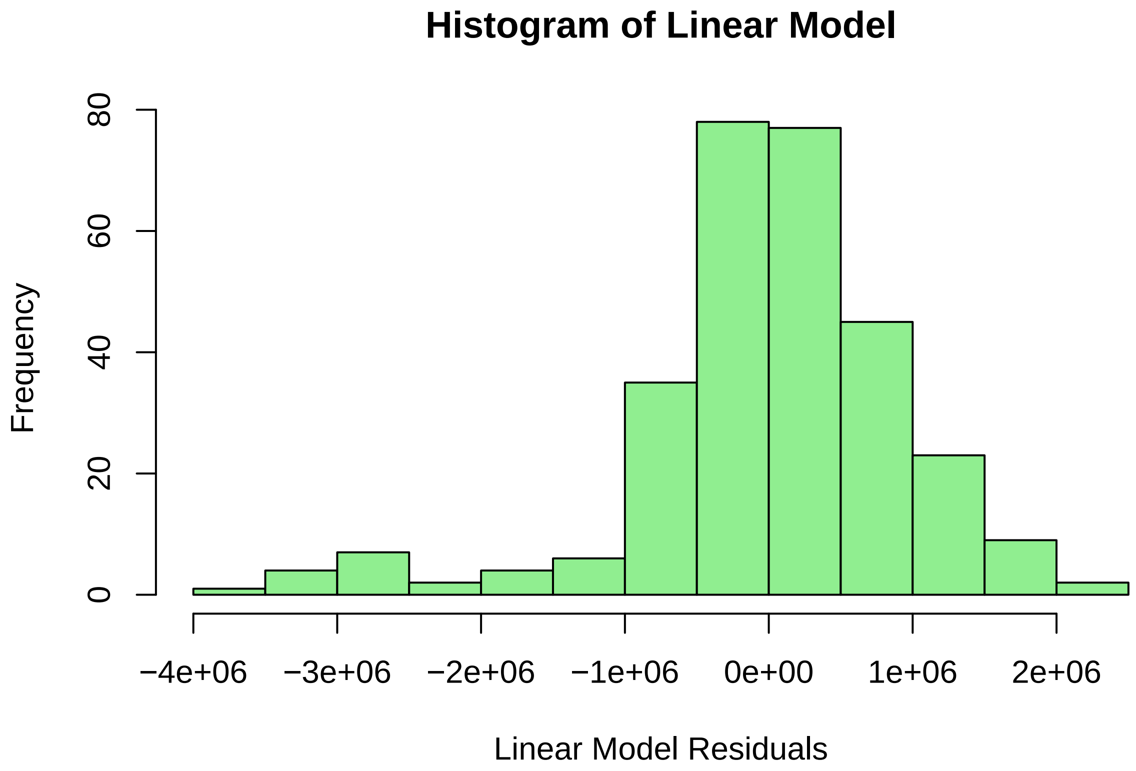




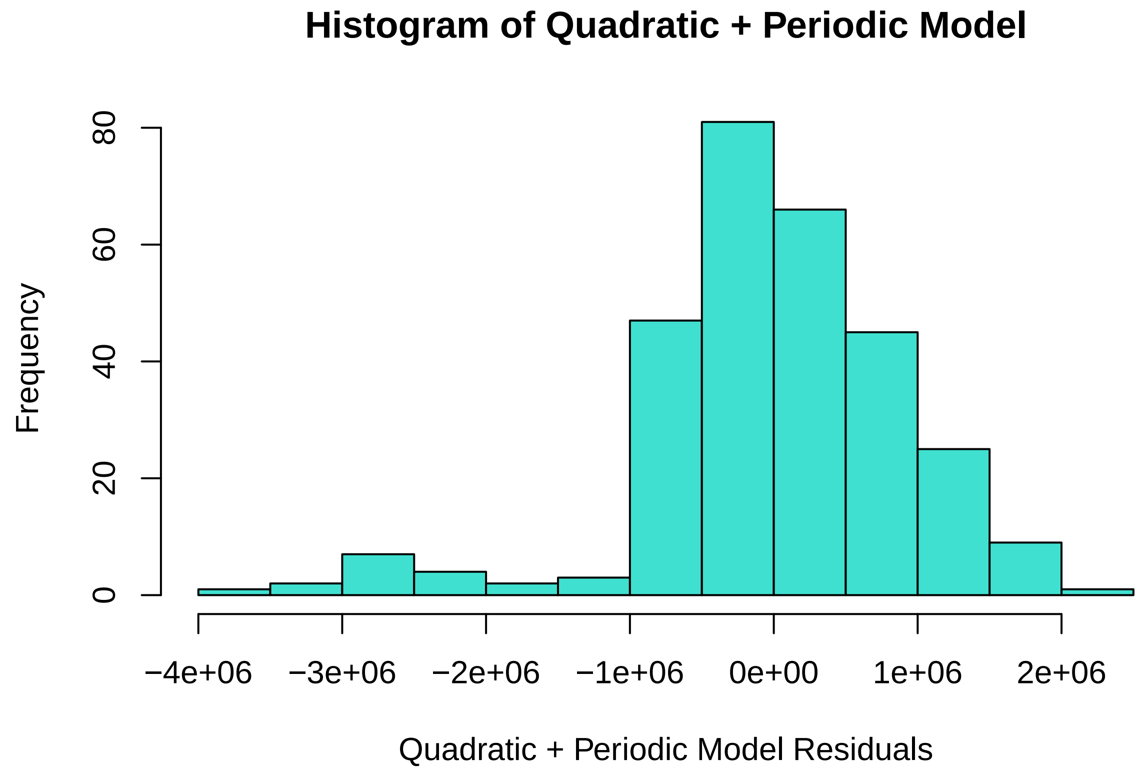
Discussion: Looking at the plots, for both the linear and nonlinear quadratic models, the residuals seem to not follow any pattern as the fitted values increase, signifying that there are no issues in that respect for either model. Both models seem to be relatively good models for the data based on these residual plots. Especially with the quadratic + periodic model, the residual plot is very chaotic, not following any pattern whatsoever

f. Plotting the Histogram of the residuals with discussion

```
# Plotting the histograms of both of the models (linear and quadratic)
hist(m1$residuals,
     main = "Histogram of Linear Model",
     xlab = "Linear Model Residuals",
     col = "lightgreen")
```



```
hist(m2$residuals,
     main = "Histogram of Quadratic + Periodic Model",
     xlab = "Quadratic + Periodic Model Residuals",
     col = "turquoise")
```



Discussion: Looking at the histogram of the residuals of both the the linear and nonlinear models, they are extremely similar, with some minor differences. Both histograms appear to be normally distributed, centered around 0 which is good for the models. There may be some potential outliers towards the lower side of both residual plots as well, and some potential left skewness in both models. Overall, the histogram of the residuals checks out for the most part.

g. Discuss the diagnostic statistics for both models

```
# Creating summaries of both models below
summary(m1)
```

```
##
## Call:
## lm(formula = ts ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3601521  -417312    66150   549003  2031546
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -83118372   15324451  -5.424 1.23e-07 ***
## t               42993       7618   5.643 3.96e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 944300 on 291 degrees of freedom
## Multiple R-squared:  0.09864,    Adjusted R-squared:  0.09555
## F-statistic: 31.85 on 1 and 291 DF,  p-value: 3.958e-08
```

```
summary(m2)
```

```
##
## Call:
## lm(formula = ts ~ t + t2 + sin.t + cos.t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3644472  -468054    -1360   589430  2090189
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.329e+10  4.698e+09  -2.830  0.00499 **
## t            1.318e+07  4.671e+06   2.821  0.00512 **
## t2          -3.265e+03  1.161e+03  -2.812  0.00526 **
## sin.t        9.692e+04  7.717e+04   1.256  0.21018
## cos.t       -8.328e+04  7.688e+04  -1.083  0.27961
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 932000 on 288 degrees of freedom
## Multiple R-squared:  0.131,  Adjusted R-squared:  0.119
## F-statistic: 10.86 on 4 and 288 DF,  p-value: 3.273e-08
```

Discussion: First, looking at the R^2 values, the linear model produces an r^2 and adjusted r^2 of 0.0984 and 0.0955, while the quadratic model produces values of 0.131 and 0.119. Based on this, the quadratic +periodic model is a better fit for the data, as more of the variation in the number of passenger arrivals by month is explained by that model.

Looking at the F, statistic, the linear model produced a value of 31.85, and the quadratic model produced an F statistic of 10.86, with slightly different degrees of freedom. These values both have an associated p value of very very small, showing the significance of both the linear and the quadratic + periodic model. The p value is a little smaller with the quadratic fit, although both are extremely small.

Looking at the T values and the associated p values, in both the linear and quadratic model the p values were very small and significant. Again, this is asserting that both models are significant, and all the predictors are significant. In this case, the linear model has more significant p values from the t distribution.

Overall, from these statistics, we can see that both models are significant. We can also see that in most respects, the quadratic + periodic model has more significant statistics, but they are very similar.

h. Select a trend model from AIC and BIC

```
# Creating the AIC and the BIC from the models
AIC(m1, m2)
```

```
##    df      AIC
## m1  3 8897.819
## m2  6 8893.099
```

```
BIC(m1, m2)
```

```
##    df     BIC
## m1  3 8908.86
## m2  6 8915.18
```

Discussion: According to the AIC and BIC, both models are extremely similar, however the AIC selects M2 as the model with a lower value, but the BIC selects M1 as the model to choose. Logically, this makes sense as the BIC penalizes extra variables more, and in the quadratic + periodic method there are more predictors. Looking at the regression of the quadratic + periodic model as well, we can see that adding the periodic components were not significant, but the quadratic terms were significant. Therefore, we will use the quadratic + periodic model, but take out the periodic component because it is not significant, leaving just the quadratic components

AIC and BIC taking out the periodic trend:

```
# run the only quadratic linear model
m3=lm(ts~t+I(t^2))
# compute the AIC and BIC
AIC(m1, m3)
```

```
##    df      AIC
## m1  3 8897.819
## m3  4 8891.884
```

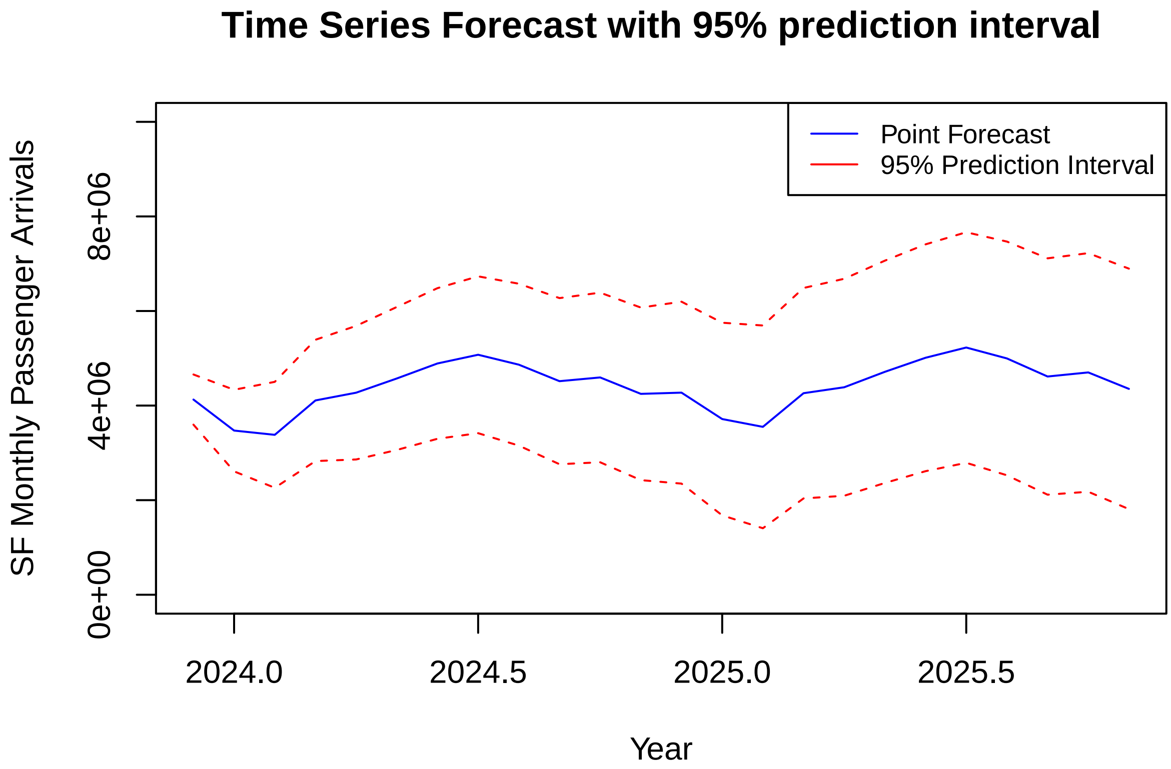
```
BIC(m1, m3)
```

```
##    df      BIC
## m1  3 8908.860
## m3  4 8906.604
```

Now we can see in this case, both the AIC and BIC agree with the quadratic model over the linear model, supporting what we had already discussed.

i. Use the preferred model to forecast h-steps ahead with prediction interval

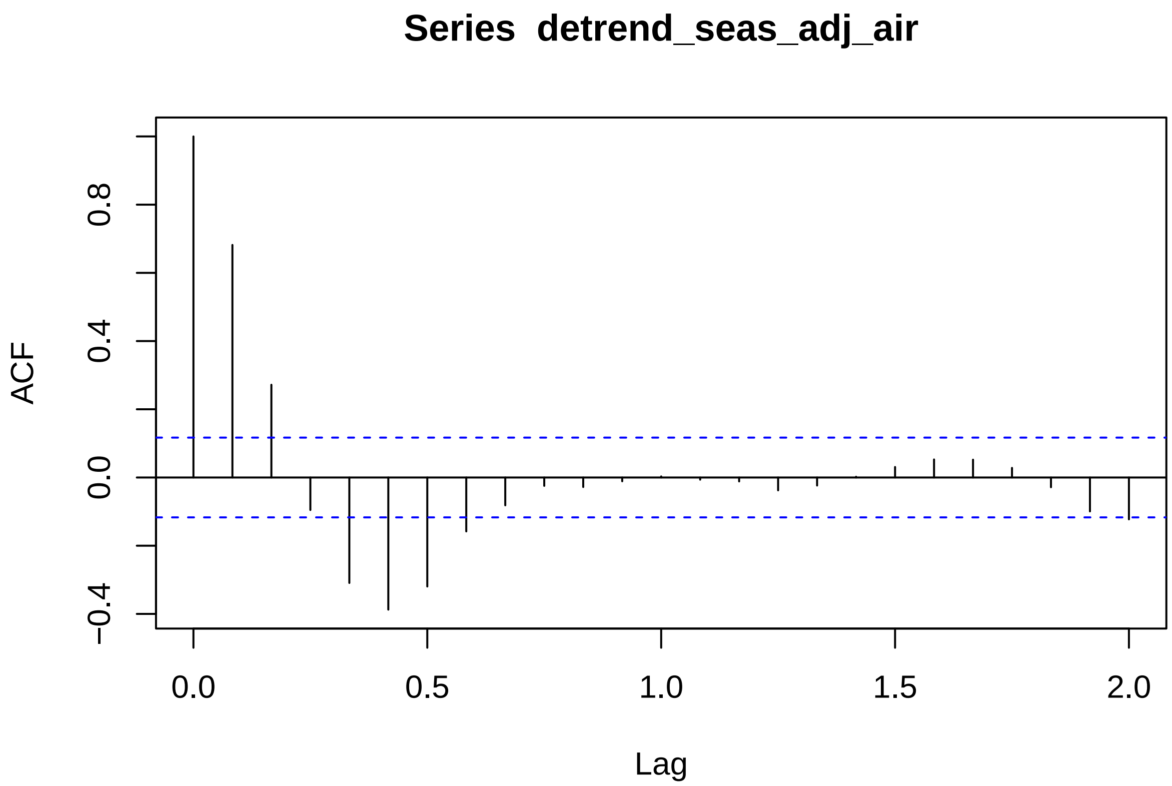
```
forecast_model <- auto.arima(ts)
forecast_result <- forecast(forecast_model, h = 24, level = 95)
plot(forecast_result[["mean"]], main = "Time Series Forecast with 95% prediction interval", xlab = "Yea
lines(forecast_result$lower[, "95%"], col = "red", lty = 2)
lines(forecast_result$upper[, "95%"], col = "red", lty = 2)
legend("topright", legend = c("Point Forecast", "95% Prediction Interval"),
       col = c("blue", "red"), lty = 1:1, cex = 0.8)
```



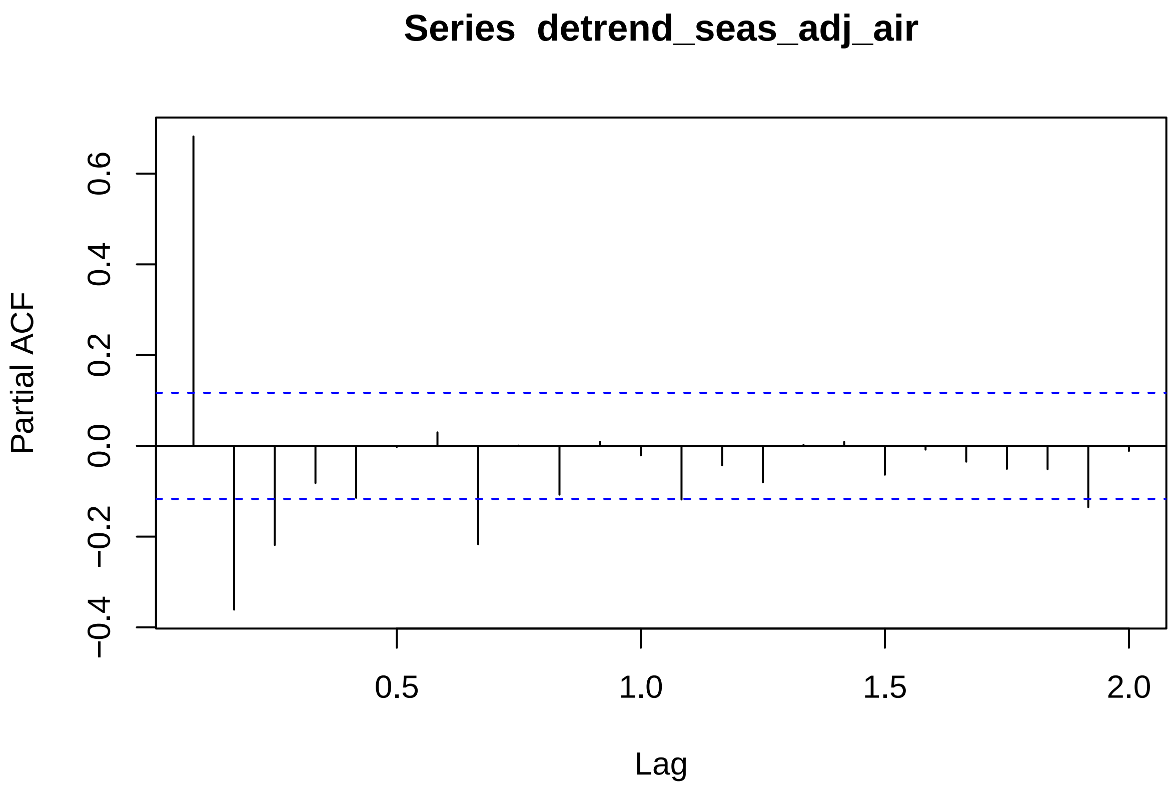
2. Trend and Seasonal Adjustments

a. Perform an additive decomposition of your series. Remove the trend and seasonality, and comment on the ACF ### and PACF of the residuals.

```
# Additive decomposition creation
dcmp_air = decompose(ts(df$total_passengers,frequency=12), "additive")
cost = ts(df$total_passengers,frequency=12)
# Removing trend and seasonality
detrend_seas_adj_air = cost - dcmp_air$trend - dcmp_air$seasonal
# Creating safeguards for NAs and then making ACF and PACF for comment
detrend_seas_adj_air <- na.omit(detrend_seas_adj_air)
acf(detrend_seas_adj_air)
```



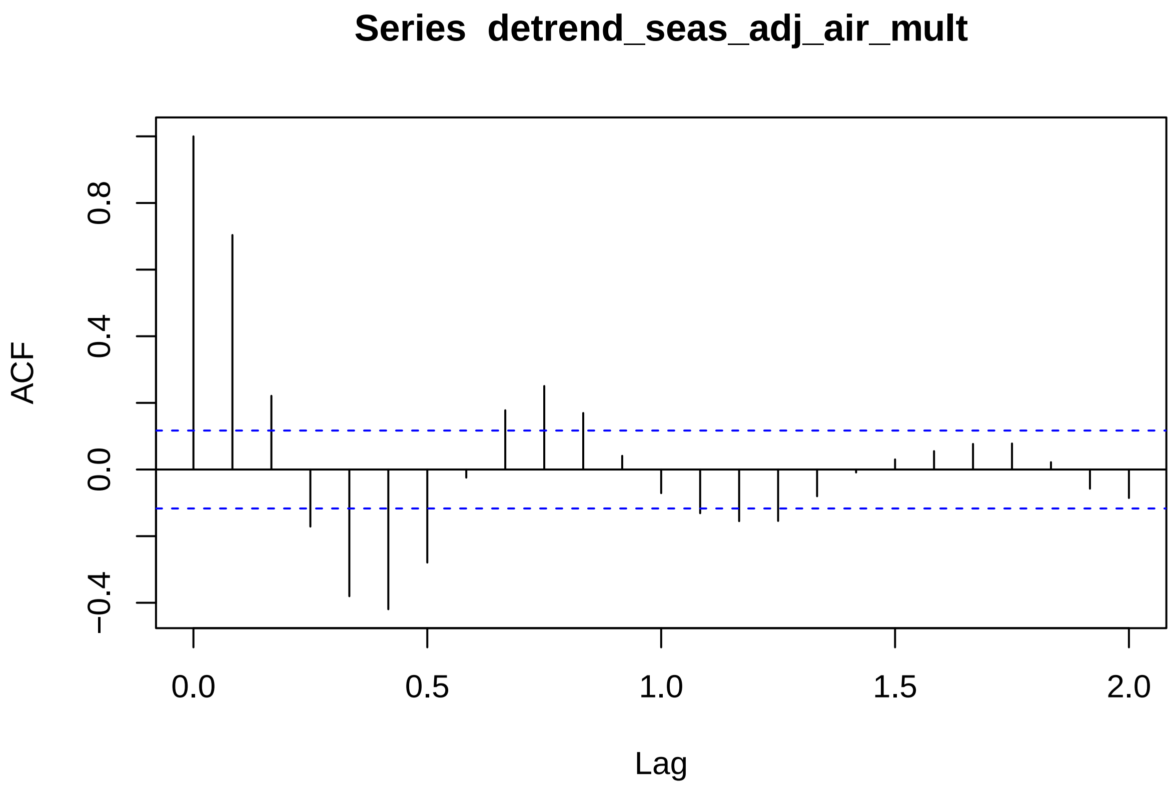
```
pacf(detrend_seas_adj_air)
```



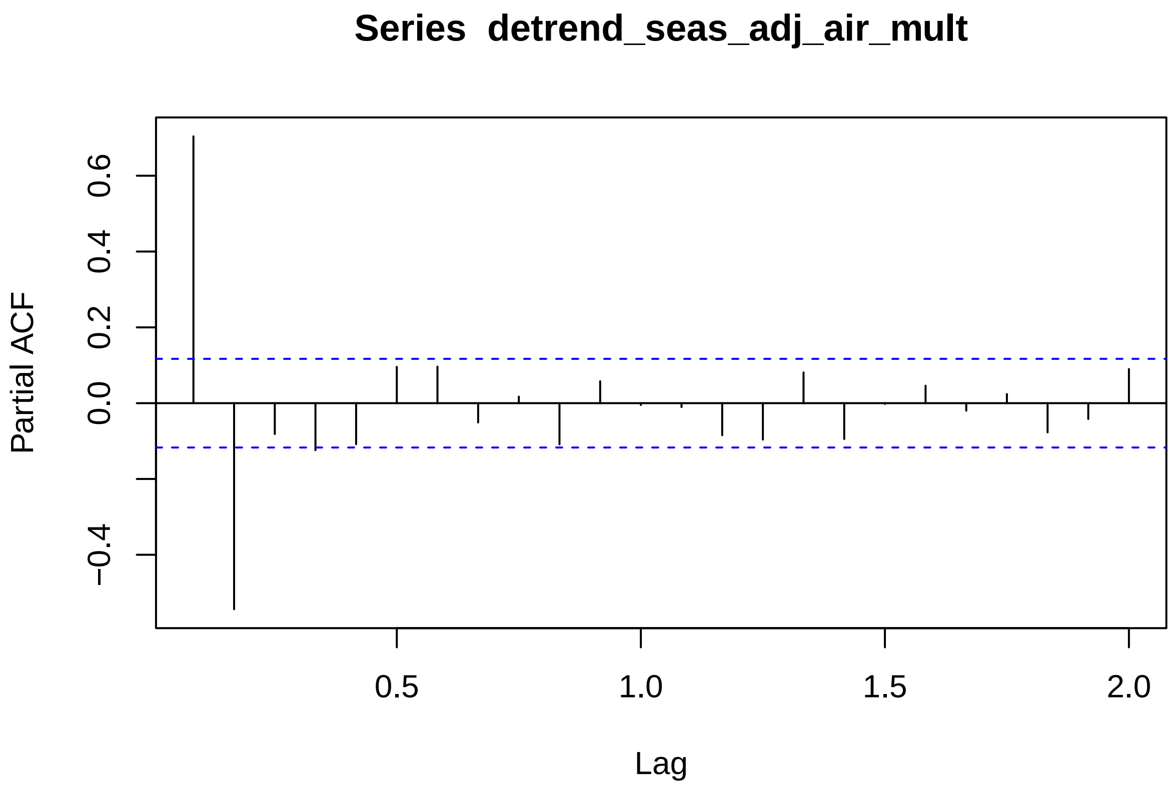
Comment: Both graphs show that there is not white noise and the regular ACF shows a clear pattern. There is evident autocorrelation in this situation.

b. Perform a multiplicative decomposition of your series. Remove the trend and seasonality, and comment on the ### ACF and PACF of the residuals

```
# Creating a multiplicative decomposition
dcmp_air_mult = decompose(ts(df$total_passengers,frequency=12), "multiplicative")
cost = ts(df$total_passengers,frequency=12)
detrend_seas_adj_air_mult = (cost / dcmp_air_mult$trend) / dcmp_air_mult$seasonal
# Similar to above code for additive decomposition
detrend_seas_adj_air_mult <- na.omit(detrend_seas_adj_air_mult)
acf(detrend_seas_adj_air_mult)
```



```
pacf(detrend_seas_adj_air_mult)
```



Comment: The graphs are very similar to the additive ones. In both situations, we see once again that there is not white noise and there is clear autocorrelation. Furthermore, we can see that from both the ACF and the PACF plots that the there are significant lags between 0-0.5 but after that period, the lags do not seem significant. In the ACF plot, more of the lags are significant pretty much all the way up to in between 1-1.5. More importantly, from the ACF, the plot implies that there is some seasonality and cyclical dynamics.

c. Which decomposition is better, additive or multiplicative? Why?

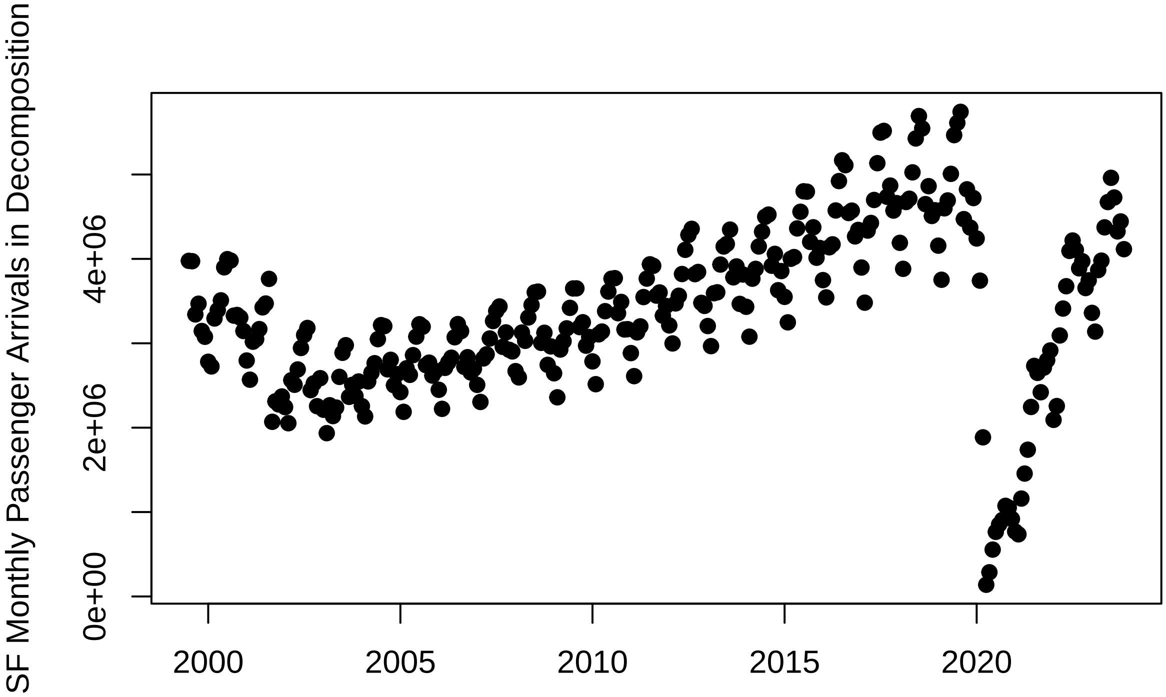
```
total_passengers_trimmed <- window(df$total_passengers, start = start(df$total_passengers) + 6, end = en
# Comparing the mean absolute error of the models to determine which is better
accuracy(detrend_seas_adj_air, total_passengers_trimmed)[1, "MAE"]
```

```
## [1] 3346382
```

```
accuracy(detrend_seas_adj_air_mult, total_passengers_trimmed)[1, "MAE"]
```

```
## [1] 3325416
```

```
# Creating the plot for the decomposition analysis
plot(df$month, df$total_passengers,
     ylab = "SF Monthly Passenger Arrivals in Decomposition",
     xlab = "Time",
     pch = 19)
```

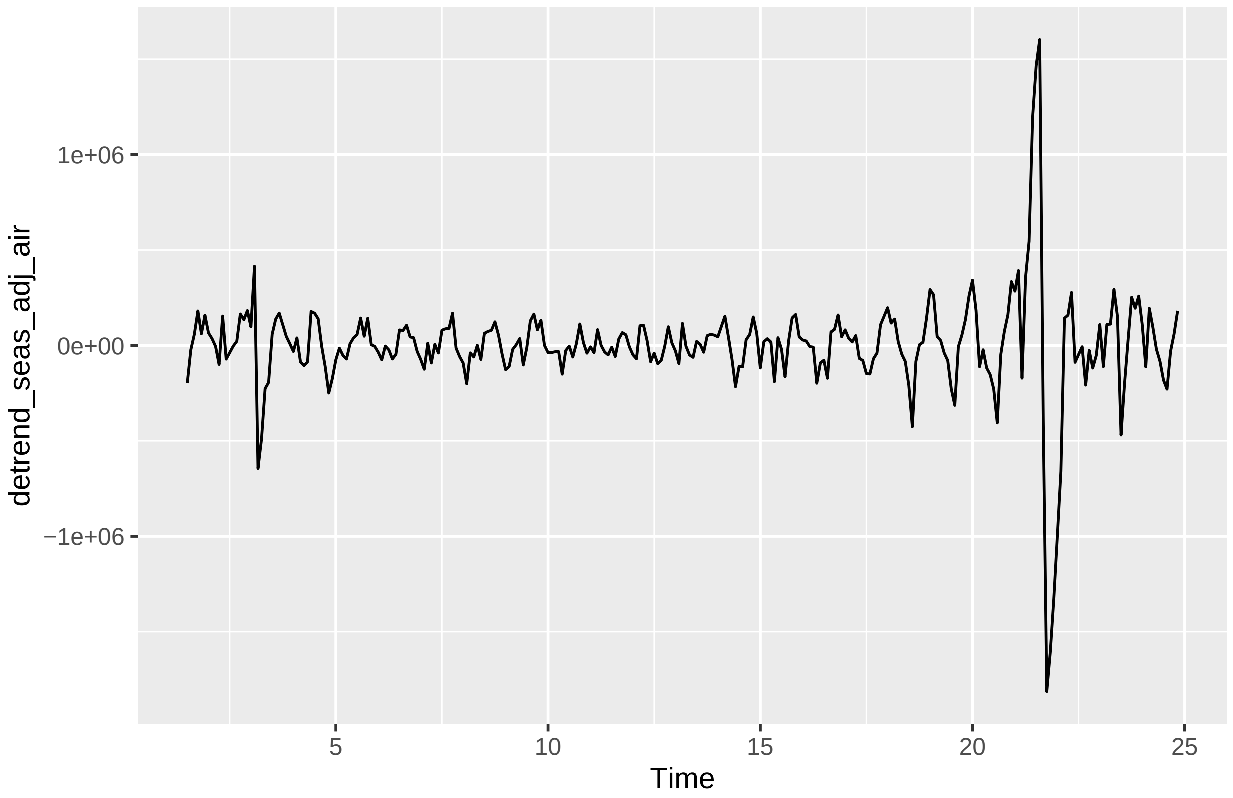


Comment: Based on the MAE (Mean Absolute Error), it seems that the multiplicative model is slightly better than the additive model. Also, when we look at the graph for total passengers, we see that it kind of fans out rather than being constant, and usually, if it fans out, a multiplicative model is the better option.

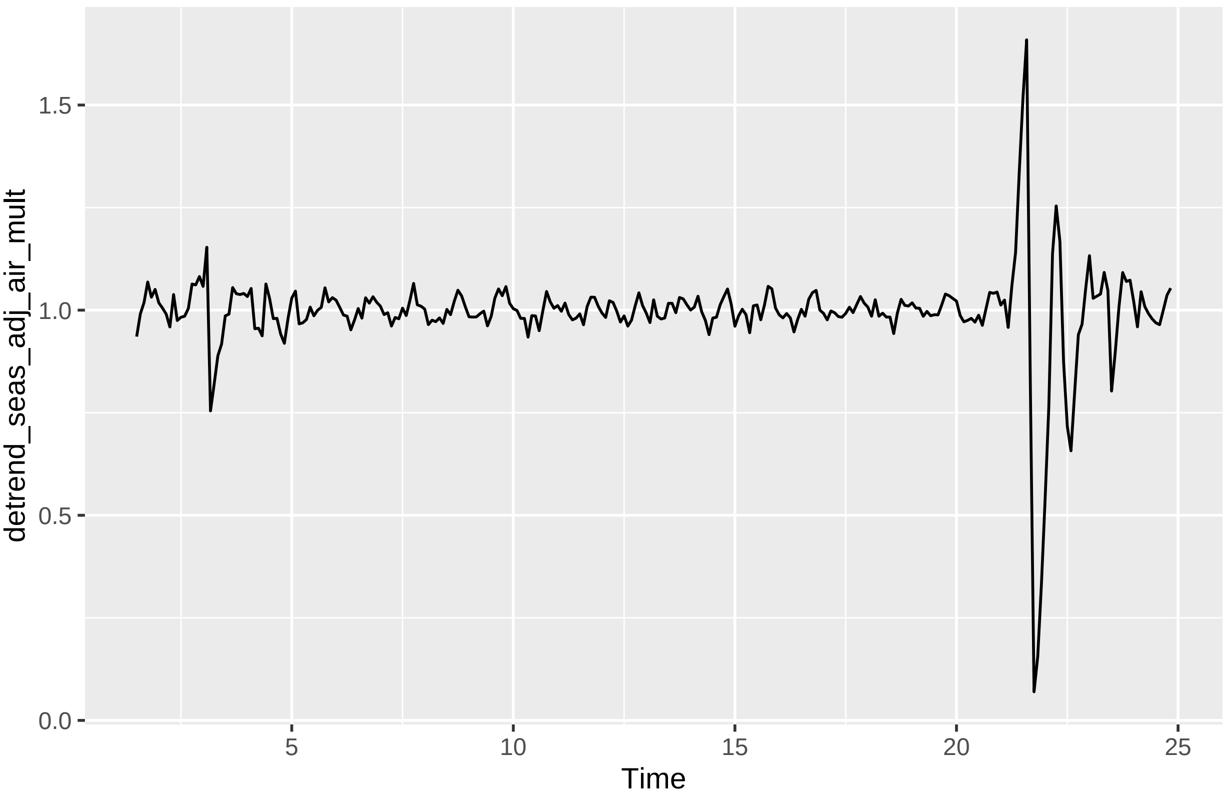
d. Based on the two decompositions, and interpretation of the random components, would your models for the

cycles be similar (additive vs. multiplicative) or very different? Why?

```
# Creating plots of the two models
autoplot(detrend_seas_adj_air)
```



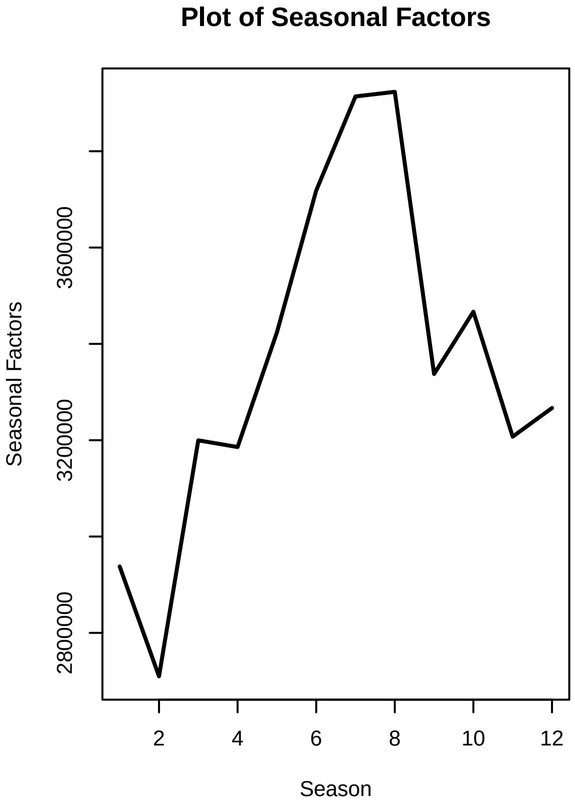
```
autoplot(detrend_seas_adj_air_mult)
```

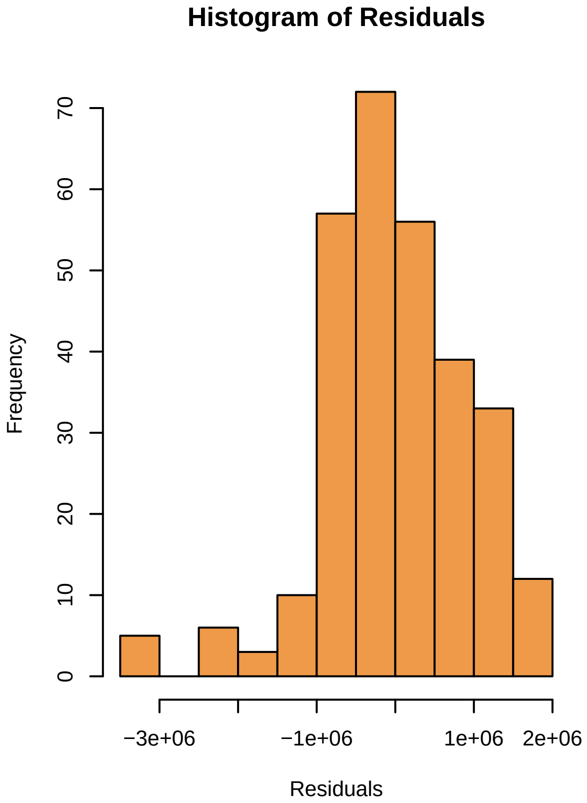


Comment: When looking at the autoplot of the additive model, we see that there's a lot of fluctuation at the start and end of the year, and it's fairly stable around the middle, so it gives a good idea on what a cycle would look like. While the second model is fairly different, it does have a similar pattern. For that reason, while the models for cyclicity wouldn't be exactly the same, it would make sense that they are pretty similar.

e. Plot the seasonal factors and comment

```
# Creating a seasonal model
seasonal_fit <- tslm(ts~season+0)
layout(matrix(c(1,1,2,2,1,1,2,2,1, 1, 2, 2, 1, 1, 2, 2), nrow = 4, ncol = 4, byrow = TRUE))
# Plotting the model along with the histogram of residuals
plot(seasonal_fit$coef,type='l',ylab='Seasonal Factors', xlab="Season",lwd=2, main="Plot of Seasonal Fa
hist(seasonal_fit$res,main="Histogram of Residuals",col="tan2", xlab = "Residuals")
```

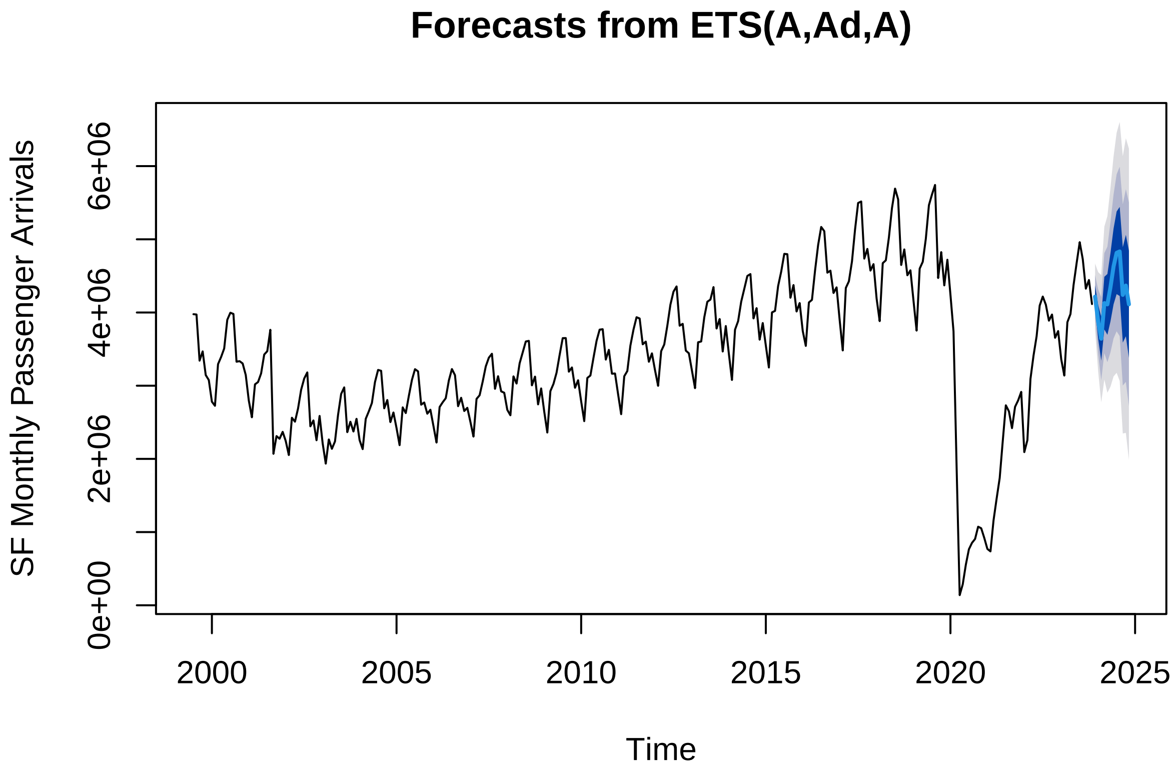




Comment: Looking at the seasonal factors there are large seasonal factors in the middle of the year and a peak in July-August and then that decreases as the year progresses. The beginning of the year is also the low point for seasonal factors, seen in February as the least. There is an increase from February to the Summer months, then back down again as Winter approaches.

f. Choosing a model and forecasting the data

```r
# Forecasting can be improved via ets, and should therefore be used as the model
fit=ets(ts)

plot(forecast(fit,level=c(50,80,95),h=12),
     ylab = "SF Monthly Passenger Arrivals",
     xlab = "Time")
```



Conclusions and Future Work

Regarding the final model, the forecast appears to follow the observed positive trend with similar fluctuations continuing after the fall seen in 2020. The data has recovered to a point in the forecast where the fluctuations and overall trend can continue as normally seen before 2020. The levels outlined in the various shades in the graph provide some prediction intervals for the data. The forecast above portrays the overall historical trend of the data with the preferred model with a decent fit. In the future one can consider imposing restrictions on the model, using the shrinkage principle, to improve forecast performance.

References

The San Francisco International Airport is the subject of this data, with the data being owned by OpenData: https://data.sfgov.org/Transportation/Air-Traffic-Passenger-Statistics/rkru-6vcg/about_data