Monthly Passengers Analysis and Forecast

Luyang Zhang, Jiabao Gao

2023-12-05

Abstract

This project addresses the impact of the COVID-19 pandemic on global travel, with a focus on the periods when restrictions imposed on international and domestic flights. We answered the question that the number of passengers that would have visited San Francisco International Airport (SFO) if the pandemic did not happen. Herein, we considered time series analysis using the pre-COVID monthly passengers data. We decomposed the time series and fitted using an Autoregressive Moving Average (ARMA) model with additional seasonal and trend components. We used spectral analysis to validate our assumptions after fitting the ARMA model. Then we made predictions for the total 24 months passengers in 2020 and 2021 and concluded that SFO could lost about 80 Million passengers within 2020 and 2021.

1. Introduction

San Francisco International Airport (SFO), as one of the major gateway in the U.S west coast, had undergone spectacular increase in visitors from year to year. According to FY 2018-2019 Financial Summary of SFO, SFO served about 57.5 million passengers in 2019. The number of international travelers increased 7.2 percent from 2018[1]. However, the sudden strike of COVID-19 brought the number of visitors a significant plunge.

The United States enforced travel restriction at the beginning of 2020 as a response to the highly contagious virus, and international air travel impacted significantly. Since the number of passengers grew fast before the pandemic, the influence of a sudden lockdown could be substantial. To address the extent of the loss, we hope to predict what the monthly passengers could be if we do not have the pandemic.

2. Data Description

The analysis is based on San Francisco International Airport monthly air traffic passenger statistics. The data is retrieved from DataSF Open Data, a data sharing project coordinated by the Government of San Francisco[2]. The raw data has 34,878 rows and 15 column. The columns are as the follows.

```
##  [1] "Activity Period"             "Activity Period Start Date"
##  [3] "Operating Airline"           "Operating Airline IATA Code"
##  [5] "Published Airline"           "Published Airline IATA Code"
##  [7] "GEO Summary"                 "GEO Region"
##  [9] "Activity Type Code"          "Price Category Code"
## [11] "Terminal"                    "Boarding Area"
## [13] "Passenger Count"             "data_as_of"
## [15] "data_loaded_at"
```

The data contains multiple information about airlines, boarding details, and terminals, but we are only interested in the number of passengers, so we use `dplyr` package to sum the passengers and aggregate it by

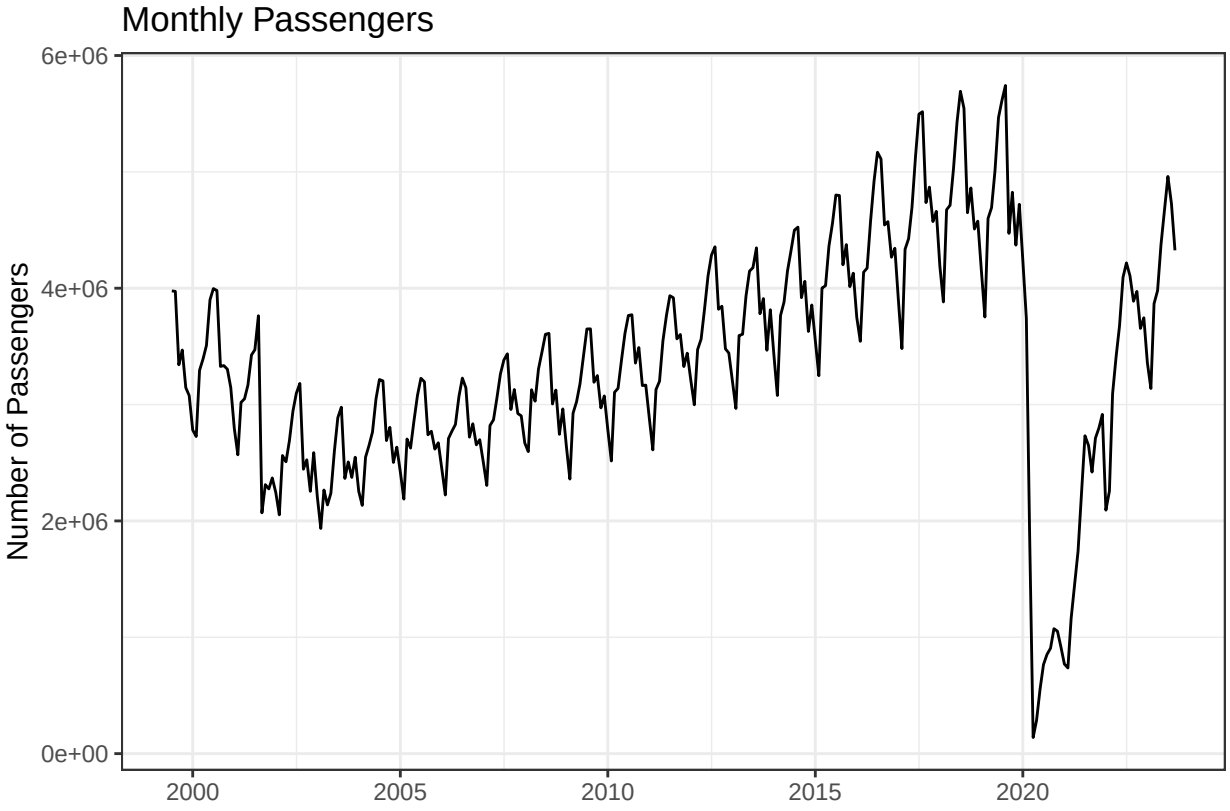
[1] FY 2018-2019 Financial Summary, https://www.flysfo.com/fy-2018-2019-financial-summary#:~:text=Fiscal%20year%202019%20passenger%20traffic,international%20enplaned%20passengers%20increased%206.7%25.

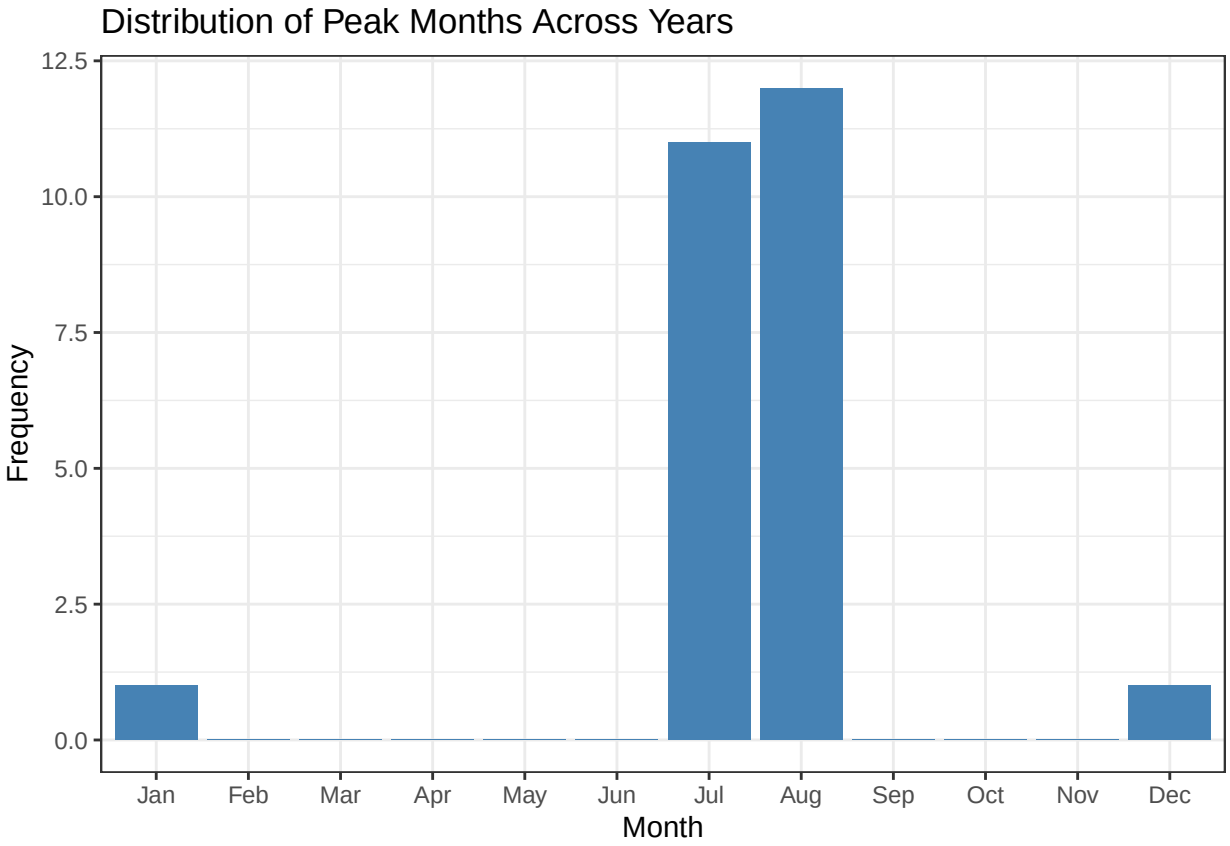
[2] Air Traffic Passenger Statistics, https://data.sfgov.org/Transportation/Air-Traffic-Passenger-Statistics/rkru-6vcg

months. The head of the aggregated dataframe is shown as the following.

Activity_Period	Monthly_Passengers
1999-07-01	3976746
1999-08-01	3972694
1999-09-01	3341964
1999-10-01	3468846
1999-11-01	3145240
1999-12-01	3077142

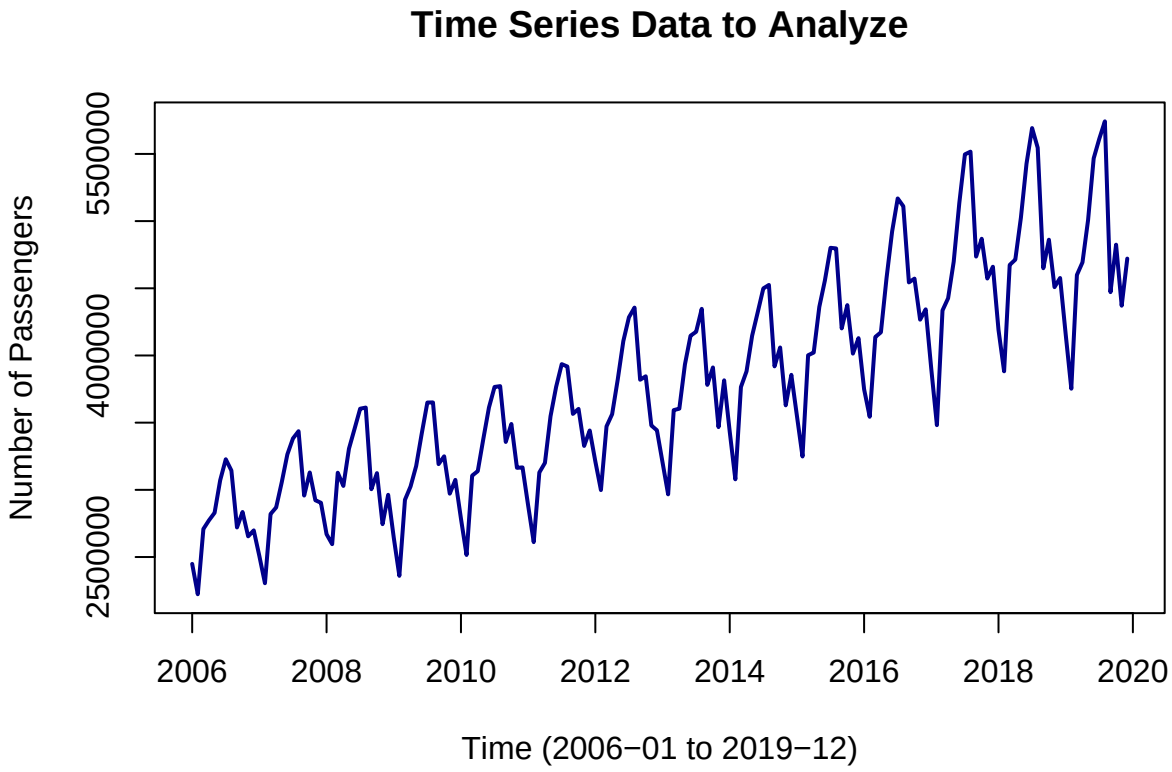
The plot shows the number of passenger plunged a lot after 2020 and is recovering since then. It can be seen that there is a clear upward trend in late 2000s and 2010s before COVID-19. The number of passengers seem increase from year to year. Also, the peaks for each year indicate it has seasonality. We further investigated that July or Auguest are the most visited months in a year.





2.2 Truncate the data and turn into a time series object

Also, since the upward trend started around 2006, we truncated the data before 2006 and turned it into a time series object to analyze. Then we made another time series containing data from Jan.2020 to Dec.2021 to compare the prediction in the later sections of the project.



3. Data Analysis

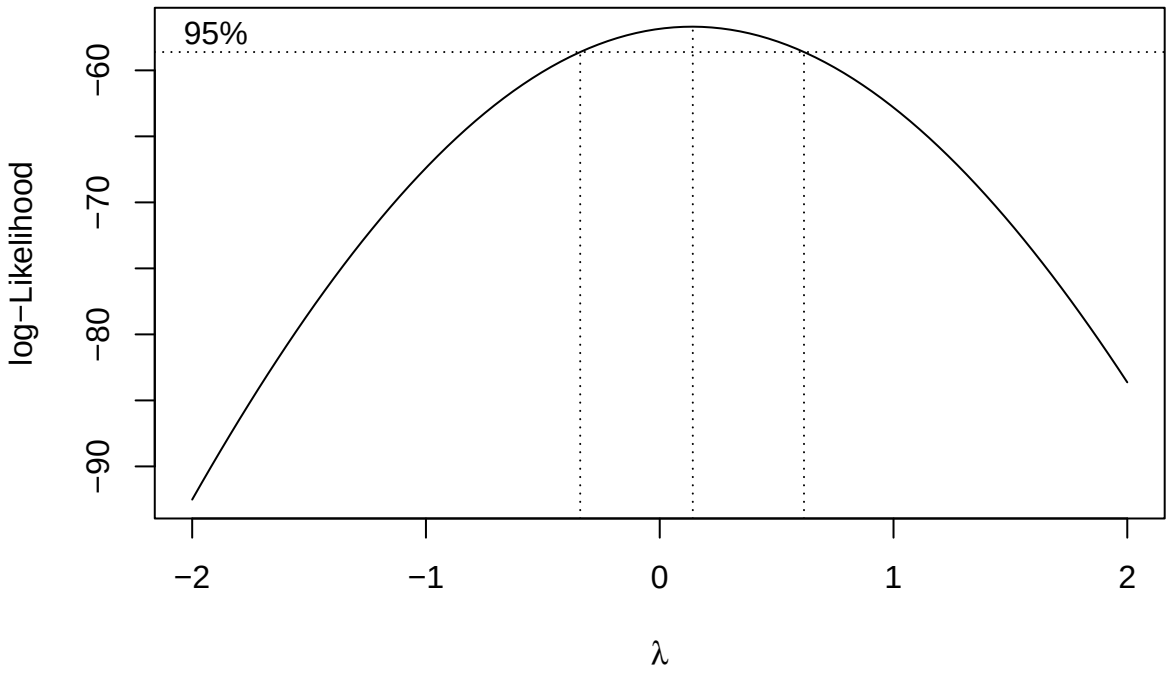
3.1 Box-Cox Transformation

First, we investigate the normality of the time series data using Box-Cox Transformation. It is a technique that re-shape the data to make it looks more like a normally distributed data. The method that the data get transformed depends on the choice of optimal value, λ. The general transformation of positive data follows this form[3]:

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0 \\ \log y, & \text{if } \lambda = 0 \end{cases} \tag{1}$$

Since zero is enclosed into the confidence interval, we applied log-transform.

[3]Box Cox Transformation: Definition, Examples,https://www.statisticshowto.com/box-cox-transformation/



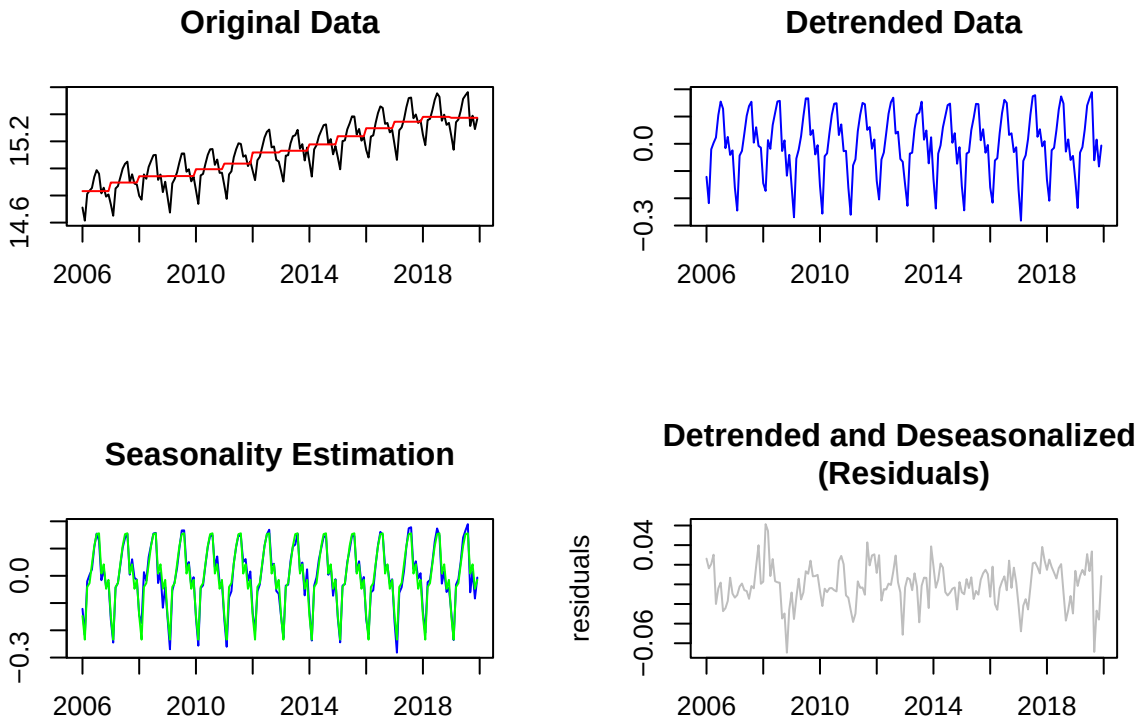
3.2 Seasonal Component Estimation

In the classical decomposition, a time series X_t can be decomposed into three parts:$X_t = m_t + s_t + Y_t$, which correspond to a trend, a seasonal, and a residuals part with mean of zero. Next, we use small trend method to remove the seasonal component of the data. We estimate the drift m_j in year j that can be regarded as the mean of monthly data k for $k = 1, 2, ..., 12$ in a year j.

$$\hat{m_j} = \frac{1}{12} \sum_{k=1}^{12} x_{j,k}$$

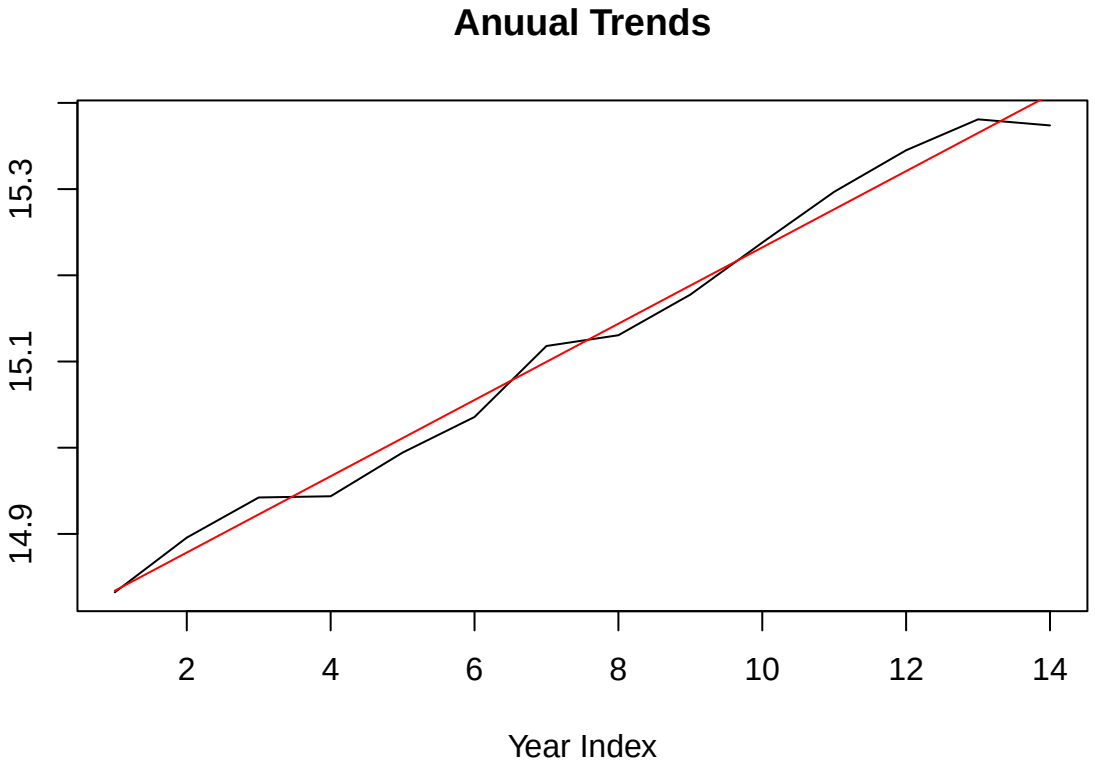
Using this, we estimated the seasonality after we remove the drift. We have 14 full cycles of the data, and the seasonality estimation is the following:

$$\hat{s_k} = \frac{1}{14} \sum_{j=1}^{14} (x_{j,k} - \hat{m_j})$$



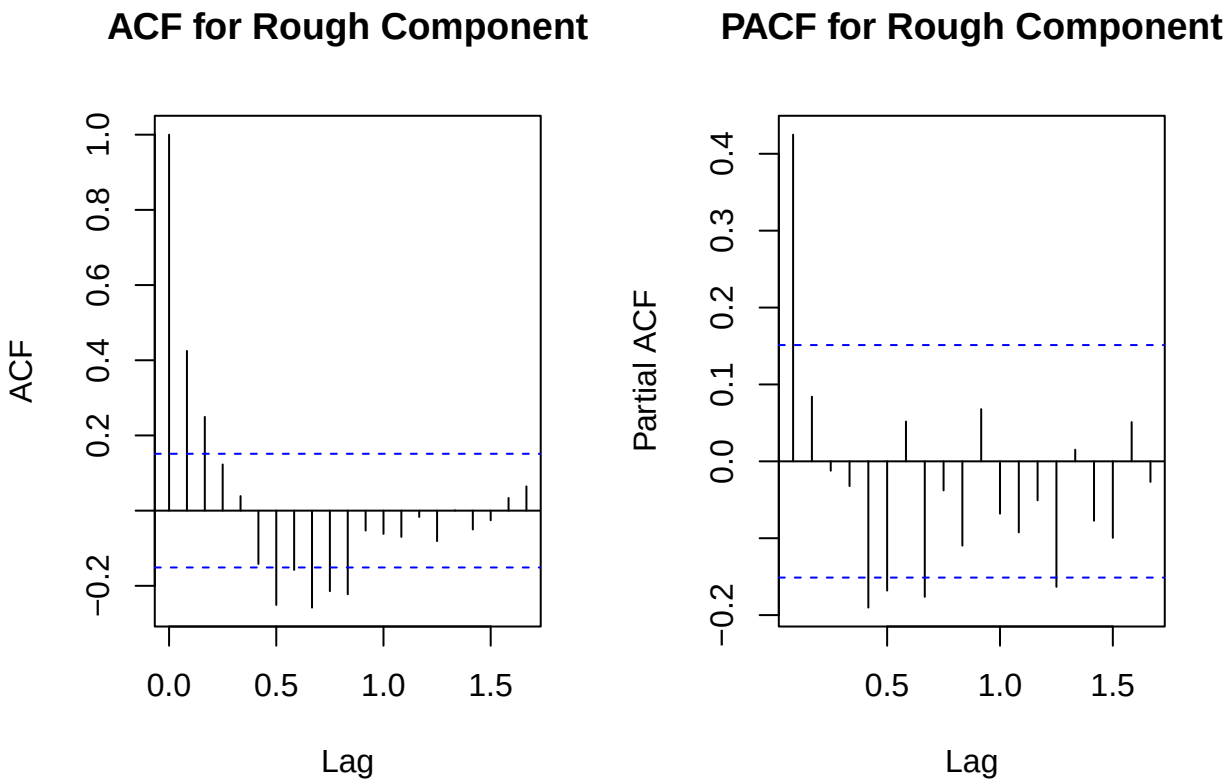
3.3 Trend Estimation

Then we estimate the trend component m_j. The small trend method has already given us the yearly trend data, but for forecasting, we need to find an average annual growth. Thus, We did a first-order polynomial regression to estimate the mean yearly increment of trend. The annual trend, starting from the year 2006, is modeled by the expression $\hat{m}_j = 0.0442j + 14.79$, where t represents the number of years elapsed since 2006. We already have the estimated trend up to 2019. Specifically, the estimated trend for 2019 is 15.37, so we can alter the expression as $\hat{m}_t = 0.0442t + 15.37$, where t represents the number of years elapsed since 2019. We will use this for the forecasting part later.



After we remove the trend and seasonality, the mean of residual becomes zero. Secondly, according to Shapiro-Wilk's test, the residuals shows normality. However, the Box-Pierce Test and Box-Ljung Test showed there is a significant evidence to say the residuals are not stationary. This aligned with the ACF and PACF plots. Several lags for both plots are significant and outside the confidence interval. Thus, we later fitted ARMA models to address this.

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.9861, p-value = 0.09361
##
##
##  Box-Pierce test
##
## data:  na.omit(res)
## X-squared = 89.993, df = 12, p-value = 4.952e-14
##
##
##  Box-Ljung test
##
## data:  na.omit(res)
## X-squared = 93.736, df = 12, p-value = 9.326e-15
```



3.4 Rough Component Estimation

The ACF and PACF plots of the residuals after removing trends and seasonality indicate that some ARMA behaviors. We used a grid search fashion to select the model with the lowest AIC scores. According to the PACF plot, lag 5, 6, 8, and 15 seems significant, but we did not consider AR(15) since this could lead to overfitting issue and parsimony of coefficients concern. We also want to test if including some moving average (MA) coefficients could increase the goodness of fit.

	AR_coef	MA_coef	AIC_scores
8	6	3	-877.2
12	8	3	-874.7
11	8	2	-867.4
4	5	3	-864.9
10	8	1	-864.8
2	5	1	-864.3
3	5	2	-862.8
6	6	1	-861.2
9	8	0	-860.5
7	6	2	-859.2
5	6	0	-857.4
1	5	0	-854.2

The results suggested that $ARMA(6, 3)$ has the lowest AIC.

Also, there is a reasonable argument to contain MA coefficients thought it is not obvious in the ACF plot. If we did not consider MA, so only autoregressive model instead, the residuals after any AR model is still not stationary, but all ARMA models' residuals are stationary. So the result from grid search aligns with our experiment of ARMA model.

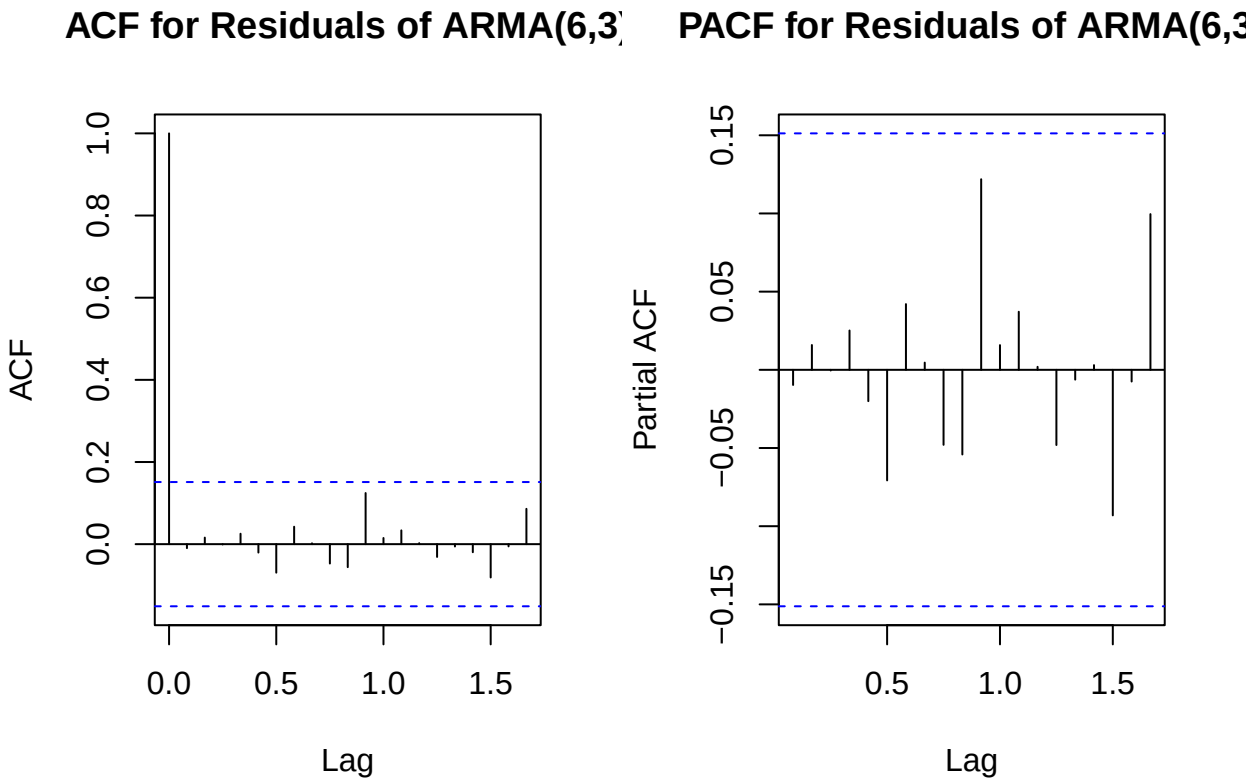
The model with coefficients is in this form:

$$Y_t - .39Y_{t-1} + .98Y_{t-2} + .62Y_{t-3} - .28Y_{t-4} - .14Y_{t-5} - .24Y_{t-6} = Z_t + .79Z_{t-1} - .79Z_{t-2} - .99Z_{t-3}$$

3.5 Model Diagnostics

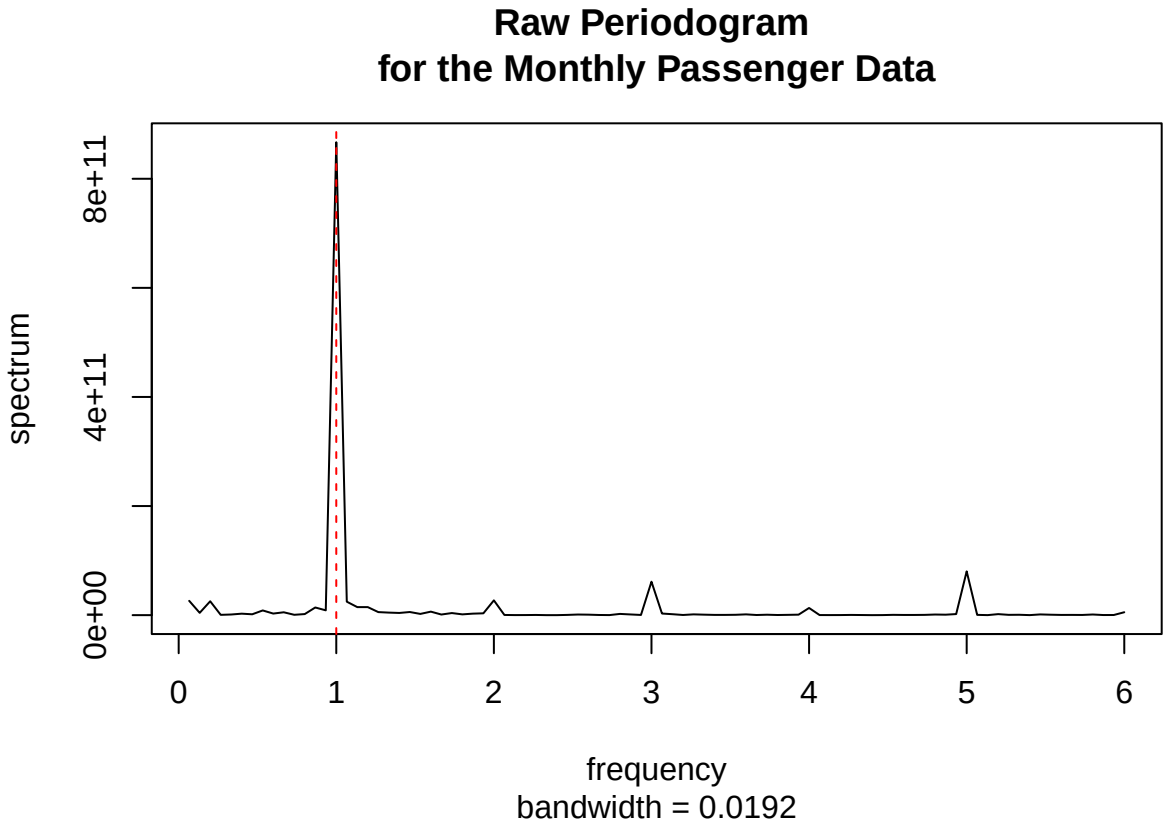
Then we analyzed the residuals for the ARMA model. It turned out that it passed the Shapiro-Wilk's test for normality and Box-Pierce Test and Box-Ljung Test for stationality. All the test do not show significant p-values. This is further validated as the ACF and PACF for the residuals have no significant ticks, suggesting a white noise behavior. Therefore, we have strong evidence to say that the ARMA(6,3) model captures enough variation in the rough component after removing the trend and seasonality.

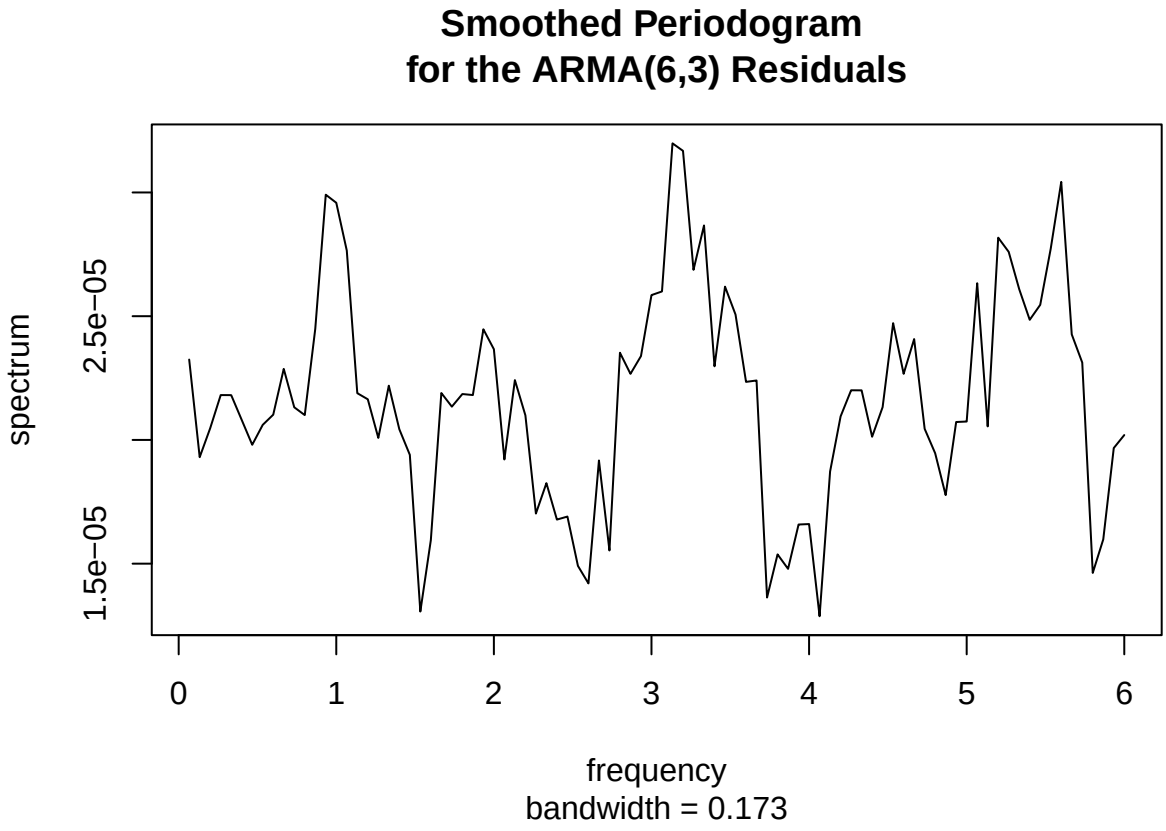
```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.98533, p-value = 0.07478
##
##
##  Box-Pierce test
##
## data:  na.omit(res)
## X-squared = 4.8924, df = 12, p-value = 0.9615
##
##
##  Box-Ljung test
##
## data:  na.omit(res)
## X-squared = 5.2424, df = 12, p-value = 0.9494
```



3.6 Spectral Analysis

From the spectral analysis, the original data has a very high peak at frequency = 1, it means the data has strong annual periodicity. Although there exists some small ups at other frequency, they are not significant compare the first one. Then after we decomposed the data and fitted an ARMA model, the residuals remaining does not show any significant peaks after applying a Daniell smoother. This indicates that our analysis eliminates major seasonality.





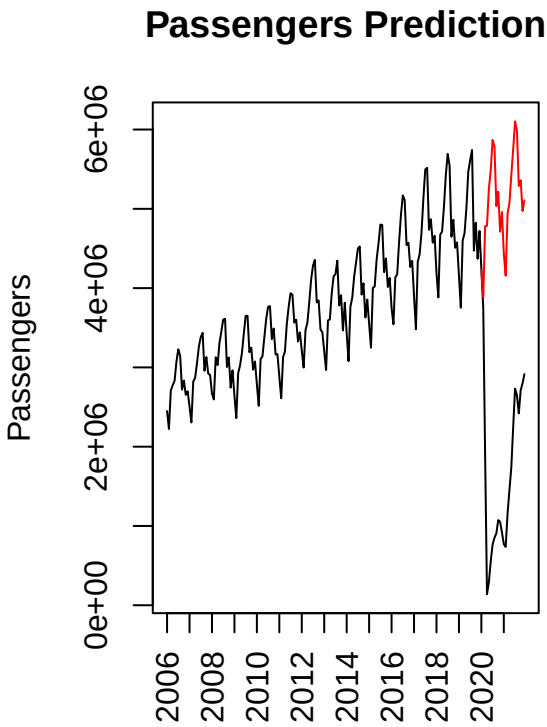
4. Discussion

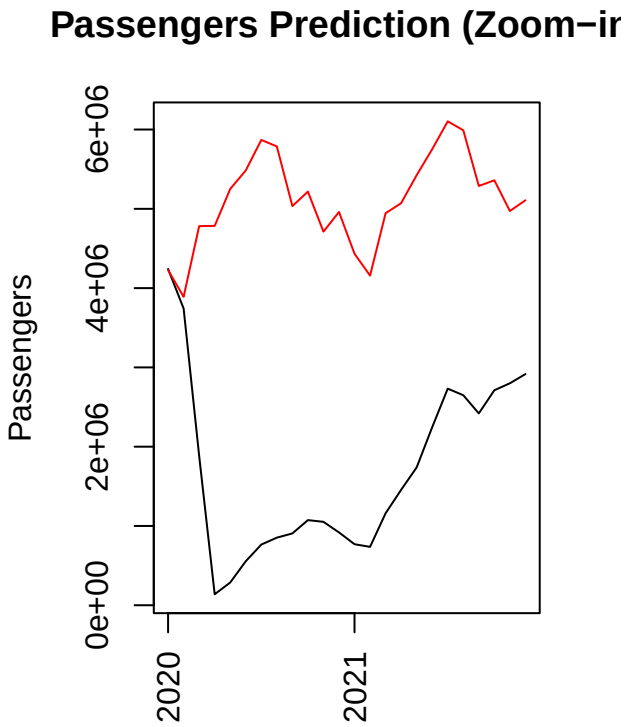
Then we predicted the next 24 month (2 years) passengers using the ARMA(6,3) model. First, we first predict the next 24 residuals $\hat{Y_{j,k}}$, or the rough component. Then we added the seasonal component from small trend method. To add the yearly trend, we first access the trend component for the log data of year 2019, which is 15.37385. We noted that the slope for the linear model is around 0.0442. This means the log data is going to increase by this amount each year. Then we add one copy of the value of slope for the first 12 predictions and two copies of the value of slope to the next 12 predictions. Lastly, we added the seasonal component $\hat{s_k}$ for each $k = 1, 2, ..., 12$. The predictions for 2020 and 2021 data is shown below:

$$\begin{aligned} \hat{X_{2020,k}} &= \hat{m_{2020}} + \hat{s_k} + \hat{Y_{2020,k}} \\ &= m_{2019} + 0.0442 + \hat{s_k} + \hat{Y_{2020,k}} \\ \hat{X_{2021,k}} &= \hat{m_{2021}} + \hat{s_k} + \hat{Y_{2021,k}} \\ &= m_{2019} + 2(0.0442) + \hat{s_k} + \hat{Y_{2021,k}} \end{aligned}$$

Then we applied exponential to transfer the log prediction back to normal.

The predictions are increasing, implying the passengers should have expected an year-to-year increase if COVID-19 did not happen. We provided the forecasting plot and a zoom-in for year 2020-21. Finally, we sum up the predictions and subtract the sum of the actual numbers. We concluded that the loss over the two years is around 81,851,571.





5.Conclusion

The number of passengers in San Francisco International Airport (SFO) has a clear year-to-year trend and seasonality within a year. We log-transformed the data and decomposed it into trend, seasonal, and residuals parts. We fitted the residuals with an ARMA(6,3), along with the predicted trends and seasonality, to forecast the monthly passengers in 2020 and 2021 with the absence of COVID-19. Then we use spectral analysis to check that the residuals from ARMA model no longer have seasonality. And the one major peak of the original data from the spectral analysis indicates that before the pandemic, the monthly passengers could be affected by seasonality, such as holidays. As the year of 2020 started, the pandemic highly influence the passenger and drastically changed the overall patterns of the historical data–since the pandemic, the gap between actual data and forecasts enlarges. We concluded that SFO could lost about 80 Million passengers in the two years.

In future studies, prediction with confidence intervals can provide a range of the predicted number of predicted passengers. Also, for using ARMA models to do future predictions, one could try to use some data as a validation set to test the in-sample variability. The inclusion of some ideas from cross-validation can give a better assessment of models.

6.References

[1] FY 2018-2019 Financial Summary. https://www.flysfo.com/fy-2018-2019-financial-summary#:~:text=Fiscal%20year%202019%20passenger%20traffic,international%20enplaned%20passengers%20increased%206.7%25.

[2] Box Cox Transformation: Definition, Examples. https://www.statisticshowto.com/box-cox-transformation/.

[3] Air Traffic Passenger Statistics. https://data.sfgov.org/Transportation/Air-Traffic-Passenger-Statistics/rkru-6vcg.

7.Appendix

```
df <- read_csv("Air_Traffic_Passenger_Statistics.csv",show_col_types = FALSE)
colnames(df)

passenger_df <- df %>%
  group_by(Activity_Period = `Activity Period`) %>%
  summarize(Monthly_Passengers = sum(`Passenger Count`))

passenger_df$Activity_Period <- as.Date(as.yearmon(as.character(passenger_df$Activity_Period),
                                                   format="%Y%m"))
passenger_df %>% head() %>% pander()

passenger_df %>% ggplot() +
  geom_line(aes(x = Activity_Period, y = Monthly_Passengers)) +
  xlab('') + ylab('Number of Passengers') +
  theme_bw() + ggtitle('Monthly Passengers')

passenger_df$Activity_Period<-as.Date(passenger_df$Activity_Period)

peak_months_each_year <- passenger_df %>%
  mutate(Year = year(Activity_Period)) %>%
  group_by(Year) %>%
  summarize(Peak_Month = Activity_Period[which.max(Monthly_Passengers)],
            Peak_Passengers = max(Monthly_Passengers))

# extract the month from the Peak_Month
peak_months_each_year$Month <- month(peak_months_each_year$Peak_Month,
                                     label = TRUE)

# create a frequency table
peak_month_distribution <- table(peak_months_each_year$Month)

# convert the table to a dataframe
peak_month_distribution_df <-  as.data.frame(peak_month_distribution)

#Rename the columns for calrity
names(peak_month_distribution_df) <- c("Month", "Frequency")

ggplot(peak_month_distribution_df, aes(x = Month, y = Frequency)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  xlab("Month") + ylab("Frequency") +
  ggtitle("Distribution of Peak Months Across Years") +
  theme_bw()


#### filter the data
precovid_df <- passenger_df %>%
  filter(Activity_Period < '2020-01-01', Activity_Period > '2005-12-01')
aftercovid_df <- passenger_df %>%
  filter(Activity_Period > '2019-12-01', Activity_Period < '2022-01-01')

# convert into time series objects
```

```r
precovid <- ts(precovid_df$Monthly_Passengers, start = 2006, frequency = 12)
aftercovid <- ts(aftercovid_df$Monthly_Passengers, start = 2020, frequency = 12)

plot.ts(precovid, main = "Time Series Data to Analyze",
        xlab = "Time (2006-01 to 2019-12)",
        ylab = 'Number of Passengers', col= 'darkblue', lwd = 2)

#### Box-cox
t = 1:length(precovid)
bcTransform = boxcox(precovid ~ t,plotit = TRUE)

lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]  #optimal lambda

#apply log transform to the dataset
precovid.bc = ts(log(precovid), start = 2006, frequency = 12)


#### Smooth Components
small_trend <- function(time_series, start_date) {

  # compute monthly average m_j1
  par(mfrow = c(2, 2))
  m_j1 = tapply(time_series, floor(time(time_series)), mean)
  m_j1 = ts(rep(m_j1, each = 12), start = start_date, frequency = 12)

  # plot original and detrended data
  ts.plot(time_series, m_j1, col = c("black", "red"),
          main = 'Original Data', xlab = '', ylab = '')
  ts.plot(time_series - m_j1, col = "blue",
          main = 'Detrended Data', xlab = '', ylab = '')

  # seasonal component s_k1
  s_k1 = tapply(time_series - m_j1, cycle(time_series), mean)
  s_k1 = ts(rep(s_k1, times = length(time_series) / 12),
            start = start_date, frequency = 12)

  # plot detrended and deseasonalized data
  ts.plot(time_series - m_j1, s_k1, col = c("blue", "green"),
          main = 'Seasonality Estimation', xlab = '', ylab = '')

  # compute residuals
  residuals = time_series - m_j1 - s_k1
  ts.plot(residuals, col = "grey",
          main = 'Detrended and Deseasonalized \n(Residuals)', xlab = '')

  return(list(m_j1 = m_j1, s_k1 = s_k1, residuals = residuals))
}

start_date <- c(2006, 1)
result1 <- small_trend(precovid.bc, start_date)

analyze_residuals <- function(res, tests = TRUE, plots = FALSE, plots_headers = NULL) {
```

```r
  # normality and stationality tests
  if (tests == TRUE) {
    print(shapiro.test(res))
    print(Box.test(na.omit(res), lag = 12, type = "Box-Pierce"))
    print(Box.test(na.omit(res), lag = 12, type = "Ljung-Box"))
  }

  # ACF and PACF plots
  if (plots == TRUE) {
    par(mfrow = c(1, 2))

    acf(na.omit(res), lag.max = 20, main = paste("ACF", plots_headers))
    pacf(na.omit(res), lag.max = 20, main = paste("PACF", plots_headers))
  }
}


#### Trend estimation
yearly_m_j1<- result1$m_j1[(seq(1, length(result1$m_j1), by = 12))]
t <- 1:length(yearly_m_j1)
#lm(yearly_m_j1~t)$coef
plot(yearly_m_j1, type = 'l', main = 'Anuual Trends',
     xlab = 'Year Index', ylab = '')
lines(fitted(lm(yearly_m_j1~t)), col = 'red')

analyze_residuals(result1$residuals, plots =T,
                  plots_headers = 'for Rough Component')


#### Rough Component
# grid seach coefficients candidates
ar_candid <- c(5, 6, 8)
ma_candid <- c(0, 1, 2, 3)

# initialization
search_result <- data.frame(AR_coef = numeric(),
                            MA_coef = numeric(), AIC_scores = numeric())

for (i in ar_candid) {
  for (j in ma_candid) {

    # fit ARMA and calculate AIC
    mod_candid <- arima(result1$residuals, order = c(i, 0, j))
    aic <- AIC(mod_candid)

    # append values to the data frame
    search_result <- rbind(search_result,
                           data.frame(AR_coef = i, MA_coef = j, AIC_scores = aic))
  }
}

search_result <- search_result[order(search_result$AIC_scores), ]
search_result %>% pander()
```

```r
res_mod <- arima(result1$residuals, order = c(6, 0, 3))
analyze_residuals(res_mod$residuals, plots = T,
                  plots_headers = 'for Residuals of ARMA(6,3)')


#### Spectral Analysis
spectrum(precovid, taper = 0, log = 'no',
         main = 'Raw Periodogram \nfor the Monthly Passenger Data')
abline(v = 1, lty = 2, col = 'red')
spectrum(res_mod$residuals, kernel("daniell",4), taper = 0, log = 'no',
         main = 'Smoothed Periodogram \nfor the ARMA(6,3) Residuals')


#### Prediction
res_mod <- arima(result1$residuals, order = c(6, 0, 3))

# a vector to store predicted values
predicted_res <- predict(res_mod, 24)$pred

# add all components and take the exponential to scale back the data
pred <- exp(predicted_res + result1$s_k1[1:12] + c(rep(0.04424486, 12),
                                                   rep(0.04424486*2, 12)) + 15.37385)

observed <- c(exp(precovid.bc), aftercovid)  # concatenate pre- and after-covid data

label_years <- seq(2006, length.out = length(observed)/12, by = 1)
par(mfrow = c(1, 2))

plot(observed, type = 'l', xaxt = "n", main = "Passengers Prediction",
     xlab = '', ylab = 'Passengers',
     ylim = c(min(aftercovid_df$Monthly_Passengers), max(pred)))

# draw predictions
lines(169:192, pred, col = 'red', type = 'l')
axis(1, at = seq(1, length(observed), by = 12), labels = label_years, las = 2)

label_years2 <- seq(2020,
                    length.out = length(aftercovid_df$Monthly_Passengers)/12, by = 1)
plot(aftercovid_df$Monthly_Passengers, type = 'l',
     xaxt = "n", main = "Passengers Prediction (Zoom-in)",
     xlab = '', ylab = 'Passengers',
     ylim = c(min(aftercovid_df$Monthly_Passengers), max(pred)))
lines(1:24, pred, col = 'red', type = 'l')
axis(1, at = seq(1, length(aftercovid_df$Monthly_Passengers), by = 12),
     labels = label_years2, las = 2)
# calculate the total loss in the two years
total_loss <- sum(pred) - sum(aftercovid_df$Monthly_Passengers)
```