Impact of COVID-19 on Worldwide Aviation

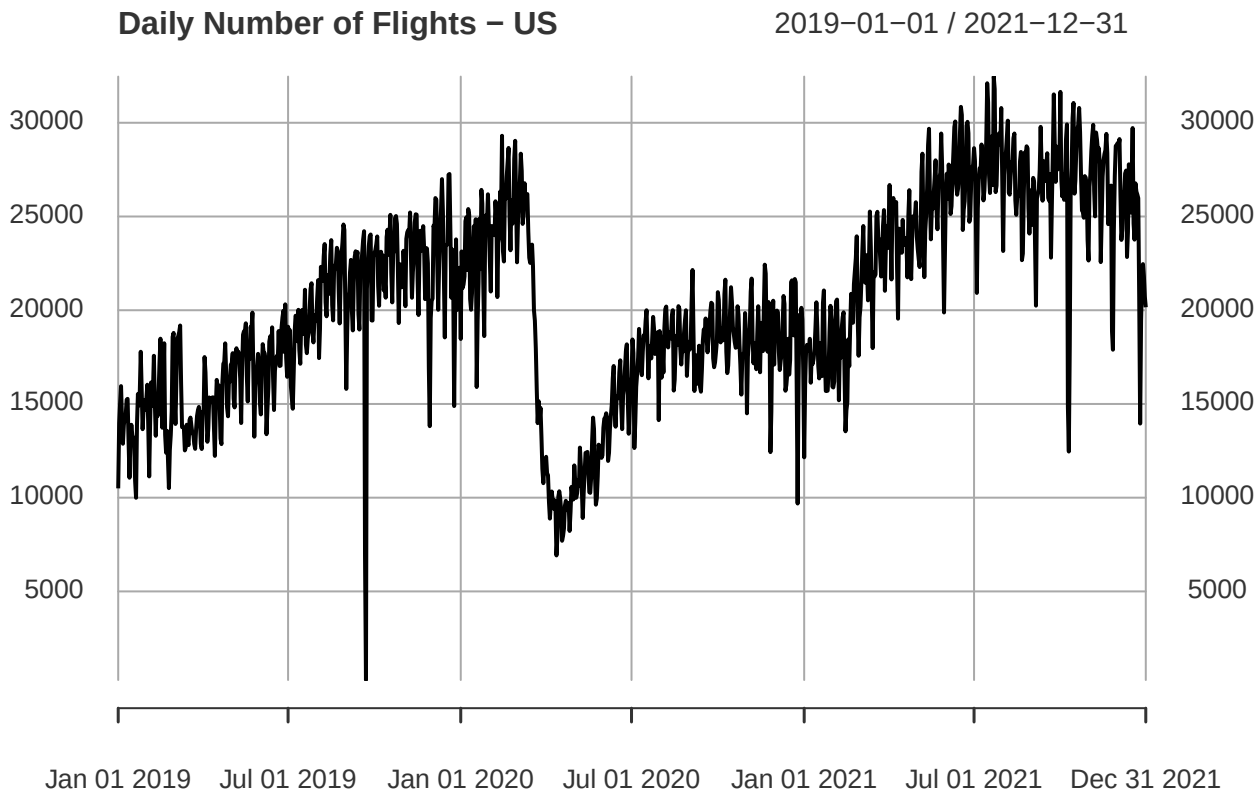
Data Visualization

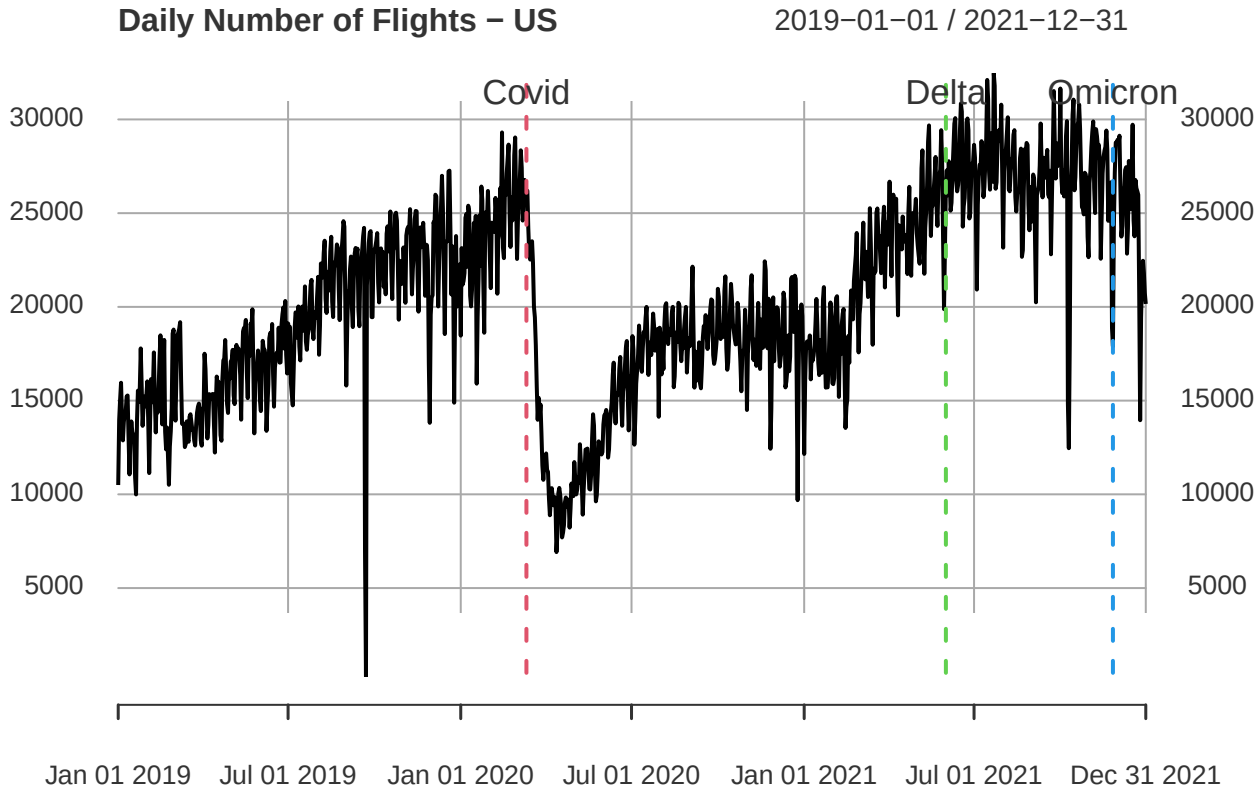
Hao Jin

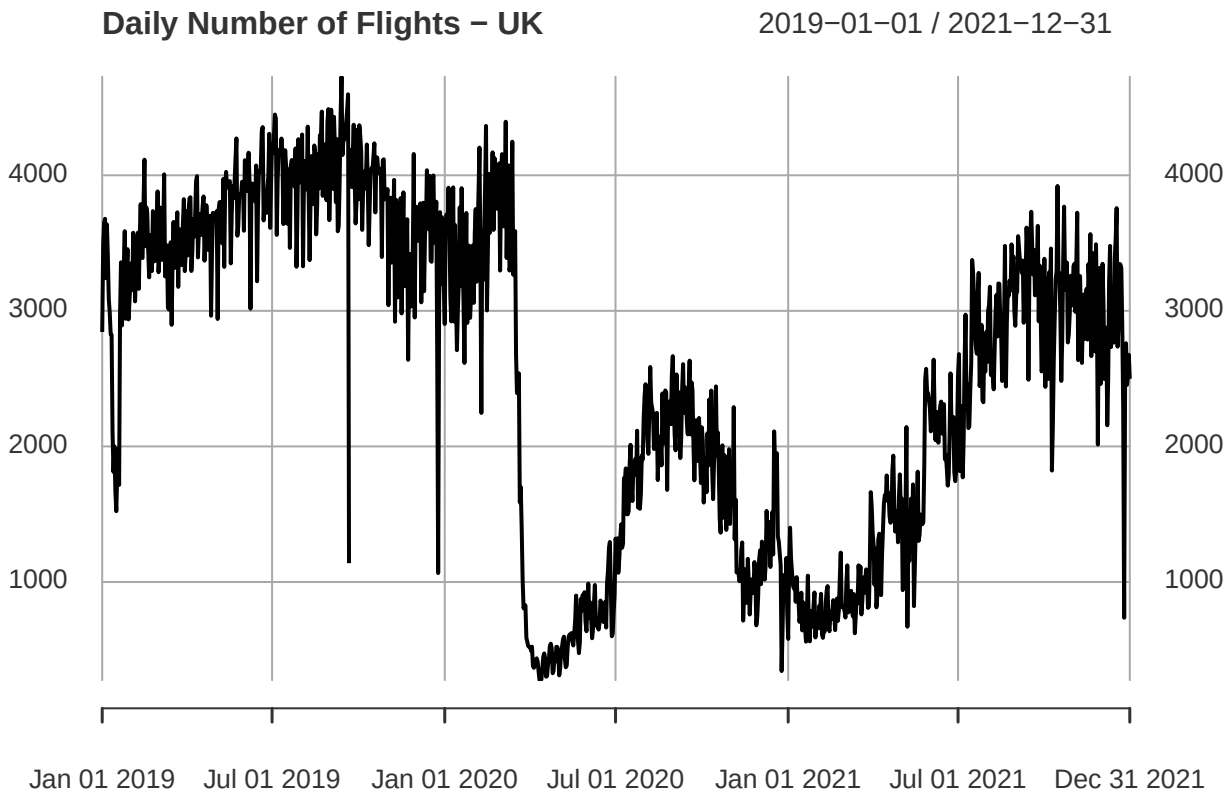
March 17, 2022

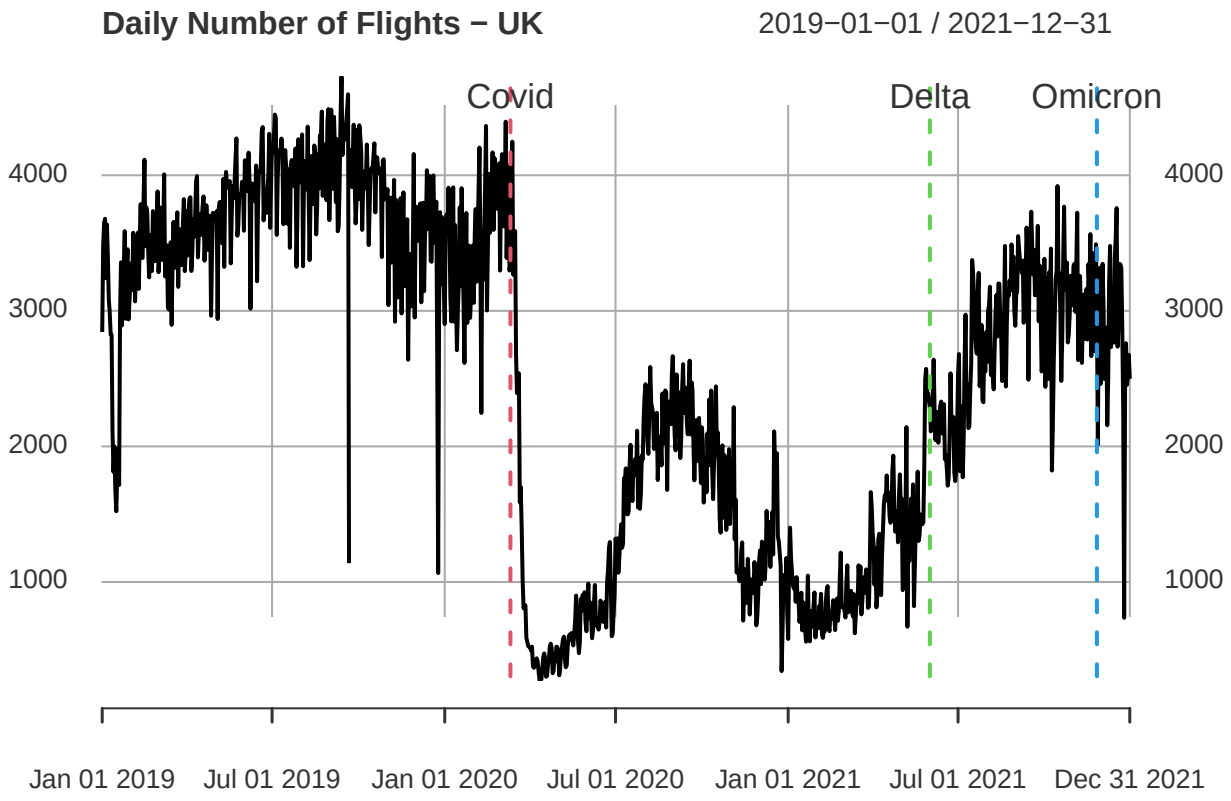
Contents

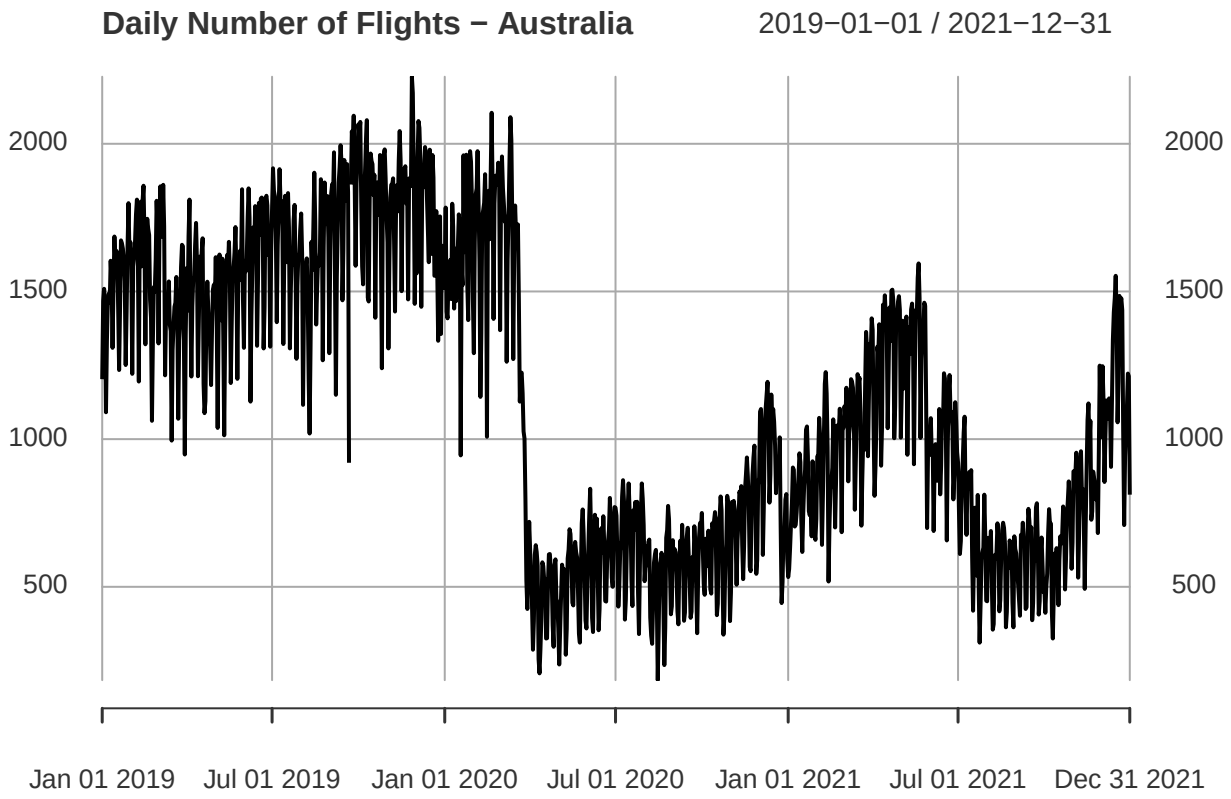
Daily Number of Flights Worldwide

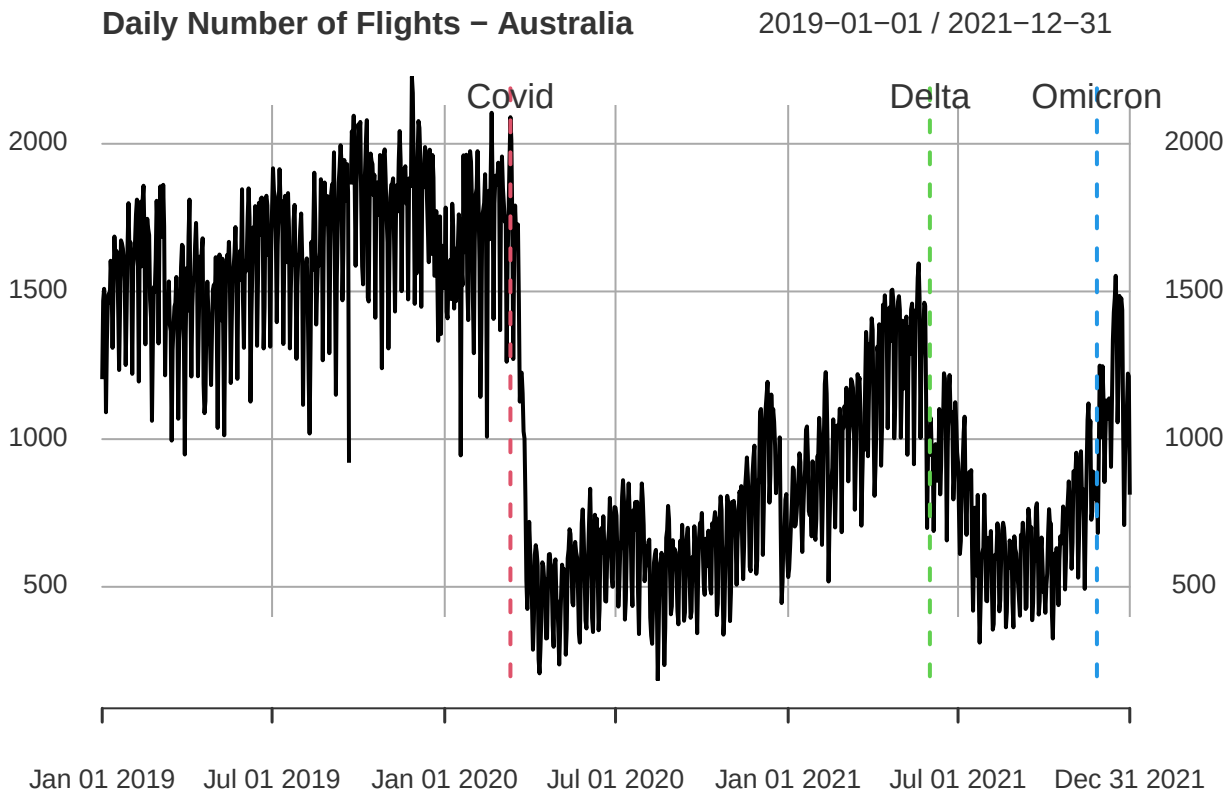


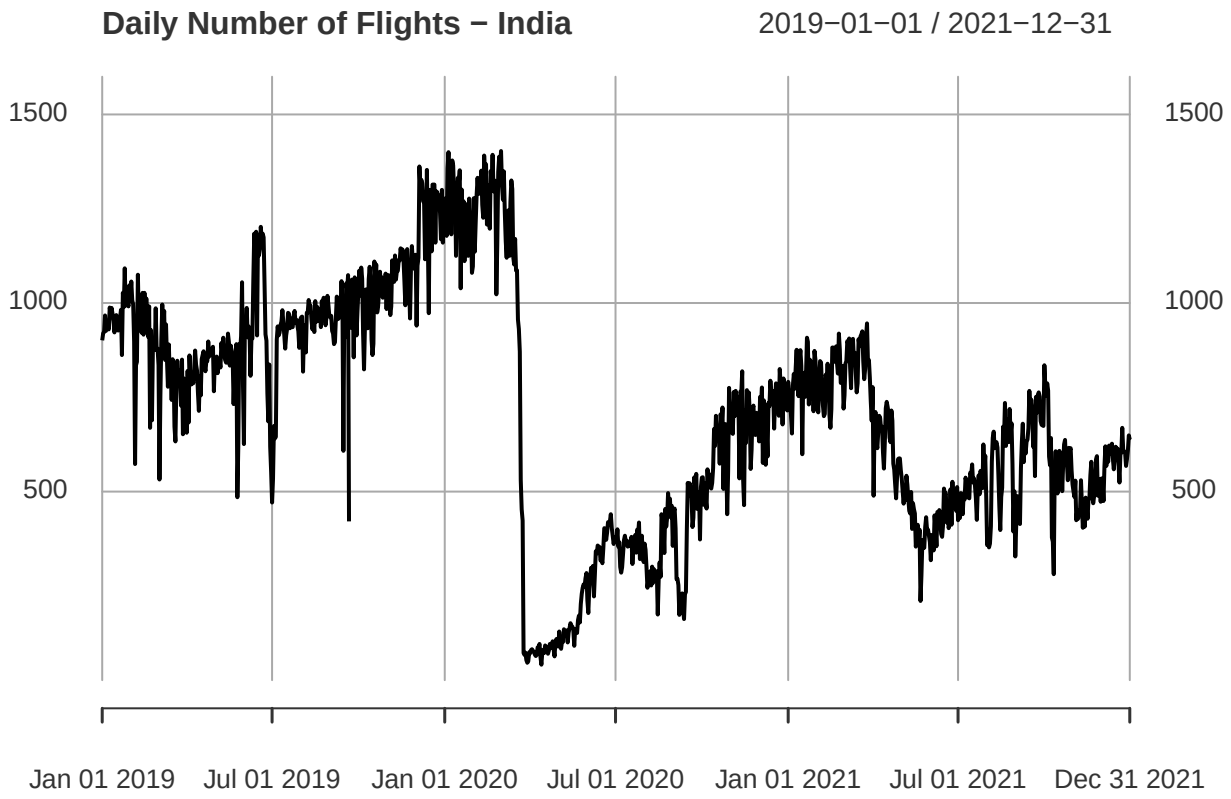


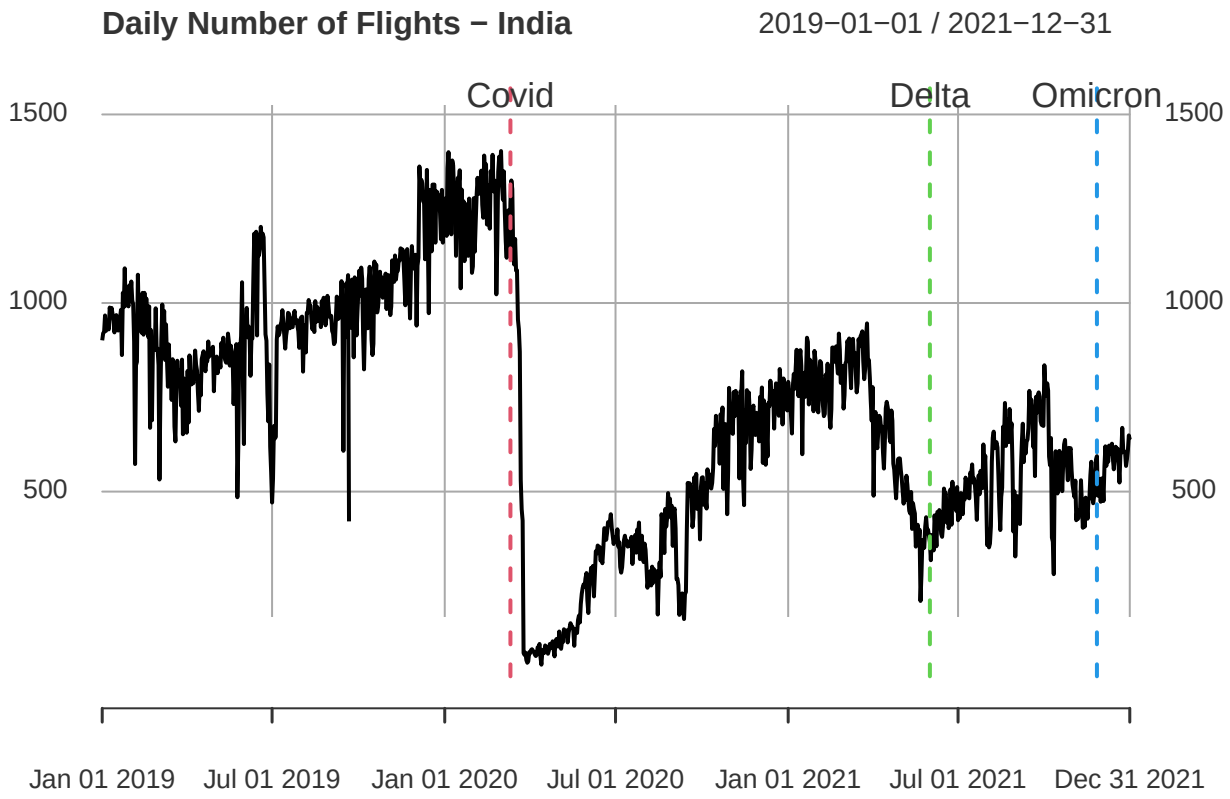

Daily Number of Flights − UK
2019−01−01 / 2021−12−31
4000
3000
2000
1000
Jan 01 2019
Jul 01 2019
Jan 01 2020
Jul 01 2020
Jan 01 2021
Jul 01 2021
Dec 31 2021

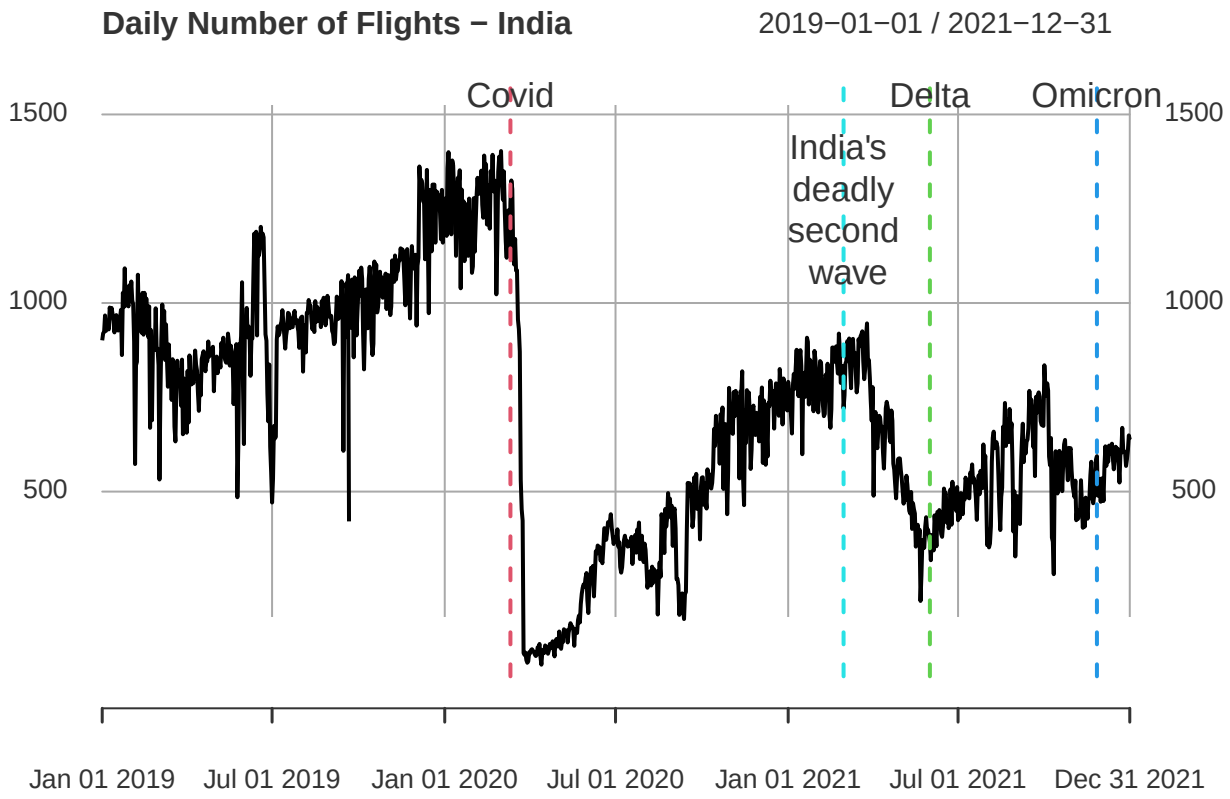

Daily Number of Flights − UK
2019−01−01 / 2021−12−31
Covid
Delta
Omicron
4000
3000
2000
1000
Jan 01 2019
Jul 01 2019
Jan 01 2020
Jul 01 2020
Jan 01 2021
Jul 01 2021
Dec 31 2021

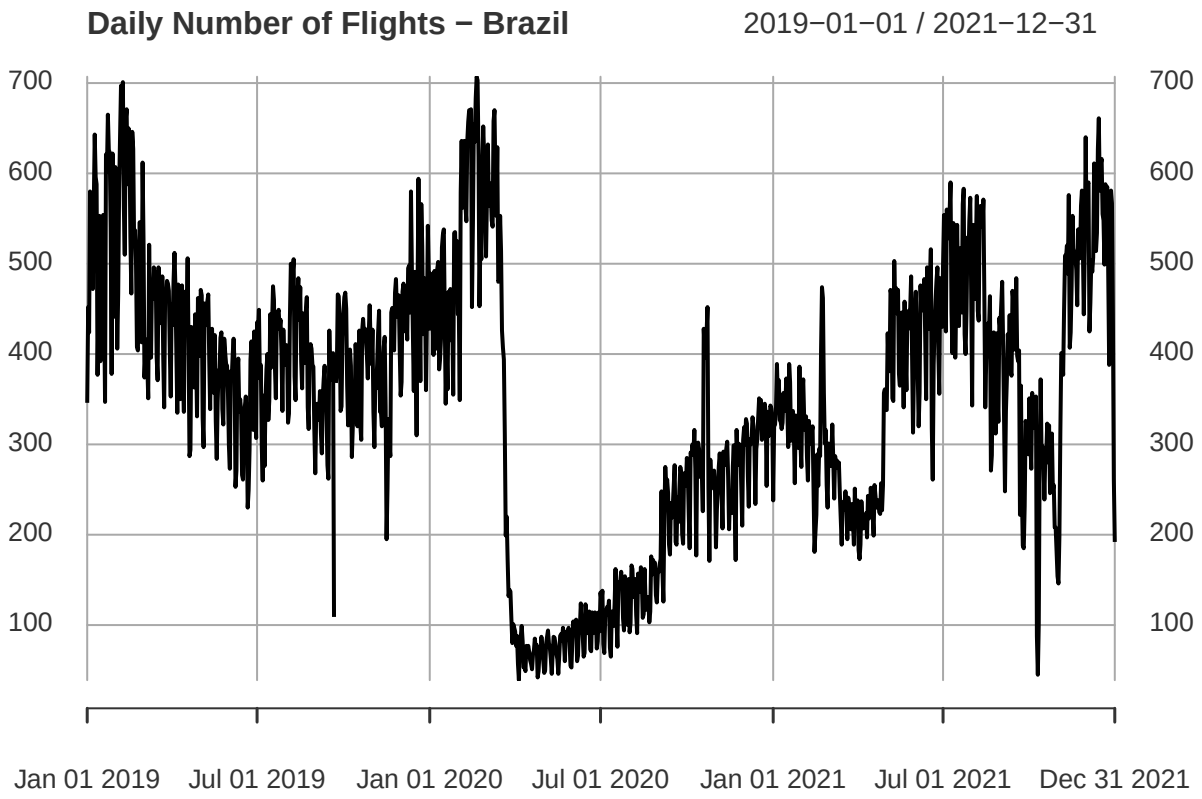

Daily Number of Flights − Australia
2019−01−01 / 2021−12−31
2000
1500
1000
500
Jan 01 2019
Jul 01 2019
Jan 01 2020
Jul 01 2020
Jan 01 2021
Jul 01 2021
Dec 31 2021

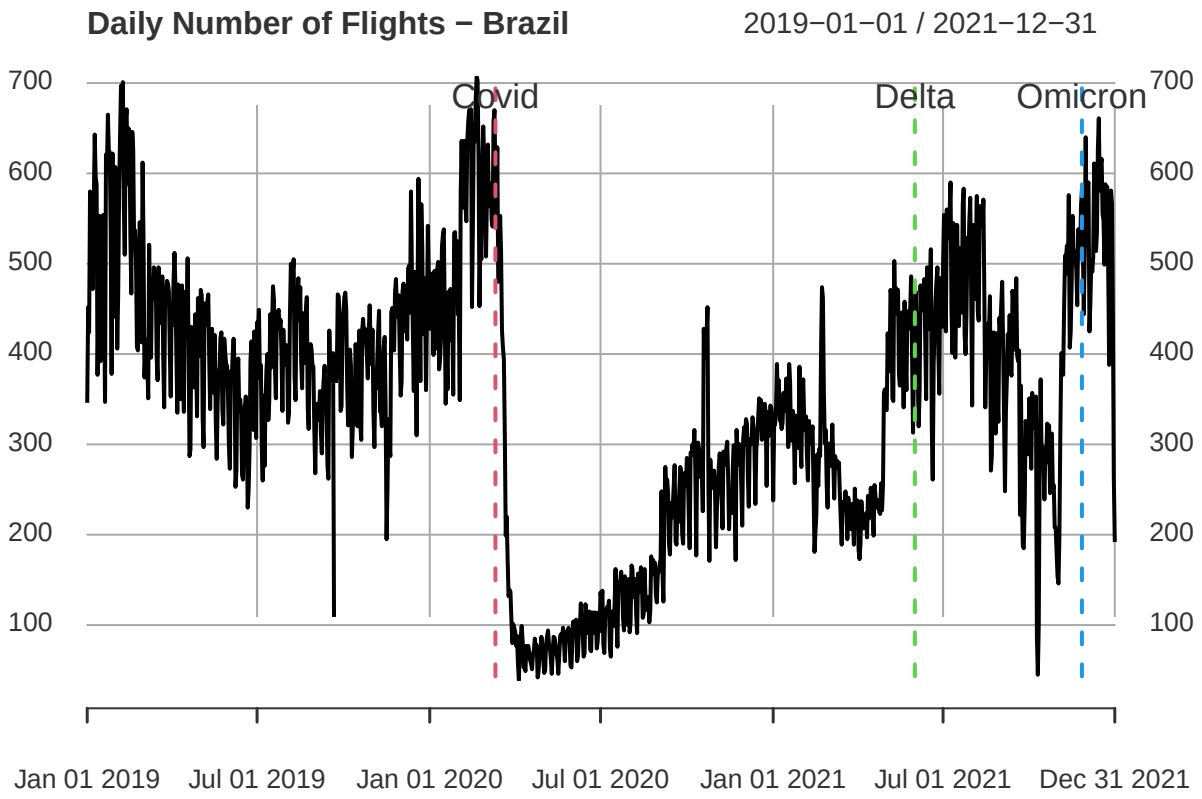

Daily Number of Flights − Australia
2019−01−01 / 2021−12−31
Covid
Delta
Omicron
2000
1500
1000
500
Jan 01 2019
Jul 01 2019
Jan 01 2020
Jul 01 2020
Jan 01 2021
Jul 01 2021
Dec 31 2021

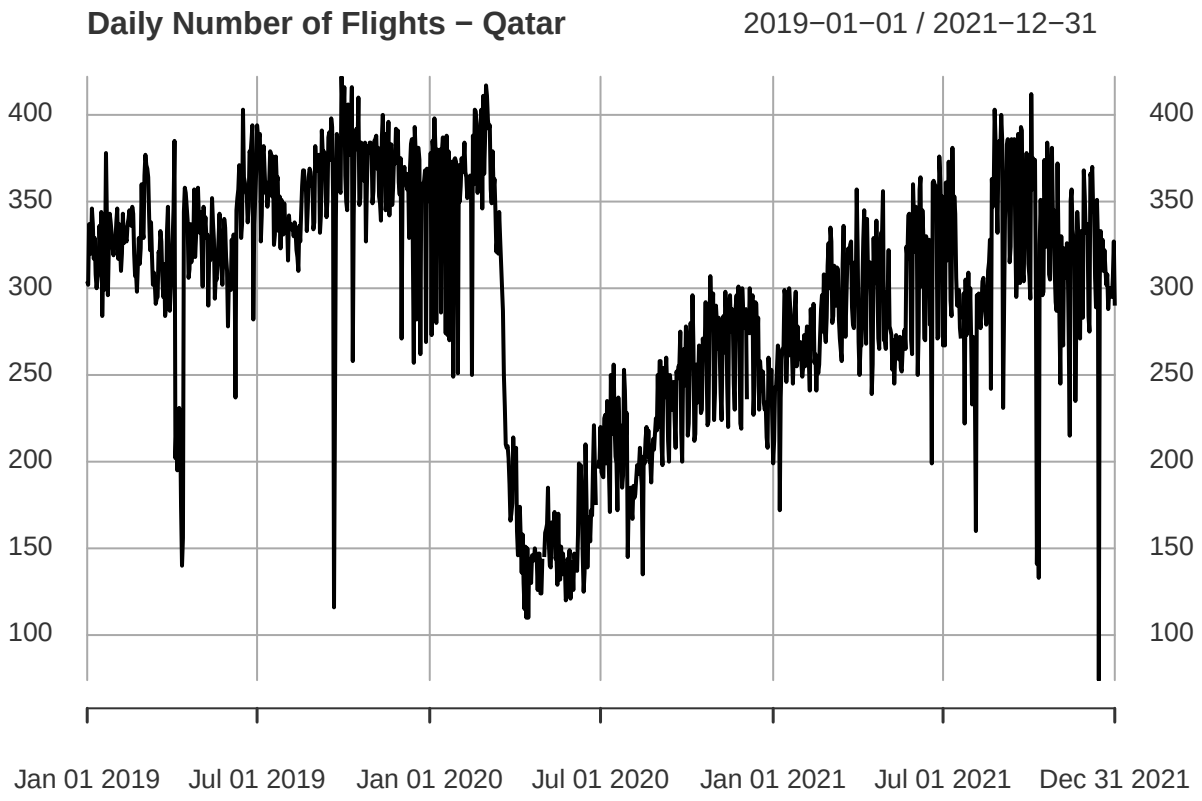

Daily Number of Flights − India
2019−01−01 / 2021−12−31
1500
1000
500
Jan 01 2019
Jul 01 2019
Jan 01 2020
Jul 01 2020
Jan 01 2021
Jul 01 2021
Dec 31 2021

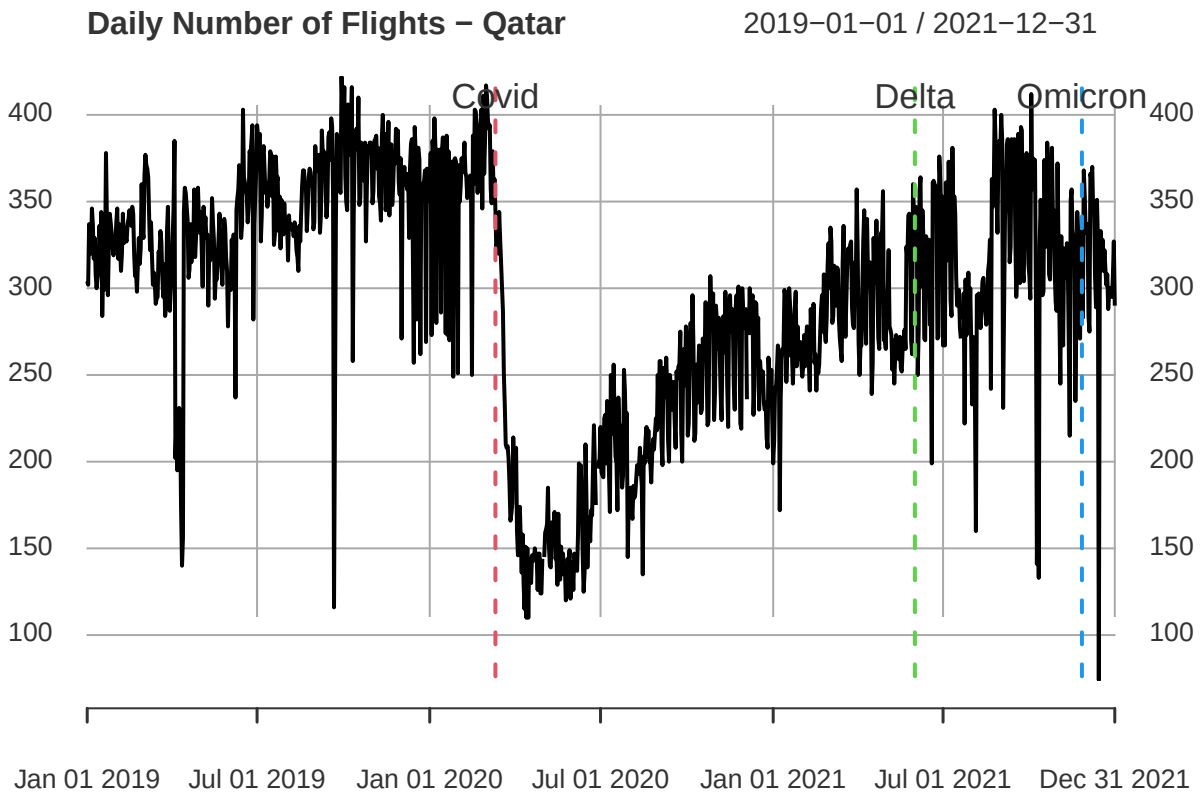

Daily Number of Flights − India
2019−01−01 / 2021−12−31
Covid
Delta
Omicron
1500
1000
500
Jan 01 2019
Jul 01 2019
Jan 01 2020
Jul 01 2020
Jan 01 2021
Jul 01 2021
Dec 31 2021

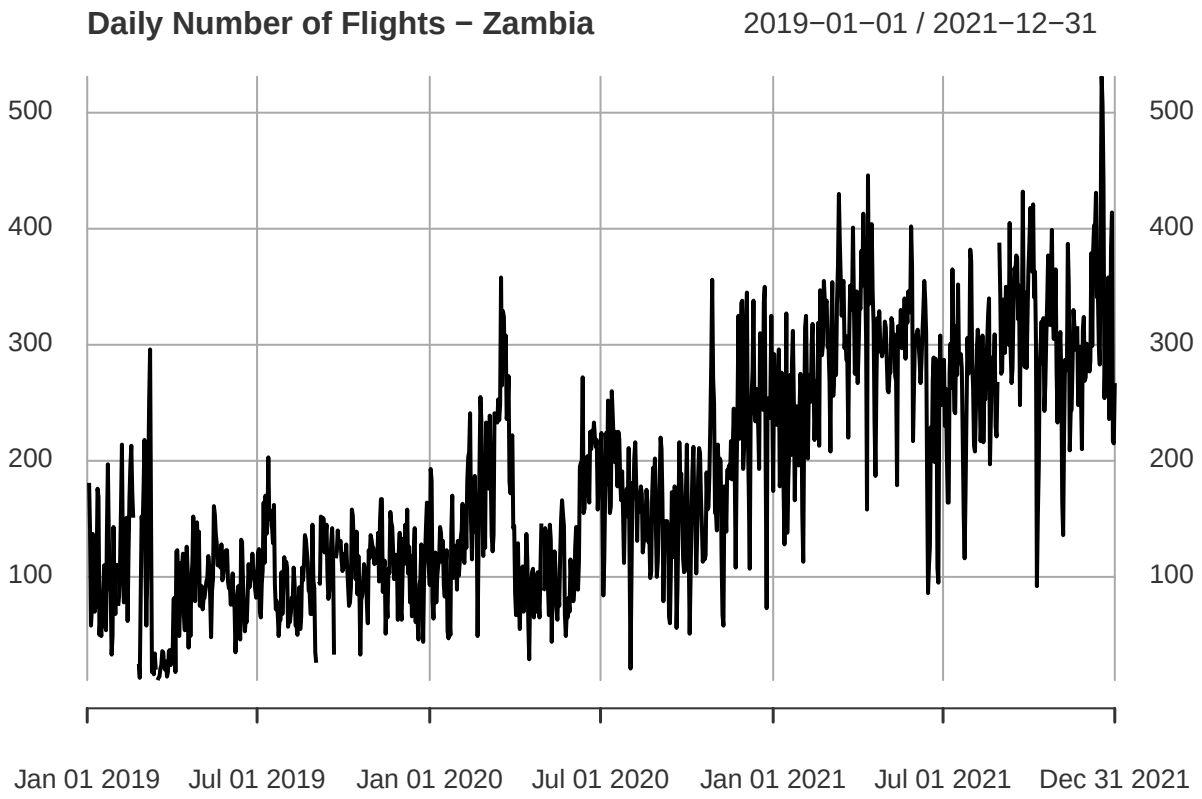

Daily Number of Flights − India
2019−01−01 / 2021−12−31
Covid
Delta
Omicron
India's deadly second wave
1500
1000
500
Jan 01 2019
Jul 01 2019
Jan 01 2020
Jul 01 2020
Jan 01 2021
Jul 01 2021
Dec 31 2021

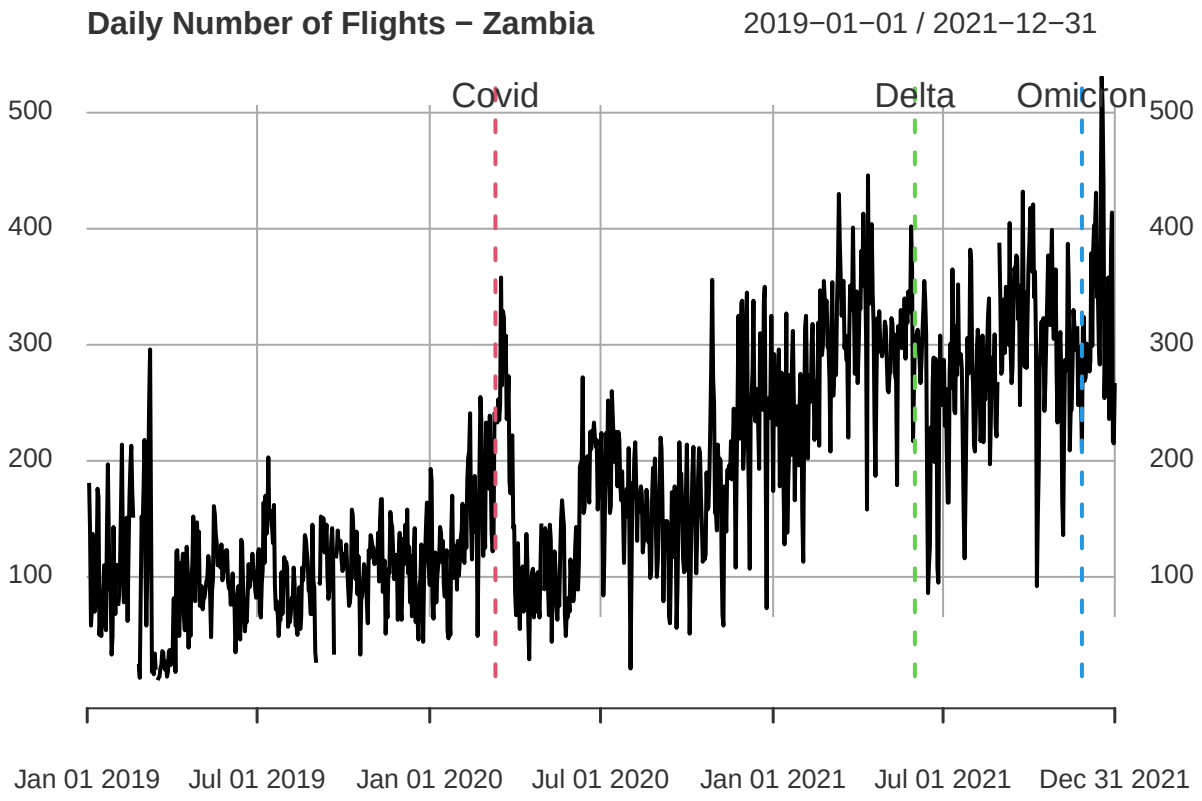

Daily Number of Flights − Brazil
2019−01−01 / 2021−12−31
700
600
500
400
300
200
100
Jan 01 2019
Jul 01 2019
Jan 01 2020
Jul 01 2020
Jan 01 2021
Jul 01 2021
Dec 31 2021


Daily Number of Flights − Brazil
2019−01−01 / 2021−12−31
Covid
Delta
Omicron
700
600
500
400
300
200
100
Jan 01 2019
Jul 01 2019
Jan 01 2020
Jul 01 2020
Jan 01 2021
Jul 01 2021
Dec 31 2021


Daily Number of Flights − Qatar
2019−01−01 / 2021−12−31
400
350
300
250
200
150
100
Jan 01 2019
Jul 01 2019
Jan 01 2020
Jul 01 2020
Jan 01 2021
Jul 01 2021
Dec 31 2021


Daily Number of Flights − Qatar
2019−01−01 / 2021−12−31
Covid
Delta
Omicron
400
350
300
250
200
150
100
Jan 01 2019
Jul 01 2019
Jan 01 2020
Jul 01 2020
Jan 01 2021
Jul 01 2021
Dec 31 2021


Daily Number of Flights − Zambia
2019−01−01 / 2021−12−31
500
400
300
200
100
Jan 01 2019
Jul 01 2019
Jan 01 2020
Jul 01 2020
Jan 01 2021
Jul 01 2021
Dec 31 2021


Daily Number of Flights − Zambia
2019−01−01 / 2021−12−31
Covid
Delta
Omicron
500
400
300
200
100
Jan 01 2019
Jul 01 2019
Jan 01 2020
Jul 01 2020
Jan 01 2021
Jul 01 2021
Dec 31 2021

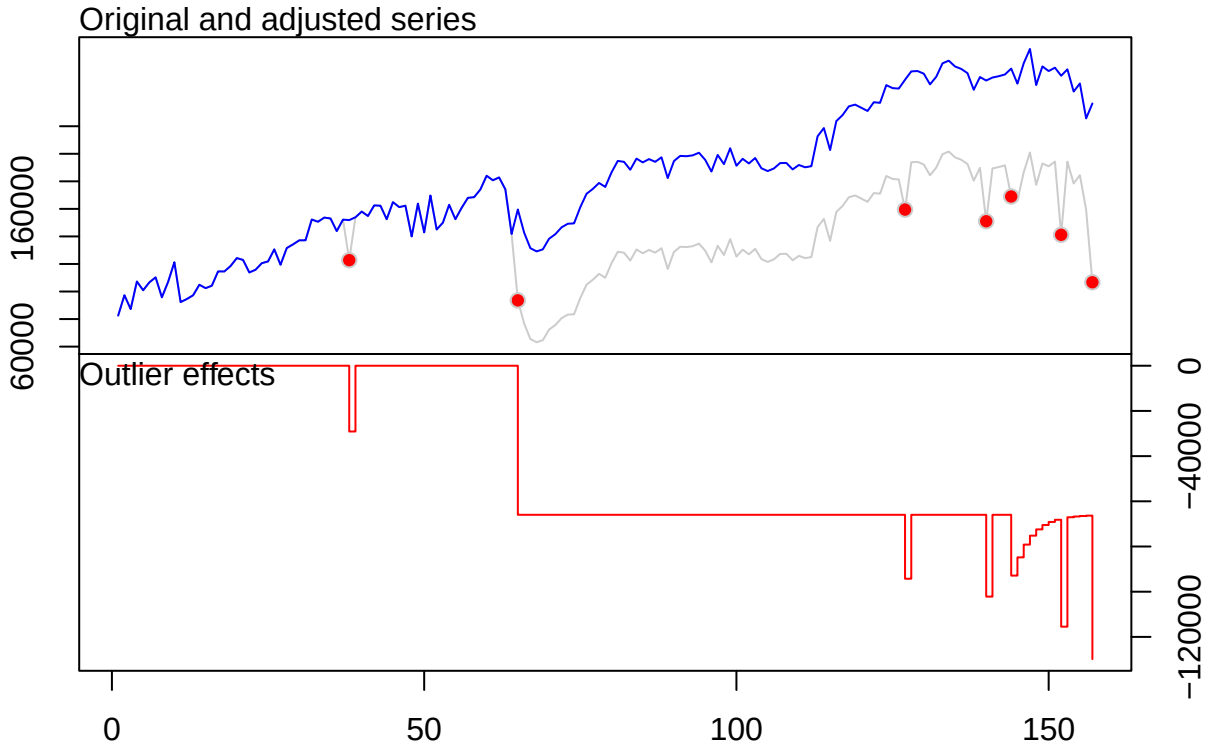
Change Point Detection

```
# Daily number of flights in the US (total: domestic + international), create xts object
US_daily_total <- data %>%
  filter(country == "United States of America") %>%
  select(day, total_flights)

US_daily_total$day <- as.Date(US_daily_total$day)
US_daily_total_xts <- xts(US_daily_total$total_flights, US_daily_total$day)
```

```
# Daily number of flights in the US (total: domestic + international), create ts object
US_daily_total <- data %>%
  filter(country == "United States of America") %>%
  select(day, total_flights)

US_daily_total_ts <- ts(US_daily_total$total_flights, start = c(2019, 1), frequency = 365)
```

```
US_weekly_total_xts <- apply.weekly(US_daily_total_xts, sum)
US_weekly_total_ts <- ts(US_weekly_total_xts, start = c(2019,1), frequency = 52)
```

```
library("tsoutliers")
US_weekly_total_ts_no_time <- ts(as.vector(US_weekly_total_ts), frequency = 1)
US_weekly_total_ts_no_time_outliers <- tso(US_weekly_total_ts_no_time)
# tiff("change_detection.tiff", units="in", width=6, height=4, res=300)
plot(US_weekly_total_ts_no_time_outliers)
```



```
# dev.off()
US_weekly_total_ts_no_time_outliers
```

```
Series: US_weekly_total_ts_no_time
Regression with ARIMA(1,1,0) errors

Coefficients:
          ar1        AO38        LS65       AO127       AO140      TC144
      -0.4282  -29094.787  -65960.868  -28305.502  -36211.771  -26908.69
s.e.   0.0784    7276.793    8517.141    7273.008    7287.801    8046.81
           AO152       AO157
      -47968.135  -63585.110
s.e.    7346.868    9156.898

sigma^2 = 84208535:  log likelihood = -1640.76
AIC=3299.51   AICc=3300.75   BIC=3326.96

Outliers:
  type ind time coefhat  tstat
1   AO  38   38  -29095 -3.998
2   LS  65   65  -65961 -7.744
3   AO 127  127  -28306 -3.892
4   AO 140  140  -36212 -4.969
5   TC 144  144  -26909 -3.344
6   AO 152  152  -47968 -6.529
7   AO 157  157  -63585 -6.944
```

Additive Outlier (AO) Innovation Outlier (IO) Level Shift (LS) Temporary change (TC) Seasonal Level Shift (SLS)

```
library(xtable)
xtable(US_weekly_total_ts_no_time_outliers$outliers)
```

```
% latex table generated in R 4.0.3 by xtable 1.8-4 package
% Thu Mar 17 17:44:46 2022
\begin{table}[ht]
\centering
\begin{tabular}{rlrrrr}
  \hline
 & type & ind & time & coefhat & tstat \\
  \hline
1 & AO &  38 & 38.00 & -29094.79 & -4.00 \\
  2 & LS &  65 & 65.00 & -65960.87 & -7.74 \\
  3 & AO & 127 & 127.00 & -28305.50 & -3.89 \\
  4 & AO & 140 & 140.00 & -36211.77 & -4.97 \\
  5 & TC & 144 & 144.00 & -26908.69 & -3.34 \\
  6 & AO & 152 & 152.00 & -47968.14 & -6.53 \\
  7 & AO & 157 & 157.00 & -63585.11 & -6.94 \\
   \hline
\end{tabular}
\end{table}
```

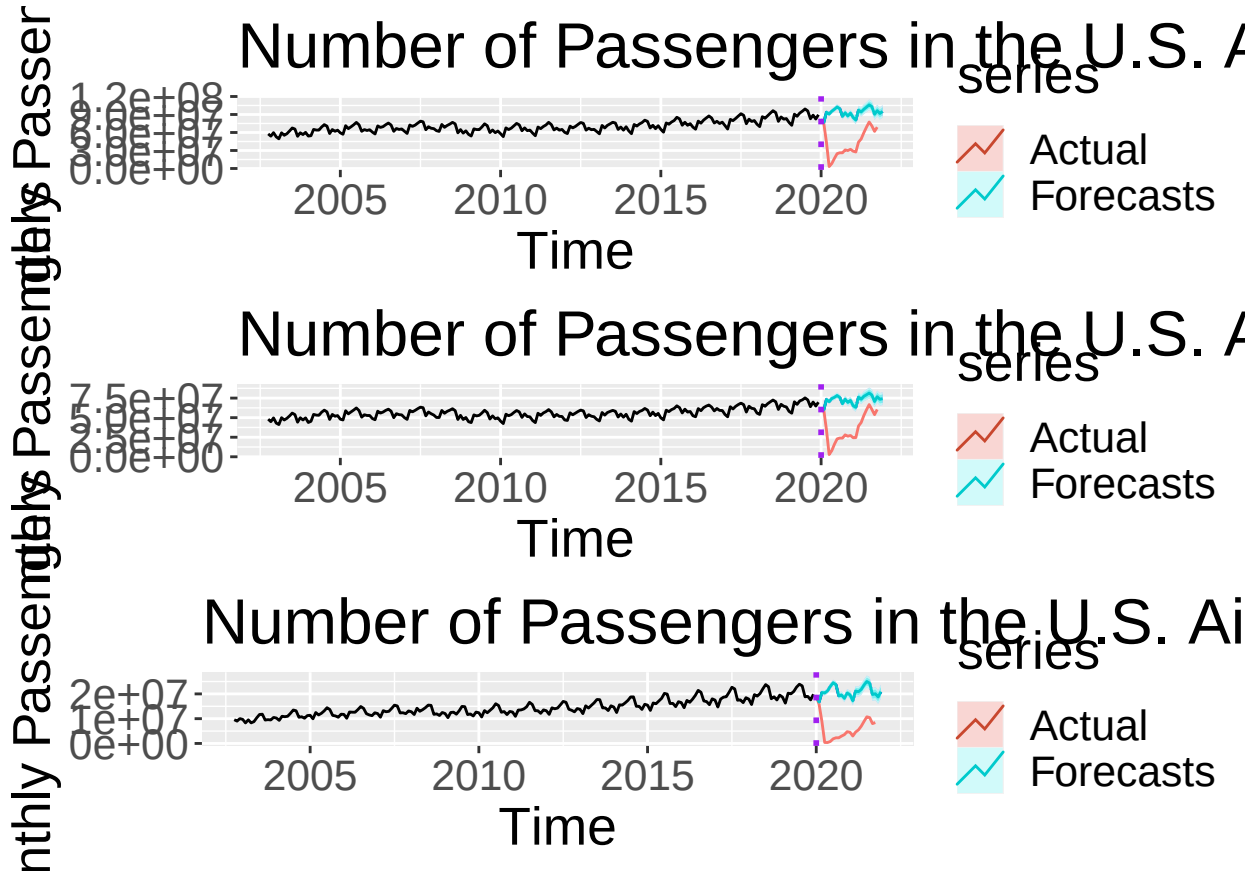
```r
library("strucchange")
breakpoints(US_weekly_total_ts_no_time~1)
```

```
     Optimal 5-segment partition:

Call:
breakpoints.formula(formula = US_weekly_total_ts_no_time ~ 1)

Breakpoints at observation number:
28 64 87 115

Corresponding to breakdates:
28 64 87 115
```

Number of Passengers in the US Airports

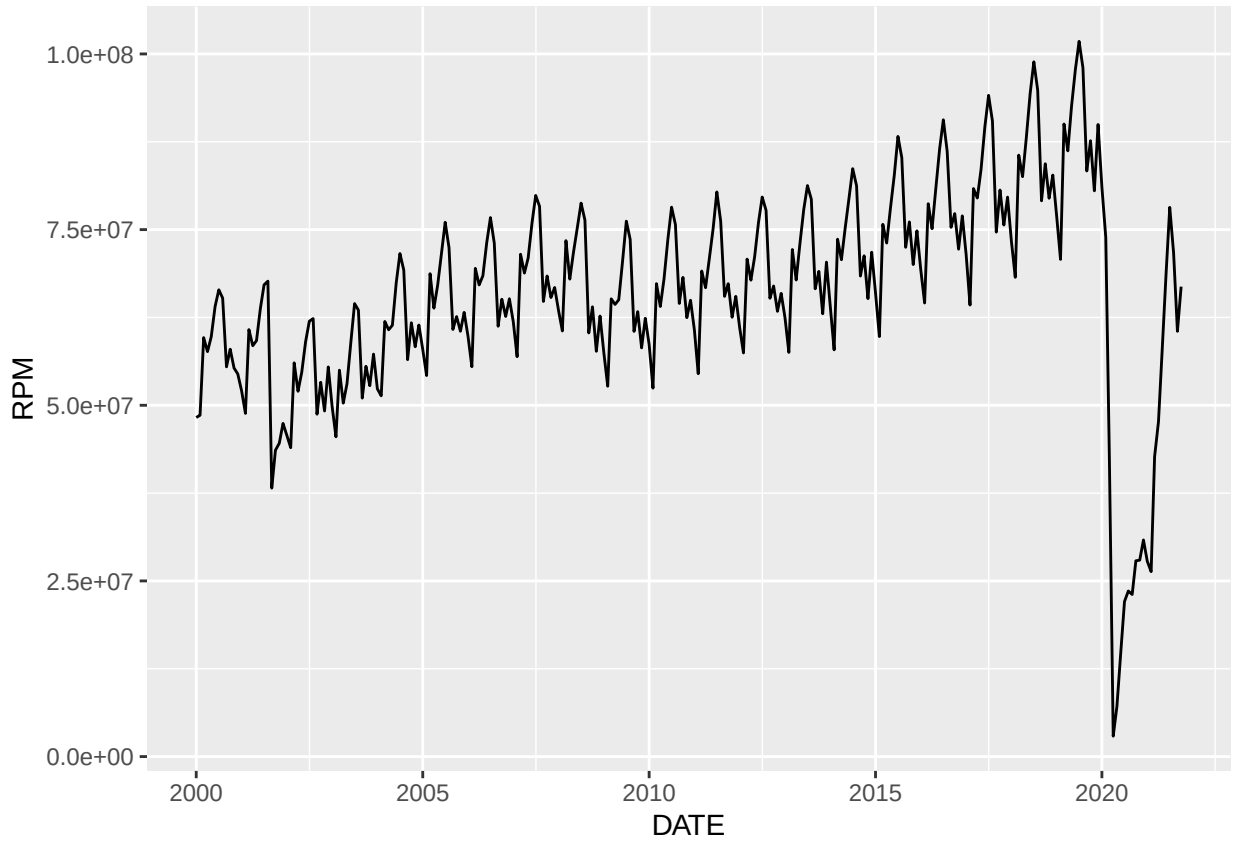
```
# ARIMA
# Total
arima_passengers_bf2020_total <- auto.arima(passengers_bf2020_total_ts)
arima_passengers_bf2020_total_forecast <- forecast(arima_passengers_bf2020_total, h = 24)

# Domestic
arima_passengers_bf2020_domestic <- auto.arima(passengers_bf2020_domestic_ts)
arima_passengers_bf2020_domestic_forecast <- forecast(arima_passengers_bf2020_domestic, h = 24)

# International
arima_passengers_bf2020_intl <- auto.arima(passengers_bf2020_intl_ts)
arima_passengers_bf2020_intl_forecast <- forecast(arima_passengers_bf2020_intl, h = 24)
```

```
# png(file="passenger_total.png", width=1000, height=600)
library(gridExtra)
total_plot <- arima_passengers_bf2020_total_forecast %>%
  autoplot() +
  autolayer(passengers_sc2020_total_ts, series = "Actual") +
  autolayer(arima_passengers_bf2020_total_forecast, series = "Forecasts", shadecols = "oldstyle") +
  geom_vline(xintercept = 2020, linetype = "dotted", color = "purple", size = 1) +
  ylab("Monthly Passengers") +
  ggtitle("Number of Passengers in the U.S. Airports - Total") +
  theme(text = element_text(size = 20))

domestic_plot <- arima_passengers_bf2020_domestic_forecast %>%
  autoplot() +
  autolayer(passengers_sc2020_domestic_ts, series = "Actual") +
  autolayer(arima_passengers_bf2020_domestic_forecast, series = "Forecasts", shadecols = "oldstyle") +
  geom_vline(xintercept = 2020, linetype = "dotted", color = "purple", size = 1) +
  ylab("Monthly Passengers") +
  ggtitle("Number of Passengers in the U.S. Airports - Domestic Only") +
  theme(text = element_text(size = 20))

intl_plot <- arima_passengers_bf2020_intl_forecast %>%
  autoplot() +
  autolayer(passengers_sc2020_intl_ts, series = "Actual") +
  autolayer(arima_passengers_bf2020_intl_forecast, series = "Forecasts", shadecols = "oldstyle") +
  geom_vline(xintercept = 2020, linetype = "dotted", color = "purple", size = 1) +
  ylab("Monthly Passengers") +
  ggtitle("Number of Passengers in the U.S. Airports - International Only") +
  theme(text = element_text(size = 20))

# png(file="passenger_domestic_intl.png", width=1000, height=900)
grid.arrange(total_plot, domestic_plot, intl_plot, nrow = 3)
```

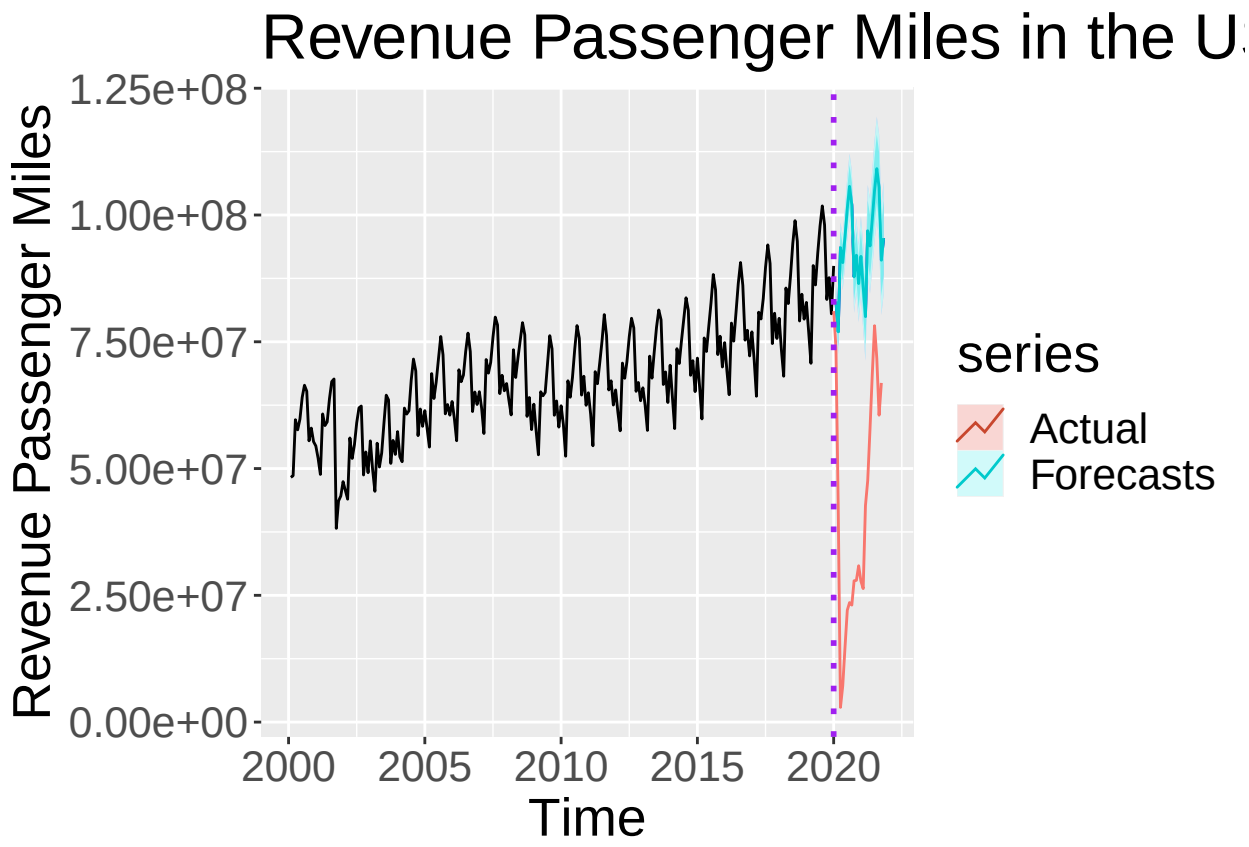



```
# dev.off()
```

```
revenue<- read_csv("RPM.csv")
ggplot(revenue, aes(DATE, RPM)) + geom_line()
```




```r
rev_bf2020 <- revenue %>% filter(DATE < as.Date("2020-01-01"))
rev_bf2020_total_ts <- ts(rev_bf2020$RPM, start = c(2000, 2), frequency = 12)

rev_sc2020 <- revenue %>% filter(DATE >= as.Date("2020-01-01"))
rev_sc2020_total_ts <- ts(rev_sc2020$RPM, start = c(2020, 1), frequency = 12)


arima_rev_bf2020 <- auto.arima(rev_bf2020_total_ts)
arima_rev_bf2020_total_forecast <- forecast(arima_rev_bf2020, h = 22)

# png(file="revenue.png", width=1000, height=300)
arima_rev_bf2020_total_forecast %>%
  autoplot() +
  autolayer(rev_sc2020_total_ts, series = "Actual") +
  autolayer(arima_rev_bf2020_total_forecast, series = "Forecasts", shadecols = "oldstyle") +
  geom_vline(xintercept = 2020, linetype = "dotted", color = "purple", size = 1) +
  ylab("Revenue Passenger Miles") +
  ggtitle("Revenue Passenger Miles in the US") +
  theme(text = element_text(size = 20))
```


Revenue Passenger Miles in the U
Revenue Passenger Miles
1.25e+08
1.00e+08
7.50e+07
5.00e+07
2.50e+07
0.00e+00
2000
2005
2010
2015
2020
Time
series
Actual
Forecasts

```
# dev.off()
```

Number of Flights in the vs. Confirmed Cases

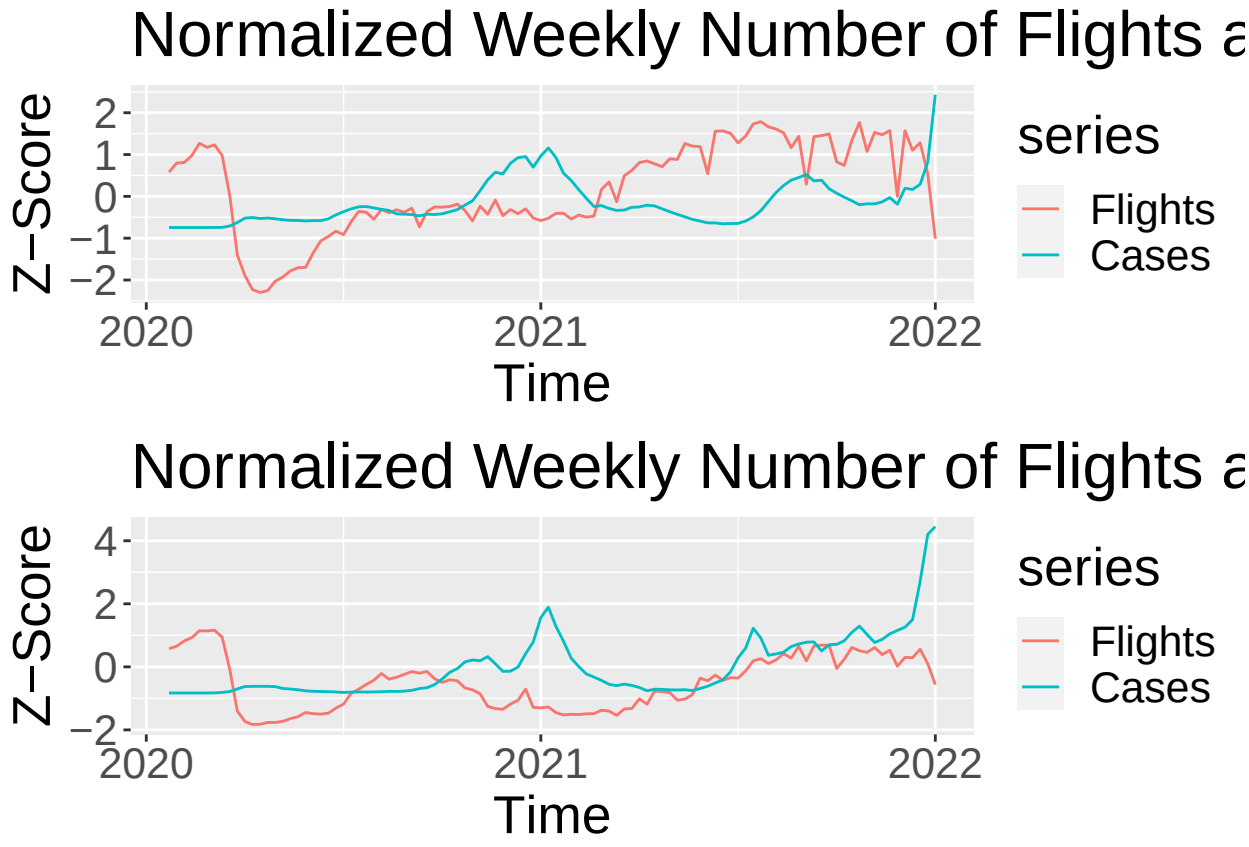
```
US_daily_covid_cum <- read_csv("US_cumulative.csv")
US_daily_covid_cum_ts <- ts(US_daily_covid_cum$each_day_US)
US_daily_covid_ts <- ts(US_daily_covid_cum_ts - stats::lag(US_daily_covid_cum_ts, -1), start = c(2020,23
# plot(US_daily_covid_ts)

# xts object
US_daily_covid_date <- as.Date(US_daily_covid_cum$X1[-1], format = "%m/%d/%y")
US_daily_covid_ts_xts <- xts(as.vector(US_daily_covid_ts), US_daily_covid_date)
```

```
US_weekly_covid_xts <- apply.weekly(US_daily_covid_ts_xts, sum)

US_weekly_total_xts_normal <- (US_weekly_total_xts - mean(US_weekly_total_xts)) / sd(US_weekly_total_xts
US_weekly_covid_xts_normal <- (US_weekly_covid_xts - mean(US_weekly_covid_xts)) / sd(US_weekly_covid_xts
```

```
US_weekly_total_ts_normal <- ts(US_weekly_total_xts_normal, start = c(2019,1), frequency = 52)
US_weekly_covid_ts_normal <- ts(US_weekly_covid_xts_normal, start = c(2020,4), frequency = 52)
```

```
df <- cbind(US_weekly_total_ts_normal, US_weekly_covid_ts_normal)
df <- tseries::na.remove(df)

# png(file="us_flights_cases.png", width=1000, height=600)
us_flights_cases <- autoplot(df) +
  ylab("Z-Score") +
  ggtitle("Normalized Weekly Number of Flights and Confirmed Cases in the US") +
  scale_x_continuous(breaks = seq(2020, 2022, by = 1)) +
  scale_colour_discrete(labels = c("Flights", "Cases")) +
  theme(text = element_text(size = 20))
# dev.off()



# Daily number of flights in the UK (total: domestic + international), create xts object
UK_daily_total <- data %>%
  filter(country == "United Kingdom") %>%
  select(day, total_flights)

UK_daily_total$day <- as.Date(UK_daily_total$day)
UK_daily_total_xts <- xts(UK_daily_total$total_flights, UK_daily_total$day)
UK_weekly_total_xts <- apply.weekly(UK_daily_total_xts, sum, na.rm=TRUE)

UK_weekly_total_ts <- ts(UK_weekly_total_xts, start = c(2019,1), frequency = 52)

UK_daily_covid_cum <- read_csv("UKCases.csv")
UK_daily_covid_cum_ts <- ts(UK_daily_covid_cum$`United Kingdom`)
UK_daily_covid_ts <- ts(UK_daily_covid_cum_ts - stats::lag(UK_daily_covid_cum_ts, -1)
                        ,start = c(2020,23), frequency = 365)
# plot(US_daily_covid_ts)

# xts object
UK_daily_covid_date <- as.Date(UK_daily_covid_cum$day[-1], format = "%m/%d/%y")
```

```r
UK_daily_covid_ts_xts <- xts(as.vector(UK_daily_covid_ts), UK_daily_covid_date)



UK_weekly_covid_xts <- apply.weekly(UK_daily_covid_ts_xts, sum)
UK_weekly_total_xts_normal <- (UK_weekly_total_xts - mean(UK_weekly_total_xts)) / sd(UK_weekly_total_xts
UK_weekly_covid_xts_normal <- (UK_weekly_covid_xts - mean(UK_weekly_covid_xts)) / sd(UK_weekly_covid_xts
UK_weekly_total_ts_normal <- ts(UK_weekly_total_xts_normal, start = c(2019,1), frequency = 52)
UK_weekly_covid_ts_normal <- ts(UK_weekly_covid_xts_normal, start = c(2020,4), frequency = 52)
df_uk <- cbind(UK_weekly_total_ts_normal, UK_weekly_covid_ts_normal)
df_uk <- tseries::na.remove(df_uk)
uk_flights_cases <- autoplot(df_uk) +
  ylab("Z-Score") +
  ggtitle("Normalized Weekly Number of Flights and Confirmed Cases in the UK") +
  scale_x_continuous(breaks = seq(2020, 2022, by = 1)) +
  scale_colour_discrete(labels = c("Flights", "Cases")) +
  theme(text = element_text(size = 20))

# png(file="us_uk_flights_cases.png", width=1000, height=600)
grid.arrange(us_flights_cases, uk_flights_cases, nrow = 2)
```




```r
# dev.off()
```

```
# UK
# confirmed_GLOBAL_raw
```