# FLT Forecasting

Stanley Hsu

2024-02-27

```r
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
rescaled = read_csv("data_rescaled.csv")
```

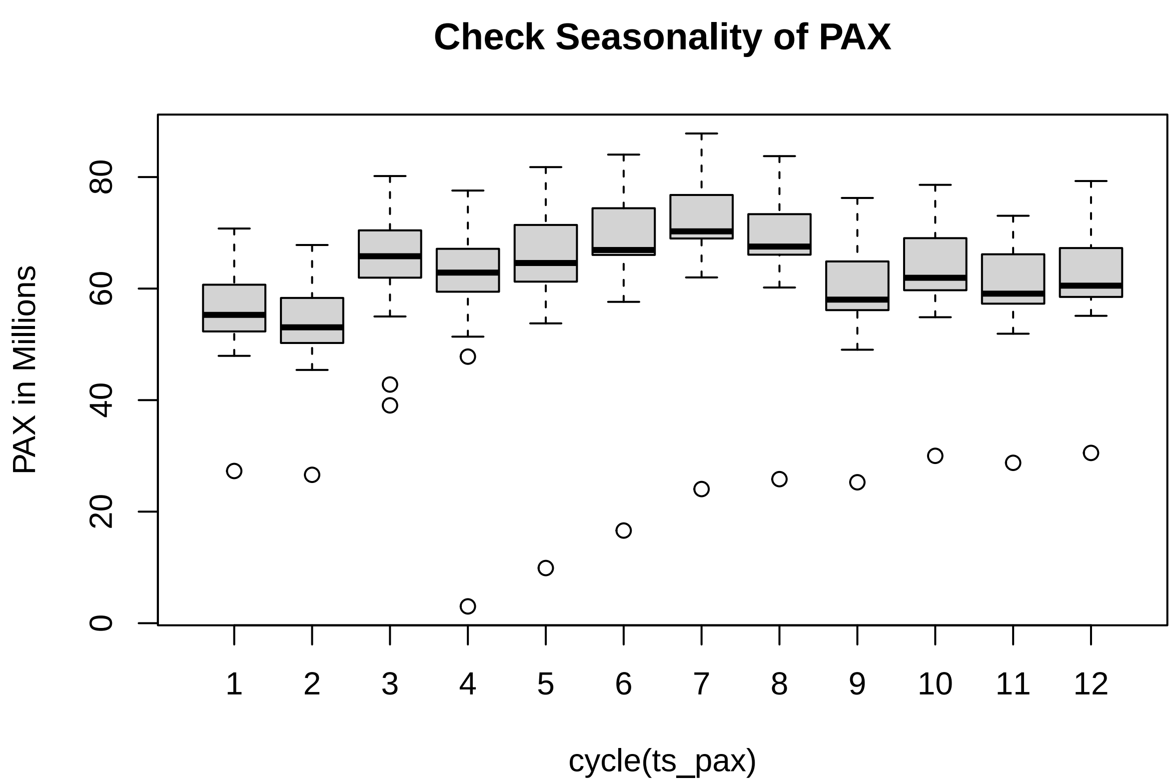
```
## New names:
## Rows: 249 Columns: 18
## -- Column specification
## -------------------------------------------------------- Delimiter: "," dbl
## (18): ...1, Year, Month, Dom_Pax (in millions), Int_Pax (in millions), P...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```r
rescaled
```

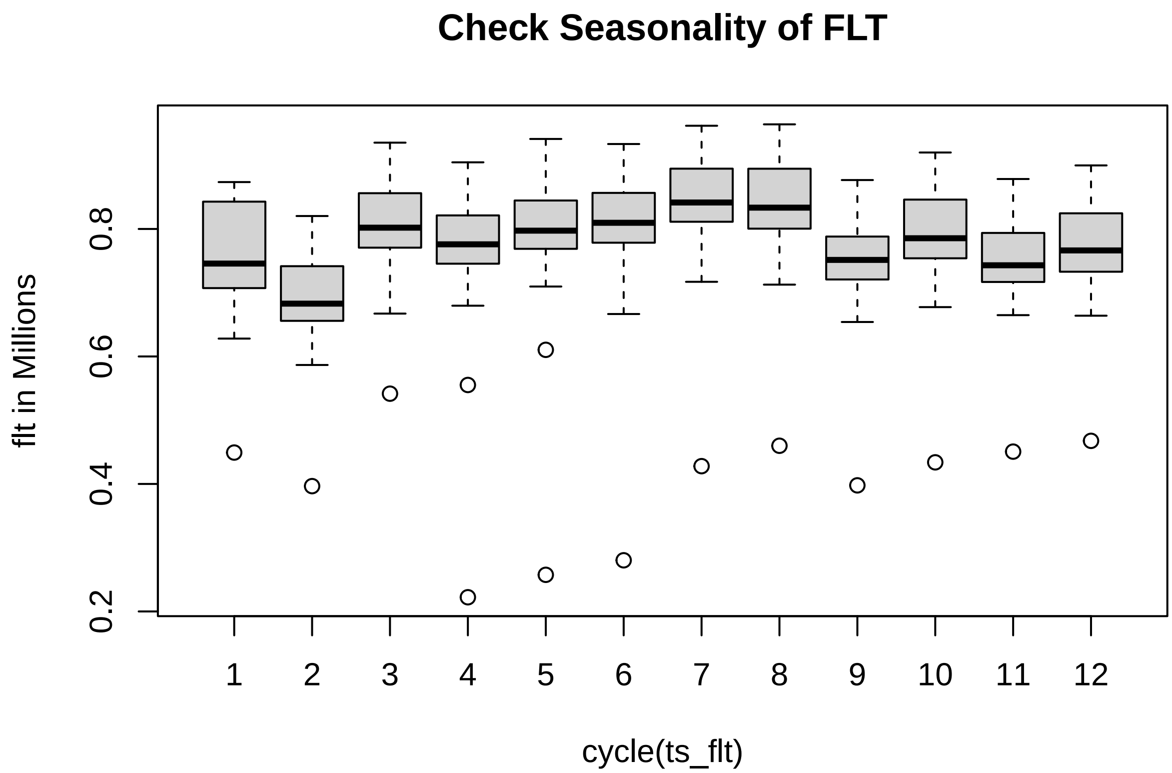
```
## # A tibble: 249 x 18
##     ...1  Year Month `Dom_Pax (in millions)` `Int_Pax (in millions)`
##    <dbl> <dbl> <dbl>                   <dbl>                   <dbl>
##  1     1  2003     1                    43.0                    4.91
##  2     2  2003     2                    41.2                    4.25
##  3     3  2003     3                    50.0                    5.01
##  4     4  2003     4                    47.0                    4.35
##  5     5  2003     5                    49.2                    4.61
##  6     6  2003     6                    52.2                    5.41
##  7     7  2003     7                    55.8                    6.19
##  8     8  2003     8                    53.9                    6.27
##  9     9  2003     9                    44.2                    4.82
```

```
## 10    10  2003    10                        49.9                        4.92
## # i 239 more rows
## # i 13 more variables: 'Pax (in millions)' <dbl>,
## #   'Dom_Flt (in millions)' <dbl>, 'Int_Flt (in millions)' <dbl>,
## #   'Flt (in millions)' <dbl>, 'Dom_RPM (in millions)' <dbl>,
## #   'Int_RPM (in millions)' <dbl>, 'RPM (in millions)' <dbl>,
## #   'Dom_ASM (in millions)' <dbl>, 'Int_ASM (in millions)' <dbl>,
## #   'ASM (in millions)' <dbl>, Dom_LF <dbl>, Int_LF <dbl>, LF <dbl>
```

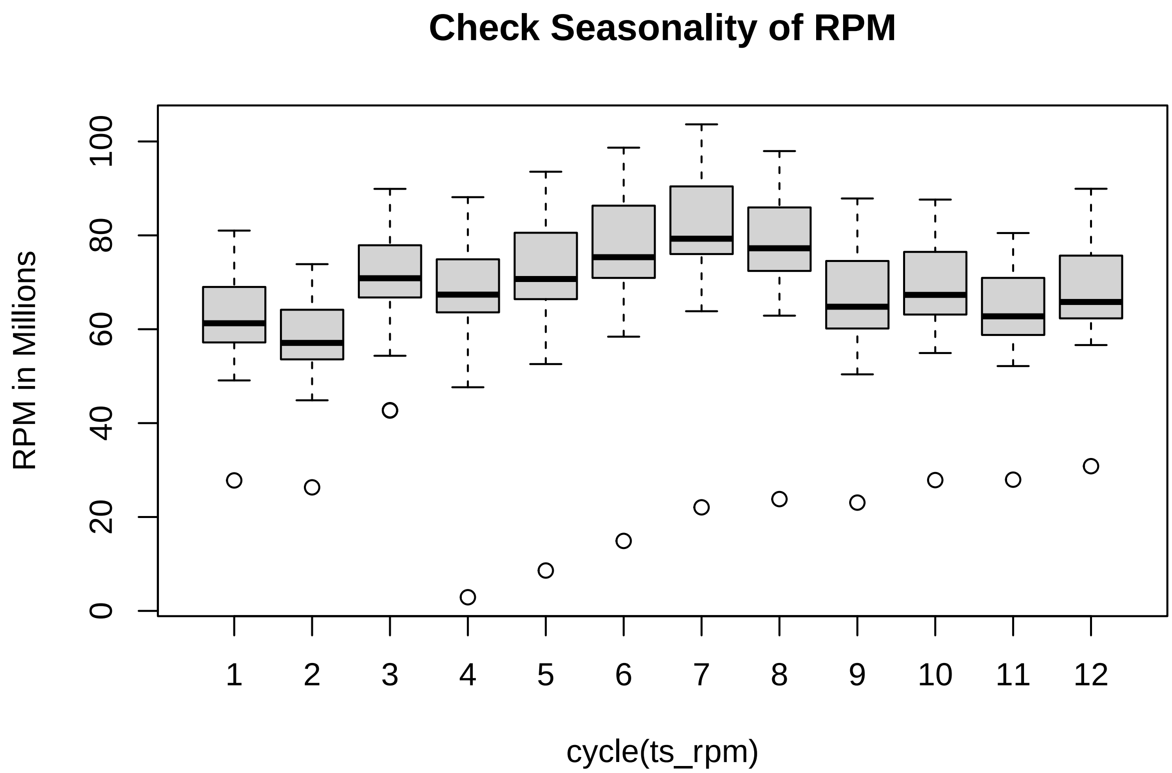
```r
ts_pax = ts(rescaled$`Pax (in millions)`, start = c(2003, 1), end = c(2023, 9),
            frequency = 12)
boxplot(ts_pax~cycle(ts_pax), main="Check Seasonality of PAX", ylab="PAX in Millions")
```



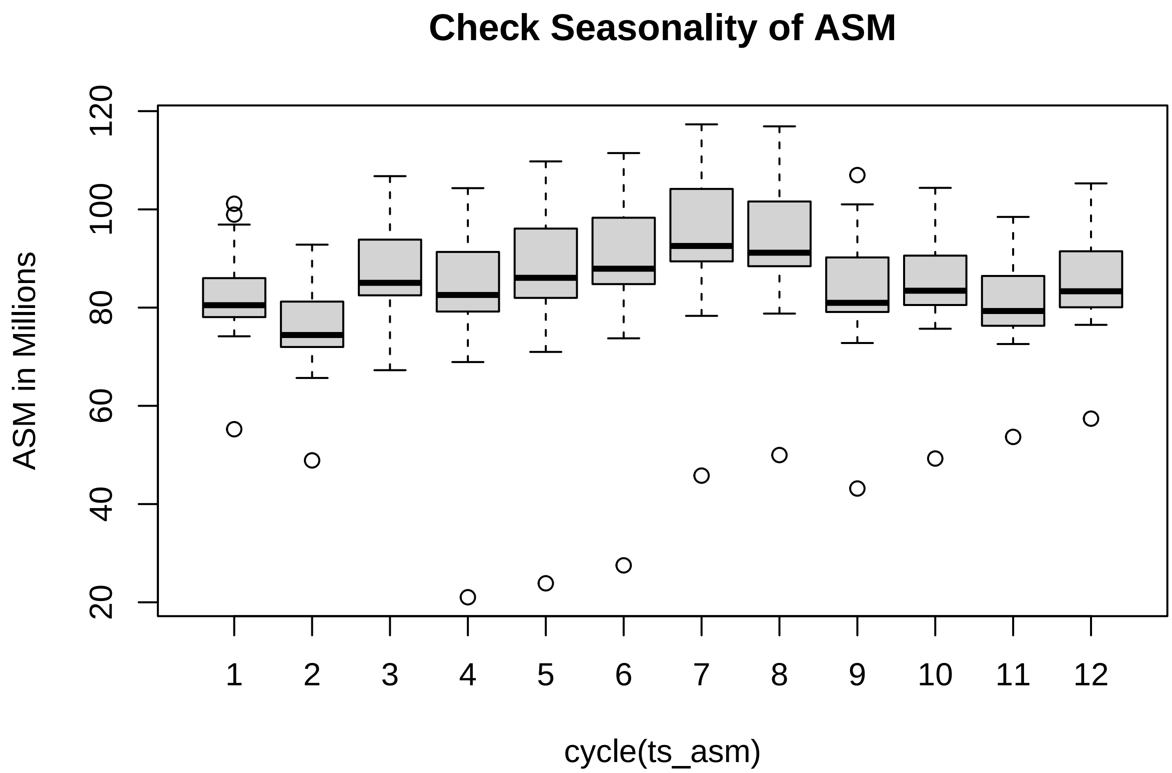

```r
ts_flt = ts(rescaled$`Flt (in millions)`, start = c(2003, 1), end = c(2023, 9),
            frequency = 12)
boxplot(ts_flt~cycle(ts_flt), main="Check Seasonality of FLT", ylab="flt in Millions")
```



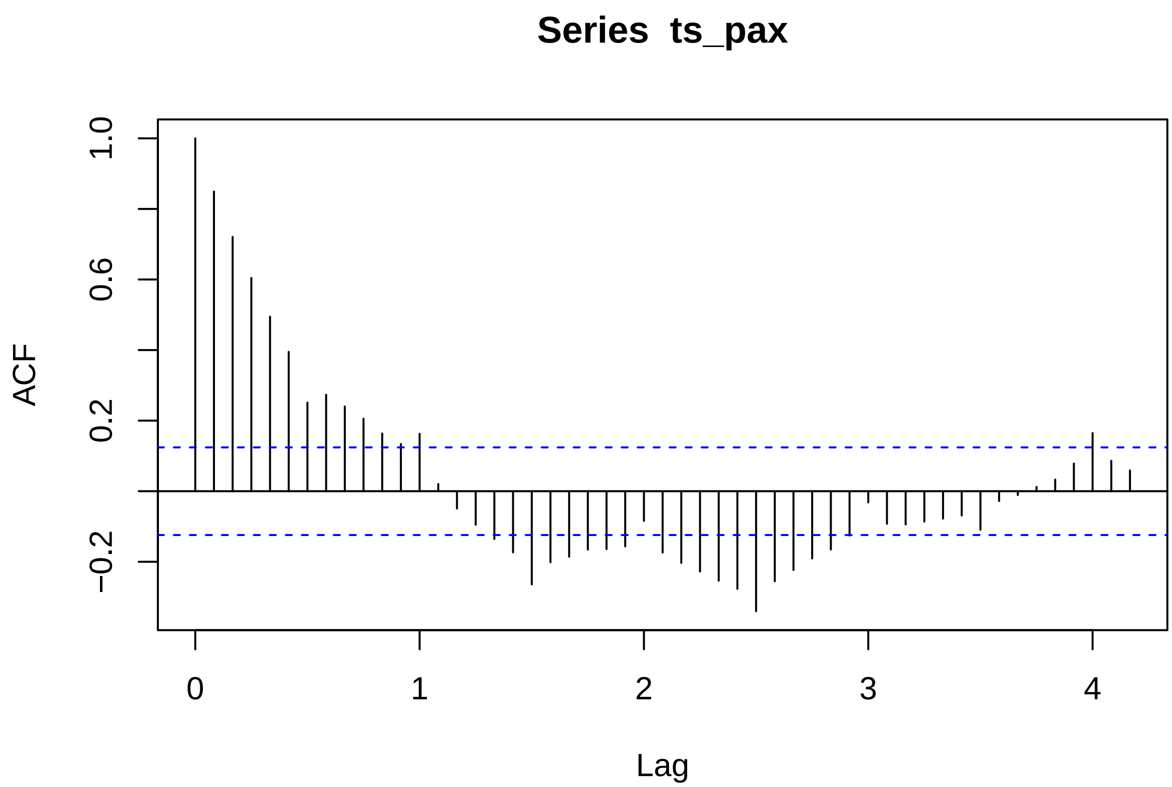

```
ts_rpm = ts(rescaled$`RPM (in millions)`, start = c(2003, 1), end = c(2023, 9),
            frequency = 12)
boxplot(ts_rpm~cycle(ts_rpm), main="Check Seasonality of RPM", ylab="RPM in Millions")
```



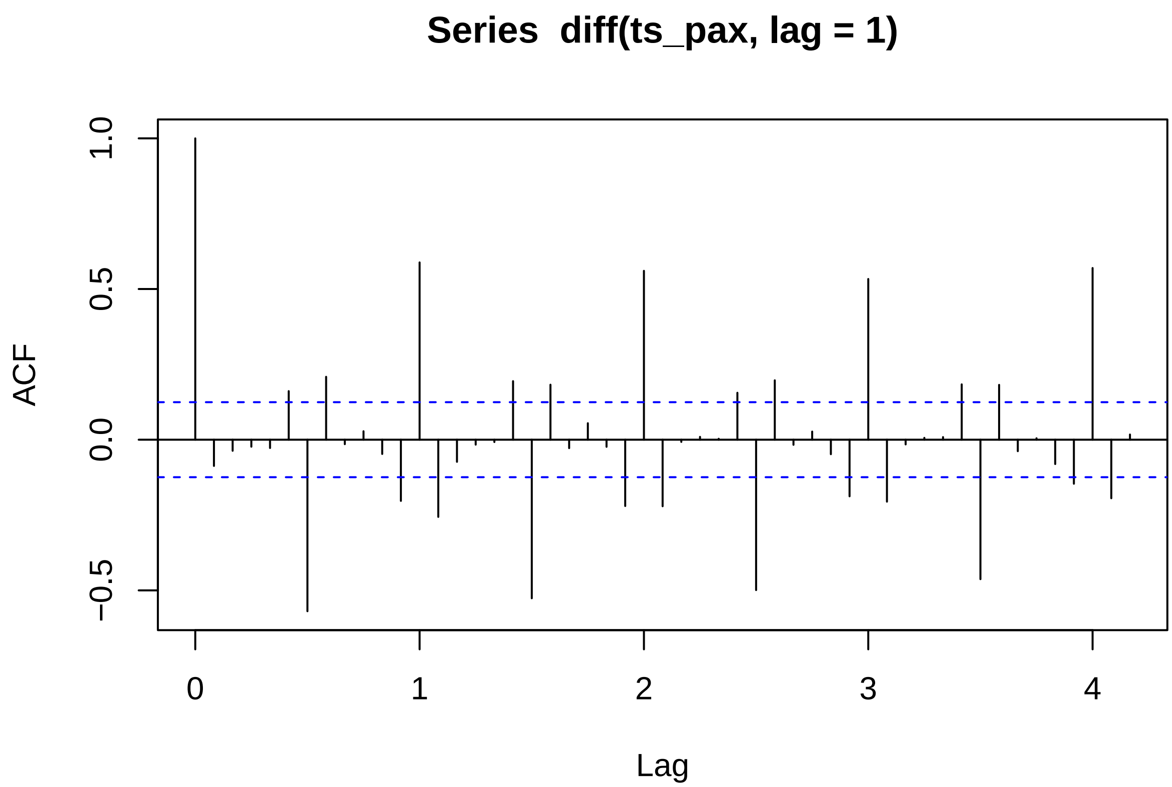

```
ts_asm = ts(rescaled$`ASM (in millions)`, start = c(2003, 1), end = c(2023, 9),
            frequency = 12)
boxplot(ts_asm~cycle(ts_asm), main="Check Seasonality of ASM", ylab="ASM in Millions")
```



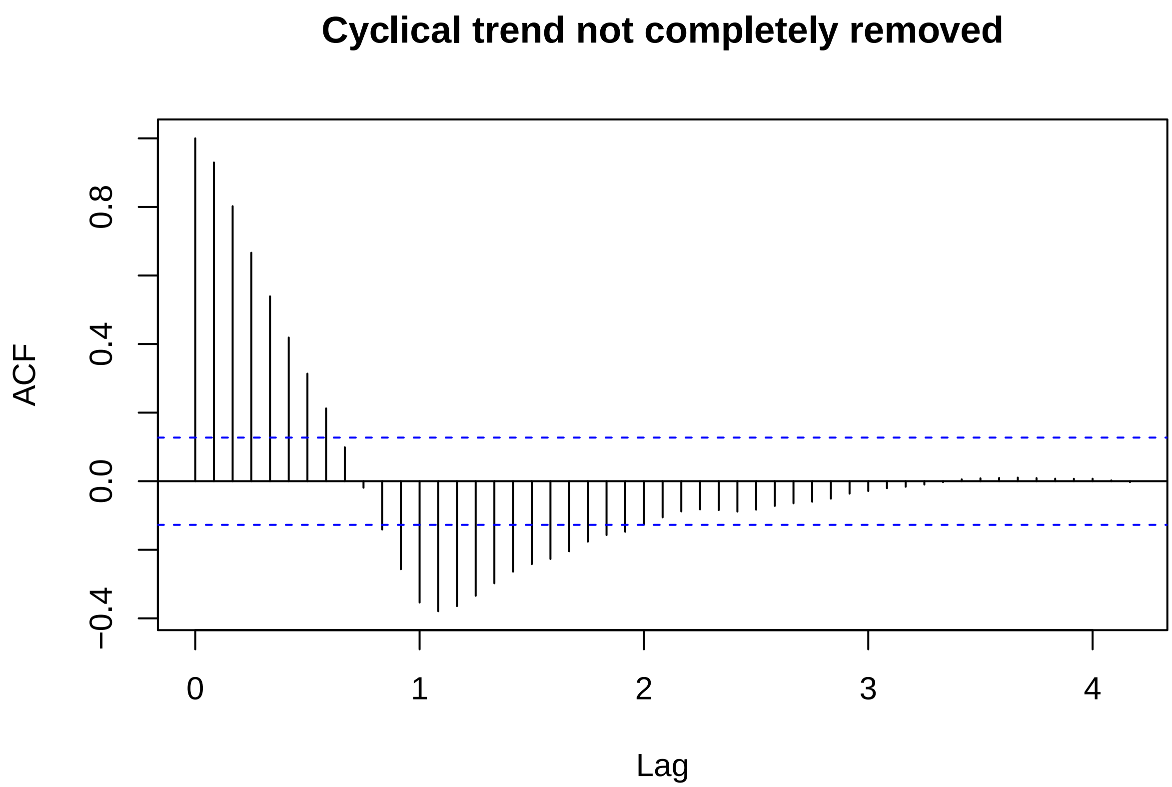

```r
acf(ts_pax, lag.max=50)
```



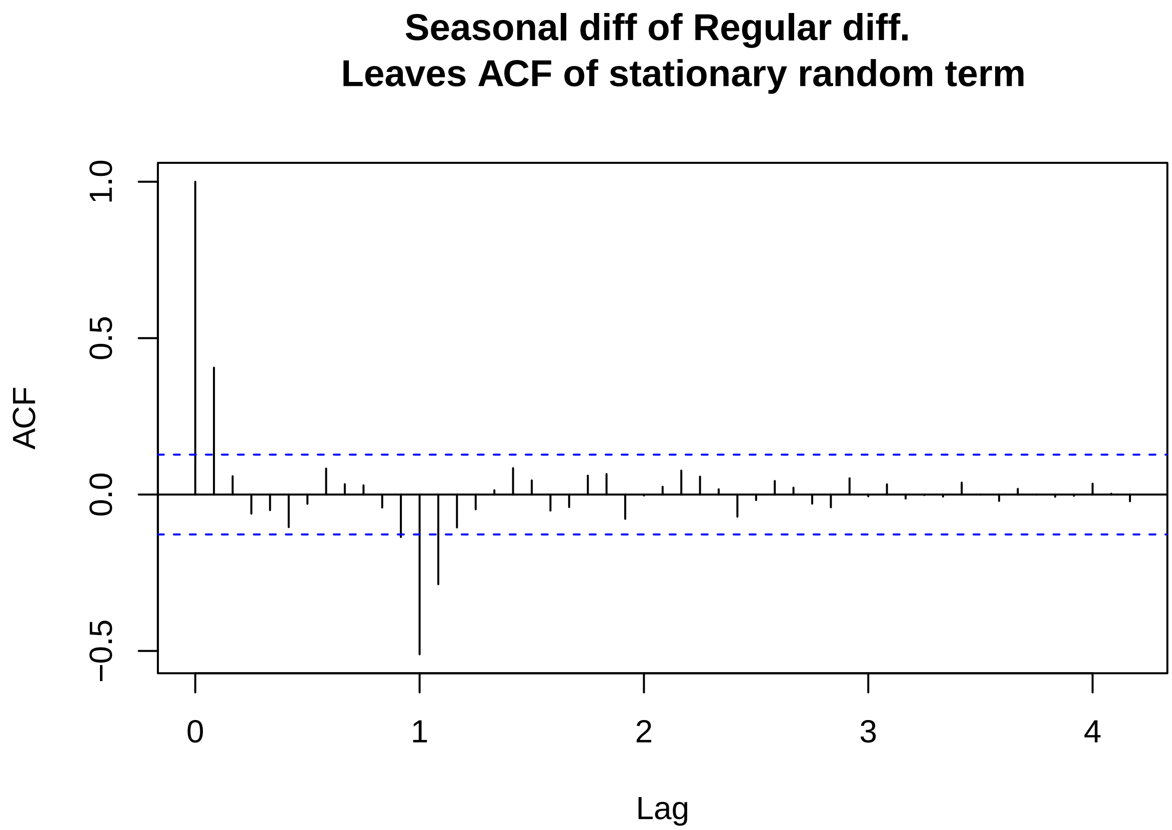

```
acf(diff(ts_pax, lag=1), lag.max=50)
```




```r
acf(diff(ts_pax, lag=12), lag.max=50, main="Cyclical trend not completely removed")
```




```
# Do seasonal differencing of the regular differencing
acf(diff(diff(ts_pax, lag=1), lag=12), lag.max=50, main="Seasonal diff of Regular diff.
    Leaves ACF of stationary random term" )
```




Forecast PAX (# of Passengers)

The idea is to use pre-covid data to train model and forecast

```
ts_pax = ts(rescaled$`Pax (in millions)`, start = c(2003, 1), end = c(2023, 9),
            frequency = 12)
# use data before 2019/1 as training
training=window(ts_pax, end=c(2019,1), frequency=12)
# use data from 2019/2 to 2020/2 as test
test=window(ts_pax, start=c(2019,2), end=c(2020,2), frequency=12)
test
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2019          63.55123 80.18159 76.39926 81.31647 83.80858 86.92585 83.75119
## 2020 70.76777 67.81321
##           Sep      Oct      Nov      Dec
## 2019 72.56932 78.60060 73.05753 79.28466
## 2020
```

```
model1=arima(training,order=c(0,1,1), seas=list(order=c(0,1,1), 12))
model1
```

```
##
## Call:
## arima(x = training, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1),
##     12))
```

```
##
## Coefficients:
##           ma1     sma1
##       -0.4600  -0.6600
## s.e.   0.0599   0.0801
##
## sigma^2 estimated as 1.37:  log likelihood = -287.27,  aic = 580.55
```

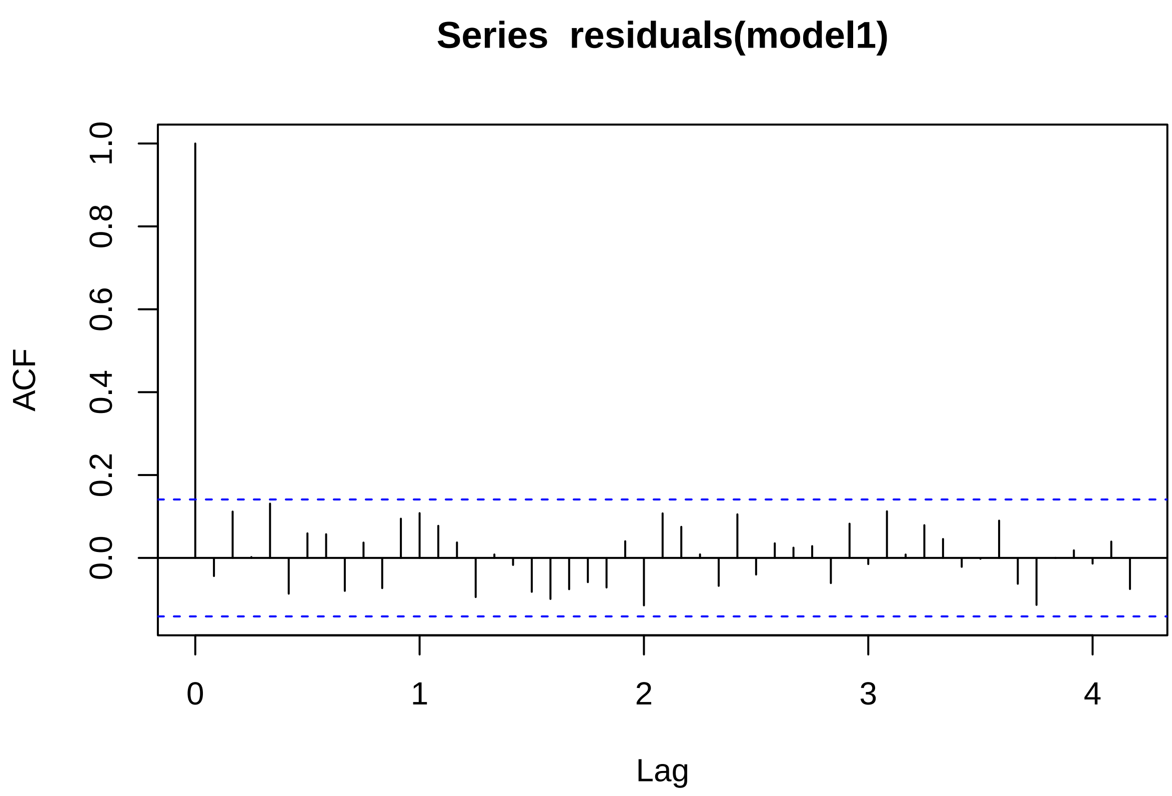
```
RMSE.of.fit = sqrt(model1$sigma2)
RMSE.of.fit
```

```
## [1] 1.17034
```

```
range(rescaled$`Pax (in millions)`)
```

```
## [1]  3.013899 87.810772
```

```
acf(residuals(model1), lag.max=50)
```

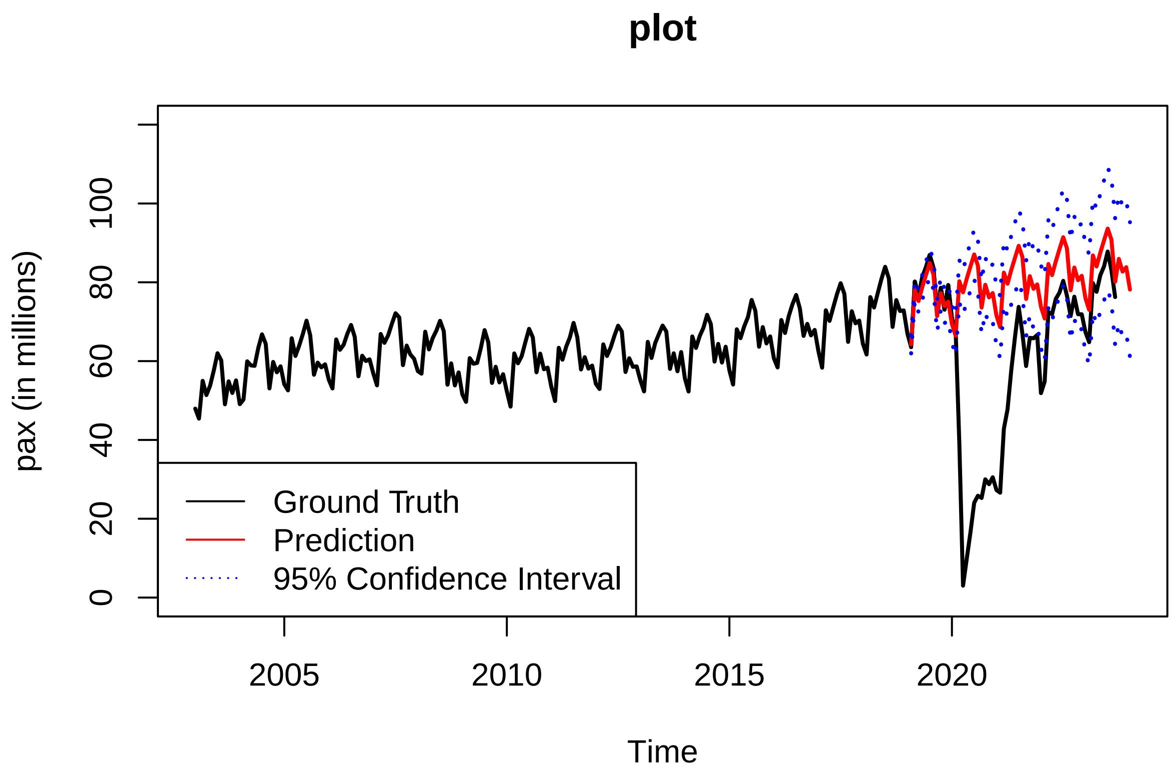



```
forecast=predict(model1,n.ahead=60,se.fit=TRUE)

pcil=ts((forecast$pred-1.96*forecast$se),start=c(2019,2),freq=12)
pciu=ts((forecast$pred+1.96*forecast$se),start=c(2019,2),freq=12)
```

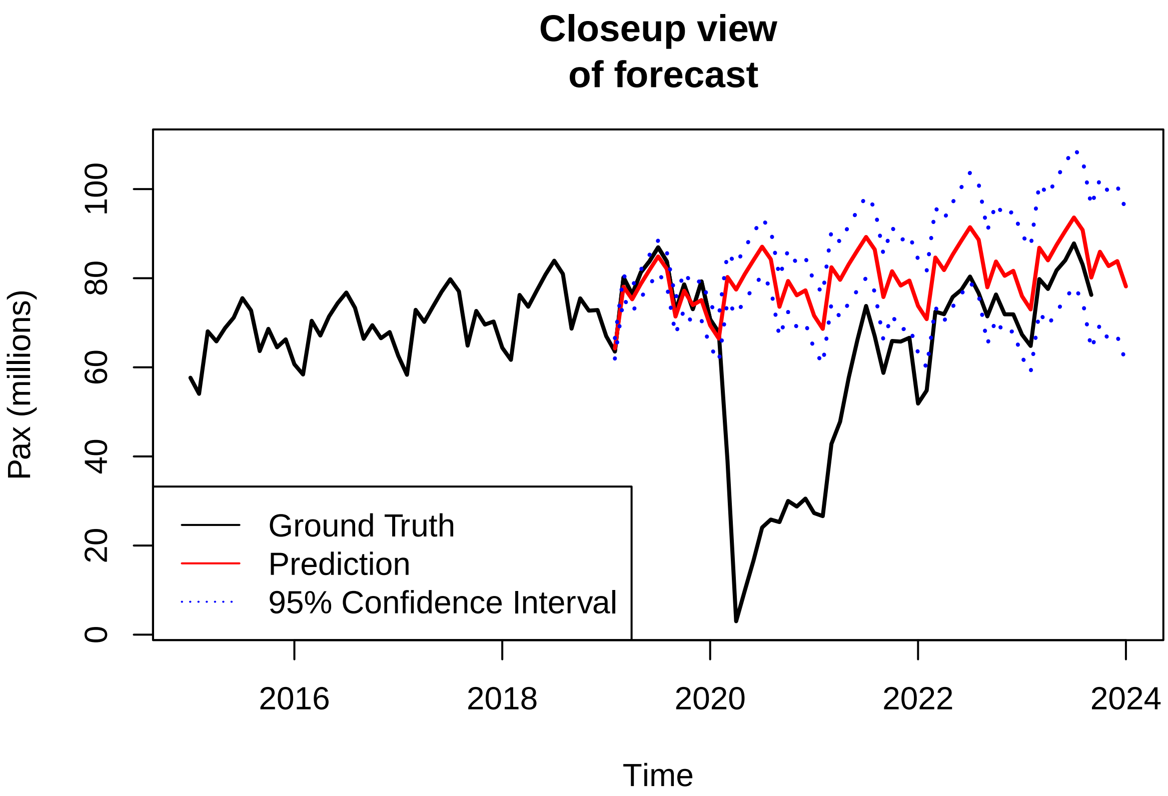
```r
predictions=ts(cbind(pcil, test, forecast$pred, pciu), start=c(2019,2),frequency=12)


ts.plot(cbind(ts_pax, test, forecast$pred, pcil, pciu), ylim = c(0, 120),
        lty=c(1,1,1,3,3),
        col=c("black","black","red","blue","blue"),main="plot",
        ylab="pax (in millions)", lwd=2)
legend("bottomleft", legend = c("Ground Truth", "Prediction", "95% Confidence Interval"),
       col = c("black", "red", "blue"), lty = c(1, 1, 3))
```




```r
#ts.plot(ts_pax, lwd = 2, ylim = c(0, 110), ylab = "PAX (in millions)", main = "Original Data")
shortts=window(ts_pax, start=c(2015,1), end=c(2023,9), freq=12)
ts.plot(cbind(shortts,test,forecast$pred,pcil,pciu),
        lty=c(1,1,1,3,3),
        col=c("black","black","red","blue","blue"),
        main="Closeup view\n of forecast", lwd=2,
        ylab="Pax (millions)")

legend("bottomleft", legend = c("Ground Truth", "Prediction", "95% Confidence Interval"),
       col = c("black", "red", "blue"), lty = c(1, 1, 3))
```




```
summary(ts_pax)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    3.014   57.665   63.899   62.602   69.447   87.811
```