DATA 598 Assignment 2

Shrusti Ghela

5/4/2022

Forecast the aus_airpassengers dataset in R package fpp3. (If you use this series, use the last 10 years as the test set.)

```
library(fpp3)
```

```
## -- Attaching packages -------------------------------------------- fpp3 0.4.0 --
```

```
## v tibble      3.1.6     v tsibble     1.1.1
## v dplyr       1.0.8     v tsibbledata 0.4.0
## v tidyr       1.2.0     v feasts      0.2.2
## v lubridate   1.8.0     v fable       0.3.1
## v ggplot2     3.3.5
```

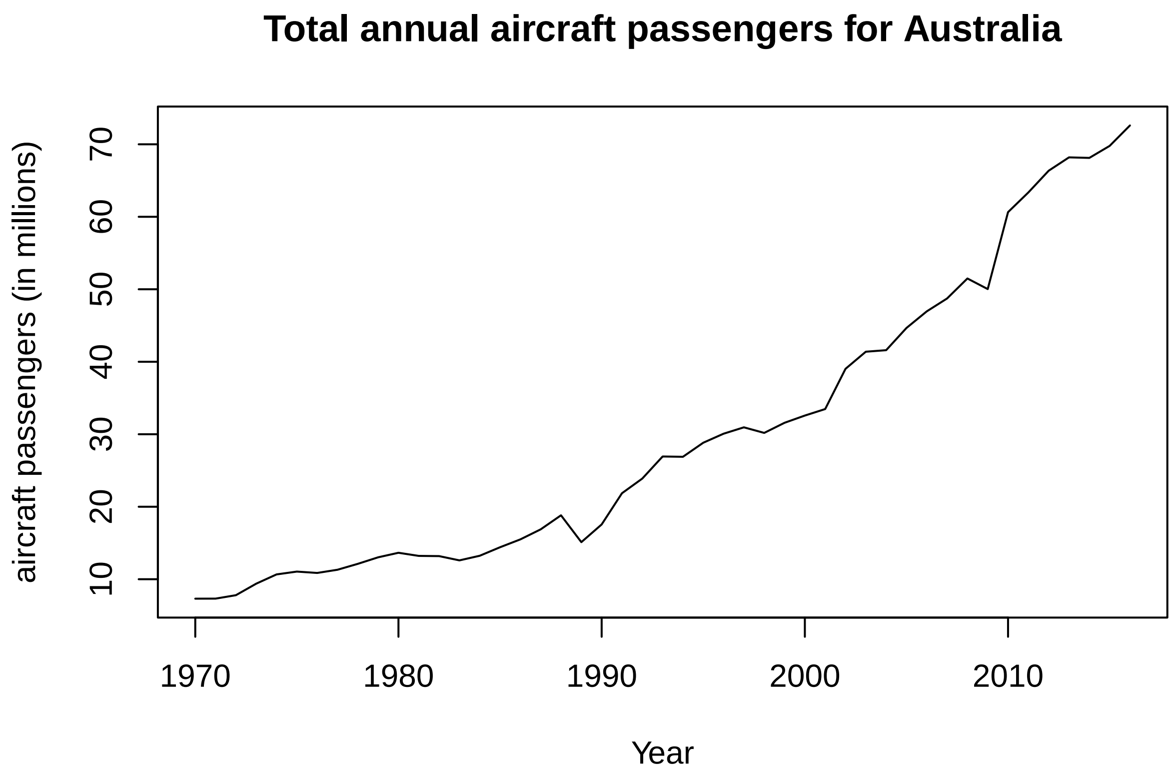
```
## -- Conflicts ------------------------------------------------ fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```
head(aus_airpassengers)
```

```
## # A tsibble: 6 x 2 [1Y]
##    Year Passengers
##   <dbl>      <dbl>
## 1  1970       7.32
## 2  1971       7.33
## 3  1972       7.80
## 4  1973       9.38
## 5  1974      10.7
## 6  1975      11.1
```

1. Split the series into train and test sets [1 point].

```
air = ts(aus_airpassengers$Passengers, freq=1, start=c(1970))
plot(air, , ylab="aircraft passengers (in millions)", xlab="Year", main="Total annual aircraft passenge
```




```
end(air)
```

```
## [1] 2016    1
```

```
air_train <- window(air, end = c(2006))
air_test <- window(air, start = c(2007))
```

2. Identify an ARIMA model for the train set and justify it. At a minimum, consider the model residuals; you may also consider other evidence [1 **point**]

By simply eyeballing, we see that the data is not stationary To verify this, check this using KPSS test.

KPSS test H_0 : Series is stationary H_α :Series is NOT stationary

```
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
kpss.test(air_train)
```

```
## Warning in kpss.test(air_train): p-value smaller than printed p-value
```

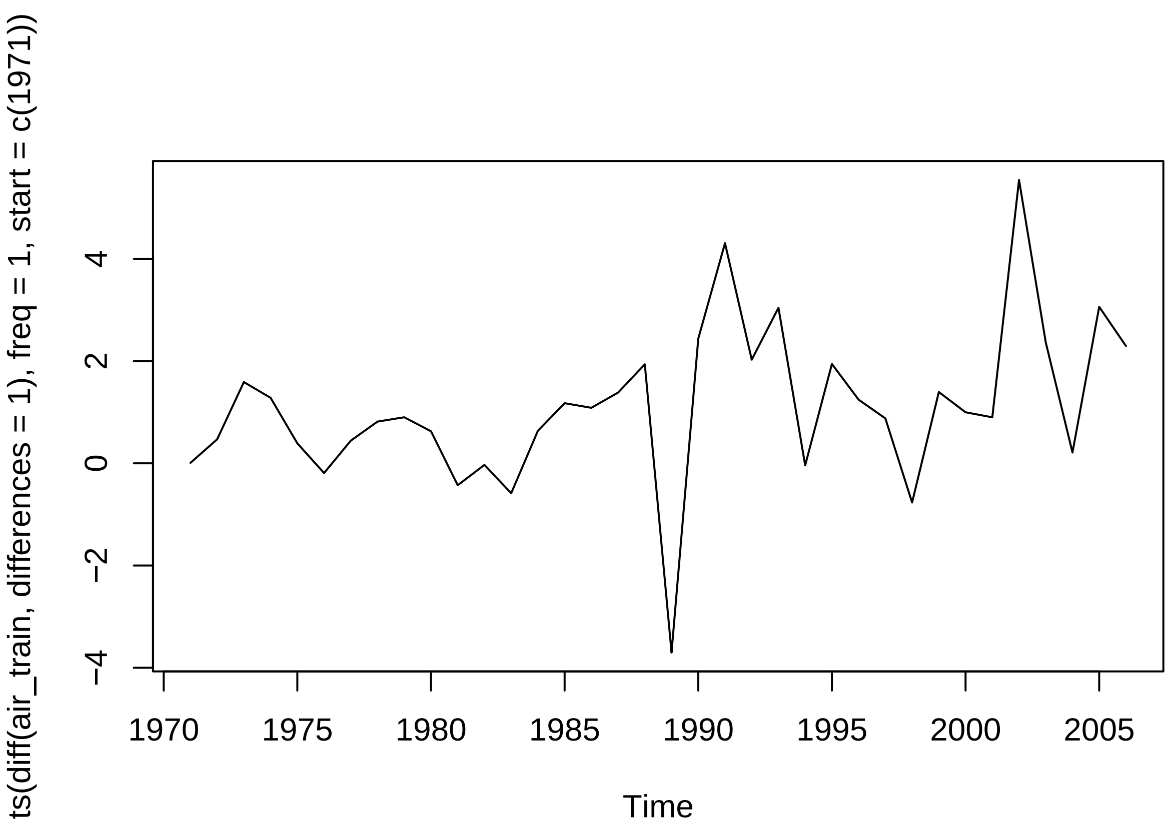
```
##
##  KPSS Test for Level Stationarity
##
## data:  air_train
## KPSS Level = 0.98017, Truncation lag parameter = 3, p-value = 0.01
```

p-value $= 0.01$, p-value < 0.05, $\implies$ we reject the null-hypothesis

```r
kpss.test(ts(diff(air_train, differences = 1), freq=1, start=c(1971)))
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  ts(diff(air_train, differences = 1), freq = 1, start = c(1971))
## KPSS Level = 0.46754, Truncation lag parameter = 3, p-value = 0.04898
```

```r
plot(ts(diff(air_train, differences = 1), freq=1, start=c(1971)))
```

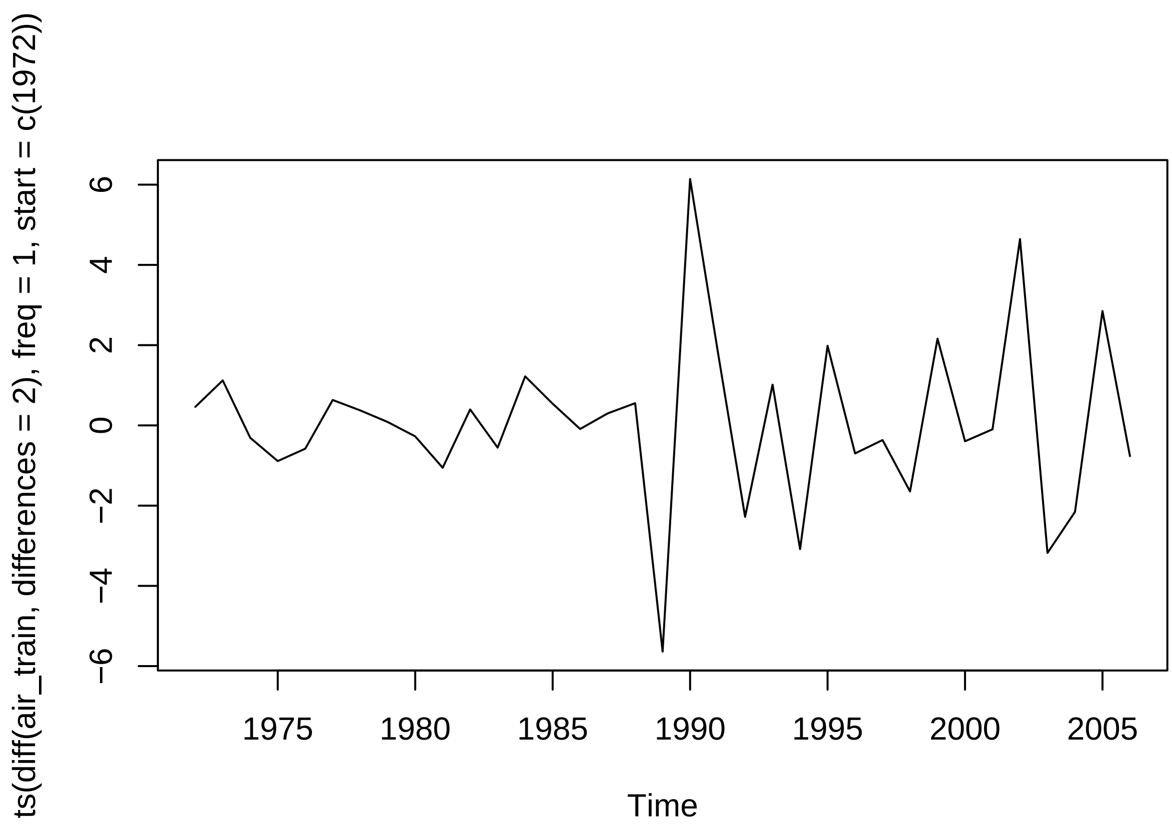


```
kpss.test(ts(diff(air_train, differences = 2), freq=1, start=c(1972)))
```

```
## Warning in kpss.test(ts(diff(air_train, differences = 2), freq = 1, start =
## c(1972))): p-value greater than printed p-value
```

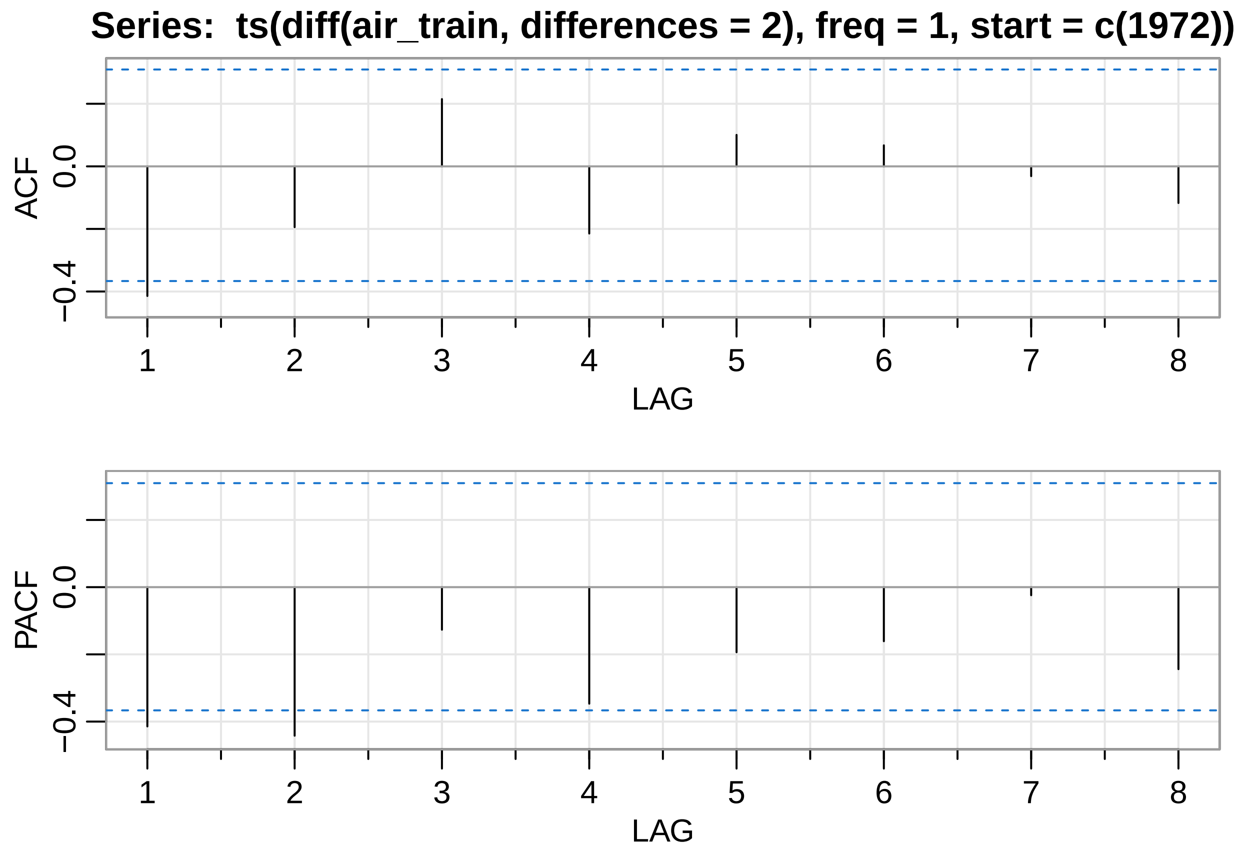
```
##
##  KPSS Test for Level Stationarity
##
## data:  ts(diff(air_train, differences = 2), freq = 1, start = c(1972))
## KPSS Level = 0.047888, Truncation lag parameter = 3, p-value = 0.1
```

```
plot(ts(diff(air_train, differences = 2), freq=1, start=c(1972)))
```



We will need difference of 2 to convert the time-series to stationary time series.

```
library(astsa)
acf2(ts(diff(air_train, differences = 2), freq=1, start=c(1972)))
```



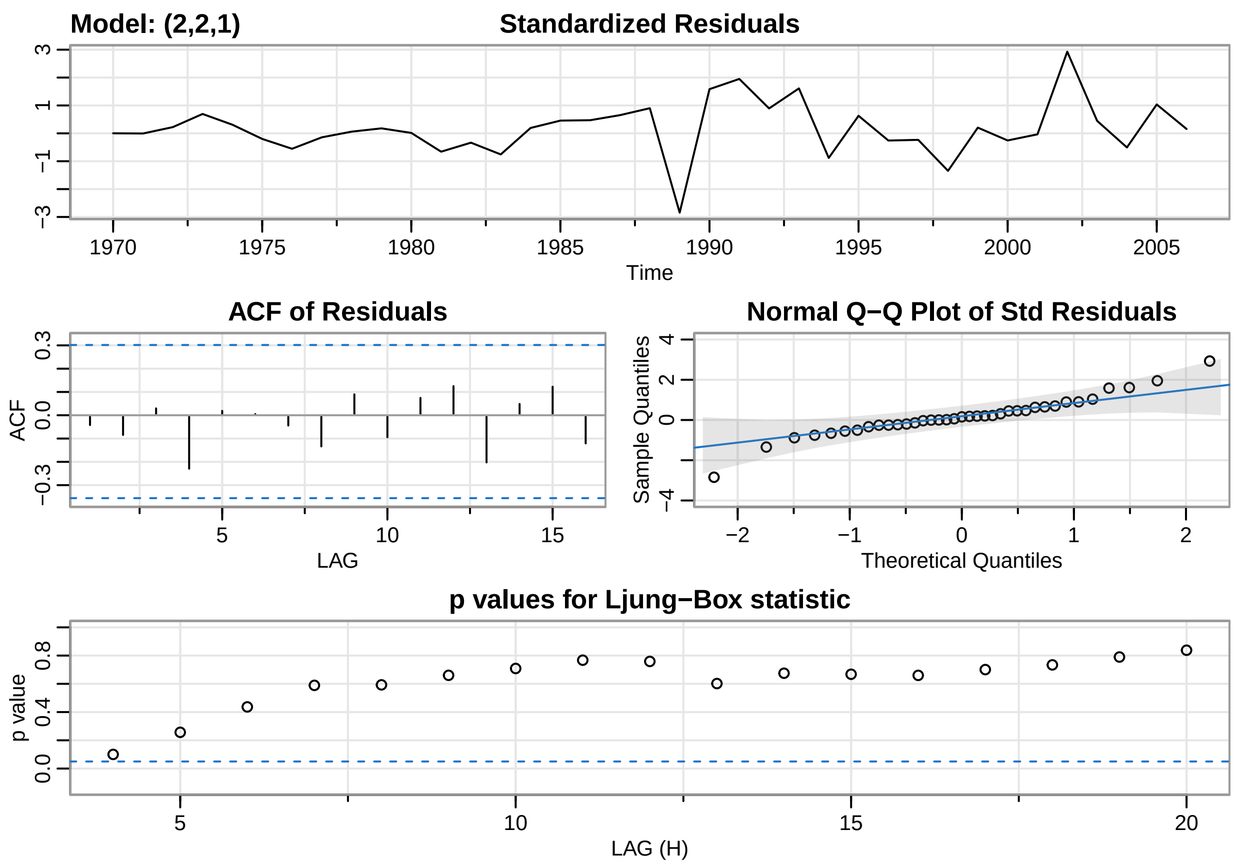

```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]
## ACF  -0.41 -0.19  0.21 -0.21  0.10  0.07 -0.03 -0.12
## PACF -0.41 -0.44 -0.13 -0.35 -0.19 -0.16 -0.02 -0.24
```

One significant lag in ACF suggests a possibility of one MA term and two significant lag in PACF suggest a possibility of two AR terms

```
sarima(air_train, p=2, d=2, q=1)
```

```
## initial  value 0.761991
## iter   2 value 0.542062
## iter   3 value 0.512535
## iter   4 value 0.496392
## iter   5 value 0.472297
## iter   6 value 0.471416
## iter   7 value 0.471222
## iter   8 value 0.471084
## iter   9 value 0.471083
## iter  10 value 0.471083
## iter  10 value 0.471083
## iter  10 value 0.471083
## final  value 0.471083
## converged
## initial  value 0.459644
## iter   2 value 0.458928
```

```
## iter   3 value 0.458153
## iter   4 value 0.458150
## iter   5 value 0.458150
## iter   5 value 0.458150
## iter   5 value 0.458150
## final  value 0.458150
## converged
```

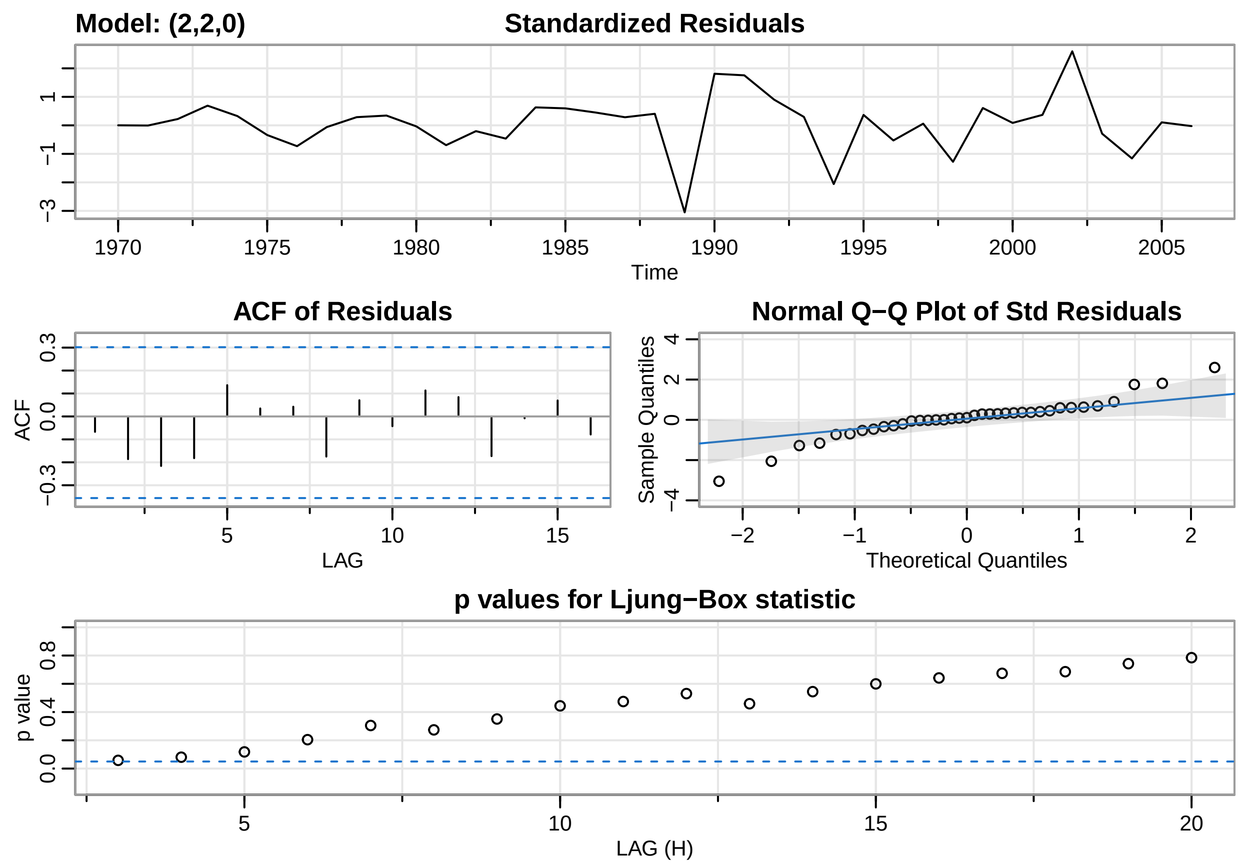



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2      ma1
##       -0.0116  -0.1769  -0.8474
## s.e.   0.1853   0.1765   0.1100
##
## sigma^2 estimated as 2.389:  log likelihood = -65.7,  aic = 139.4
##
## $degrees_of_freedom
## [1] 32
##
```

```
## $ttable
##     Estimate     SE t.value p.value
## ar1  -0.0116 0.1853 -0.0624  0.9507
## ar2  -0.1769 0.1765 -1.0021  0.3238
## ma1  -0.8474 0.1100 -7.7071  0.0000
##
## $AIC
## [1] 3.982748
##
## $AICc
## [1] 4.004867
##
## $BIC
## [1] 4.160502
```

```r
sarima(air_train, p=2, d=2, q=0)
```

```
## initial  value 0.761991
## iter   2 value 0.595719
## iter   3 value 0.566984
## iter   4 value 0.551359
## iter   5 value 0.550934
## iter   6 value 0.550933
## iter   6 value 0.550933
## final  value 0.550933
## converged
## initial  value 0.538469
## iter   2 value 0.538278
## iter   3 value 0.538186
## iter   4 value 0.538180
## iter   5 value 0.538180
## iter   5 value 0.538180
## iter   5 value 0.538180
## final  value 0.538180
## converged
```

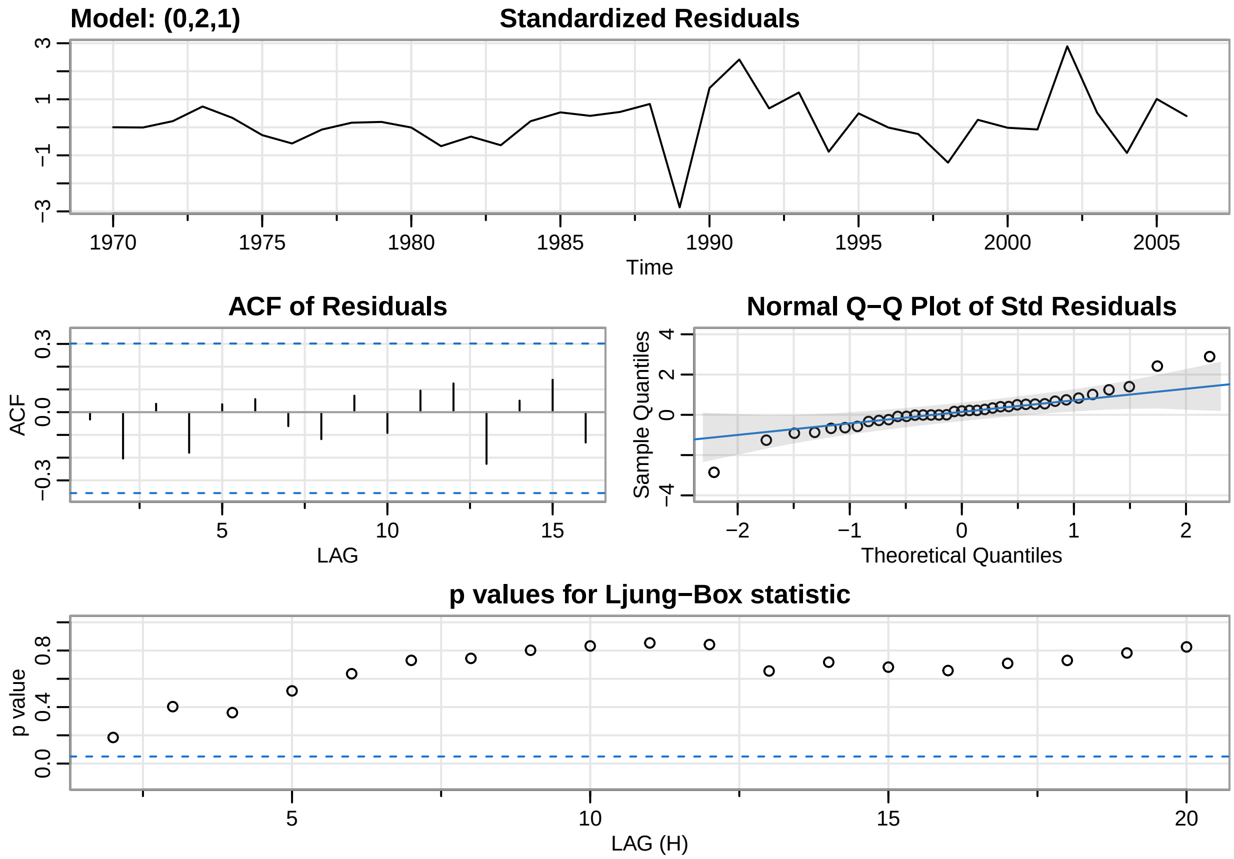



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2
##       -0.5873  -0.4422
## s.e.   0.1498   0.1499
##
## sigma^2 estimated as 2.883:  log likelihood = -68.5,  aic = 143
##
## $degrees_of_freedom
## [1] 33
##
## $ttable
##     Estimate     SE t.value p.value
## ar1  -0.5873 0.1498 -3.9206  0.0004
## ar2  -0.4422 0.1499 -2.9495  0.0058
##
## $AIC
## [1] 4.085666
##
## $AICc
```

```
## [1] 4.09638
##
## $BIC
## [1] 4.218982
```

```
sarima(air_train, p=0, d=2, q=1)
```

```
## initial  value 0.737389
## iter   2 value 0.586561
## iter   3 value 0.490723
## iter   4 value 0.473105
## iter   5 value 0.464085
## iter   6 value 0.462221
## iter   7 value 0.461574
## iter   8 value 0.461374
## iter   9 value 0.461346
## iter  10 value 0.461346
## iter  10 value 0.461346
## final  value 0.461346
## converged
## initial  value 0.472235
## iter   2 value 0.472171
## iter   3 value 0.472169
## iter   3 value 0.472169
## iter   3 value 0.472169
## final  value 0.472169
## converged
```




```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1
##       -0.8807
## s.e.   0.0829
##
## sigma^2 estimated as 2.464:  log likelihood = -66.19,  aic = 136.38
##
## $degrees_of_freedom
## [1] 34
##
## $ttable
##     Estimate     SE t.value p.value
## ma1  -0.8807 0.0829 -10.621       0
##
## $AIC
## [1] 3.8965
##
## $AICc
## [1] 3.899963
```

```
##
## $BIC
## [1] 3.985377
```

Looking at these 3 models, the aic is least for the (0,2,1) model. Looking at the plots as well, we see that for the (0,2,1) model, the Ljung-Box graph suggests higher p-value for all the lags. Compared to the other models, this is the best one.

Hence, the ARIMA model that I would select would be the (0,2,1) model based on the evidences provided by these models.

Verifying this result by auto.arima()

```
library(forecast)
```

```
##
## Attaching package: 'forecast'

## The following object is masked from 'package:astsa':
##
##     gas
```

```
auto.arima(air_train)
```

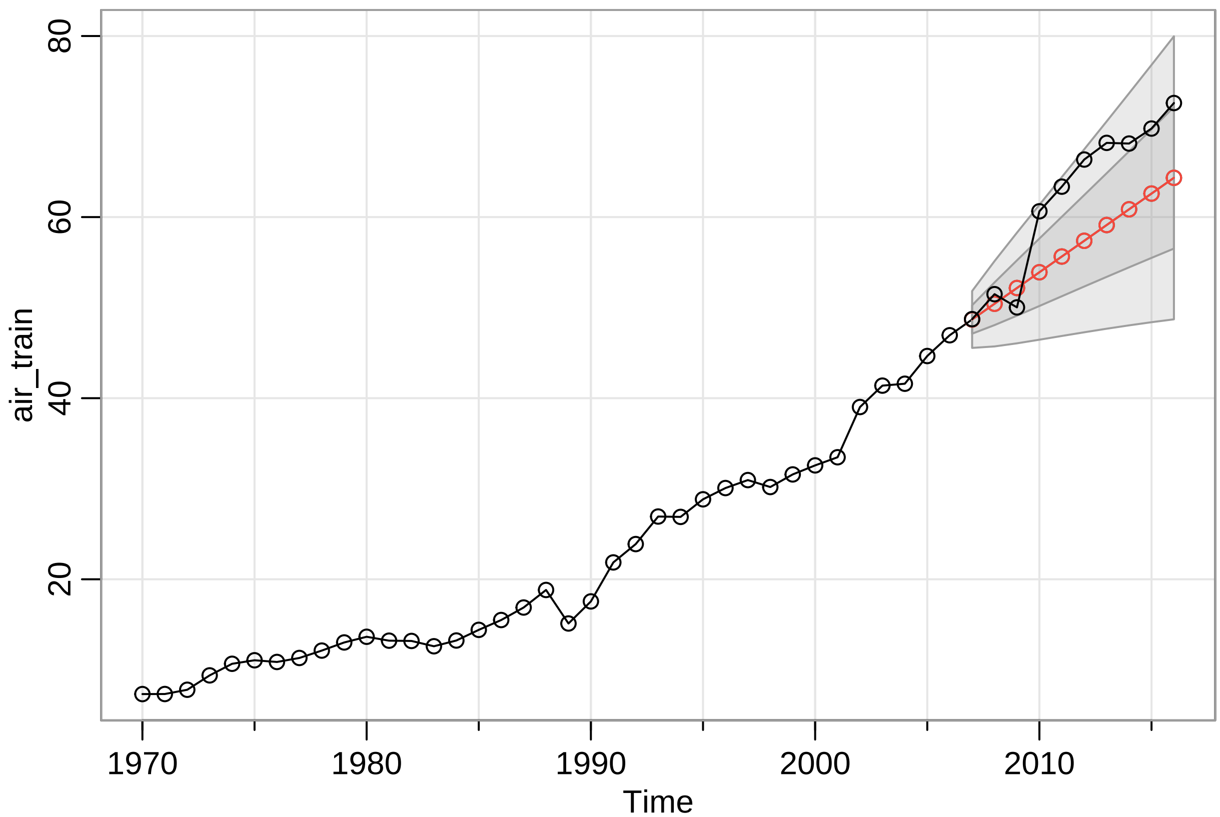
```
## Series: air_train
## ARIMA(0,2,1)
##
## Coefficients:
##           ma1
##       -0.8807
## s.e.   0.0829
##
## sigma^2 = 2.536:  log likelihood = -66.19
## AIC=136.38   AICc=136.75   BIC=139.49
```

Yayy! I came to the same result!

3. Generate a forecast from the model in #1 for the time period of the test set [1 point].

4. Plot the train set, the forecast, and the test set [1 point].

```
air_for = sarima.for(air_train, 10, 0, 2, 1)
lines(air_test, type='o')
```




```
air_for
```

```
## $pred
## Time Series:
## Start = 2007
## End = 2016
## Frequency = 1
##  [1] 48.69035 50.42893 52.16751 53.90608 55.64466 57.38324 59.12181 60.86039
##  [9] 62.59897 64.33754
##
## $se
## Time Series:
## Start = 2007
## End = 2016
## Frequency = 1
##  [1] 1.569611 2.355872 3.054412 3.724468 4.387128 5.052248 5.725093 6.408698
##  [9] 7.104889 7.814786
```

5. Calculate error metrics for the forecast compared to the test set [1 point].

```
data <- data.frame(pred = air_for$pred, actual = air_test)
data
```

```
##        pred   actual
```

```
## 1  48.69035 48.72884
## 2  50.42893 51.48843
## 3  52.16751 50.02697
## 4  53.90608 60.64091
## 5  55.64466 63.36031
## 6  57.38324 66.35527
## 7  59.12181 68.19795
## 8  60.86039 68.12324
## 9  62.59897 69.77935
## 10 64.33754 72.59770
```

```
mean((data$actual - data$pred)^2)
```

```
## [1] 44.60059
```