Time Series Anlaysis on Airpassenger Data

Weiling Li

2/12/2020

This the a tryout of time series analysis on the famous Air Passenger Dataset.

I followed all the steps introduced by *kimnewzealand* hoping to get more used to time series analysis

Data loading

```r
data("AirPassengers")

AP <- AirPassengers

class(AP)
```

```
## [1] "ts"
```

EDA

```r
AP
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949 112 118 132 129 121 135 148 148 136 119 104 118
## 1950 115 126 141 135 125 149 170 170 158 133 114 140
## 1951 145 150 178 163 172 178 199 199 184 162 146 166
## 1952 171 180 193 181 183 218 230 242 209 191 172 194
## 1953 196 196 236 235 229 243 264 272 237 211 180 201
## 1954 204 188 235 227 234 264 302 293 259 229 203 229
## 1955 242 233 267 269 270 315 364 347 312 274 237 278
## 1956 284 277 317 313 318 374 413 405 355 306 271 306
## 1957 315 301 356 348 355 422 465 467 404 347 305 336
## 1958 340 318 362 348 363 435 491 505 404 359 310 337
## 1959 360 342 406 396 420 472 548 559 463 407 362 405
## 1960 417 391 419 461 472 535 622 606 508 461 390 432
```

```r
sum(is.na(AP))
```

```
## [1] 0
```

```r
frequency(AP)
```

```
## [1] 12
```

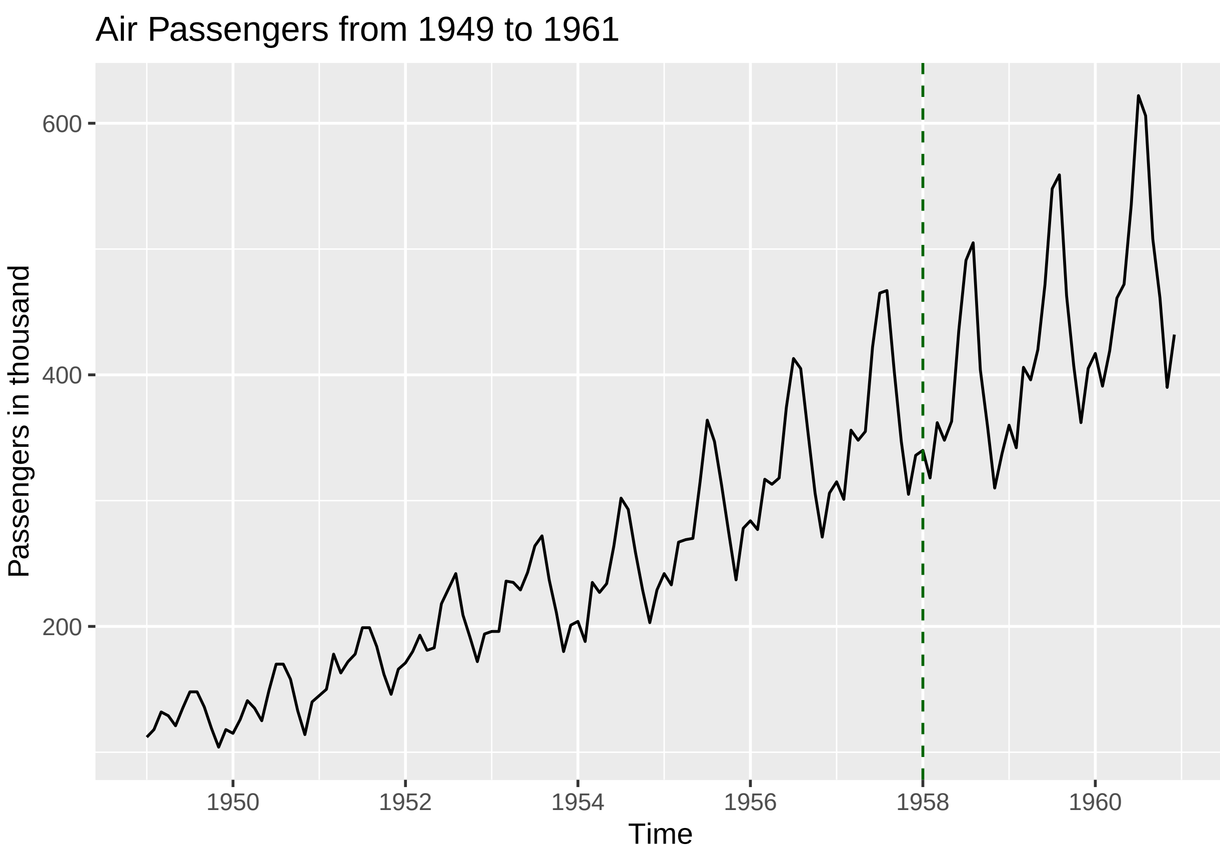
```r
cycle(AP)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949   1   2   3   4   5   6   7   8   9  10  11  12
## 1950   1   2   3   4   5   6   7   8   9  10  11  12
```

```
## 1951   1   2   3   4   5   6   7   8   9  10  11  12
## 1952   1   2   3   4   5   6   7   8   9  10  11  12
## 1953   1   2   3   4   5   6   7   8   9  10  11  12
## 1954   1   2   3   4   5   6   7   8   9  10  11  12
## 1955   1   2   3   4   5   6   7   8   9  10  11  12
## 1956   1   2   3   4   5   6   7   8   9  10  11  12
## 1957   1   2   3   4   5   6   7   8   9  10  11  12
## 1958   1   2   3   4   5   6   7   8   9  10  11  12
## 1959   1   2   3   4   5   6   7   8   9  10  11  12
## 1960   1   2   3   4   5   6   7   8   9  10  11  12
```

Plot the ts using ggfortify

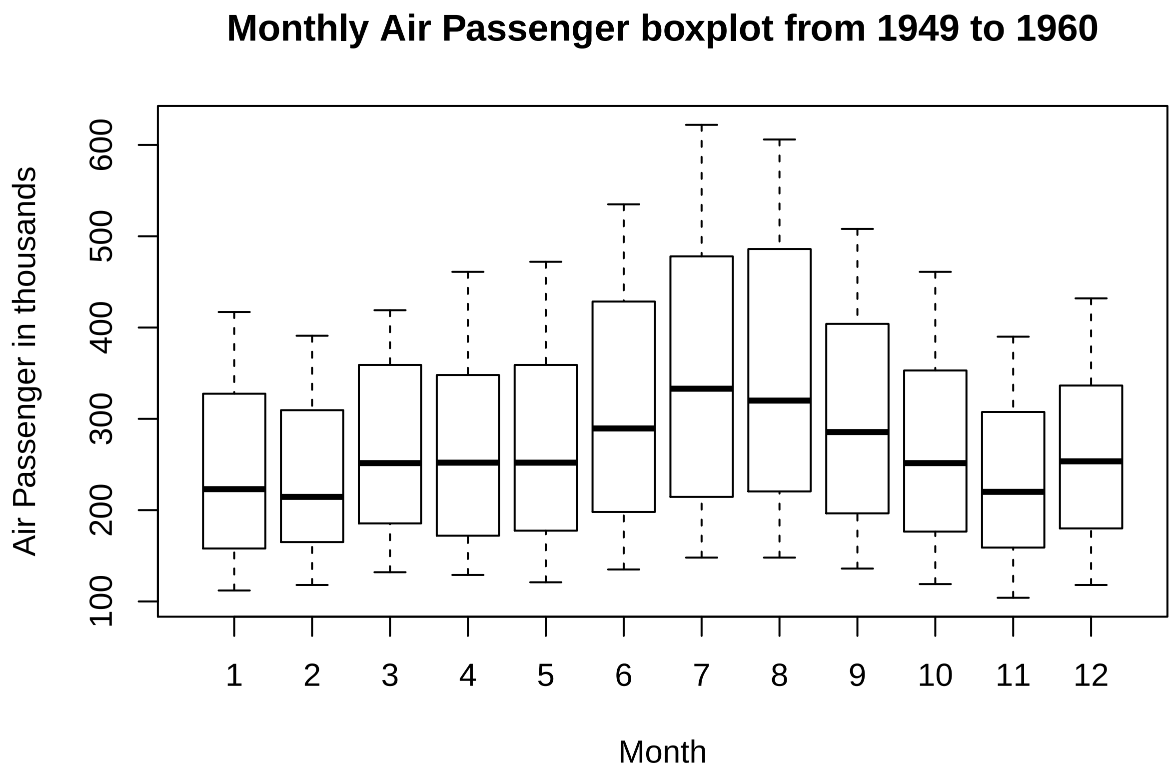
```r
autoplot(AP)+labs(x = 'Time', y = 'Passengers in thousand', title = "Air Passengers from 1949 to 1961")
```



As stated in the original post, there is a strong seasonality in the dataset, but we can also observe an increasing trend on the local means and the variability. The seasonality can be observed using boxplot on cycle. From now on, the tutorial went on to analyze the data with the full dataset and predict into 3 years future. However, because there are no actual data available to validate the outcome, we are not able to access the prediction power.

In order to resolve this issue, we broke the data into two parts: training set(1949 to 1957) & test set(1958 to 1961). After modeling we will compare the prediction results against the actual data to obtain prediction power.

```r
boxplot(AP~cycle(AP),xlab = 'Month', ylab = 'Air Passenger in thousands', main = 'Monthly Air Passenger
```



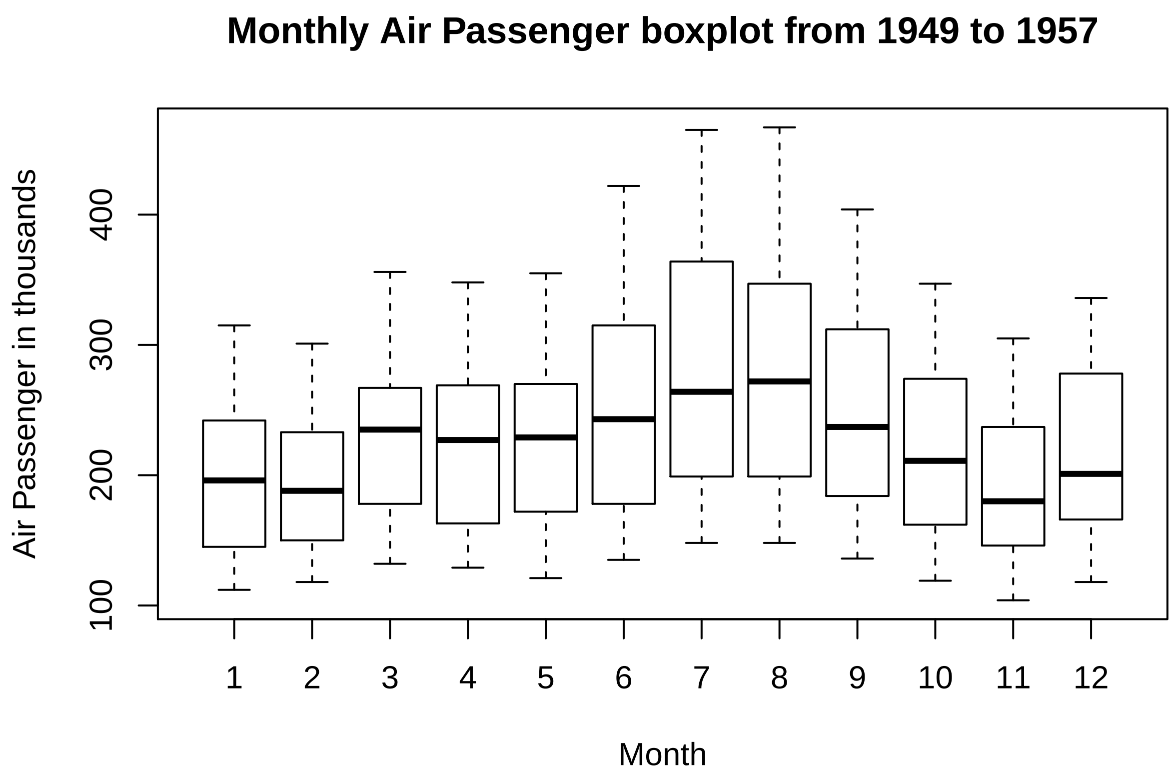

```r
##break the ts into two parts
train <- head(AP,9*12)
test <- tail(AP, 3*12)

class(train)
```

```
## [1] "ts"
```

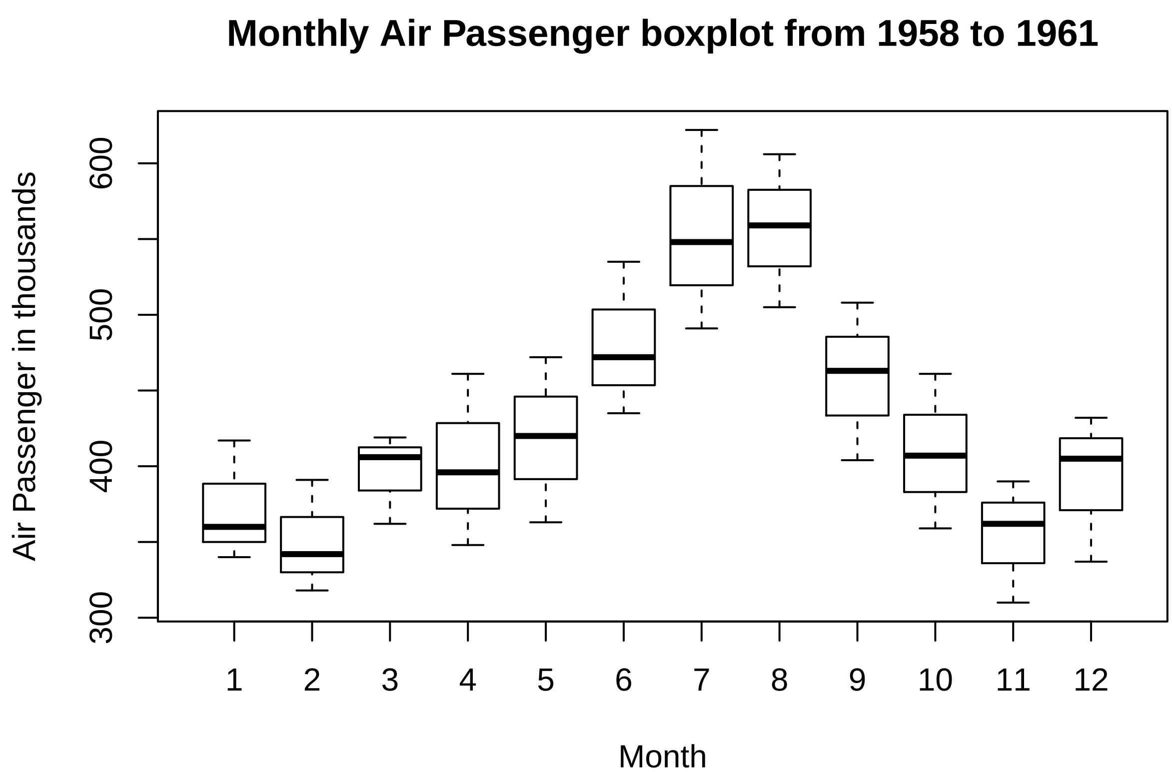
```r
class(test)
```

```
## [1] "ts"
```

```r
# boxplot for train
boxplot(train~cycle(train),xlab = 'Month', ylab = 'Air Passenger in thousands', main = 'Monthly Air Pas
```




```r
# boxplot for test
boxplot(test~cycle(test),xlab = 'Month', ylab = 'Air Passenger in thousands', main = 'Monthly Air Passe
```



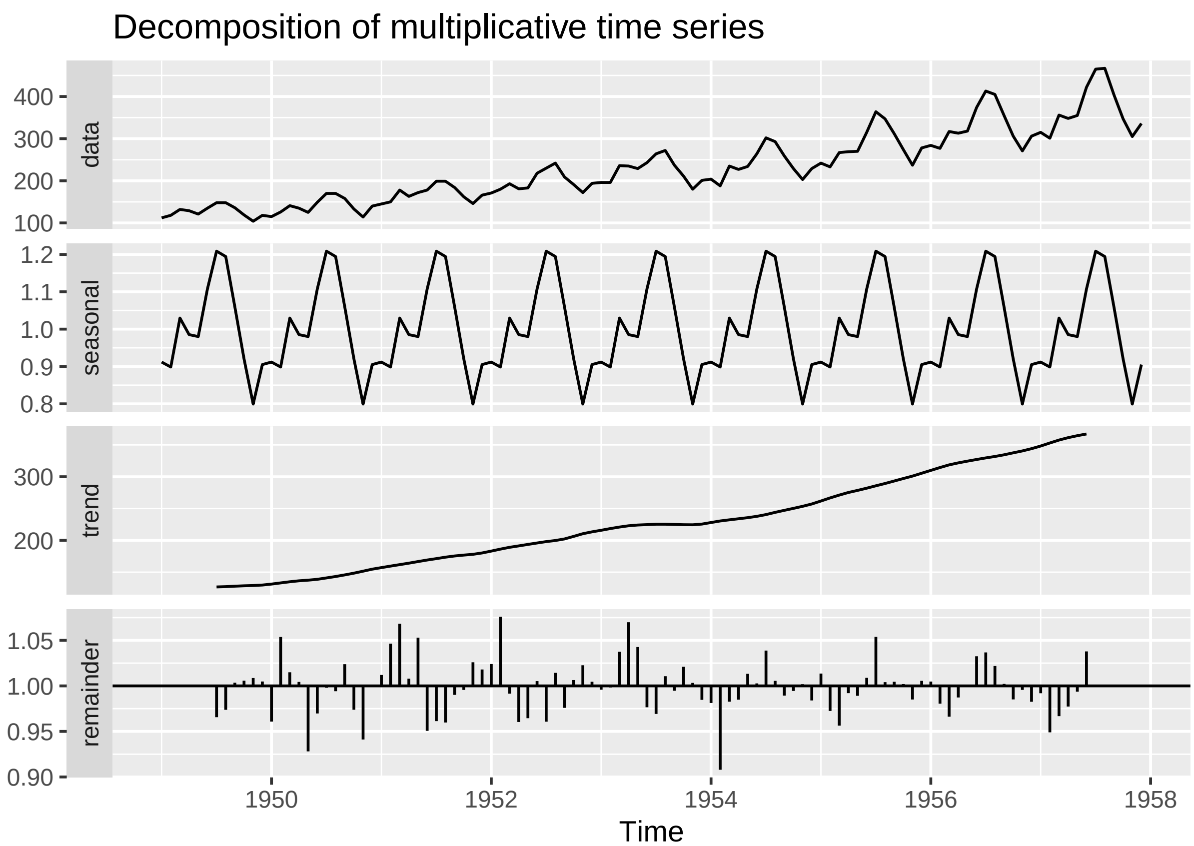
From the tutorial, the author says that the model appears to be multiplicative. After checking a post on **R-blogger**. It is the line plot shown the multiplicative relationship. The explaination is quoted as follows:

```
How these three components interact determines the difference between a multiplicative and an additive t

In a multiplicative time series, the components multiply together to make the time series. If you have a

In an additive time series, the components add together to make the time series. If you have an increasi
```

In our line plot, there is a clear trend that everything gets exaggerated so it is more likely a multiplicative series. In the blog, Steph layed out a very intuitive model selection via the quality of residuals that can take human bias out of the equation. For this simple case, I would use my intuition of the data and select multiplicative.

Decompose the series

```r
decomposeAP <- decompose(train,"multiplicative")
autoplot(decomposeAP)
```



In this plot, we could confirm the trend and seasonality. and the remainder does not seems to be a function of time. But in order to fit an arima model, we need to ensure that the time series to be stationary.

Test for Stationary

The author proposed 2 tests. `adf.test` from tseries library and autocorrelation using `acf` from forecast.

we shall try the tseries 1st:

```r
adf.test(train)
```

```
## Warning in adf.test(train): p-value smaller than printed p-value

##
```

```
##  Augmented Dickey-Fuller Test
##
## data:  train
## Dickey-Fuller = -4.4465, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

with p-value smaller than 0.01, we reject the null hypothesis, thus accept the alternative hypothesis that the series is stationary.

The following part using acf is confusion. But the author reached the same result. However, I am not able to understand the reasoning behind. For this case we will skipped.

But question is raised: what if the series is not stationary?

Next, we move on to fit ts model and check the prediction against the real data

Fit Arima and Make Predictions

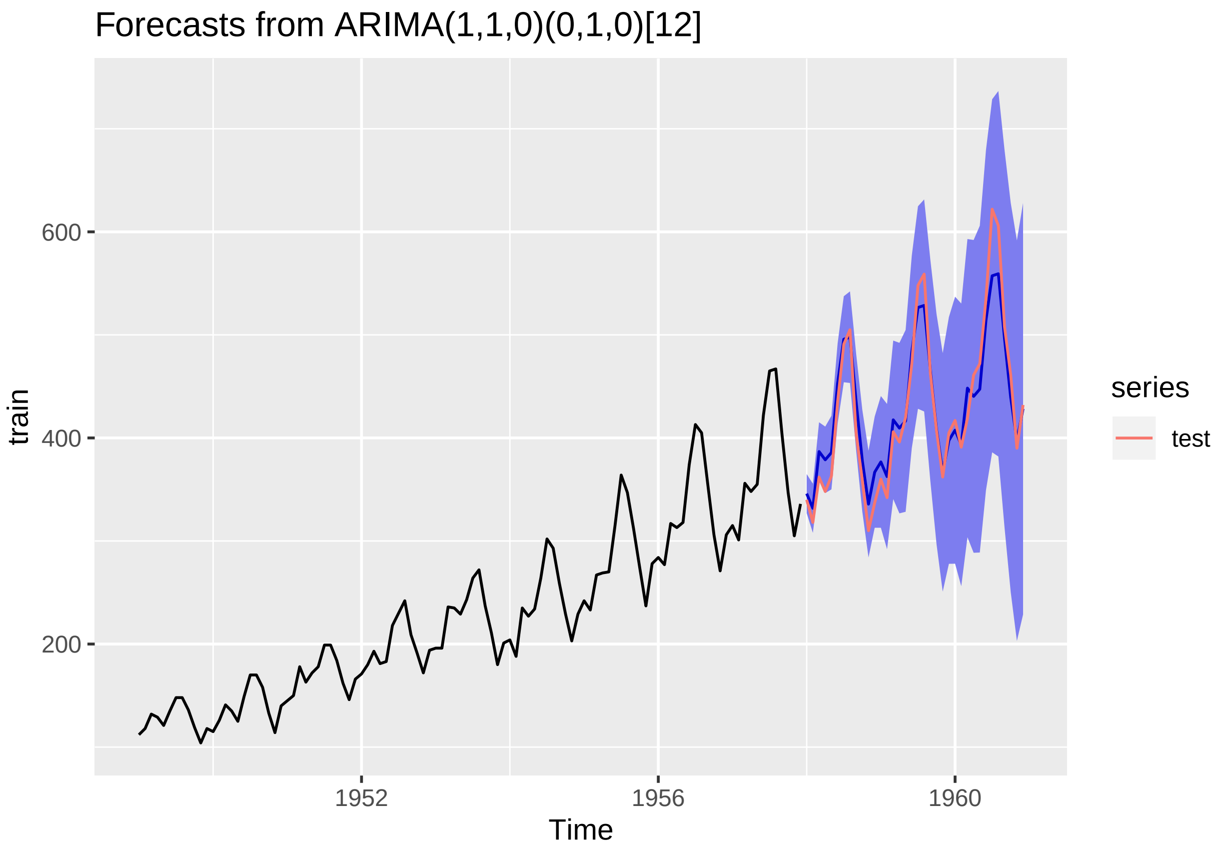
```r
arimaAP <- auto.arima(train)
arimaAP
```

```
## Series: train
## ARIMA(1,1,0)(0,1,0)[12]
##
## Coefficients:
##           ar1
##       -0.2411
## s.e.   0.0992
##
## sigma^2 estimated as 93.74:  log likelihood=-350
## AIC=704   AICc=704.13   BIC=709.11
```

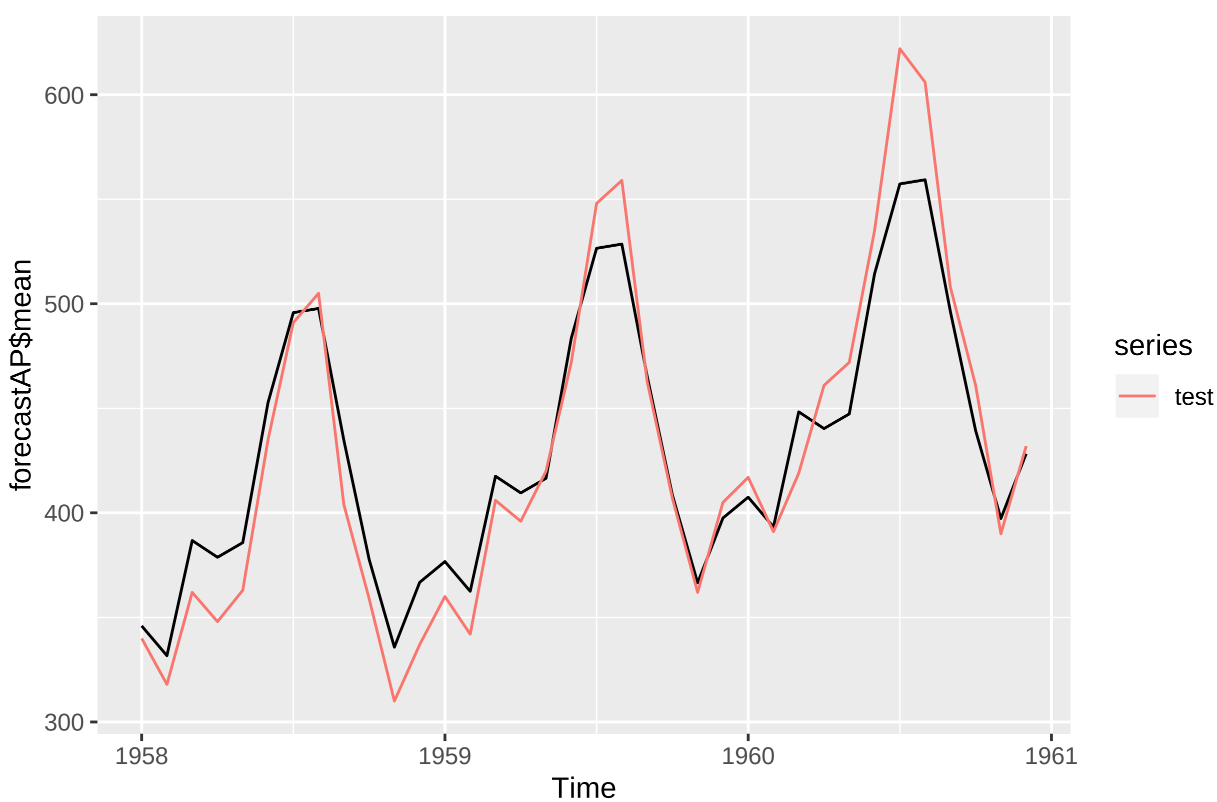
```r
ggtsdiag(arimaAP)
```

Forecast Series

```r
forecastAP <- forecast(arimaAP,level = c(95),h = 36)
autoplot(forecastAP)+autolayer(test)
```




```r
autoplot(forecastAP$mean)+autolayer(test)
```



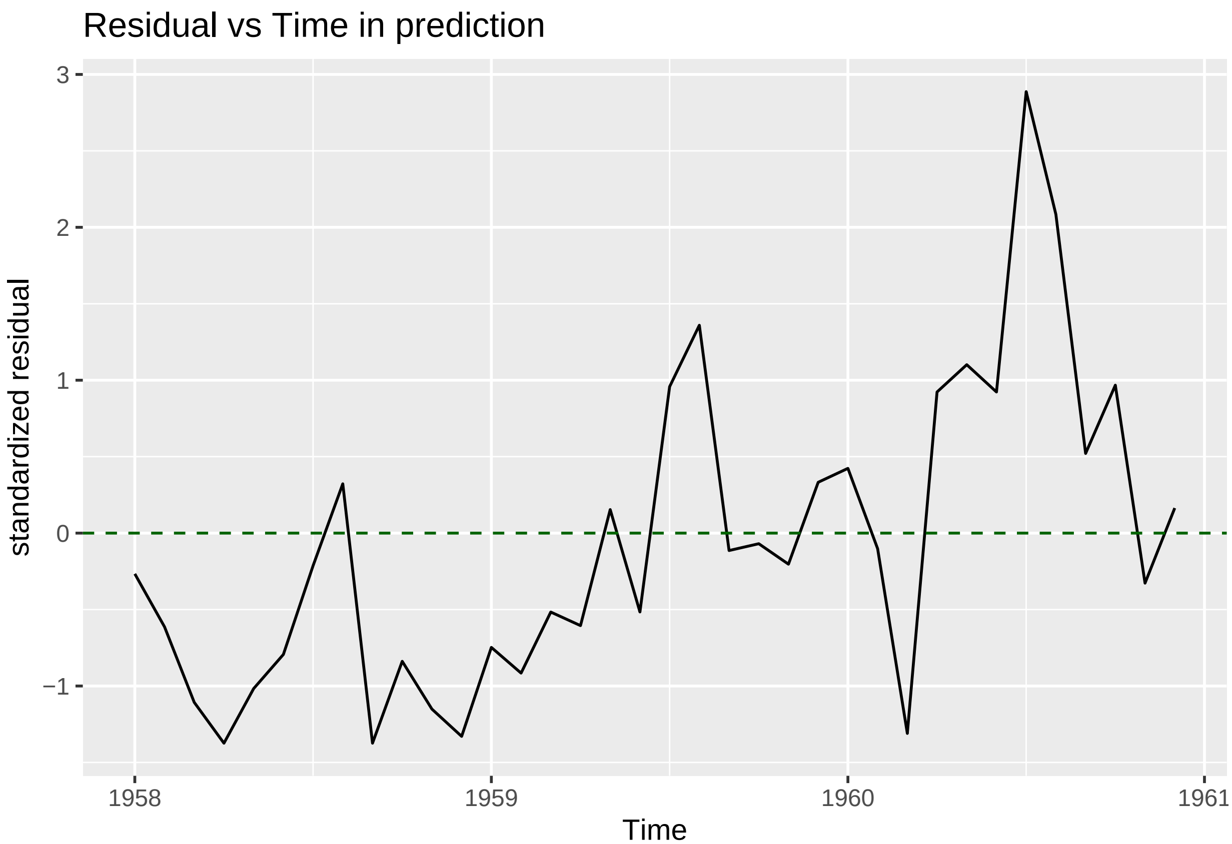
calculate mean sqrt error:

```r
pred <- as.matrix(forecastAP$mean)
true <- as.matrix(test)

sqrt(mean((pred - true)^2))
```

```
## [1] 22.13223
```

Averagly, the prediction is off by 22 people. Pretty impressive. However. the residual may be showing an increasing variance. Which is not captured by the model.

```r
autoplot((test - forecastAP$mean)/sd(test - forecastAP$mean))+
  geom_hline(yintercept = 0,color = 'darkgreen', linetype = "dashed")+
  labs(x = 'Time', y = 'standardized residual', title = "Residual vs Time in prediction")
```



I think tuning the model relied on understanding the auto.arima coefs.