Monthly Number of Passengers Boarding Airplanes in Portugal

Carlos Pedro Pinto$^{1[201404221]}$

[1] Faculty of Economics of the University of Porto

Abstract. This report was performed in the context of Time Series Forecasting Methods belonging to the Master in Modeling, Data Analysis and Decision Support Systems from the Faculty of Economics of the University of Porto. In this report it will be analyzed a time series representing the monthly number of passengers boarding airplanes in Portugal by performing an exploratory data analysis, and applying different methods for forecasting: decomposition methods, smoothing methods and ARIMA models. Finally, to understand what the best forecasting method was it was performed statistical measures to compare de accuracy of the model.

Keywords: Time Series, Time Series Analysis, Decomposition Methods, Smoothing Methods, Holt-Winters, ARIMA Models, Accuracy, R Software.

1 Introduction

Since ancient times, human beings have been trying to predict the future, and the methods used, have evolved and are being applied to multiple areas being one of them the forecasting of time series, where the goal is to build mathematical models capable of predicting the future values.

Time series are data points that occur in successive order over some period, and for this report it was selected a monthly time series with the number of passengers boarding airplanes in Portugal since January 2004 to December 2019.

This report will start with a time series analysis, where the goal is to have an overview of the dataset as well as understand some components of the time series.

The next step will be to apply three different methods of forecasting, namely smoothing methods, decomposition methods and ARIMA models.

Finally, it will be compared the forecasting of each model refereed above, using accuracy measures to define the best model.

2 Time Series Analysis

The first step of the project was to plot the time series to have an overview of the information, allowing to describe the main statistical features. By looking to the figure 1, we can observe that the time series as a positive trend in all the period, there is a seasonal component, that from the figure below it is not possible to define the value of this component and it appears to be non-stationarity. The last point that is possible to observe is that the amplitude of each season increases over the time.

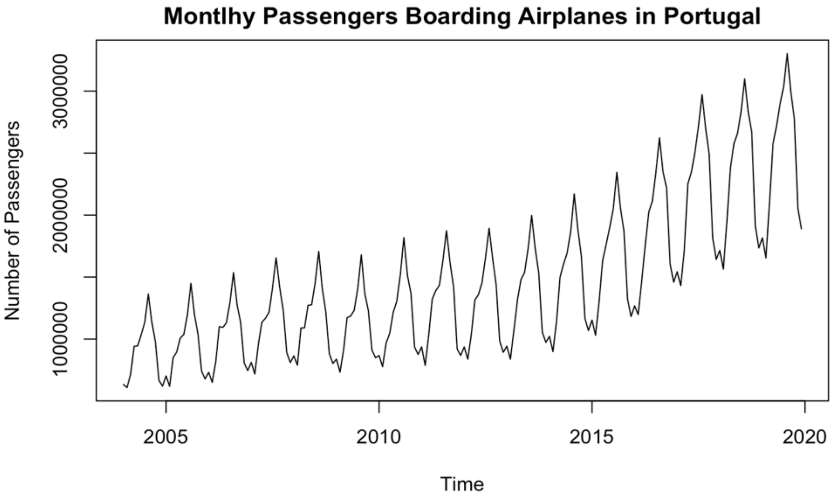


Fig. 1. Time Series plot of the Number of Passengers Boarding Airplanes since 2004 to 2019

Considering that the time series has a component of seasonality it was plot two graphics in figure 2 using the functions (monthplot and seasonplot) to understand how the dataset behaves across each month and years. Looking to the monthly plot we can see that each month has a similar behavior being the month of August the one with the highest number of passengers and February the one with the lowest in average. Also, when analyzing the seasonal plot, we can see that each year has a similar line in the graph. From these two plots we can conclude that we have yearly seasonality. The final conclusion from the plots below is that there is evidence of changes in variance in the all period, since in the latest years the difference between the month with the highest number of passengers and the lowest month is much higher than in the earlier years in analysis, meaning that in the decomposition and smoothing methods, the most suited methods will be the multiplicative one.

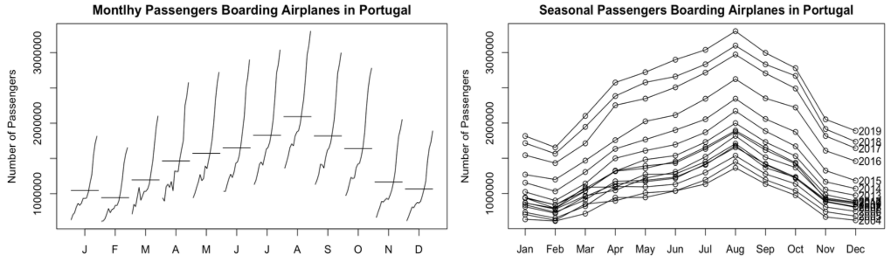


Fig. 2. Monthly and Seasonal plot of the Number of Passengers Boarding Airplanes since 2004 to 2019

The final analysis was done recurring to a box plot to detect any outliers and as expected the number of outliers present in this time series (2) is insignificant.

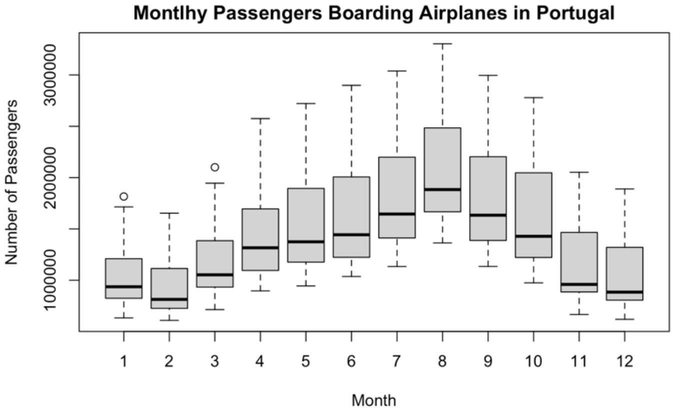
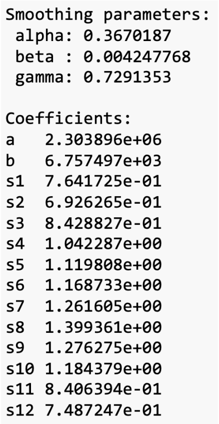


Fig. 3. Box plot of Monthly the Number of Passengers Boarding Airplanes since 2004 to 2019

3 Smoothing Methods

Although there are multiple methods of smoothing, the most appropriated method for the time series in analysis is the Holt-Winters exponential smoothing multiplicative method since the data presents trend, seasonality and there is evidence of changes in variance.

The data set was divided in two, train and test, and using the R function "HoltWinters" with the train data we obtained the parameters of the method as well as the coefficients.



```
Smoothing parameters:
 alpha: 0.3670187
 beta : 0.004247768
 gamma: 0.7291353

Coefficients:
a   2.303896e+06
b   6.757497e+03
s1  7.641725e-01
s2  6.926265e-01
s3  8.428827e-01
s4  1.042287e+00
s5  1.119808e+00
s6  1.168733e+00
s7  1.261605e+00
s8  1.399361e+00
s9  1.276275e+00
s10 1.184379e+00
s11 8.406394e-01
s12 7.487247e-01
```

Fig. 4. Smoothing Parameters and Coefficients of Holt-Winters Exponential Smoothing Multiplicative Method

The smoothing parameters of the model give us some insights of the model. Starting with alpha that indicates the weight used in the level component of the smoothed estimated and is similar to a moving average of the observations, it has a value close to 0,37 indicating that weight of the data is higher for the older observations in the prediction used by this model. The second parameter, beta that indicates the weight used in the trend component is almost equal to zero, meaning that the fittet trend was good for the entire window of data. The last one, gamma, used in the seasonal component of the smoothed estimate, has a value closer to one (approximately 0,73) indicating that the most recent data has a higher weight.

Regarding the coefficients we notice that the month where the number of passengers has the highest value also has the highest coefficient (August, s8).

In the figure 5 we can see the plot of the forecast using the Holt-Winters exponential smoothing multiplicative method.

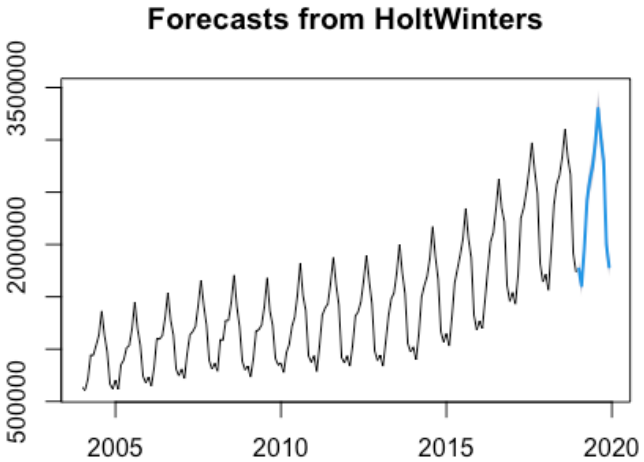


Fig. 5. Plot of the Number of Passengers Forecast using Holt-Winters addictive method.

4 Decomposition Methods

In order to deconstruct the time series into several components: trend-cycle, seasonal and random, we can use decomposition methods leading to a better understanding of the data and improving the forecasting methods.

4.1 Classical Decomposition

The first decomposition method used was the classical multiplicative decomposition, this is the simplest method and forms the starting point for most of other methods

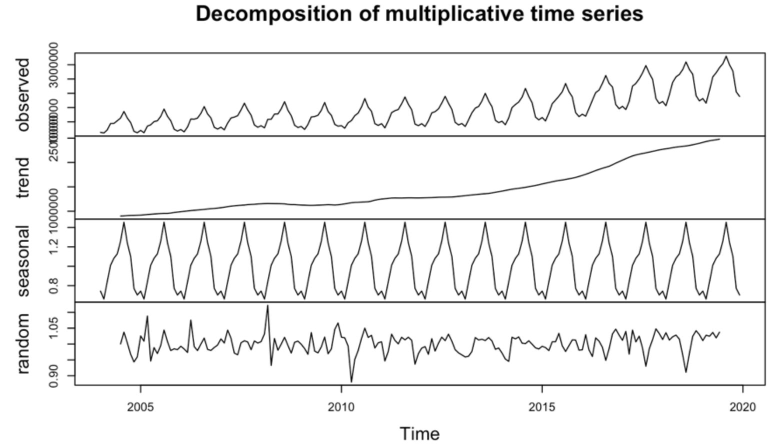
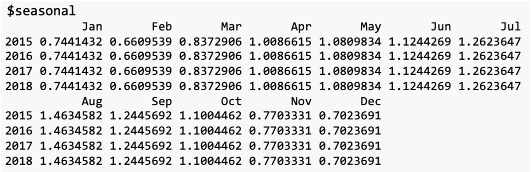


Fig. 6. Decomposition of the time series using Classical Decomposition Multiplicative method

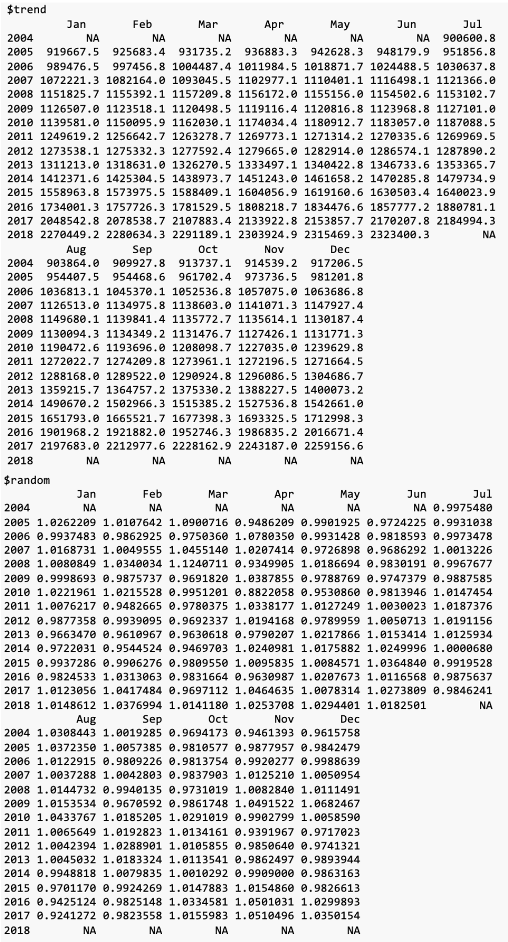
As previously stated, there is evidence of changes in variance, in the time series, hence it was applied the multiplicative decomposition method that is displayed in the figure 6.

Analyzing the output of R in the first graph we see the data observed, in the second it is possible to observe a positive trend that intensifies in the last years, the third one we observe the seasonality that doesn't appear to change from year to year, being constant in all the period analyzed. Finally, on the last graphic the random seems to have a random behavior, similar to a white noise plot, proving that the multiplicative method is a good fit.



```
$seasonal
           Jan       Feb       Mar       Apr       May       Jun       Jul
2015 0.7441432 0.6609539 0.8372906 1.0086615 1.0809834 1.1244269 1.2623647
2016 0.7441432 0.6609539 0.8372906 1.0086615 1.0809834 1.1244269 1.2623647
2017 0.7441432 0.6609539 0.8372906 1.0086615 1.0809834 1.1244269 1.2623647
2018 0.7441432 0.6609539 0.8372906 1.0086615 1.0809834 1.1244269 1.2623647
           Aug       Sep       Oct       Nov       Dec
2015 1.4634582 1.2445692 1.1004462 0.7703331 0.7023691
2016 1.4634582 1.2445692 1.1004462 0.7703331 0.7023691
2017 1.4634582 1.2445692 1.1004462 0.7703331 0.7023691
2018 1.4634582 1.2445692 1.1004462 0.7703331 0.7023691
```

Fig. 7. Decomposition results (seasonal) from R

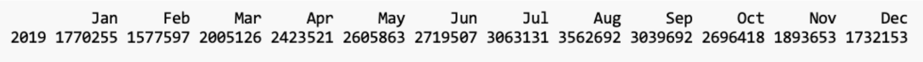


```
$trend
          Jan       Feb       Mar       Apr       May       Jun       Jul
2004       NA        NA        NA        NA        NA        NA  900600.8
2005  919667.5  925683.4  931735.2  936883.3  942628.3  948179.9  951856.8
2006  989476.5  997456.8 1004487.4 1011984.5 1018871.7 1024488.5 1030637.8
2007 1072221.3 1082164.0 1093045.5 1102977.1 1110401.1 1116498.1 1121366.0
2008 1151825.7 1155392.1 1157209.8 1156172.0 1155156.0 1154502.6 1153102.7
2009 1126507.0 1123518.1 1120498.5 1119116.4 1120816.8 1123968.8 1127101.0
2010 1139581.0 1150095.9 1162030.1 1174034.4 1180912.7 1183057.0 1187088.5
2011 1249619.2 1256642.7 1263278.7 1269773.1 1271314.2 1270335.6 1269969.5
2012 1273538.1 1275332.3 1277592.4 1279665.0 1282914.0 1286574.1 1287890.2
2013 1311213.0 1318631.0 1326270.5 1333497.1 1340422.8 1346733.6 1353365.7
2014 1412371.6 1425304.5 1438973.7 1451243.0 1461658.2 1470285.8 1479734.9
2015 1558963.8 1573975.5 1588409.1 1604056.9 1619160.6 1630503.4 1640023.9
2016 1734001.3 1757726.3 1781529.5 1808218.7 1834476.6 1857777.2 1880781.1
2017 2048542.8 2078538.7 2107883.4 2133922.8 2153857.7 2170207.8 2184994.3
2018 2270449.2 2280634.3 2291189.1 2303924.9 2315469.3 2323400.3        NA
          Aug       Sep       Oct       Nov       Dec
2004  903864.0  909927.8  913737.1  914539.2  917206.5
2005  954407.5  954468.6  961702.4  973736.5  981201.8
2006 1036813.1 1045370.1 1052536.8 1057075.0 1063686.8
2007 1126513.0 1134975.8 1138603.0 1141071.3 1147927.4
2008 1149680.1 1139841.4 1135772.7 1135614.1 1130187.4
2009 1130094.3 1134349.2 1131476.7 1127426.1 1131771.3
2010 1190472.6 1193696.0 1208098.7 1227035.0 1239629.8
2011 1272022.7 1274209.8 1273961.1 1272196.5 1271664.5
2012 1288168.0 1289522.0 1290924.8 1296086.5 1304686.7
2013 1359215.7 1364757.2 1375330.2 1388227.5 1400073.2
2014 1490670.2 1502966.3 1515385.2 1527536.8 1542661.0
2015 1651793.0 1665521.7 1677398.3 1693325.5 1712998.3
2016 1901968.2 1921882.0 1952746.3 1986835.2 2016671.4
2017 2197683.0 2212977.6 2228162.9 2243187.0 2259156.6
2018        NA        NA        NA        NA        NA
$random
           Jan       Feb       Mar       Apr       May       Jun       Jul
2004        NA        NA        NA        NA        NA        NA 0.9975480
2005 1.0262209 1.0107642 1.0900716 0.9486209 0.9901925 0.9724225 0.9931038
2006 0.9937483 0.9862925 0.9750360 1.0780350 0.9931428 0.9818593 0.9973478
2007 1.0168731 1.0049555 1.0455140 1.0207414 0.9726898 0.9686292 1.0013226
2008 1.0080849 1.0340034 1.1240711 0.9349905 1.0186694 0.9830191 0.9967677
2009 0.9998693 0.9875737 0.9691820 1.0387855 0.9788769 0.9747379 0.9887585
2010 1.0221961 1.0215528 0.9951201 0.8822058 0.9530860 0.9813946 1.0147454
2011 1.0076217 0.9482665 0.9780375 1.0338177 1.0127249 1.0030023 1.0187376
2012 0.9877358 0.9939095 0.9692337 1.0194168 0.9789959 1.0050713 1.0191156
2013 0.9663470 0.9610967 0.9630618 0.9790207 1.0217866 1.0153414 1.0125934
2014 0.9722031 0.9544524 0.9469703 1.0240981 1.0175882 1.0249996 1.0000680
2015 0.9937286 0.9906276 0.9809550 1.0095835 1.0084571 1.0364840 0.9919528
2016 0.9824533 1.0313063 0.9831664 0.9630987 1.0207673 1.0116568 0.9875637
2017 1.0123056 1.0417484 0.9697112 1.0464635 1.0078314 1.0273809 0.9846241
2018 1.0148612 1.0376994 1.0141180 1.0253708 1.0294401 1.0182501        NA
           Aug       Sep       Oct       Nov       Dec
2004 1.0308443 1.0019285 0.9694173 0.9461393 0.9615758
2005 1.0372350 1.0057385 0.9810577 0.9877957 0.9842479
2006 1.0122915 0.9809226 0.9813754 0.9920277 0.9988639
2007 1.0037288 1.0042803 0.9837903 1.0125210 1.0050954
2008 1.0144732 0.9940135 0.9731019 1.0082840 1.0111491
2009 1.0153534 0.9670592 0.9861748 1.0491522 1.0682467
2010 1.0433767 1.0185205 1.0291019 0.9902799 1.0058590
2011 1.0065649 1.0192823 1.0134161 0.9391967 0.9717023
2012 1.0042394 1.0288901 1.0105855 0.9850640 0.9741321
2013 1.0045032 1.0183324 1.0113541 0.9862497 0.9893944
2014 0.9948818 1.0079835 1.0010292 0.9909000 0.9863163
2015 0.9701170 0.9924269 1.0147883 1.0154860 0.9826613
2016 0.9425124 0.9825148 1.0334581 1.0501031 1.0299893
2017 0.9241272 0.9823558 1.0155983 1.0510496 1.0350154
2018        NA        NA        NA        NA        NA
```

Fig. 8. Decomposition results (trend and random) from R

The decomposition method follows the behavior of $Y_t = S_t * T_t * E_t$ using the values present in the figure 7 and 8. In this figure we can also see that the seasonal component is the same across the years, being the highest in August and lowest in February as expected since these months are the ones with the highest and lowest number of passengers, and the values of trend increase with the time.

In order to forecast using this method it was used the trend component of the decomposition and applied the Holt method to predict the trend values for 2019, after getting these values it was multiplied each one of them by the correspondent seasonal component in order to have the final prediction.



```
         Jan     Feb     Mar     Apr     May     Jun     Jul     Aug     Sep     Oct     Nov     Dec
2019 1770255 1577597 2005126 2423521 2605863 2719507 3063131 3562692 3039692 2696418 1893653 1732153
```

Fig. 9. Forecast of Number of Passengers from Classical Decomposition on R

4.2 Seasonal-Trend Decomposition using Loess (STL)

Other decomposition method that was implemented was the STL method since it is more complex than the classical decomposition, producing better results. The first step for the forecasting it was to divide the dataset into two the training (with all the data except the last 12 months) and testing (containing only the last twelve months) allowing us to forecast and evaluate the accuracy of the model. Also, since the STL method can only be used for additive models, it was applied the logarithm function to the time series.

The first output of this method was the plot the result of "stl" present in figure 10 where we can observe siliar information regarding the trend and seasonality as in the classical decomposition and that the remainder, appears to have a mean around zero and a random behavior, similar to a white noise plot, indicating that the model selected has properly fitted the data.

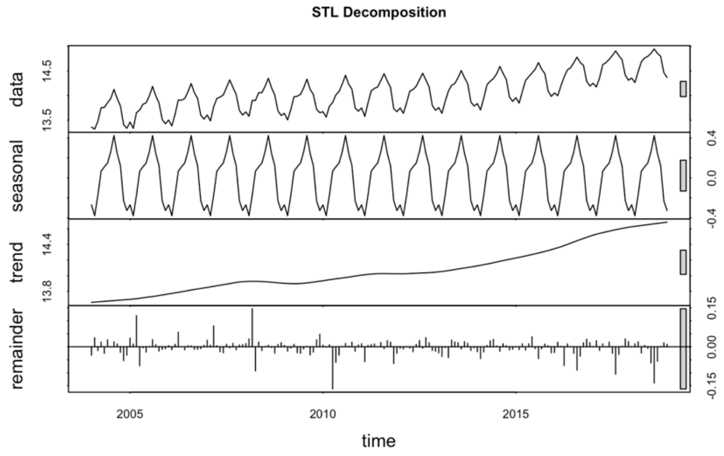


Fig. 10. Decomposition of the time series using STL Decomposition method

Using the R function "stl" it was possible to obtain the plot of the estimation on figure 11 and the forecasted values present on the figure 12. Since the time series needed to be transformed by using the logarithmic function, to get the real values of the forecasting we need to apply the exponential function to the results.

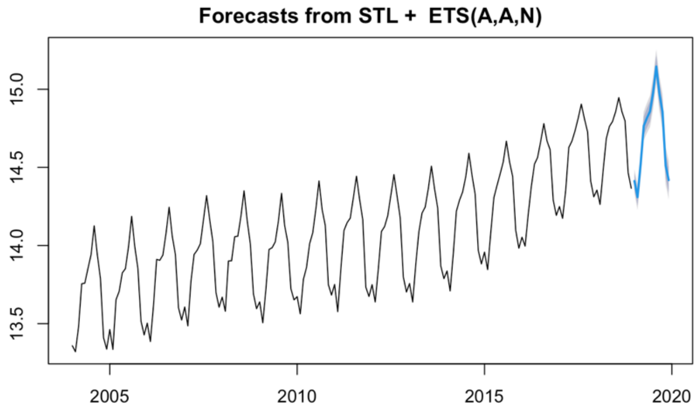
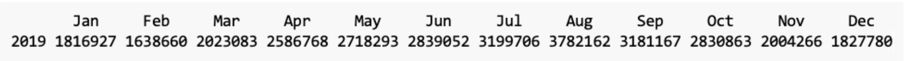


Fig. 11. Plot of the Number of Passengers Forecast using STL method.



```
         Jan     Feb     Mar     Apr     May     Jun     Jul     Aug     Sep     Oct     Nov     Dec
2019 1816927 1638660 2023083 2586768 2718293 2839052 3199706 3782162 3181167 2830863 2004266 1827780
```

Fig. 12. Forecast of Number of Passengers from STL Decomposition on R

5 ARIMA Models

After using methods that assume that the observations are not correlated, it was decided to use a method (ARIMA) that considers the correlation between observations, since in reality this is the most typical behavior.

These models can be defined using three parameters: p is the order of the autoregressive model; d is the degree of differencing and q is the order of the moving average. In this case, since the time series has seasonality, it will be used the seasonal ARIMA that has 4 more parameters to be defined (P, D, Q) the first free represent the same information as the ones referred but applied to the seasonal component and the last parameter m, represents the number of periods in each season. The correct definition of this parameters will conduct to a better forecasting increasing the accuracy of the model.

As previously stated, the time series has seasonality and trend, it doesn’t appear to be stationary, and the variance increases with time. The first step of this analysis was to perform the autocorrelation and partial autocorrelation of the time series, present in figure 13, where there are multiple values outside the rejection zone, the values of the ACF don’t go to zero proving the non-stationarity and there is a visible pattern proving that it has seasonality.

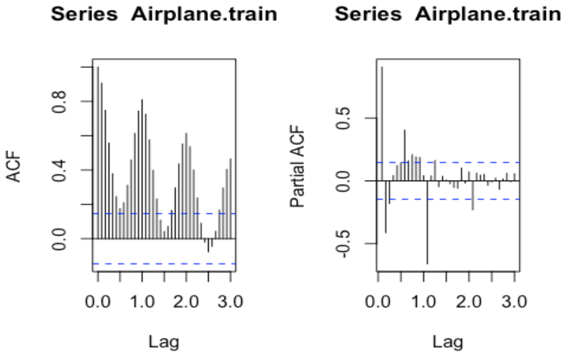


Fig. 13. ACF and PACF of the Number of Passengers Time Series

The next step was to stabilize the variance of the time series by applying taking logarithms, and as observed in figure 14 the time series still contains seasonality and trend; hence it is not stationary.

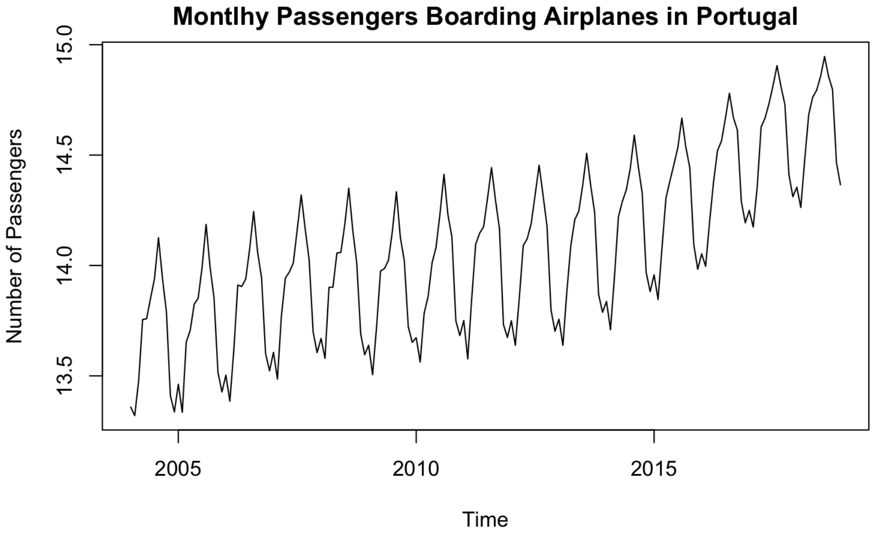


Fig. 14. Plot of the Logarithm Number of Passengers Boarding Airplanes in Portugal

Since we want to transform the time series into stationary, the next step was to perform a seasonal difference (D=1) obtaining the following plot where the seasonal component doesn't appear to be present.

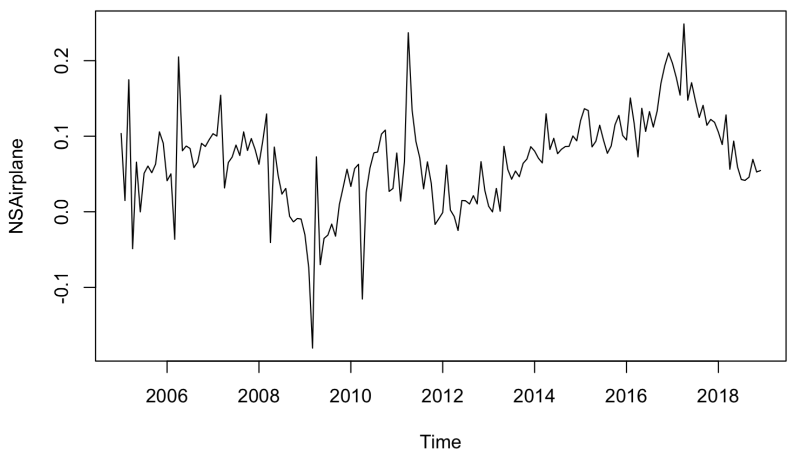
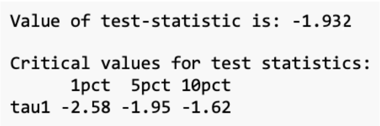


Fig. 15. Plot of the Number of Passengers Time Series with one seasonal difference

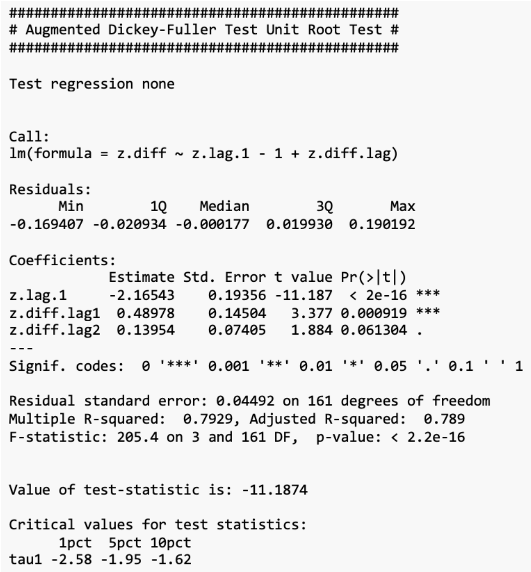
After plotting the data, in order to understand if the time series is now stationary it was applied a unit root test called Augmented Dickey-Fuller test, where the null hypotheses is the time series is nonstationary. The results obtain on this test didn't allow us to reject the null hypotheses since the value of the test statistic -1,932 is higher than the critical values of 1pct and 5pct.



```
Value of test-statistic is: -1.932

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Fig. 16. Result of the Augmented Dickey-Fuller unit root test for D=1

The removal of the seasonal component was important, but since we still don't have a stationary version of the time series it was applied a difference using lag=1 (d=1). After applying the difference, the unit root test was once again made and this time the value of the test statistics is lower than the critical values and with a p-value close to 0, rejecting the null hypotheses, the time series is now considered to be stationary and there is no need to apply any more differences (d=1 and D=1).



```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.169407 -0.020934 -0.000177  0.019930  0.190192 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
z.lag.1     -2.16543    0.19356 -11.187  < 2e-16 ***
z.diff.lag1  0.48978    0.14504   3.377 0.000919 ***
z.diff.lag2  0.13954    0.07405   1.884 0.061304 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04492 on 161 degrees of freedom
Multiple R-squared:  0.7929,	Adjusted R-squared:  0.789 
F-statistic: 205.4 on 3 and 161 DF,  p-value: < 2.2e-16


Value of test-statistic is: -11.1874 

Critical values for test statistics: 
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Fig. 17. Result of the Augmented Dickey-Fuller unit root test for D=1 and d=1

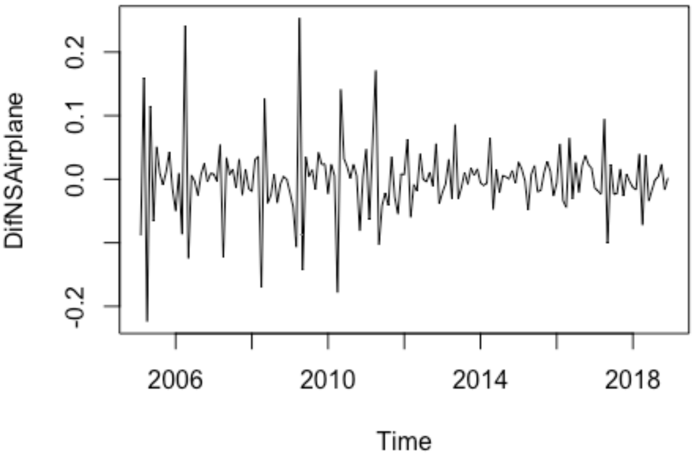


Fig. 18. Plot of the Number of Passengers Boarding Airplanes in Portugal with for D=1 and d=1

With the parameters m, d and D defined, it is necessary to define the remain parameters. To define them, the ACF and PACF of the stationary data was plotted and we can observe in the ACF there is a spike in the non-seasonal component and the PACF tails of so we will consider q=1. Then on the seasonal component we can see that the PACF tails off and the ACF also tails off which may represent an ARMA(1,1), so we will assume that P=1 and Q=1. The remain parameters will be 0, having the following model SARIMA(0,1,1)(1,1,1)12.

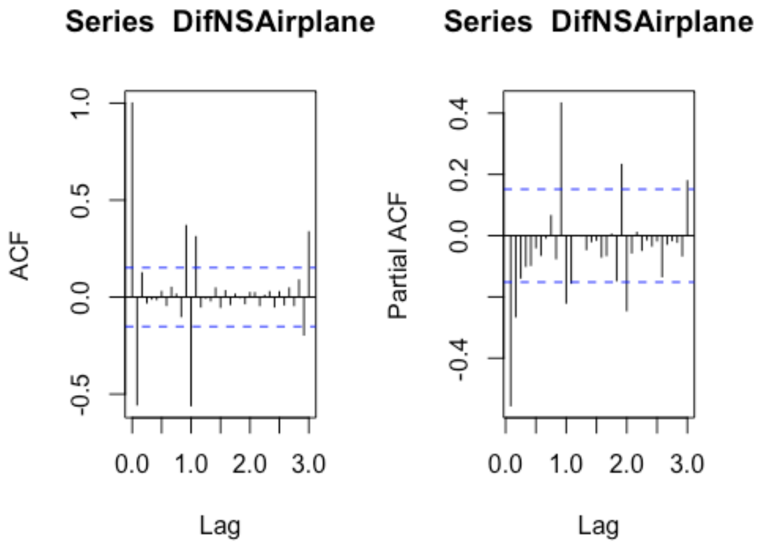
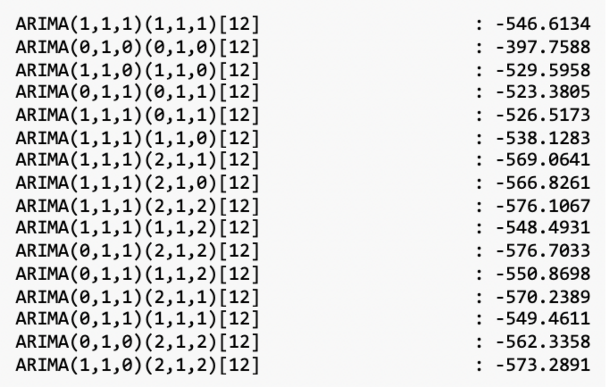


Fig. 19. ACF and PACF of the Number of Passengers Time Series with D=1 and d=1

To understand what the best parameters for the SARIMA model would be, we have performed tested models with other parameters and observed the aic of each one being the one with the lowest value the model that fits the best the time series. With the aic -576,7033 the model SARIMA(0,1,1)(2,1,2) was the best so for this project, both the one selected with the help of ACF and PACF and the model with the lowest aic were considered.



```
ARIMA(1,1,1)(1,1,1)[12]                    : -546.6134
ARIMA(0,1,0)(0,1,0)[12]                    : -397.7588
ARIMA(1,1,0)(1,1,0)[12]                    : -529.5958
ARIMA(0,1,1)(0,1,1)[12]                    : -523.3805
ARIMA(1,1,1)(0,1,1)[12]                    : -526.5173
ARIMA(1,1,1)(1,1,0)[12]                    : -538.1283
ARIMA(1,1,1)(2,1,1)[12]                    : -569.0641
ARIMA(1,1,1)(2,1,0)[12]                    : -566.8261
ARIMA(1,1,1)(2,1,2)[12]                    : -576.1067
ARIMA(1,1,1)(1,1,2)[12]                    : -548.4931
ARIMA(0,1,1)(2,1,2)[12]                    : -576.7033
ARIMA(0,1,1)(1,1,2)[12]                    : -550.8698
ARIMA(0,1,1)(2,1,1)[12]                    : -570.2389
ARIMA(0,1,1)(1,1,1)[12]                    : -549.4611
ARIMA(0,1,0)(2,1,2)[12]                    : -562.3358
ARIMA(1,1,0)(2,1,2)[12]                    : -573.2891
```

Fig. 20. Values of aic for different SARIMA models

Before using the two models for forecasting it was analyzed the residuals of the models in the figure 21 and 22.

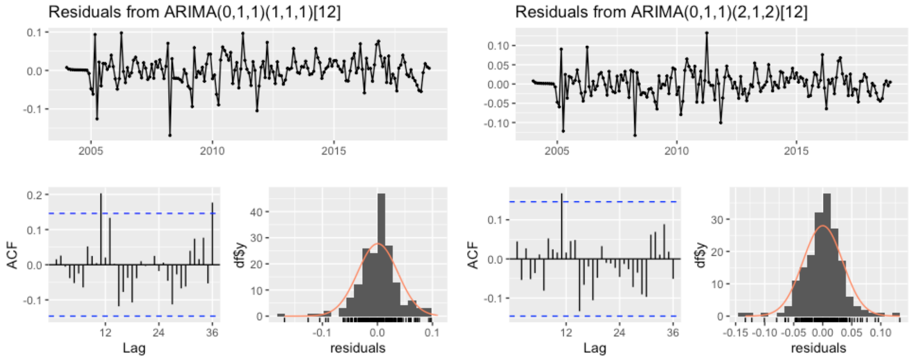
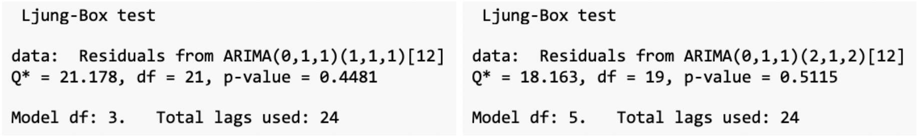


Fig. 21. Residuals from the SARIMA models used in this project



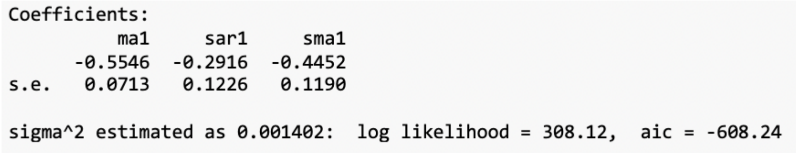
```
 Ljung-Box test

data:  Residuals from ARIMA(0,1,1)(1,1,1)[12]
Q* = 21.178, df = 21, p-value = 0.4481

Model df: 3.   Total lags used: 24
```

```
 Ljung-Box test

data:  Residuals from ARIMA(0,1,1)(2,1,2)[12]
Q* = 18.163, df = 19, p-value = 0.5115

Model df: 5.   Total lags used: 24
```

Fig. 22. Ljung-Box for Residuals from the SARIMA models used in this project

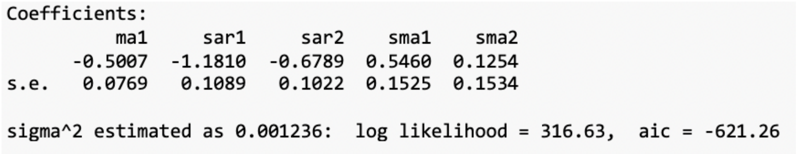
The analysis of the residuals of both models allows us to understand that on the second model there's only one spike on the ACF and the mean value is closer to zero, being the residuals more similar to a white noise in this model. Finally, the Ljung-Box test shows us that the residuals in both models do not have autocorrelation.

In order to forecast using these models, and as in the other models the data set was divided into training and test and with the training test and the SARIMA modesl above described we obtained the coefficients below.



```
Coefficients:
          ma1     sar1     sma1
      -0.5546  -0.2916  -0.4452
s.e.   0.0713   0.1226   0.1190

sigma^2 estimated as 0.001402:  log likelihood = 308.12,  aic = -608.24
```

Fig. 23. Coefficients of SARIMA(0,1,1)(1,1,1)12



```
Coefficients:
          ma1     sar1     sar2    sma1    sma2
      -0.5007  -1.1810  -0.6789  0.5460  0.1254
s.e.   0.0769   0.1089   0.1022  0.1525  0.1534

sigma^2 estimated as 0.001236:  log likelihood = 316.63,  aic = -621.26
```

Fig. 24. Coefficients of SARIMA(0,1,1)(2,1,2)12

Finally using the coefficients from the models, it was possible to forecast the last 12 months (the year 2019), obtaining the plots represented in figure 25 and 26.

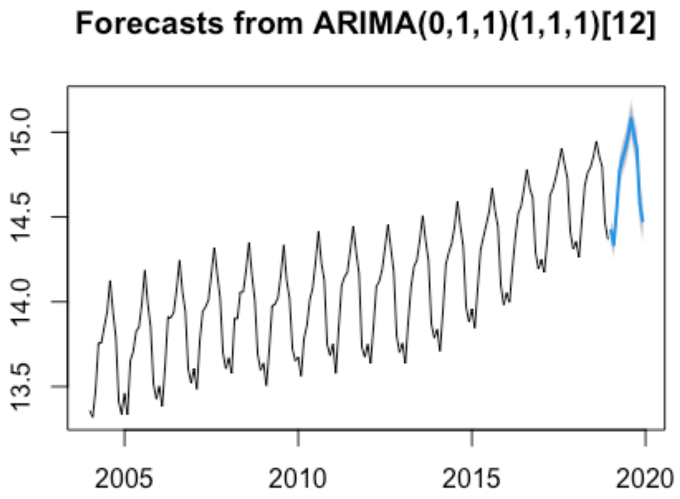


Fig. 25. Plot of the Number of Passengers Forecast using SARIMA(0,1,1)(1,1,1)12

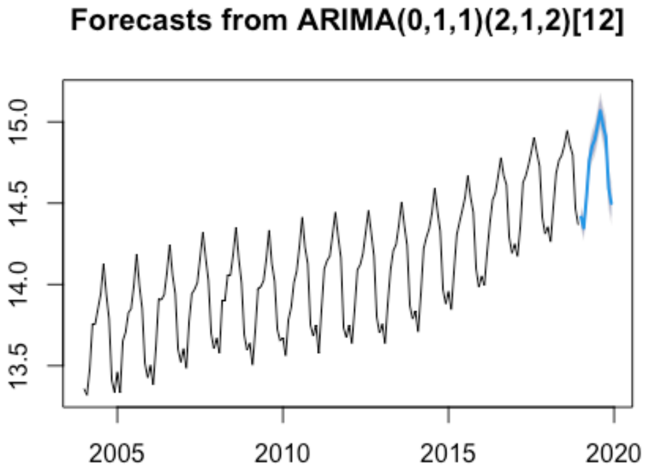


Fig. 26. Plot of the Number of Passengers Forecast using SARIMA(0,1,1)(2,1,2)12

6 Compare Results

There are many statistical measures to evaluate accuracy of the models used by comparing the values observed with the values predicted. The methods that will be used on this report is the mean squared error (MSE) and the mean absolute percentage error (MAPE) being the last one the easiest to read. In the table 1 it is possible to observe the real value of the number of passengers boarding airplanes in Portugal, the forecasting of the Holt-Winters, Classical Decomposition method, STL Decomposition method and SARIMA models as well as the MSE and MAPE of each one of them.

Table 1. Comparison of the different forecasting models

Month	**Observed**	**Holt-Winters**	**Classical Decomposition**	**STL Decomposition**	**SARIMA (0,1,1)(1,1,1)12**	**SARIMA (0,1,1)(2,1,2)12**
Jan 2019	1816042	1765738	1770255	1816927	1829797	1820356
Feb 2019	1653622	1605100	1577597	1638660	1684955	1700044
Mar 2019	2100750	1959002	2005126	2023083	2061948	2059370
Apr 2019	2574859	2429493	2423521	2586768	2602163	2581079
May 2019	2721621	2617758	2605863	2718293	2803334	2813130
Jun 2019	2899081	2740027	2719507	2839052	2954281	2932145
Jul 2019	3038557	2966284	3063131	3199706	3205788	3192391
Aug 2019	3303479	3299631	3562692	3782162	3547034	3499981
Sep 2019	2995654	3018025	3039692	3181167	3202982	3190916
Oct 2019	2779066	2808721	2696418	2830863	2970127	2993116
Nov 2019	2051939	1999233	1893653	2004266	2141044	2174030
Dec 2019	1889375	1785698	1732153	1827780	1938365	1979314
	MSE	8.528.284.582	17.659.048.880	25.690.324.927	15.855.964.698	15.217.453.007
	MAPE	3,14	4,76	3,39	3,67	3,81

With the table above it is possible to conclude that the model tested that fits the best for the time series analyzed was the Holt-Winters method since the MSE and MAPE have the lowest values and the worst model it was the Classical Decomposition with the highest MAPE. Analyzing the two SARIMA models used is also interesting to see that the model that was designed by looking to the ACF and PACF has performed better when looking to the mean absolute percentage error.

7 Conclusion

The goal of this report was to test different forecasting methods to predict the monthly number of passengers boarding airplanes in Portugal, and after applying all the test it was compared the accuracy of each model.

During the project it was possible to understand that it is important to have an overview of the time series since the beginning, in order to use that information in the selection of models, for example addictive or multiplicative.

In the case of smoothing methods, the weight of the most recent data or the oldest is very determinant in the accuracy of the forecasting.

It was also important to understand that the time series is composed by different components: seasonal, trend (trend and cycle) and random. Being the separation of this components very important to select the model that fits the best.

In the ARIMA models it is important to transform the time series into stationary and analyze the ACF and PACF, as well as the residuals, very carefully in order to select the model with the highest accuracy.

Finally in the case of the dataset analyzed it was stated that the model that fits the best was the Holt-Winters method and the worst the Classical Decomposition.

References

1. Monthly Number of Passengers Boarding Airplanes in Portugal, Retrieved 14 June 2022, from https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_indicadores&indOcorrCod=0000861&contexto=bd&selTab=tab2
2. Forecasting Principles and Practice, Retrieved 15 June 2022, from https://otexts.com/fpp2/
3. Wei, W.W.S.; Time Series Analysis - Univariate and Multivariate Methods, Pearson/Addison-Wesley, 2006. ISBN: 0-321-32216-9
4. Cryer, Jonathan D.; Time series analysis : with applications in R, 2009

Appendices

A- R Script

1- Import Data

```
setwd("~/Desktop/FEP/1.2/2. MPST/Assignment/2nd Assignment") library(forecast)
library(forecast)
library(urca)
library(tseries)

Airplane.total<-ts(scan("AirplaneTotal.txt"),start = c(2004,1), end= c(2019,12),deltat=1/12)
Airplane.train <- ts(scan("AirplaneTrain.txt"), start = c(2004,1), end= c(2018,12),deltat=1/12)
Airplane.test <- ts(scan("AirplaneTest.txt"), start = c(2019,1), end= c(2019,12),deltat=1/12)
```

2- Time Series Analisys

```
plot(Airplane.total,ylab= "Number of Passengers", main= "Montlhy Passengers Boarding Airplanes in Portugal")
monthplot(Airplane.total,ylab= "Number of Passengers", main= "Montlhy Passengers Boarding Airplanes in Portugal")
boxplot(Airplane.total~cycle(Airplane.total),xlab="Month", ylab= "Number of Passengers", main= "Montlhy Passengers Boarding Airplanes in Portugal")
seasonplot(Airplane.total,year.labels=TRUE, ylab= "Number of Passengers", main= "Seasonal Passengers Boarding Airplanes in Portugal", xlab = NULL)
summary(Airplane.total)
```

3- Smoothing Methods

```
HW<-HoltWinters(Airplane.train, seasonal="multiplicative")
HW
forHW<-forecast(HW,h=12)
pHW<-forHW$mean
predict(HW,n.ahead=12)
plot(forHW)
```

4- Decomposition Methods

```
#Classical Decomposition
decomp<-decompose(Airplane.train, type="m")
plot(decomp)
decomp
trend.no.NA.m <- na.remove(decomp$trend)
forecast.holt.m <- predict(HoltWinters(trend.no.NA.m,gamma=FALSE), n.ahead=18)
forDecompNoTrend <- ts(forecast.holt.m[7:18],start=c(2019,1),deltat=1/12)
pDecomp <- decomp$seasonal[1:12] * forDecompNoTrend
pDecomp

#STL
STL <- stl(log(Airplane.train),s.window="periodic" ,robust=T, inner=2, outer=20)
plot(STL, main="STL Decomposition")
STL
forSTL<-forecast(STL, h=12)
pSTL<-exp(forSTL$mean)
pSTL
plot(forSTL)
```

5- Arima Model

```
#Plot ACF and PACF with K< n/4
par(mfrow=c(1,2))
acf(Airplane.train,lag.max = 36)
pacf(Airplane.train,lag.max = 36)
par(mfrow=c(1,1))

#In order to reduce the Variance of the time series we start by applying the log
Airplane<- log(Airplane.train)
plot(Airplane, ylab= "Number of Passengers", main= "Montlhy Passengers Boarding Airplanes in Portugal")
par(mfrow=c(1,2))
acf(Airplane,lag.max = 36)
pacf(Airplane,lag.max = 36)
par(mfrow=c(1,1))
summary(ur.df(Airplane, type="none", lags=2))

#Since it has Seasonality, we need to take seasonal difference
NSAirplane<-diff(Airplane,lag=12)
plot(NSAirplane)
par(mfrow=c(1,2))
acf(NSAirplane,lag.max = 36)
pacf(NSAirplane,lag.max = 36)
```

```
par(mfrow=c(1,1))

#Perform an unit root test to see if the time series is stationary - We have Selected the ADF test
summary(ur.df(NSAirplane, type="none", lags=2))

#Since H0 is not rejected we need to perform a difference
DifNSAirplane<-diff(NSAirplane)
plot(DifNSAirplane)
#Perform once again the unit root test
summary(ur.df(DifNSAirplane, type="none", lags=2))

#Plot ACF and PACF in order to understand where it cuts and tails off
par(mfrow=c(1,2))
acf(DifNSAirplane,lag.max = 36)
pacf(DifNSAirplane,lag.max = 36)
par(mfrow=c(1,1))

#Confirm Which Arima Model is the best
auto.arima(Airplane,d=1,D=1, max.p=1,max.q=1,max.P=2,max.Q=2, max.order=6,ic="aic",trace=T)
#SARIMA Model
SARIMAMan<-arima(Airplane, order=c(0,1,1),list(order=c(1,1,1),period=12))
summary(SARIMAMan)
checkresiduals(SARIMAMan)

forSARIMAMan <- forecast(SARIMAMan,h=12)
forSARIMAMan
pSARIMAMan<-exp(forSARIMAMan$mean)
pSARIMAMan
plot(forSARIMAMan)

#Sarima Best
SARIMABest<-arima(Airplane, order=c(0,1,1),list(order=c(2,1,2),period=12))
summary(SARIMABest)
checkresiduals(SARIMABest)

forSARIMABest <- forecast(SARIMABest,h=12)
forSARIMABest
pSARIMABest<-exp(forSARIMABest$mean)
pSARIMABest
plot(forSARIMABest)
```

6-Compare the Models

```
sum((pHW-Airplane.test)^2)/12
100/12*sum(abs(pHW-Airplane.test)/Airplane.test)
sum((pDecomp-Airplane.test)^2)/12
100/12*sum(abs(pDecomp-Airplane.test)/Airplane.test)
sum((pSTL-Airplane.test)^2)/12
100/12*sum(abs(pSTL-Airplane.test)/Airplane.test)
sum((pSARIMAMan-Airplane.test)^2)/12
100/12*sum(abs(pSARIMAMan-Airplane.test)/Airplane.test)
sum((pSARIMABest-Airplane.test)^2)/12
100/12*sum(abs(pSARIMABest-Airplane.test)/Airplane.test)

cbind(Airplane.test,pHW,pDecomp,pSTL,pSARIMAMan,pSARIMABest)
```