**Date: /01/2020**

**Practical No. 5**

**AIM: Time-series forecasting**

**Theory:**

The act of forecasting consists in saying that (something) will happen in the future or to predict (something, such as weather) after looking at the information that is available. Forecasting is a form of prediction. In time series analysis we often want to predict something in the future on the basis of what we have observed in the past. The evaluation of the accuracy of our prediction is an important part of model evaluation. Poor accuracy means that our prediction will be scarcely helpful.

Making predictions about the future is called extrapolation in the classical statistical handling of time series data. Descriptive models can borrow for the future (i.e. to smooth or remove noise), they only seek to best describe the data. An important distinction in forecasting is that the future is completely unavailable and must only be estimated from what has already happened. The skill of a time series forecasting model is determined by its performance at predicting the future. This is often at the expense of being able to explain why a specific prediction was made, confidence intervals and even better understanding the underlying causes behind the problem.

The example is based on the dataset “AirPassengers”

> # load the necessary library

> library(forecast)

This is forecast 8.10

Stackoverflow is a great place to get help on R issues:

http://stackoverflow.com/tags/forecasting+r.

Warning message:

package ‘forecast’ was built under R version 3.5.3

> data(AirPassengers)

> AirPassengers

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

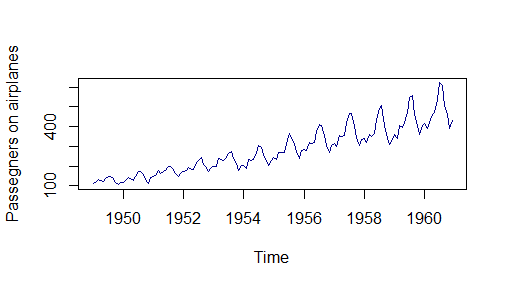
### data are assigned to a convenient vector

### this is an easy way to avoid changing the code every time

series <- AirPassengers

# plot the series

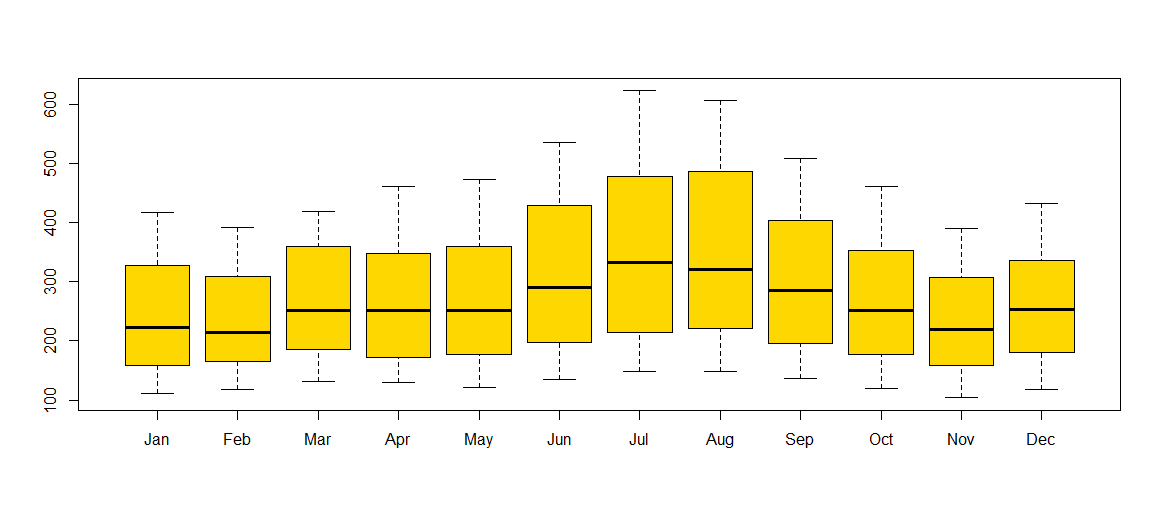
plot(series, col="darkblue", ylab="Passegners on airplanes")



# plot the seasonal distribution of the series

windows(width=800,height=350) # set the window with the dimensions you need

boxplot(split(series, cycle(series)), names = month.abb, col = "gold")



Important Inferences

1. The year on year trend clearly shows that the #passengers have been increasing without fail.

2. The variance and the mean value in July and August is much higher than rest of the months.

3. Even though the mean value of each month is quite different their variance is small. Hence, we have strong seasonal effect with a cycle of 12 months or less.

The accuracy of forecasts can only be determined by considering how well a model performs on new data that were not used when fitting the model.

The size of the test set is typically about 20% of the total sample

So, we will split the series in a training set and a test set

### training set

### use data from 1949 to 1956 for forecasting

sr = window(series, start = 1949, end = c(1956, 12))

### test set

### use remaining data from 1957 to 1960 to test accuracy

ser = window(series, start = 1957, end = c(1960, 12))

######################################################################

# plot training set

######################################################################

# plot forecasting for 5 years according to four methods

# lwd does modify the line width relative to the default (default=1)

# The command 'plot.conf=FALSE' avoids the plotting of the prediction intervals.

plot(meanf(sr,h=48), col=4, plot.conf=FALSE, main="AirPassengers", ylab="", xlab="Months", ylim = c(100,600), lwd=2)

lines(rwf(sr,h=48)$mean, col=2, lwd=2)

lines(rwf(sr,drift=TRUE,h=48)$mean, col=3, lwd=2)

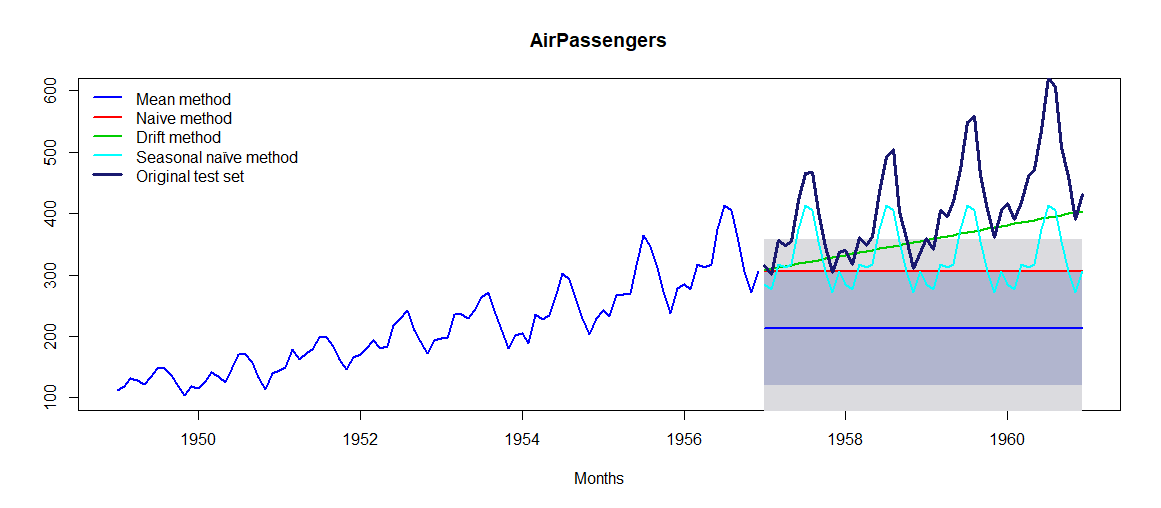
lines(snaive(sr,h=48)$mean, col=5, lwd=2)

# the test set

lines(ser, col="midnightblue", lwd = 3)

# legend

legend ("topleft", lty=1, col=c(4,2,3,5, "midnightblue"), lwd = c(2,2,2,2,3), legend=c("Mean method","Naive method","Drift method", "Seasonal naïve method", "Original test set"),bty="n")



# accuracy for forecasting of sr (forecasted data) on ser (original data used as test set)

# the best model had the lowest error (particularly the MAPE, Mean absolute percentage error)

# Mean method

accuracy(meanf(sr,h=48), ser)

# Naive method

accuracy(rwf(sr,h=48), ser)

# Drift method

accuracy(rwf(sr,drift=TRUE,h=48), ser)

# Seasonal naïve method

accuracy(snaive(sr,h=48), ser)

######################################################################

# plot test set only with the predictions

######################################################################

# calculate the forecasting

sr.mean <- meanf(sr,h=48)$mean

sr.naive <- rwf(sr,h=48)$mean

sr.drift <- rwf(sr,drift=TRUE,h=48)$mean

sr.seas <- snaive(sr,h=48)$mean

# plot the test set

plot(ser, col="midnightblue", main="AirPassengers", ylab="", xlab="Months", ylim = c(200,600), lwd = 2)

# plot forecasting for 4 years according to four methods

lines(sr.mean, col=4)

lines(sr.naive, col=2)

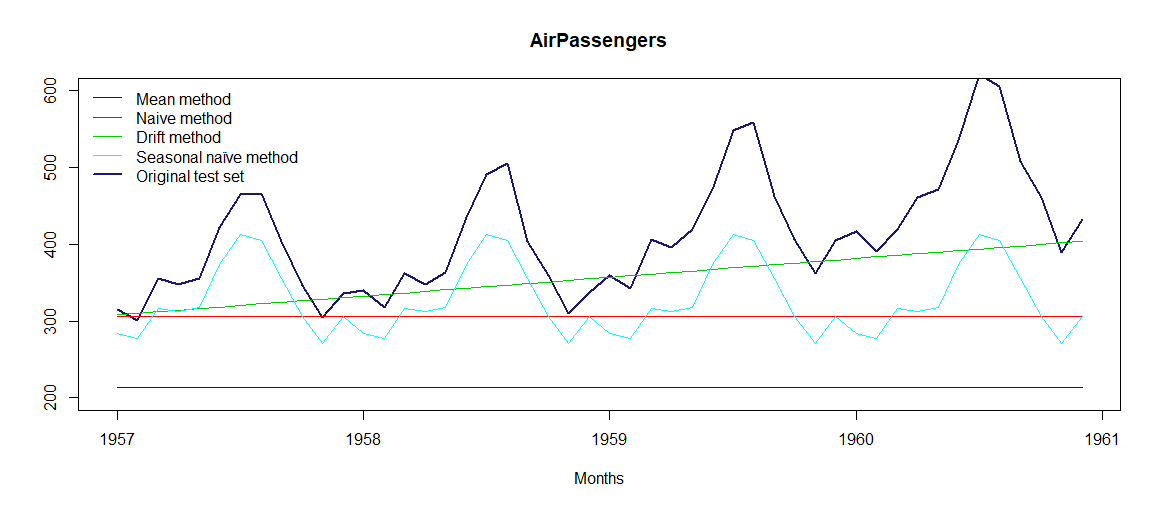
lines(sr.drift, col=3)

lines(sr.seas, col=5)

# legend

legend("topleft", lty=1, col=c(4,2,3,5, "midnightblue"), lwd = c(1,1,1,1,2),

legend=c("Mean method","Naive method","Drift method", "Seasonal naïve method", "Original test set"),bty="n"



It is rather obvious than none of these methods produce a good forecast of the series.

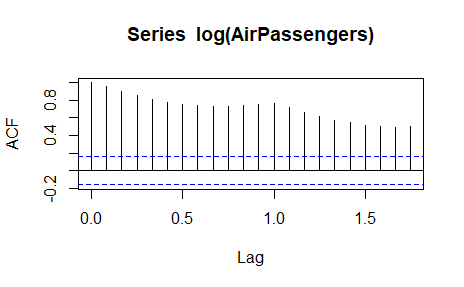
The mean method and the naive method do not detect nor the trend neither the seasonality in the series. The drift method does detect the trend but not the seasonality, while the seasonal naïve method does the reverse. The best method, on the basis of the Mean absolute percentage error (MAPE) is the drift method, which in my view suggests that the trend is more important than the seasonality in this series.

Exploring data becomes most important in a time series model – without this exploration, you will not know whether a series is stationary or not. As in this case we already know many details about the kind of model we are looking out for. Let’s now take up a few time series models and their characteristics. We will also take this problem forward and make a few predictions.

**Auto – correlation Function(ACF):** ACF is a plot of total correlation between different lag functions.

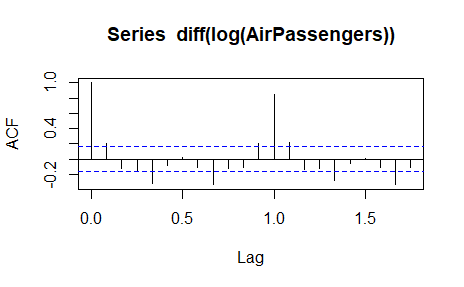
Following are the ACF plots for the series:

> acf(log(AirPassengers))



Clearly, the decay of ACF chart is very slow, which means that the population is not stationary. We have already discussed above that we now intend to regress on the difference of logs rather than log directly. Let’s see how ACF curve come out after regressing on the difference.

> acf(diff(log(AirPassengers)))



> (fit <- arima(log(AirPassengers), c(0, 1, 1),seasonal = list(order = c(0, 1, 1), period = 12)))

Call:

arima(x = log(AirPassengers), order = c(0, 1, 1), seasonal = list(order = c(0,

1, 1), period = 12))

Coefficients:

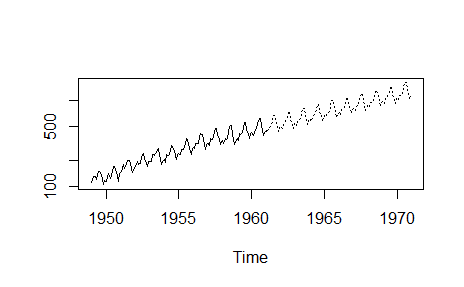
ma1 sma1

-0.4018 -0.5569

s.e. 0.0896 0.0731

sigma^2 estimated as 0.001348: log likelihood = 244.7, aic = -483.4

> pred <- predict(fit, n.ahead = 10\*12)



**Entire Code:**

# load the necessary library

library(forecast)

data(AirPassengers)

# inspect the series

AirPassengers

### data are assigned to a convenient vector

### this is a easy way to avoid changing the code every time

series <- AirPassengers

# plot the series

plot(series, col = "darkblue", ylab = "Passegners on airplanes")

# plot the seasonal distribution of the series

windows(width = 800, height = 350) # set the window with the dimensions you need

boxplot(split(series, cycle(series)), names = month.abb, col = "gold")

### training set

### use data from 1949 to 1956 for forecasting

sr = window(series, start = 1949, end = c(1956, 12))

### test set

### use remaining data from 1957 to 1960 to test accuracy

ser = window(series, start = 1957, end = c(1960, 12))

######################################################################

# plot training set

######################################################################

# plot forecasting for 5 years according to four methods

# lwd does modify the line width relative to the default (default=1)

# The command 'plot.conf=FALSE' avoids the plotting of the prediction intervals.

plot(meanf(sr,h=48), col=4, plot.conf=FALSE, main="AirPassengers", ylab="", xlab="Months", ylim = c(100,600), lwd=2)

lines(rwf(sr,h=48)$mean, col=2, lwd=2)

lines(rwf(sr,drift=TRUE,h=48)$mean, col=3, lwd=2)

lines(snaive(sr,h=48)$mean, col=5, lwd=2)

# the test set

lines(ser, col="midnightblue", lwd = 3)

# legend

legend ("topleft", lty=1, col=c(4,2,3,5, "midnightblue"), lwd = c(2,2,2,2,3), legend=c("Mean method","Naive method","Drift method", "Seasonal naïve method", "Original test set"),bty="n")

# accuracy for forecasting of sr (forecasted data) on ser (original data used as test set)

# the best model had the lowest error (particularly the MAPE, Mean absolute percentage error)

# Mean method

accuracy(meanf(sr,h=48), ser)

# Naive method

accuracy(rwf(sr,h=48), ser)

# Drift method

accuracy(rwf(sr,drift=TRUE,h=48), ser)

# Seasonal naïve method

accuracy(snaive(sr,h=48), ser)

######################################################################

# plot test set only with the predictions

######################################################################

# calculate the forecasting

sr.mean <- meanf(sr,h=48)$mean

sr.naive <- rwf(sr,h=48)$mean

sr.drift <- rwf(sr,drift=TRUE,h=48)$mean

sr.seas <- snaive(sr,h=48)$mean

# plot the test set

plot(ser, col="midnightblue", main="AirPassengers", ylab="", xlab="Months", ylim = c(200,600), lwd = 2)

# plot forecasting for 4 years according to four methods

lines(sr.mean, col=4)

lines(sr.naive, col=2)

lines(sr.drift, col=3)

lines(sr.seas, col=5)

# legend

legend("topleft", lty=1, col=c(4,2,3,5, "midnightblue"), lwd = c(1,1,1,1,2), legend=c("Mean method","Naive method","Drift method", "Seasonal naïve method", "Original test set"),bty="n")

########################################################################

# for ARIMA; Hyndman suggest to use auto-arima without stepwise

########################################################################

library(fpp)

trainData <- sr

testData <- ser

# The default value in auto.arima() is test="kpss".

# A KPSS test has a null hypothesis of stationarity

# In general, all the defaults are set to the values that give the best forecasts on average.

# CAUTION! Takes a while to compute

arimaMod <- auto.arima(trainData, stepwise=FALSE, approximation=FALSE)

arimaMod.Fr <-forecast(arimaMod,h=48)

# plot of the prediction and of the test set

plot(arimaMod.Fr)

lines(testData, col="red")

legend("topleft",lty=1,bty = "n",col=c("red","blue"),c("testData","ARIMAPred"))

# plot of the test set and its prediction only

AR.mean <-forecast(arimaMod,h=48)$mean

plot(testData, main="AirPassengers", ylab="", xlab="Months", col="darkblue")

lines(AR.mean, col="red")

# accuracy

accuracy(arimaMod.Fr,testData)

# test residues of arima

tsdisplay(residuals(arimaMod))

acf(log(AirPassengers))

acf(diff(log(AirPassengers)))

(fit <- arima(log(AirPassengers), c(0, 1, 1),seasonal = list(order = c(0, 1, 1), period = 12)))

pred <- predict(fit, n.ahead = 10\*12)

ts.plot(AirPassengers,2.718^pred$pred, log = "y", lty = c(1,3))

**Console:**

> # load the necessary library

> library(forecast)

> data(AirPassengers)

>

> # inspect the series

> AirPassengers

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

>

> ### data are assigned to a convenient vector

> ### this is a easy way to avoid changing the code every time

> series <- AirPassengers

>

> # plot the series

> plot(series, col = "darkblue", ylab = "Passegners on airplanes")

>

> # plot the seasonal distribution of the series

> windows(width = 800, height = 350) # set the window with the dimensions you need

>

> boxplot(split(series, cycle(series)), names = month.abb, col = "gold")

>

> ### training set

> ### use data from 1949 to 1956 for forecasting

>

> sr = window(series, start = 1949, end = c(1956, 12))

>

> ### test set

> ### use remaining data from 1957 to 1960 to test accuracy

>

> ser = window(series, start = 1957, end = c(1960, 12))

>

> ######################################################################

> # plot training set

> ######################################################################

>

> # plot forecasting for 5 years according to four methods

>

> # lwd does modify the line width relative to the default (default=1)

> # The command 'plot.conf=FALSE' avoids the plotting of the prediction intervals.

>

> plot(meanf(sr,h=48), col=4, plot.conf=FALSE, main="AirPassengers", ylab="", xlab="Months", ylim = c(100,600), lwd=2)

Warning messages:

1: In plot.window(xlim, ylim, log, ...) :

"plot.conf" is not a graphical parameter

2: In title(main = main, xlab = xlab, ylab = ylab, ...) :

"plot.conf" is not a graphical parameter

3: In axis(1, ...) : "plot.conf" is not a graphical parameter

4: In axis(2, ...) : "plot.conf" is not a graphical parameter

5: In box(...) : "plot.conf" is not a graphical parameter

> lines(rwf(sr,h=48)$mean, col=2, lwd=2)

> lines(rwf(sr,drift=TRUE,h=48)$mean, col=3, lwd=2)

> lines(snaive(sr,h=48)$mean, col=5, lwd=2)

>

> # the test set

> lines(ser, col="midnightblue", lwd = 3)

>

> # legend

> legend ("topleft", lty=1, col=c(4,2,3,5, "midnightblue"), lwd = c(2,2,2,2,3), legend=c("Mean method","Naive method","Drift method", "Seasonal naïve method", "Original test set"),bty="n")

>

> # accuracy for forecasting of sr (forecasted data) on ser (original data used as test set)

> # the best model had the lowest error (particularly the MAPE, Mean absolute percentage error)

>

> # Mean method

> accuracy(meanf(sr,h=48), ser)

ME RMSE MAE MPE MAPE

Training set -9.442172e-15 71.54266 59.07378 -11.67626 30.88522

Test set 1.997708e+02 214.34087 199.77083 46.59890 46.59890

MASE ACF1 Theil's U

Training set 2.022910 0.9281748 NA

Test set 6.840909 0.7915464 4.358243

> # Naive method

> accuracy(rwf(sr,h=48), ser)

ME RMSE MAE MPE MAPE

Training set 2.042105 23.33328 18.69474 0.5235867 8.715554

Test set 107.479167 132.60994 107.72917 23.5372013 23.620076

MASE ACF1 Theil's U

Training set 0.6401785 0.2471930 NA

Test set 3.6890542 0.7915464 2.527928

>

> # Drift method

> accuracy(rwf(sr,drift=TRUE,h=48), ser)

ME RMSE MAE MPE MAPE

Training set -1.121902e-14 23.24375 18.61186 -0.535584 8.71285

Test set 5.744759e+01 87.66931 63.89189 11.751252 13.72947

MASE ACF1 Theil's U

Training set 0.6373404 0.2471930 NA

Test set 2.1878999 0.7296203 1.645578

>

> # Seasonal naïve method

> accuracy(snaive(sr,h=48), ser)

ME RMSE MAE MPE MAPE MASE

Training set 28.79762 32.73323 29.20238 12.48988 12.68524 1.000000

Test set 85.22917 97.80092 85.22917 19.58538 19.58538 2.918569

ACF1 Theil's U

Training set 0.7740303 NA

Test set 0.8753624 1.932458

>

> ######################################################################

> # plot test set only with the predictions

> ######################################################################

>

> # calculate the forecasting

>

> sr.mean <- meanf(sr,h=48)$mean

> sr.naive <- rwf(sr,h=48)$mean

> sr.drift <- rwf(sr,drift=TRUE,h=48)$mean

> sr.seas <- snaive(sr,h=48)$mean

>

> # plot the test set

> plot(ser, col="midnightblue", main="AirPassengers", ylab="", xlab="Months", ylim = c(200,600), lwd = 2)

>

> # plot forecasting for 4 years according to four methods

> lines(sr.mean, col=4)

> lines(sr.naive, col=2)

> lines(sr.drift, col=3)

> lines(sr.seas, col=5)

>

> # legend

> legend("topleft", lty=1, col=c(4,2,3,5, "midnightblue"), lwd = c(1,1,1,1,2), legend=c("Mean method","Naive method","Drift method", "Seasonal naïve method", "Original test set"),bty="n")

>

> ########################################################################

> # for ARIMA; Hyndman suggest to use auto-arima without stepwise

> ########################################################################

>

> library(fpp)

>

> trainData <- sr

> testData <- ser

>

> # The default value in auto.arima() is test="kpss".

> # A KPSS test has a null hypothesis of stationarity

> # In general, all the defaults are set to the values that give the best forecasts on average.

>

> # CAUTION! Takes a while to compute

>

> arimaMod <- auto.arima(trainData, stepwise=FALSE, approximation=FALSE)

> arimaMod.Fr <-forecast(arimaMod,h=48)

>

> # plot of the prediction and of the test set

>

> plot(arimaMod.Fr)

> lines(testData, col="red")

> legend("topleft",lty=1,bty = "n",col=c("red","blue"),c("testData","ARIMAPred"))

>

> # plot of the test set and its prediction only

>

> AR.mean <-forecast(arimaMod,h=48)$mean

>

> plot(testData, main="AirPassengers", ylab="", xlab="Months", col="darkblue")

> lines(AR.mean, col="red")

>

> # accuracy

> accuracy(arimaMod.Fr,testData)

ME RMSE MAE MPE MAPE

Training set 0.7965016 8.55075 6.092191 0.2873468 2.768068

Test set -7.8983100 26.10435 21.079698 -2.6868389 5.185983

MASE ACF1 Theil's U

Training set 0.2086197 0.0175687 NA

Test set 0.7218486 0.6551648 0.5771478

>

> # test residues of arima

> tsdisplay(residuals(arimaMod))

> acf(log(AirPassengers))

> acf(diff(log(AirPassengers)))

> (fit <- arima(log(AirPassengers), c(0, 1, 1),seasonal = list(order = c(0, 1, 1), period = 12)))

Call:

arima(x = log(AirPassengers), order = c(0, 1, 1), seasonal = list(order = c(0,

1, 1), period = 12))

Coefficients:

ma1 sma1

-0.4018 -0.5569

s.e. 0.0896 0.0731

sigma^2 estimated as 0.001348: log likelihood = 244.7, aic = -483.4

> pred <- predict(fit, n.ahead = 10\*12)

> ts.plot(AirPassengers,2.718^pred$pred, log = "y", lty = c(1,3))

While plotting if you get error, use following command

Error in plot.new() : figure margins too large

> par(mar=c(1,1,1,1))