Time Series & Forecasting, End Semester Practical Exam

Pranav Gopalkrishna, 1940223

2022-05-10

# INITIAL DATA PROCESSING

setwd("~/Documents/Study/computerScience/programming/r/data/")  
data = read.csv("dailyTotalFemaleBirths.csv")

## Viewing data

head(data)

## Date Births  
## 1 1959-01-01 35  
## 2 1959-01-02 32  
## 3 1959-01-03 30  
## 4 1959-01-04 31  
## 5 1959-01-05 44  
## 6 1959-01-06 29

## Converting date strings to dates

data$Date = as.Date(data$Date)

## Summarisation

summary(data)

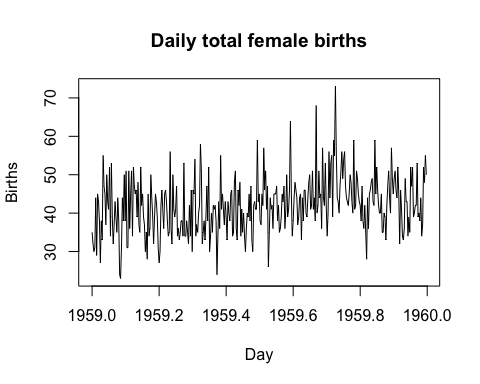
## Date Births   
## Min. :1959-01-01 Min. :23.00   
## 1st Qu.:1959-04-02 1st Qu.:37.00   
## Median :1959-07-02 Median :42.00   
## Mean :1959-07-02 Mean :41.98   
## 3rd Qu.:1959-10-01 3rd Qu.:46.00   
## Max. :1959-12-31 Max. :73.00

The data is taken for the whole year, daily.

#========================

# CONVERTING TO TIME SERIES

z = ts(data$Births, start=c(1959, 1), end=c(1959, 365), frequency=365)  
# frequency=365 => 1 time unit divided into 365 partitions.  
# Here, 1 time unit is 1 year.  
# start = c(1959, 1) => starting from the 1st partition of the year 1959.  
# end = c(1959, 365) => ending at the 365th partition of the year 1959.  
ts.plot(z,  
 main="Daily total female births",  
 xlab="Day",  
 ylab="Births")



#========================

# CHECKING STATIONARITY

library(tseries)

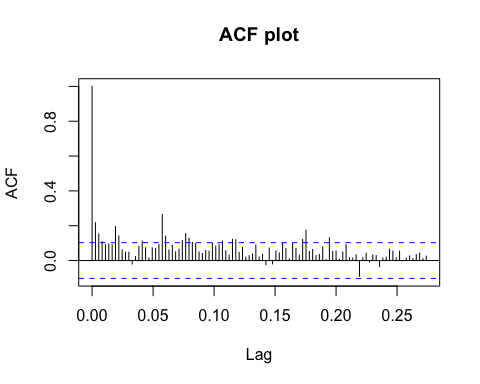
adf.test(z, alternative="stationary")

## Warning in adf.test(z, alternative = "stationary"): p-value smaller than printed  
## p-value

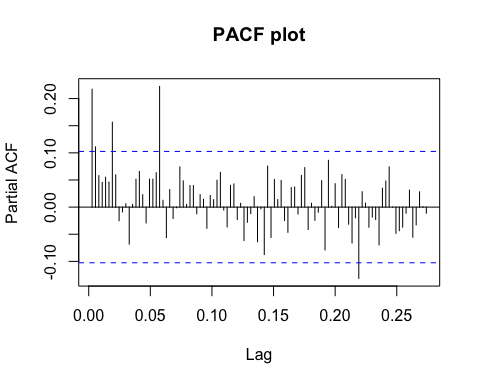
##   
## Augmented Dickey-Fuller Test  
##   
## data: z  
## Dickey-Fuller = -5.1042, Lag order = 7, p-value = 0.01  
## alternative hypothesis: stationary

From the augmented Dickey-Fuller test, we observe that the time series can be said to be stationary for upto 7 lags i.e. observations display stationarity when considering observations separated by at most 7 lags from each other.

acf(z,  
 main="ACF plot", 100)



pacf(z,  
 main="PACF plot", 100)



From ACF, we can observe largely insignificant autocorrelation between observations separated by upto 25 lags. The occasional significant autocorrelations may be chalked down to error.

#\_\_\_\_\_\_

Even from the PACF, we can observe largely insignificant autocorrelation between observations separated by upto 25 lags. This indicates that even after correcting for the effects of the lags between two lagged observations, the autocorrelation between observations is largely insignificant.

#\_\_\_\_\_\_

Both of the above, along with the fact that the data seems to display constant mean and finite variance, suggests moderate to strong stationarity.

#========================

# SUITABLE ARMA MODEL

Given the largely insignificant autocorrelation between observations, autoregression may be entirely absent from this time series process. Due to this, and due to its moderate to strong stationarity, we conclude that a moving average model may be more suitable for this time series. Hence, for the time series process model ARIMA(p, d, q), we put p = 0, d = 0, q = 1.

library(forecast)

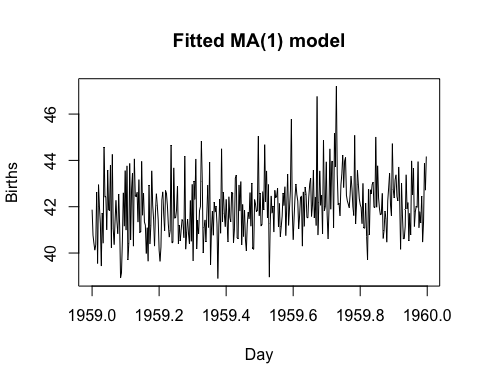
## Creating a MA(1) model…

model = auto.arima(z,  
 max.p=0,  
 max.d=0,  
 start.q=2,  
 max.q=1)  
summary(model)

## Series: z   
## ARIMA(0,0,1) with non-zero mean   
##   
## Coefficients:  
## ma1 mean  
## 0.1782 41.9807  
## s.e. 0.0465 0.4436  
##   
## sigma^2 = 52.07: log likelihood = -1238.27  
## AIC=2482.54 AICc=2482.61 BIC=2494.24  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0.002765522 7.196082 5.63229 -3.002113 13.93504 NaN 0.02284435

## Plotting fitted values

ts.plot(model$fitted,  
 main="Fitted MA(1) model",  
 xlab="Day",  
 ylab="Births")



#========================

# FITTING BEST NON-STATIONARY MODEL

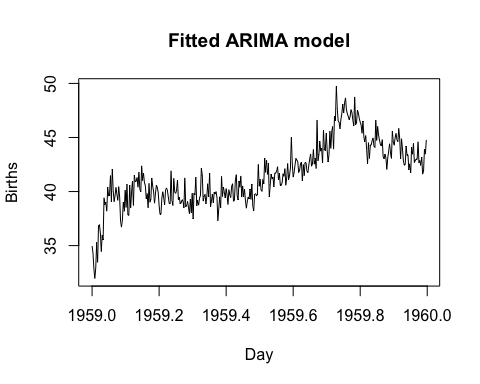
## Creating an ARIMA(p, d, q) model…

nonStationaryModel = auto.arima(z)  
summary(nonStationaryModel)

## Series: z   
## ARIMA(0,1,2)   
##   
## Coefficients:  
## ma1 ma2  
## -0.8478 -0.1079  
## s.e. 0.0497 0.0496  
##   
## sigma^2 = 49.49: log likelihood = -1226.79  
## AIC=2459.57 AICc=2459.64 BIC=2471.26  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0.4174241 7.0058 5.493027 -1.683678 13.3417 NaN 0.0009956016

## Plotting fitted values

ts.plot(nonStationaryModel$fitted,  
 main="Fitted ARIMA model",  
 xlab="Day",  
 ylab="Births")



#========================

# PREFERRED MODEL

Our preferred model would be MA(1), because not only does it visually match the time plot of the data better, but also, we are note preforming any differencing to make the data stationary, since the time series was already concluded to be sufficiently stationary. However, the fitted non-stationary model does not consider this conclusion, and performs differencing of order 1 on the time series, which creates discrepancies between the actual data and the fitted values.

#========================

# TESTING MODEL ADEQUACY

residuals = model$residuals

## Checking mean of residuals

mean(residuals)

## [1] 0.002765522

## Testing if the true mean of the residuals can be said to be 0

# (true mean => mean of the residuals in general, considering our residuals as a sample)  
t.test(residuals, mu=0)

##   
## One Sample t-test  
##   
## data: residuals  
## t = 0.0073322, df = 364, p-value = 0.9942  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.7389543 0.7444853  
## sample estimates:  
## mean of x   
## 0.002765522

We may accept null hypothesis, given significance level = 0.05. Hence, we may conclude that true mean of residuals= 0

## Testing autocorrelation

Box.test(residuals)

##   
## Box-Pierce test  
##   
## data: residuals  
## X-squared = 0.19048, df = 1, p-value = 0.6625

We may accept null hypthesis, given significance level = 0.05. Hence, we may conclude that autocorrelation residuals= 0

## Testing normality

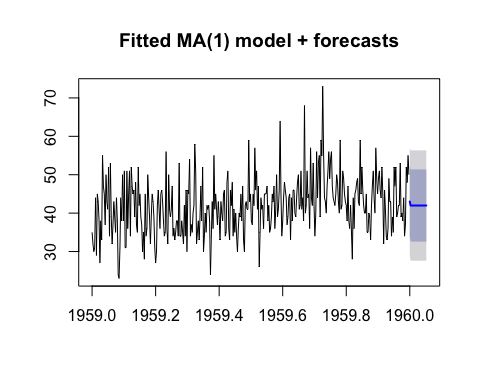
shapiro.test(residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: residuals  
## W = 0.98416, p-value = 0.0004924

We may reject null hypthesis, given significance level = 0.05. Hence, we may conclude that residuals are not normally distributed.

Due to non-normal distribution of the residuals, we may assume that the residuals may be caused by more than random error, hence indicating that our model may not be the fittest possible model for our data.

#========================  
forecasted = forecast(model, h=20)  
plot(forecasted)



## Printing the next 20 forecasted values (mean forecasted values)

for(f in forecasted$mean){print(f)}

## [1] 43.01944  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066  
## [1] 41.98066