Time Series Recreation

Read data from “A Little Book of R for Time Series” example.

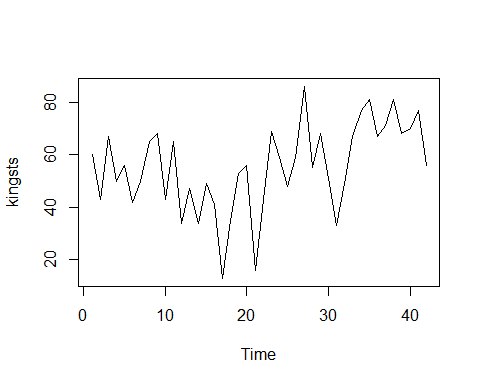
kings <- scan("http://robjhyndman.com/tsdldata/misc/kings.dat",skip=3)

Turn the df into a ts object.

kingsts <- ts(kings)

Plot the data to get an idea of the structure.

plot.ts(kingsts)

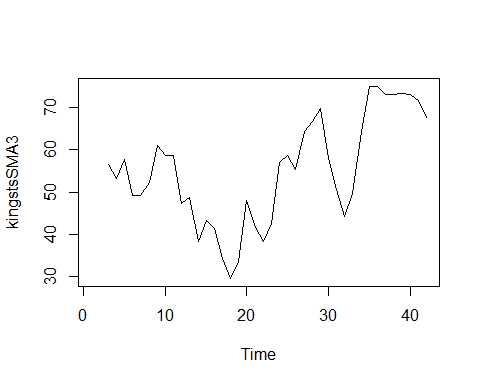


Import the TTR library and smooth out the data using a moving average since the data appears to be non-seasonal.

library("TTR")

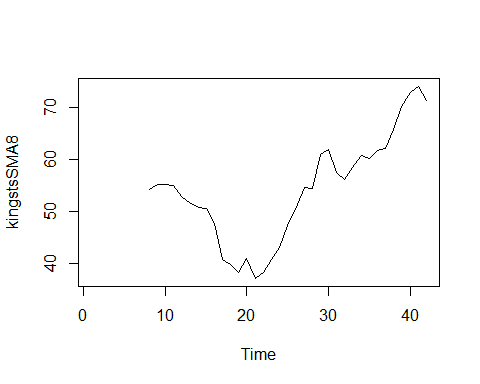
## Registered S3 method overwritten by 'xts':  
## method from  
## as.zoo.xts zoo

kingstsSMA3 <- SMA(kingsts,n=3)  
plot.ts(kingstsSMA3)



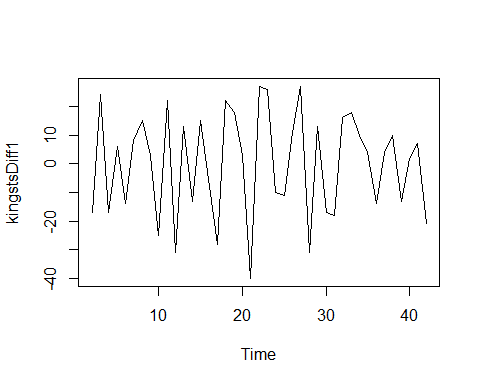
Change the order to a higher order to smooth out the random fluctuations further.

library("TTR")  
kingstsSMA8 <- SMA(kingsts,n=8)  
plot.ts(kingstsSMA8)



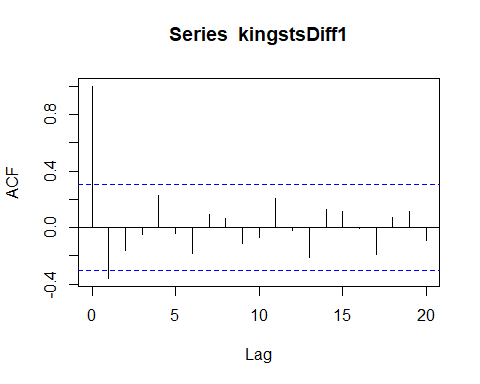
Change methods and use differencing.

kingstsDiff1 <- diff(kingsts, differences = 1)  
plot.ts(kingstsDiff1)



This now shows us that the data is stationary and an ARIMA model can be applied.

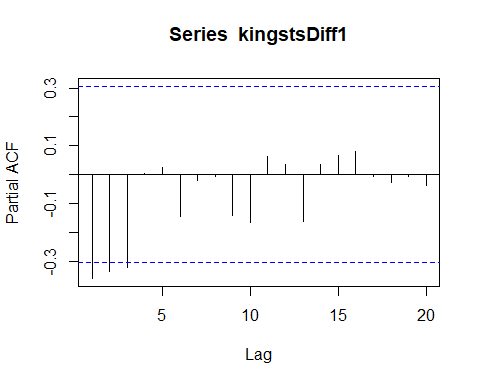
acf(kingstsDiff1, lag.max = 20)



acf(kingstsDiff1, lag.max = 20, plot = FALSE)

##   
## Autocorrelations of series 'kingstsDiff1', by lag  
##   
## 0 1 2 3 4 5 6 7 8 9   
## 1.000 -0.360 -0.162 -0.050 0.227 -0.042 -0.181 0.095 0.064 -0.116   
## 10 11 12 13 14 15 16 17 18 19   
## -0.071 0.206 -0.017 -0.212 0.130 0.114 -0.009 -0.192 0.072 0.113   
## 20   
## -0.093

pacf(kingstsDiff1, lag.max = 20)



pacf(kingstsDiff1, lag.max = 20, plot = FALSE)

##   
## Partial autocorrelations of series 'kingstsDiff1', by lag  
##   
## 1 2 3 4 5 6 7 8 9 10   
## -0.360 -0.335 -0.321 0.005 0.025 -0.144 -0.022 -0.007 -0.143 -0.167   
## 11 12 13 14 15 16 17 18 19 20   
## 0.065 0.034 -0.161 0.036 0.066 0.081 -0.005 -0.027 -0.006 -0.037

Here we can see that we shoud use a Q of 1 or a P of 3. The principle of parsimony led the author’s to choose the ARMA(0,1) model. Including the single differencing we come out to ARIMA(0,1,1).

kingsArima011 <- arima(kingsts, order = c(0,1,1))  
kingsArima011

##   
## Call:  
## arima(x = kingsts, order = c(0, 1, 1))  
##   
## Coefficients:  
## ma1  
## -0.7218  
## s.e. 0.1208  
##   
## sigma^2 estimated as 230.4: log likelihood = -170.06, aic = 344.13

Import the “forecast” package and forecast then plot the ARIMA model’s predictions.

library("forecast")

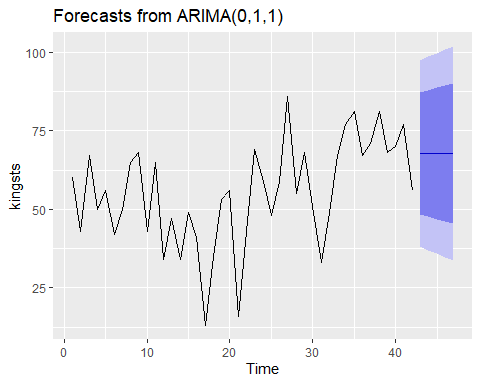
## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':  
## method from   
## fitted.fracdiff fracdiff  
## residuals.fracdiff fracdiff

kingstsForecast <- forecast(kingsArima011, h=5)  
kingstsForecast

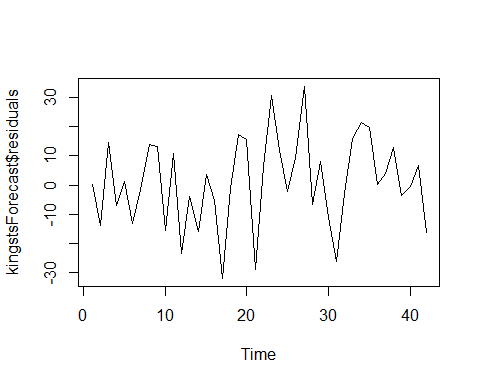
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 43 67.75063 48.29647 87.20479 37.99806 97.50319  
## 44 67.75063 47.55748 87.94377 36.86788 98.63338  
## 45 67.75063 46.84460 88.65665 35.77762 99.72363  
## 46 67.75063 46.15524 89.34601 34.72333 100.77792  
## 47 67.75063 45.48722 90.01404 33.70168 101.79958

autoplot(kingstsForecast)



The authors also plot the variance and residuals to determine if the errors or normally distributed with a mean of zero and constant variance. I am unable to run the plotForecastErrors function and cannot find documentation for it in the forecast package.

plot.ts(kingstsForecast$residuals)



#plotForecastErrors(kingstsForecast$residuals)