Lab 6

Pstat 174/274

Feb 13, 2020

Model Estimation

Summary of `R` commands for estimation:

1. To difference a time series at lag d:

   ```
   diff(data,lag = d)
   ```

2. To estimate parameters of an AR model:

   ```
   ar(data, aic = TRUE, order.max = NULL, method = c("..."))
   ```

   If `aic = TRUE`, Akaike's Information Criterion is used to select the order of the AR model to fit. If `aic = FALSE`, the model of order equal to `order.max` is fitted. `method` could be set equal to `"yule-walker"` or `"mle"` for different estimation methods of AR parameters.

3. To estimate parameters of an ARMA model:

   ```
   arima(data, order = c(p, 0, q), method = c("..."))
   ```

   The middle number in `order` is the number of times d the data should be differenced before fitting an ARMA model. For example, `order = c(2, 1, 2)` would fit an ARMA(2,2) once the original time series has been differenced (e.g., to remove the trend component). For regular ARMA models, put 0 for this number. Refer to help file (type `?arima`) for different methods of estimation.

4. To compare models using AICC:

   ```
   AICc(fittedModel)
   ```

   The `qpcR` package needs to be downloaded and installed in `R` first before using this function. This function will give the Akaike's Information Criterion corrected for bias given a fitted model object from `arima()`. We want to select the model with the *lowest* AICC.

5. To predict future observations given a fitted ARMA model:

   ```
   predict(fittedModel, number of future observations to forecast)
   ```

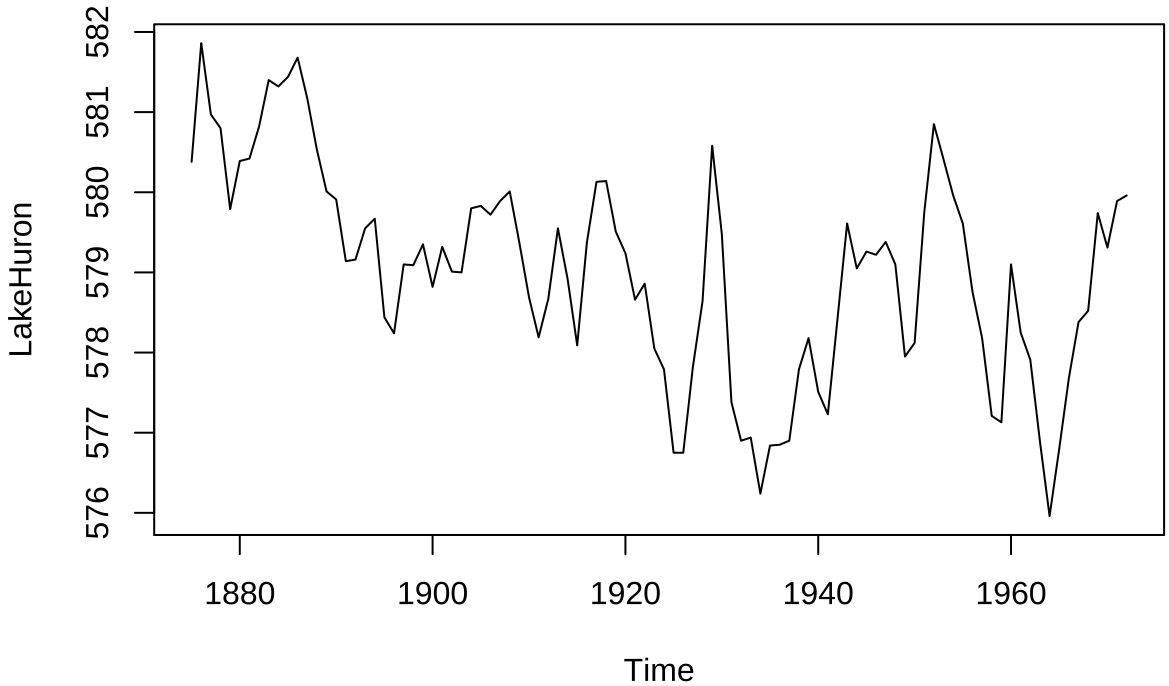
Lake Huron data

1. Analyze the Lake Huron data (preloaded data set in `R`). Load the data set using the command

   ```
   data(LakeHuron)
   ```

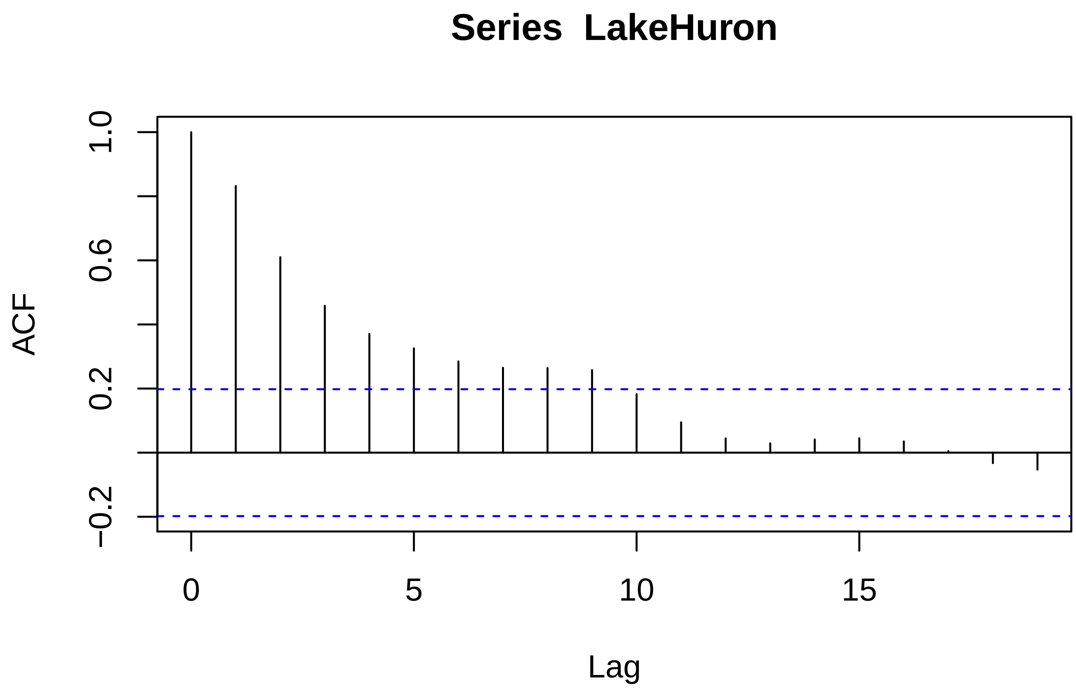
   and plot the data set using

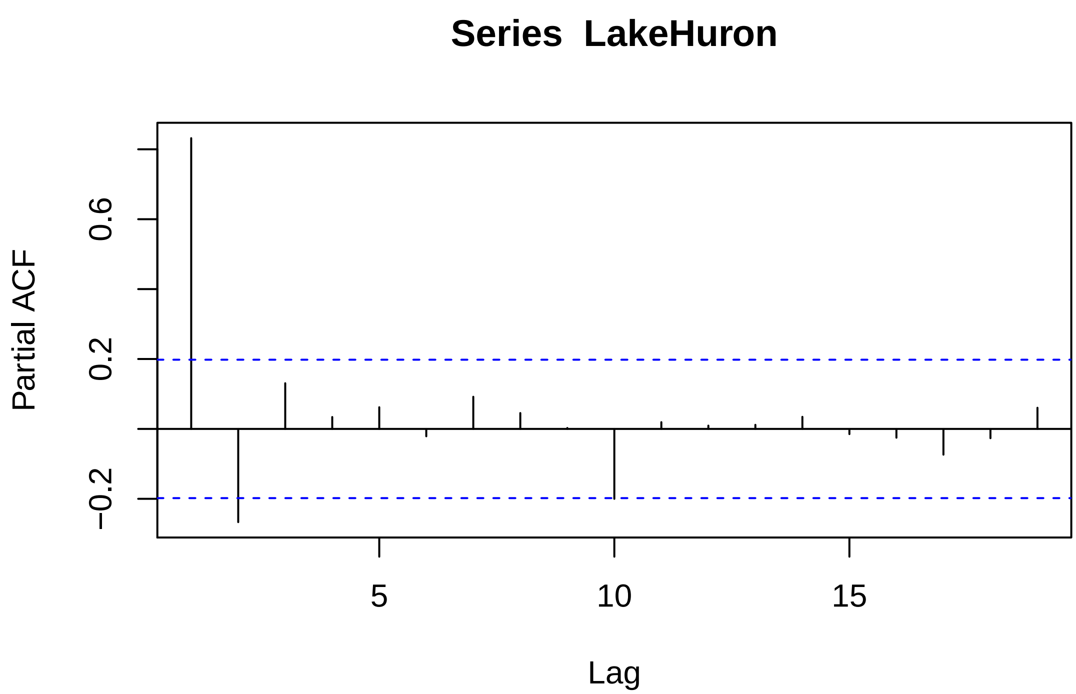
   ```
   ts.plot(LakeHuron)
   ```



2. Plot the ACF and PACF. What kind of model do they suggest?

```
op <- par(mfrow=c(2,1))
acf(LakeHuron)
pacf(LakeHuron)
par(op)
```





3. Consider some possible AR models that might fit the data and perform preliminary Yule-Walker estimation of the model parameters (Hint: use the function `ar()`).

```
# Preliminary estimation using Yule-Walker
ar(LakeHuron, method="yule-walker")
```

```
##
## Call:
## ar(x = LakeHuron, method = "yule-walker")
```

```
##
## Coefficients:
##       1        2
##  1.0538  -0.2668
##
## Order selected 2  sigma^2 estimated as  0.5075
```

4. Consider some possible MA models that might fit the data and perform preliminary estimation of model parameters using the innovations algorithm (see `lab6RCode.r` for using the innovations algorithm using the function `innovations.algorithm()` in `innovations.r` file).

```r
# Preliminary estimation using innovations algorithm
source("innovations.r")
?acf
acvf = acf(LakeHuron, plot=FALSE, lag.max = length(LakeHuron))$acf[,1,1] * var(LakeHuron)
m = length(acvf)
lh.ia = innovations.algorithm(m+1, acvf)
lh.ia$thetas[9,1:9] # Preliminary estimates of coefficients for MA(9)
```

```
## [1] 1.0810868 0.7691600 0.5218429 0.3315308 0.3085947 0.2479798 0.1932361
## [8] 0.1612645 0.2576989
```

```r
lh.ia$thetas[10,1:10] # Preliminary estimates of coefficients for MA(10)
```

```
##  [1] 1.0816255 0.7781248 0.5367164 0.3291559 0.3160040 0.2513755 0.2051537
##  [8] 0.1441070 0.3431868 0.1827401
```

```r
lh.ia$thetas[11,1:11] # Preliminary estimates of coefficients for MA(11)
```

```
##  [1] 1.08549783 0.77865745 0.54561580 0.34486948 0.31306321 0.26042628
##  [7] 0.20956724 0.16011877 0.31610604 0.33734448 0.09479822
```

5. Fit the different models under consideration using maximum likelihood estimation and compare the model fits using AICC (Hint: use `arima()` for estimation and `AICc()` in `R` library `qpcR` for model comparison - you will need to install this package into `R` first). Which model is preferred?

```r
?arima
# Fit models using maximum likelihood estimation
fit_ar2 = arima(LakeHuron, order = c(2,0,0), method = "ML")
fit_ma9 = arima(LakeHuron, order = c(0,0,9), method = "ML")
fit_ma10 = arima(LakeHuron, order = c(0,0,10), method = "ML")
fit_ma11 = arima(LakeHuron, order = c(0,0,11), method = "ML")
```

Compare models using AICC

```r
# install.packages("qpcR")
library(qpcR)
AICc(fit_ar2)
```

```
## [1] 215.5218
```

```r
AICc(fit_ma9)
```

```
## [1] 228.0829
```

```r
AICc(fit_ma10)
```

```
## [1] 228.9409
```

```r
AICc(fit_ma11)
```

```
## [1] 230.2403
```

Dow Jones Index

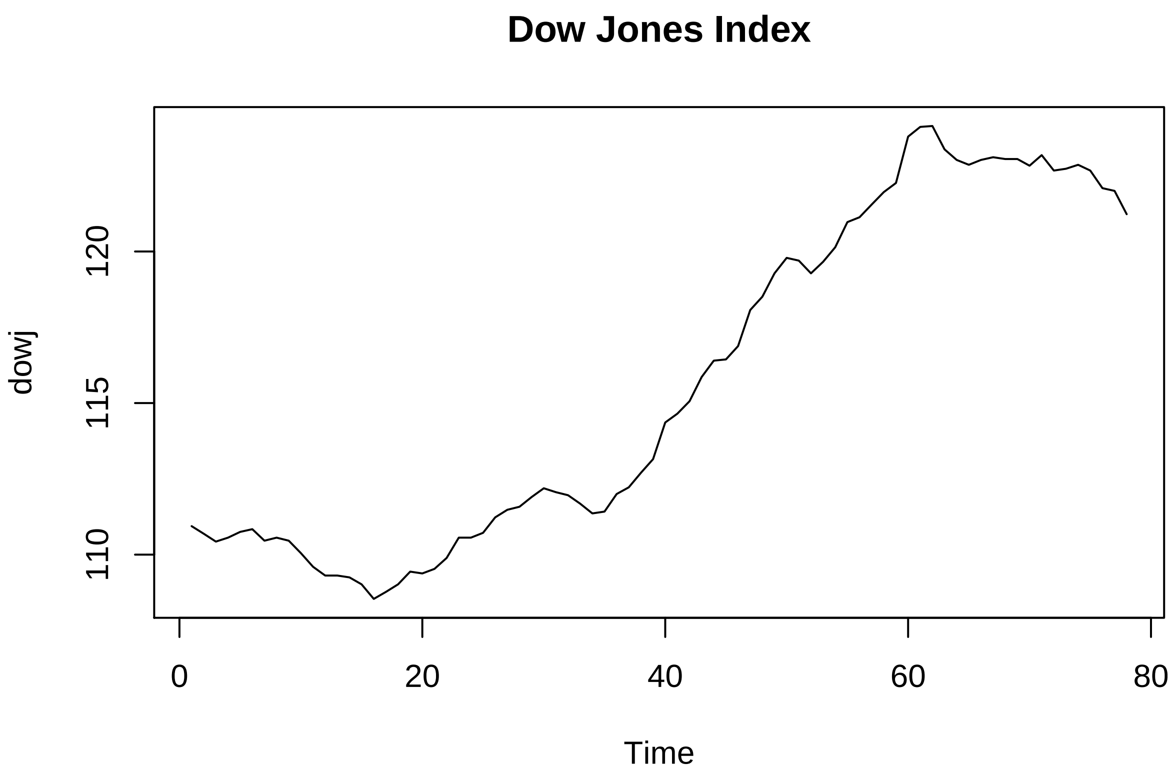
Exploratory Data Analysis

1. Analyze the Dow Jones Index data by downloading `dowj.txt` from Gauchospace. Move the file into `R`'s working directory and load the data set into R using the command `scan("dowj.txt")`.

```
# Load data
dowj_data <- scan("dowj.txt")
```

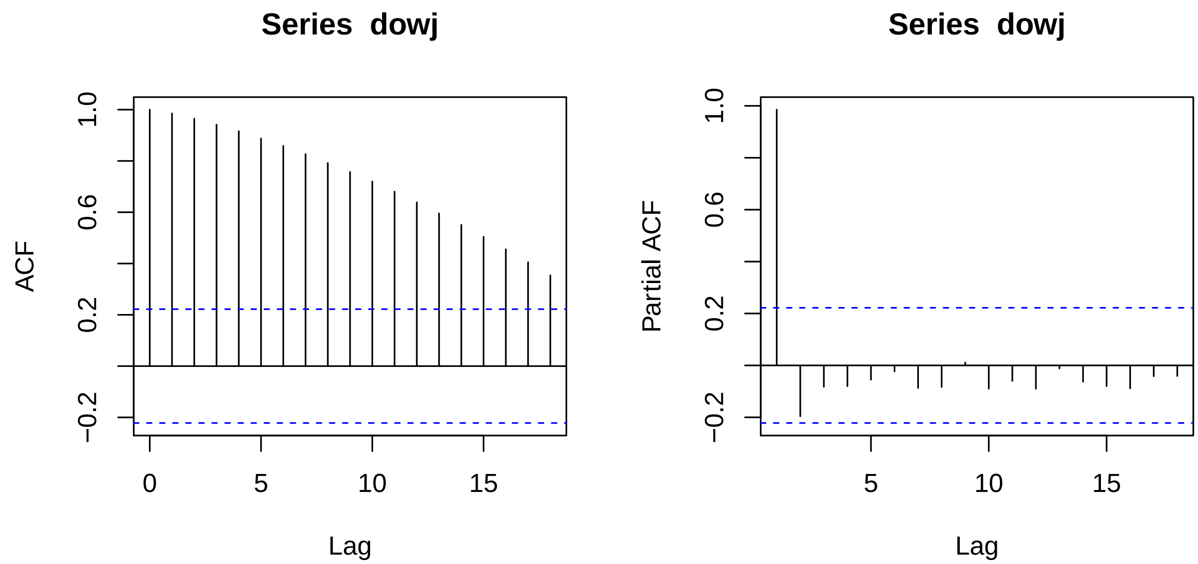
2. Plot the time series. What do you notice?

```
dowj <- ts(dowj_data)
# Plot data
ts.plot(dowj,main = "Dow Jones Index")
```



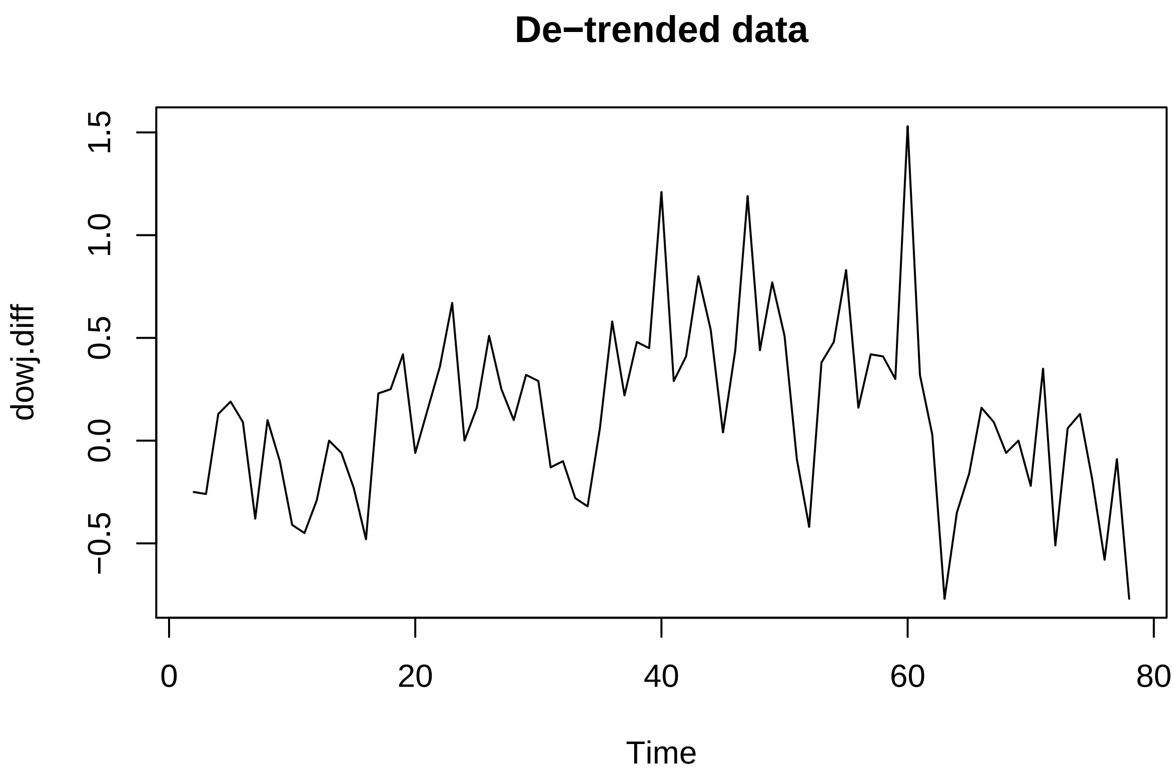
3. Make the data stationary. What procedures were used?

```
op <- par(mfrow=c(1,2))
acf(dowj)
pacf(dowj)
```




```r
par(op)
```

```r
dowj.diff <- diff(dowj,1)
ts.plot(dowj.diff, main = "De-trended data")
```

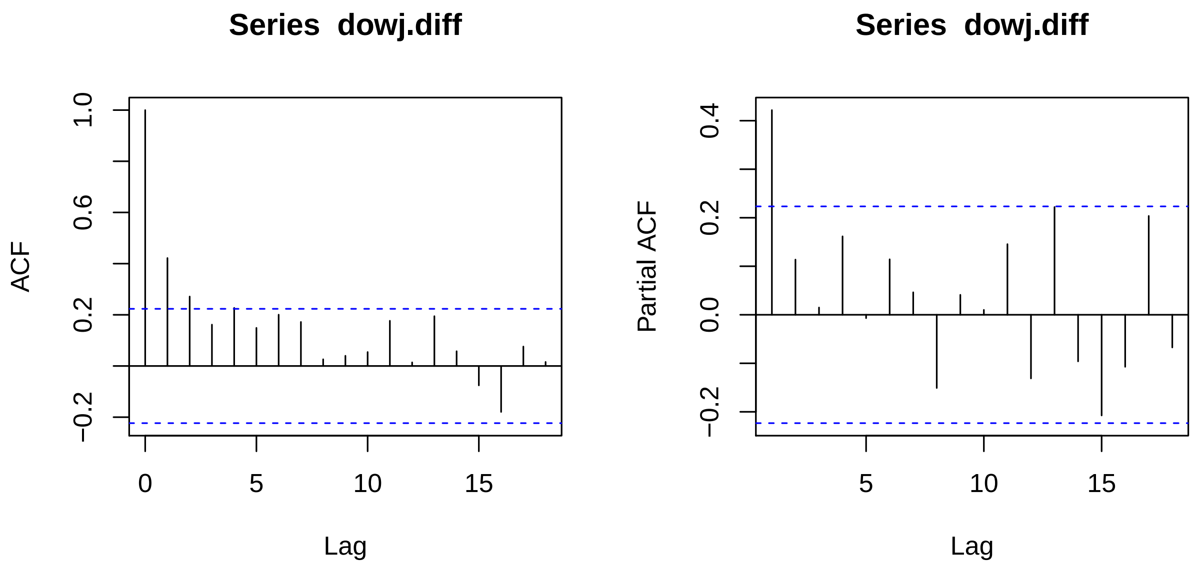


Model Identification

4. Plot the ACF and PACF. What models do they suggest?

```r
op <- par(mfrow=c(1,2))
acf(dowj.diff)
```

```r
pacf(dowj.diff)
```




```r
par(op)
```

Model Estimation

5. Fit the AR model suggested by the sample PACF and estimate the coefficients using Yule-Walker estimation.

```r
(fit.ar <- ar(dowj.diff, method="yw"))
```

```
##
## Call:
## ar(x = dowj.diff, method = "yw")
##
## Coefficients:
##      1
## 0.4219
##
## Order selected 1  sigma^2 estimated as  0.1518
```

6. Construct 95% confidence intervals for the estimated AR coefficients (Hint: obtain the asymptotic variance of the estimated coefficient from the fitted `ar()` object using `fittedObject$asy.var.coef`).

```r
# 95% CI for phi1
ar1.se <- sqrt(fit.ar$asy.var.coef)
c(fit.ar$ar - 1.96*ar1.se, fit.ar$ar + 1.96*1.96*ar1.se)
```

```
## [1] 0.2166839 0.8240603
```

7. Fit different ARMA models using maximum likelihood estimation and compare the model fits using AICC (Hint: use `arima()` for estimation and `AICc()` in `library(qpcR)` for model comparison - you will need to install this package into `R` first). Which model is preferred?

```r
library(qpcR)
# Calculate AICc for ARMA models with p and q running from 0 to 5
aiccs <- matrix(NA, nr = 6, nc = 6)
dimnames(aiccs) = list(p=0:5, q=0:5)
```

```r
for(p in 0:5)
{
  for(q in 0:5)
  {
    aiccs[p+1,q+1] = AICc(arima(dowj.diff, order = c(p,0,q), method="ML"))
  }
}
```

```
## Warning in log(s2): NaNs produced
```

```r
aiccs
```

```
##    q
## p          0        1        2        3        4        5
##   0 90.49584 81.14893 78.99601 81.17501 80.36057 82.70169
##   1 76.45411 77.04523 78.34399 80.21849 81.48857 83.13492
##   2 77.55328 79.49979 81.56072 81.16088 82.77818 85.09205
##   3 79.69841 81.76437 80.79646 82.25071 84.61907 76.69106
##   4 79.62134 81.83544 83.86231 84.60000 76.04349 86.21979
##   5 81.95313 84.24831 86.18115 87.16679 89.81553 88.75433
```

```r
(aiccs==min(aiccs))
```

```
##    q
## p       0     1     2     3     4     5
##   0 FALSE FALSE FALSE FALSE FALSE FALSE
##   1 FALSE FALSE FALSE FALSE FALSE FALSE
##   2 FALSE FALSE FALSE FALSE FALSE FALSE
##   3 FALSE FALSE FALSE FALSE FALSE FALSE
##   4 FALSE FALSE FALSE FALSE  TRUE FALSE
##   5 FALSE FALSE FALSE FALSE FALSE FALSE
```

Model Diagnostics

8. Perform diagnostics on the chosen model fit. Do the residuals appear to be white noise? Are they normally distributed?

```r
# Pick AR(1) and perform residual analysis:
fit = arima(dowj, order=c(1,1,0), method="ML")

# Test for independence of residuals
Box.test(residuals(fit), type="Ljung")
```

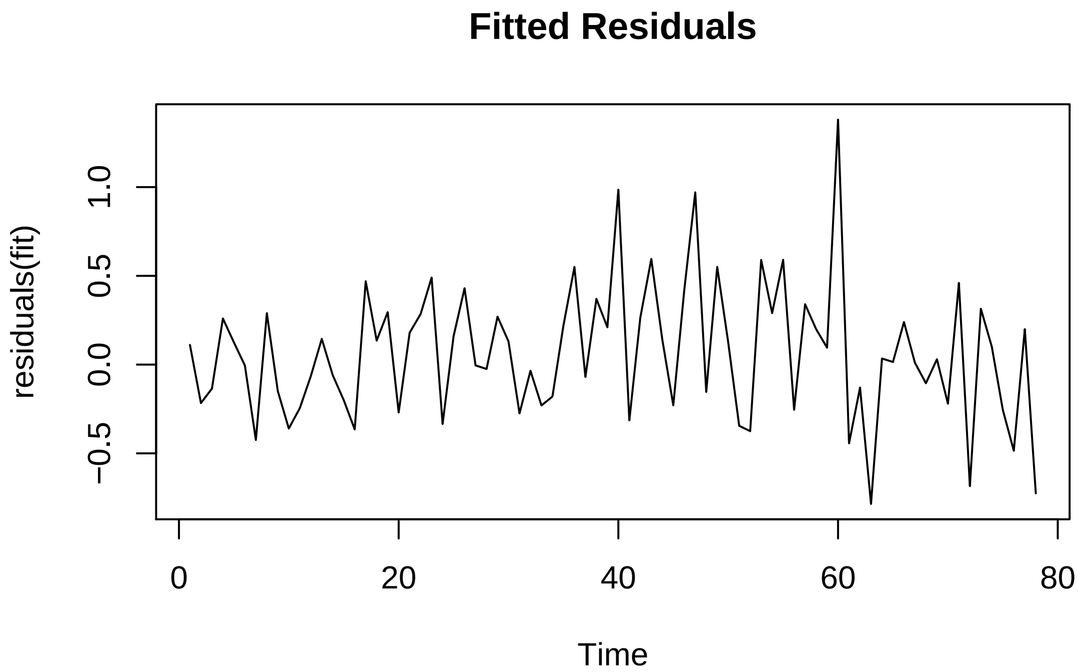
```
##
##  Box-Ljung test
##
## data:  residuals(fit)
## X-squared = 0.87778, df = 1, p-value = 0.3488
```

```r
# Test for normality of residuals
shapiro.test(residuals(fit))
```

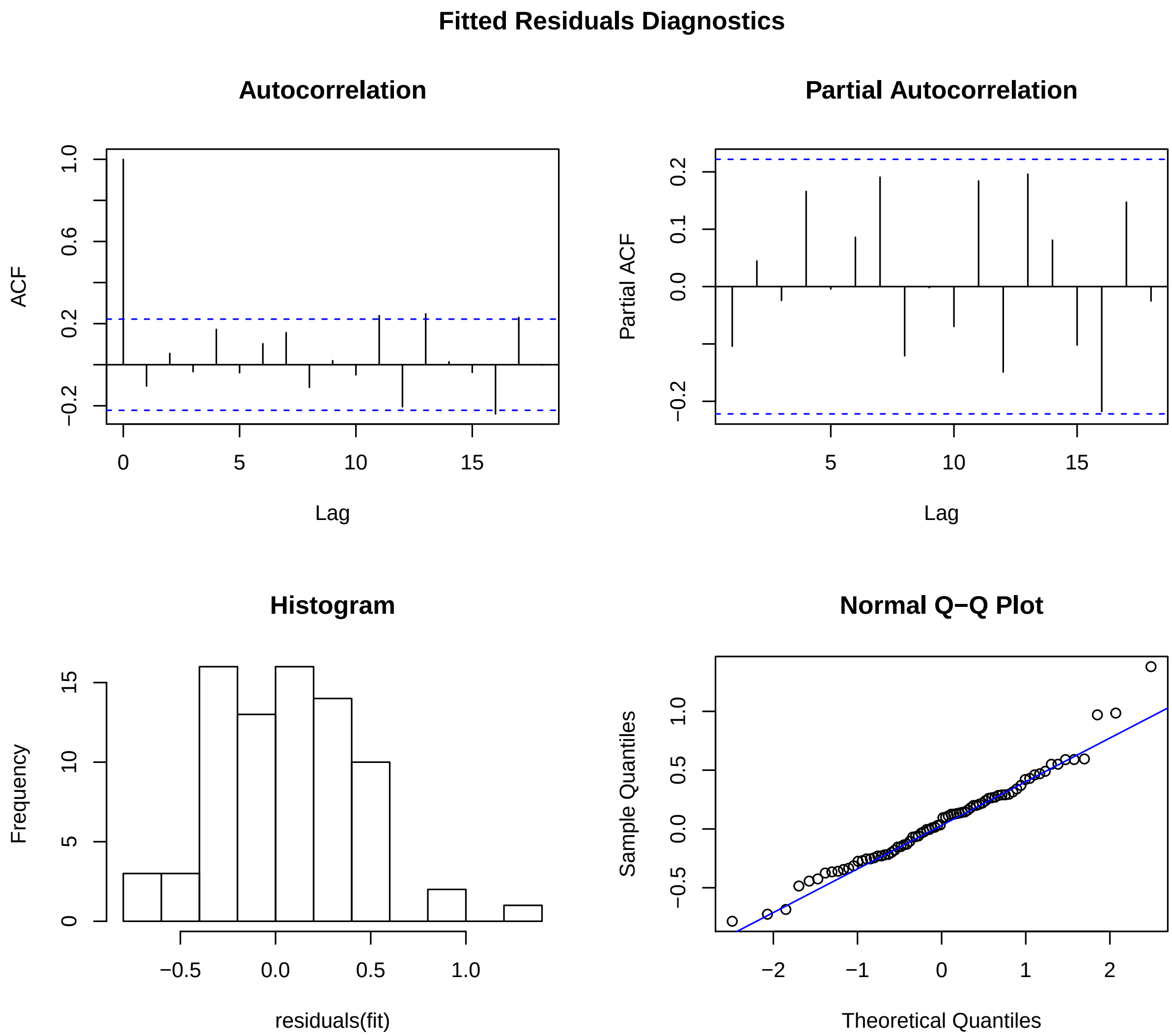
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit)
```

```
## W = 0.97422, p-value = 0.1144
```

```r
ts.plot(residuals(fit),main = "Fitted Residuals")
```




```r
par(mfrow=c(1,2),oma=c(0,0,2,0))
# Plot diagnostics of residuals
op <- par(mfrow=c(2,2))
# acf
acf(residuals(fit),main = "Autocorrelation")
# pacf
pacf(residuals(fit),main = "Partial Autocorrelation")
# Histogram
hist(residuals(fit),main = "Histogram")
# q-q plot
qqnorm(residuals(fit))
qqline(residuals(fit),col ="blue")
# Add overall title
title("Fitted Residuals Diagnostics", outer=TRUE)
```

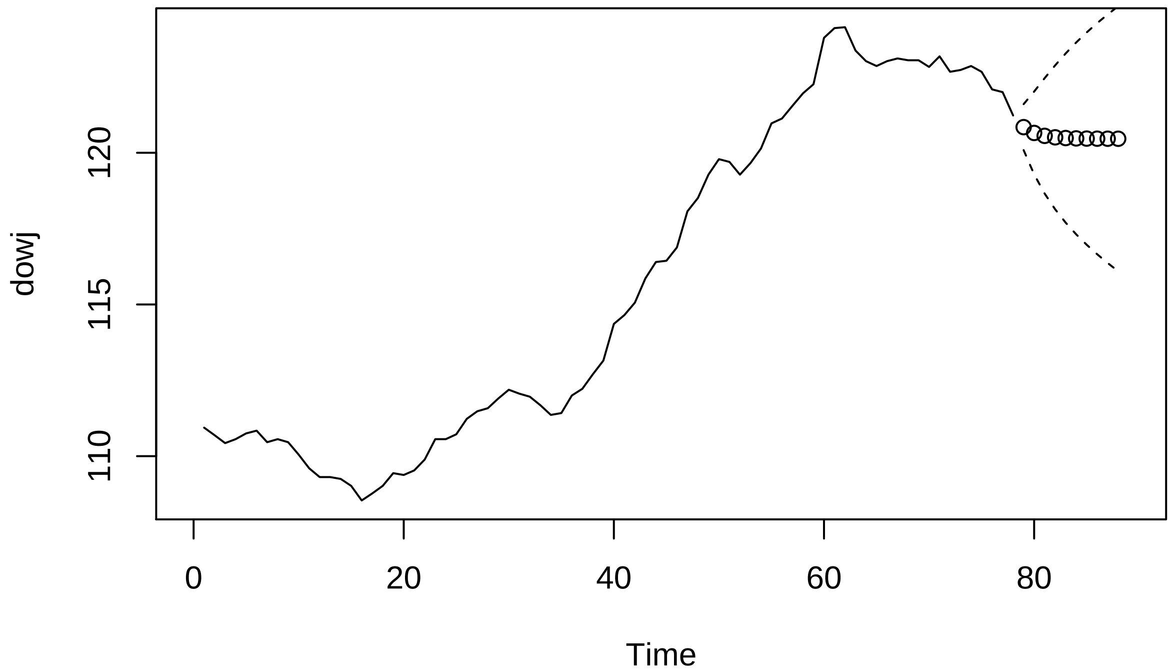



```
par(op)
```

Data Forecasting

9. Forecast the next 10 observations using your model.

```
# Predict 10 future observations and plot
mypred <- predict(fit, n.ahead=10)
ts.plot(dowj, xlim=c(0,89))
points(79:88,mypred$pred)
lines(79:88,mypred$pred+1.96*mypred$se,lty=2)
lines(79:88,mypred$pred-1.96*mypred$se,lty=2)
```


dowj
110
115
120
0
20
40
60
80
Time