Lab 6-Model Selection, Forecasting and Review

Pstat 174/274

Summary of R Commands

- To estimate parameters of an AR model:

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

- To estimate parameters of an MA or ARMA model:

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

- To compare models using AICC:

```
AICc(fittedModel)
```

- To difference a time series at lag d:

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

- To predict future observations given a fitted ARMA model:

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

Model Identification, Estimation and Diagnostics

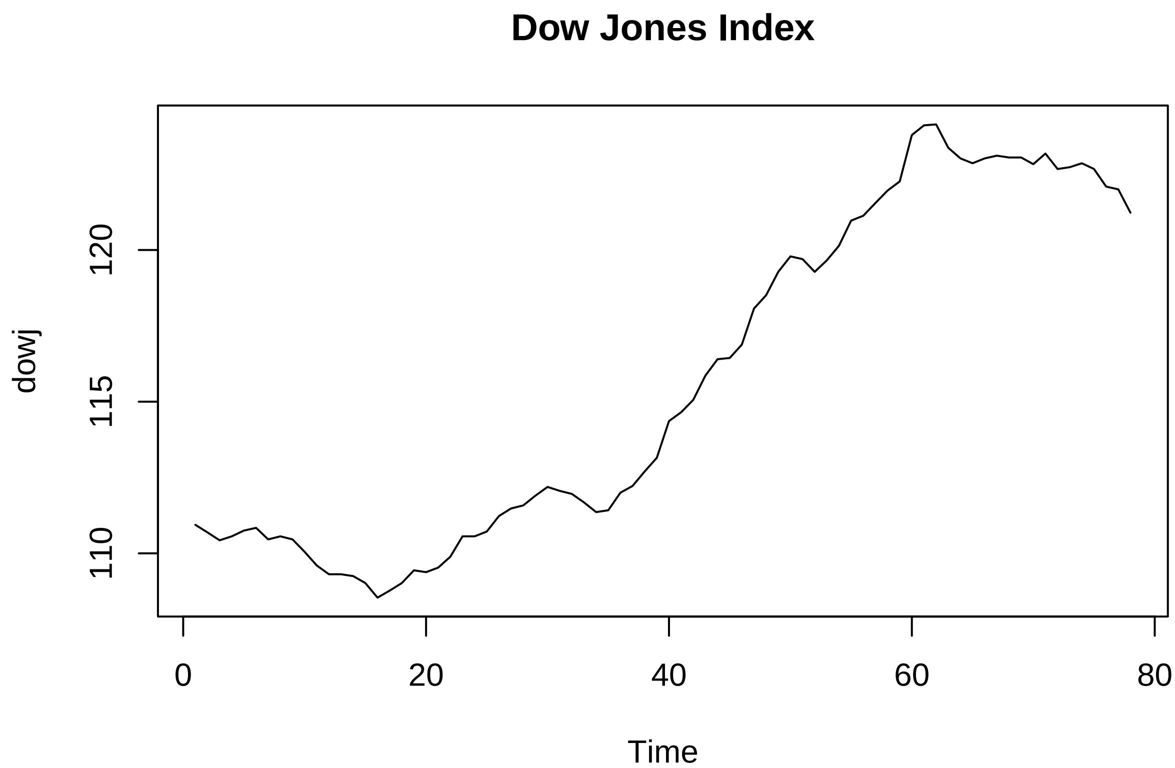
Exploratory Data Analysis

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")
```

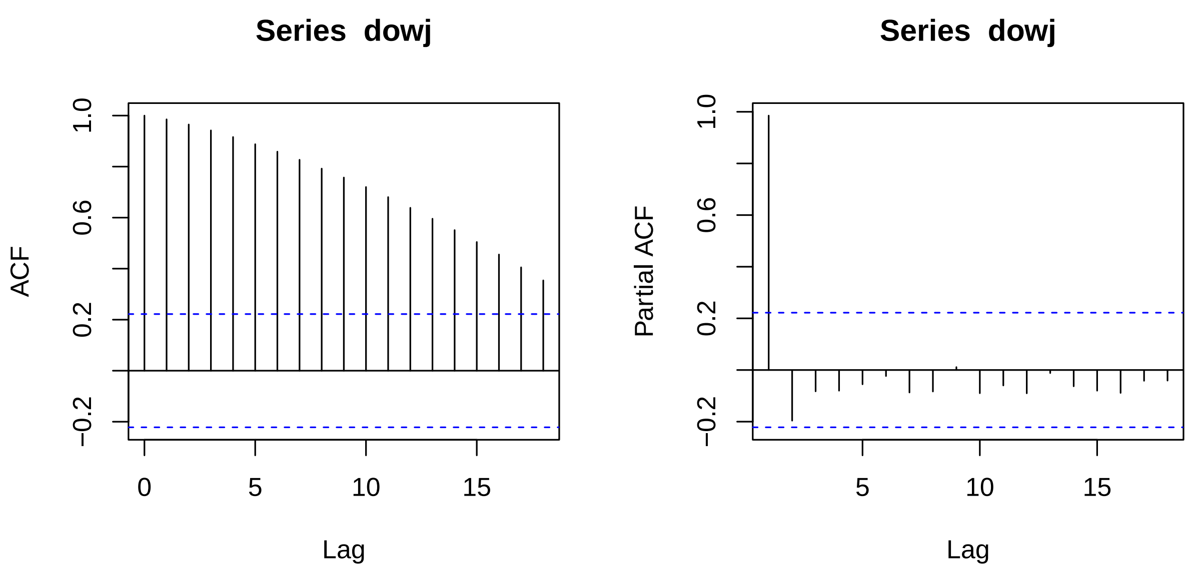
2. Plot the time series. What do you notice?

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

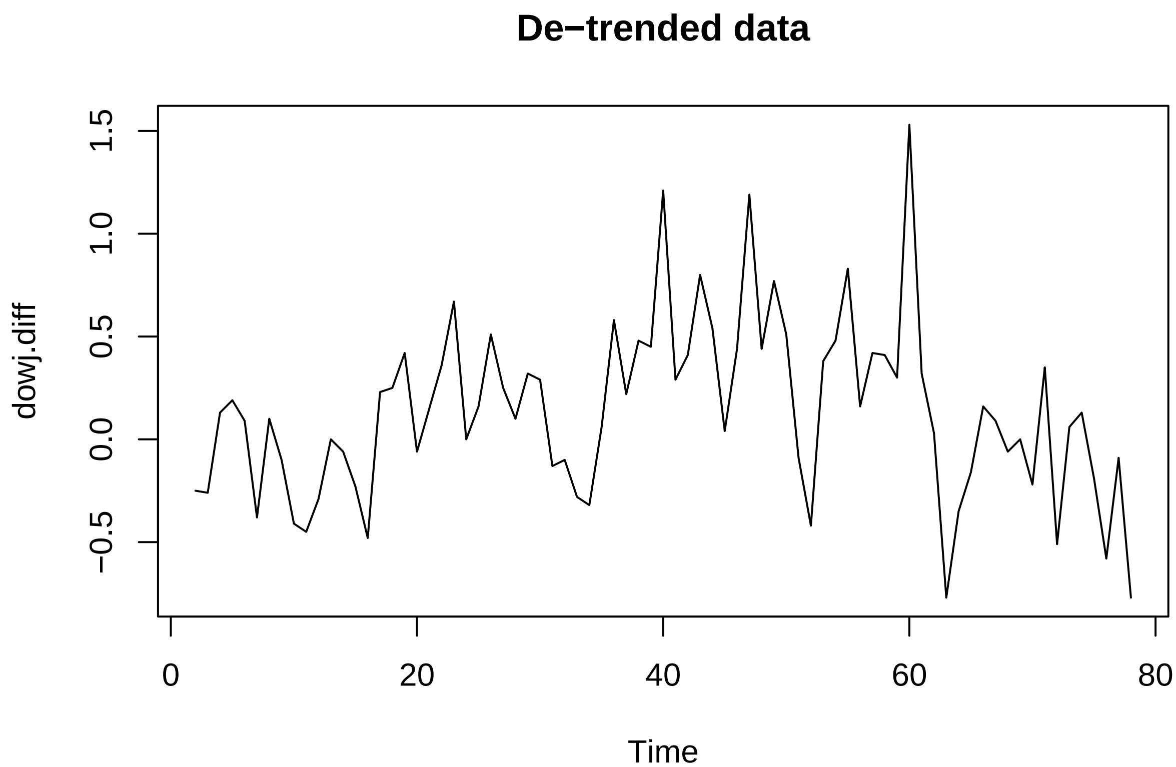


3. Make the data stationary. What procedures were used?

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



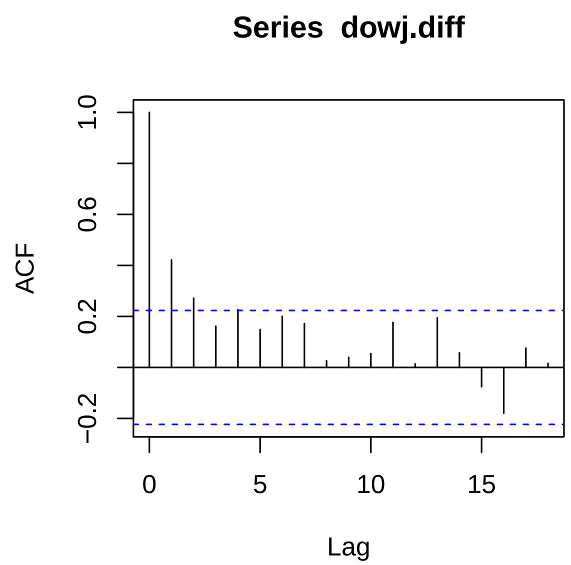
```r
dowj.diff <- diff(dowj,1)
ts.plot(dowj.diff, main = "De-trended data")
```

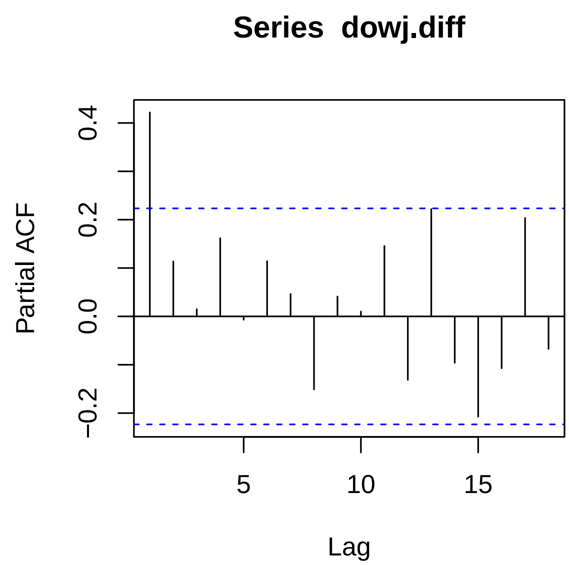


Model Identification

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

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





Model Estimation

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

```
(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`).

```
# 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?

```
library(qpcR)
```

```
## Warning: package 'qpcR' was built under R version 4.3.3
```

```
## Warning: package 'minpack.lm' was built under R version 4.3.3
```

```
## Warning: package 'rgl' was built under R version 4.3.3
```

```
## Warning: package 'robustbase' was built under R version 4.3.3
## Warning: package 'Matrix' was built under R version 4.3.2
```

```r
# 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)
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?

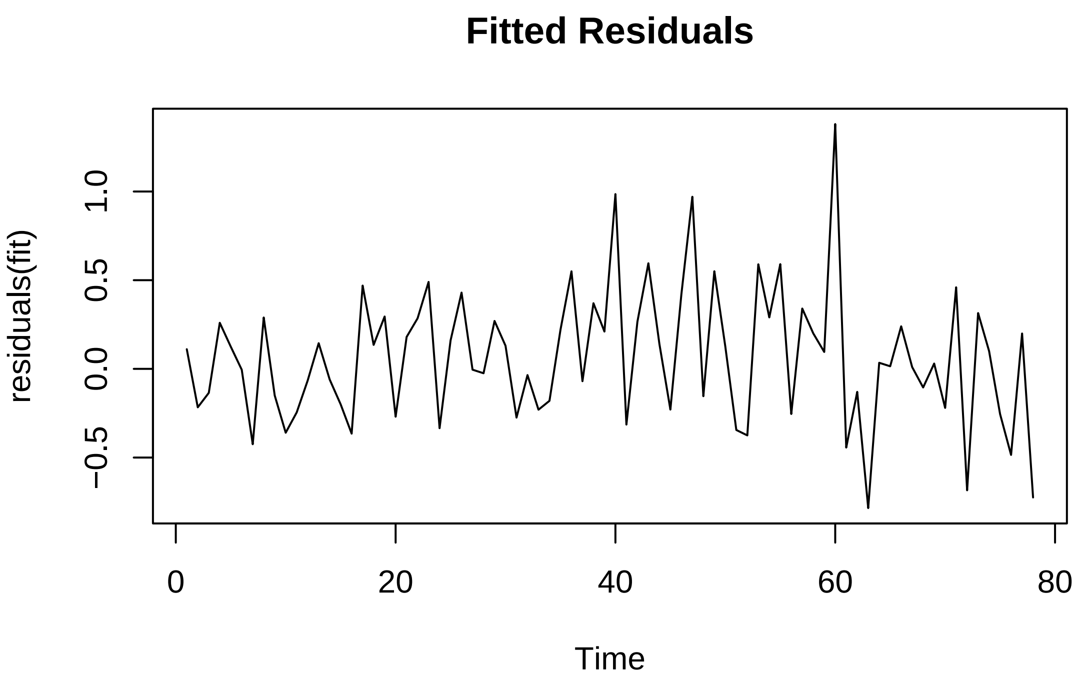
```r
# Pick AR(1) and perform residual analysis:
fit = arima(dowj, order=c(1,1,0), method="ML")

# 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
```

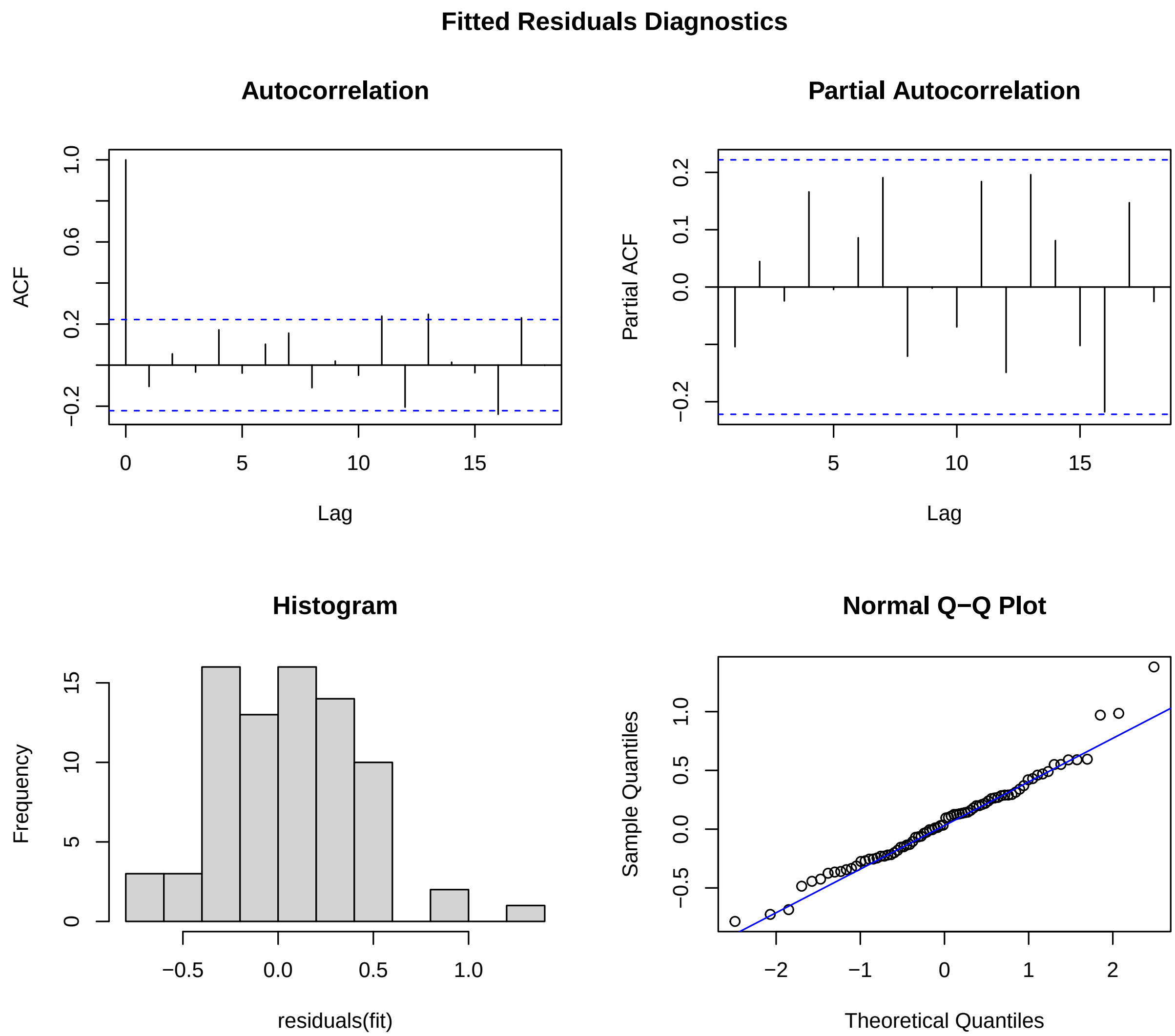
```r
# Test for normality of residuals
shapiro.test(residuals(fit))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit)
## W = 0.97422, p-value = 0.1144
```

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



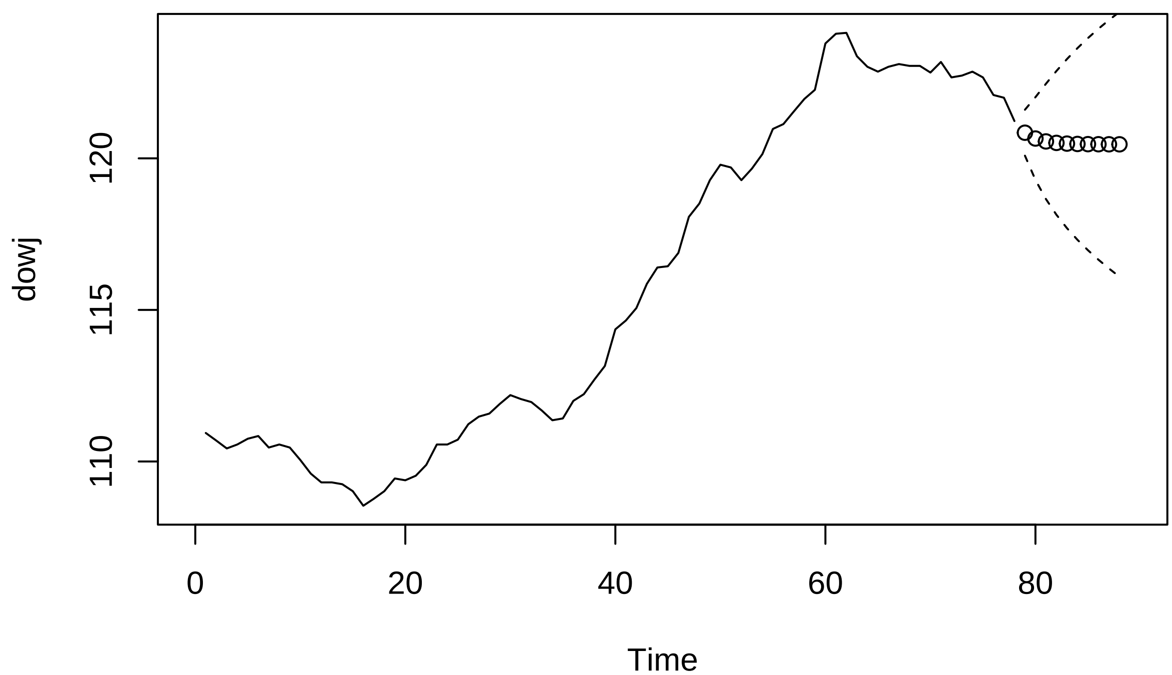

```
par(mfrow=c(1,2),oma=c(0,0,2,0))
# Plot diagnostics of residuals
op <- par(mfrow=c(2,2))
# acf
acf(residuals(fit),main = "Autocorrelation")
# pacf
pacf(residuals(fit),main = "Partial Autocorrelation")
# Histogram
hist(residuals(fit),main = "Histogram")
# q-q plot
qqnorm(residuals(fit))
qqline(residuals(fit),col ="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