Time Series Analysis

ARMA Models: Data Examples

Nicoleta Serban, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

IBM Stock Price: ARMA Modeling

Georgia Tech

About This Lesson

IBM Stock Price Prediction

International Business Machines (IBM)*:* Four companies came to form the Computing-Tabulating-Recording Company in 1911 which later became IBM in 1933. *What are important events after 1960?*

- 1964: the first computer system family;
- 1974: UPC was developed
- 1981: financial swaps (in collaboration with Word Bank)
- 1993: US$8 billion loss
- 2005: selling personal computing business
- 2014: selling x86 server division
- 2015: acquiring Merge Healthcare & Weather Company
- 2016: acquiring Truven

IBM Stock Price

Stock Price:

- Perceived company’s worth
- Multiplied by number of shares give the total company’s worth
- Affected by a number of things including volatility in the market, current economic conditions, and popularity of the company

Study Objective:

- Develop a model to predict IBM stock price given that no major events are to be released

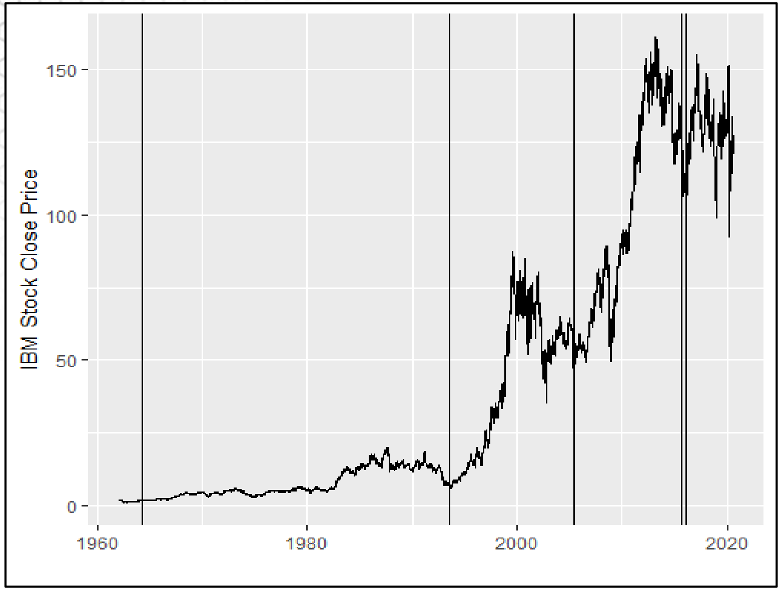
Time Series Data:

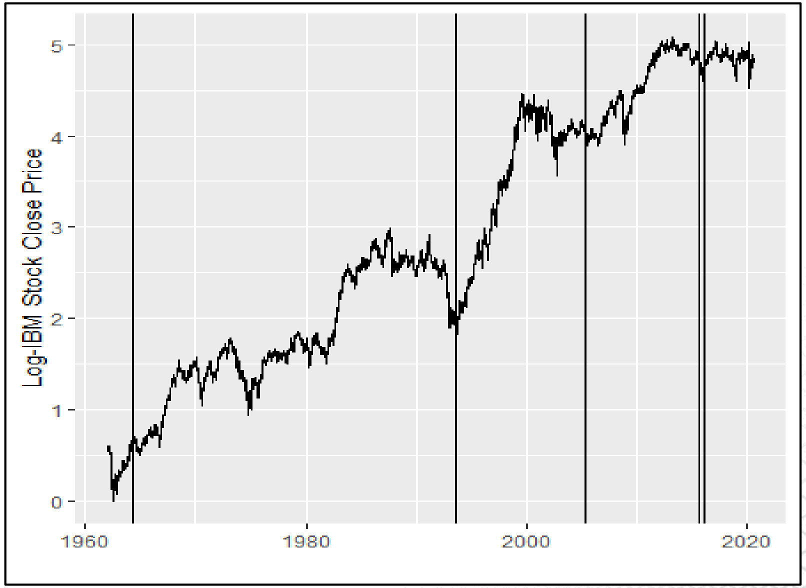
- Daily stock price from January 29th 1962 until August 26th 2020
- High, Low, Close, Adj.Close

Time Series Plots

```r
## Read Data
ibmdata = read.table("IBMstockprice.txt",header=T)
ibm.date = as.Date(as.character(ibmdata$Date), format="%Y-%m-%d")
ibmdata$Date = ibm.date
attach(ibmdata)
truven = which(Date=="2016-02-18")
library(ggplot2)
ggplot(ibmdata, aes(Date, Adj.Close)) + geom_line() + xlab("") + ylab("IBM Stock Close Price")+geom_vline(xintercept = as.numeric(Date[truven]))
## IBM Stock Price: Non-constant variance => Transform
AdjClose.tr = log(Adj.Close)
ggplot(ibmdata, aes(Date, AdjClose.tr)) + geom_line() + xlab("") + ylab("Log-IBM Stock Close Price") +geom_vline(xintercept = as.numeric(Date[truven]))
```

Time Series Plots

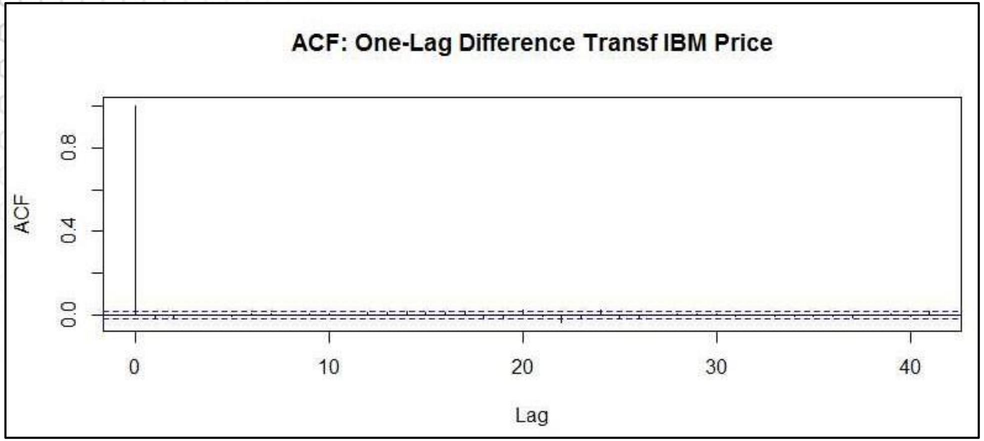


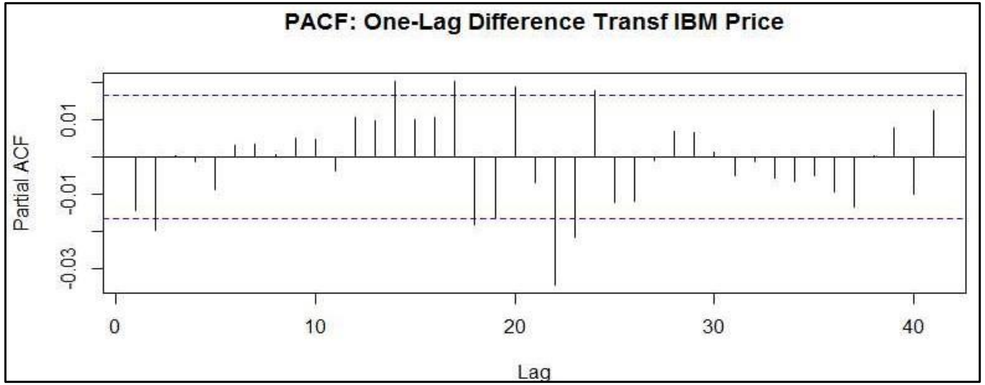


Assessing Dependence & Stationarity

```
ts.price = ts(AdjClose.tr,start=c(1962,1,29),frequency=365.25)
## Differencing to Remove Trend
diff.ts.price = diff(ts.price)
acf(diff.ts.price,main="ACF: One-Lag Difference Transf IBM Price")
pacf(diff.ts.price,main="PACF: One-Lag Difference Transf IBM Price")
```

Assessing Dependence & Stationarity



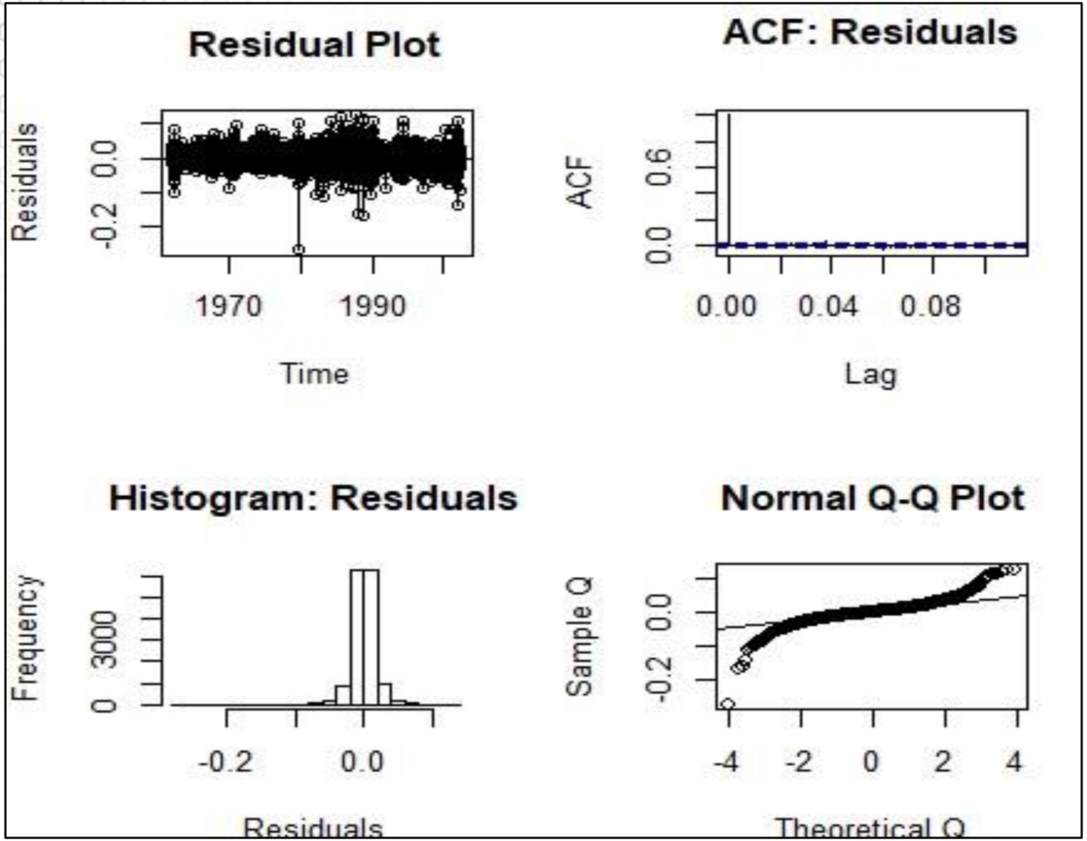


ARIMA Modeling

```
## Order selection – AIC
n = length(ts.price)
norder = 6
p = c(1:norder)-1; q = c(1:norder)-1
aic = matrix(0,norder,norder)
for(i in 1:norder){
   for(j in 1:norder){
    modij = arima(ts.price,order = c(p[i],1,q[j]), method='ML')
    aic[i,j] = modij$aic-2*(p[i]+q[j]+1)+2*(p[i]+q[j]+1)*n/(n-p[i]-q[j]-2)
   }
}
…
final_model = arima(ts.price, order = c(porder,1,qorder), method = "ML")
```

porder=0 & qorder=1

ARIMA Modeling: Residual Analysis



Testing for Uncorrelated Residuals

```
> Box.test(final_model$resid, lag = (porder+qorder+
1), type = "Box-Pierce", fitdf = (porder+qorder))
	Box-Pierce test

data:  final_model$resid
X-squared = 1.5201, df = 1, p-value = 0.2176

> Box.test(final_model$resid, lag = (porder+qorder+
1), type = "Ljung-Box", fitdf = (porder+qorder))
	Box-Ljung test

data:  final_model$resid
X-squared = 1.5206, df = 1, p-value = 0.2175
```

Summary