UN2102 Lab 8

Salim M'jahad

03/22/2018

Goals and Learning Objectives

The goal of this lab is to investigate the predictability of a few financial objects over time. We begin with an introduction to the **ramdom walk** process and the **autoregressive** model (please see the file **Lab 8 Background** for this information). In this lab, you will estimate the autoregressive statistical parameter ϕ for each year of the SP500 weekly closing prices. You will then perform the same task using closing prices from the Dow Jones Industrial Average. To accomplish these tasks, utilize the **Split/Apply/Combine** strategy using functions from the **plyr** or **apply** family.

The Autoregressive Model and Estimation

The famous Autoregressive Lag 1 Model has two statistical parameters. The first parameter is the autoregressive coefficient ϕ and the second is the drift μ. The autoregressive lag-1 process is given by the formula

$$Y_t = \mu + \phi(Y_{t-1} - \mu) + Z_t, \quad Z_t \overset{iid}{\sim} N(0, \sigma^2). \tag{1}$$

The above model can be expressed as

$$Y_t = \mu(1 - \phi) + \phi Y_{t-1} + Z_t, \quad Z_t \overset{iid}{\sim} N(0, \sigma^2).$$

To estimate the model parameters we can use common techniques, i.e., least squares.

$$Q(\mu, \phi) = \sum_{t=2}^{n} \big(Y_t - (\mu + \phi(Y_{t-1} - \mu))\big)^2. \tag{2}$$

The minimum value of Q is achieved at the least squares estimators $\hat{\mu}$ and $\hat{\phi}$.

Interpretation: The closer the estimated ϕ is to 1 gives an indication on how predictable the time series is. If ϕ is exactly 1, then the series is a random walk and hence, is not predictable. Due to random chance, estimated ϕ values will never equal 1. If $\hat{\phi}$ is very close to 1, then the time series is hard to predict.

One fun technique to easily estimate ϕ is to line the data set up as displayed below. Consider the following table for $n = 100$ cases.

Y_{t-1}	Y_t
x_1	x_2
x_2	x_3
x_3	x_4
x_4	x_5
x_5	x_6
$\vdots$	$\vdots$
x_{98}	x_{99}
x_{99}	x_{100}

The first column of the dataframe is the first $n-1$ cases and the second column are cases 2 through n. After constructing the two columns, you can use ordinary linear regression techniques to estimate ϕ, i.e., regress Y_t on the variable Y_{t-1}. The estimated slope $\hat{\beta}_1$ is thus the estimated AR(1) coefficient $\hat{\phi}$. Note that this new dataframe only has $n-1$ cases.

Tasks:

1) Write a function called **phi.hat** that estimates the autoregressive parameter ϕ. The input should be a data vector **Y** and the output should be the two estimated parameters $\hat{\mu}$ and $\hat{\phi}$. I recommend using the **lm()** function to complete this task.

```
# Gabriel will help students with this in class
setwd("/Users/salimmjahad/Desktop/STAT_COMP/lab8")
dji <- read.csv("DJIDaily.csv", as.is = TRUE)
head(dji)
```

```
##         Date    Open    High     Low   Close Adj.Close   Volume
## 1 1990-01-02 2748.72 2811.65 2732.51 2810.15   2810.15 20680000
## 2 1990-01-03 2814.20 2834.04 2786.26 2809.73   2809.73 23620000
## 3 1990-01-04 2804.39 2821.46 2766.42 2796.08   2796.08 24370000
## 4 1990-01-05 2786.90 2810.15 2758.11 2773.25   2773.25 20290000
## 5 1990-01-08 2761.73 2803.97 2753.41 2794.37   2794.37 16610000
## 6 1990-01-09 2792.88 2810.79 2760.03 2766.00   2766.00 15800000
```

```
library(ggplot2)
#ggplot(dji)+geom_line(aes(x=c(1:7111), y=Close))

phi.hat <- function(vec) {
  len <- length(vec)
  bound <- cbind(vec[1:len-1],vec[2:len])
  res <- coef(lm(bound[,1] ~ bound[,2]))
  return(res)
}

phi.hat(dji$Close)
```

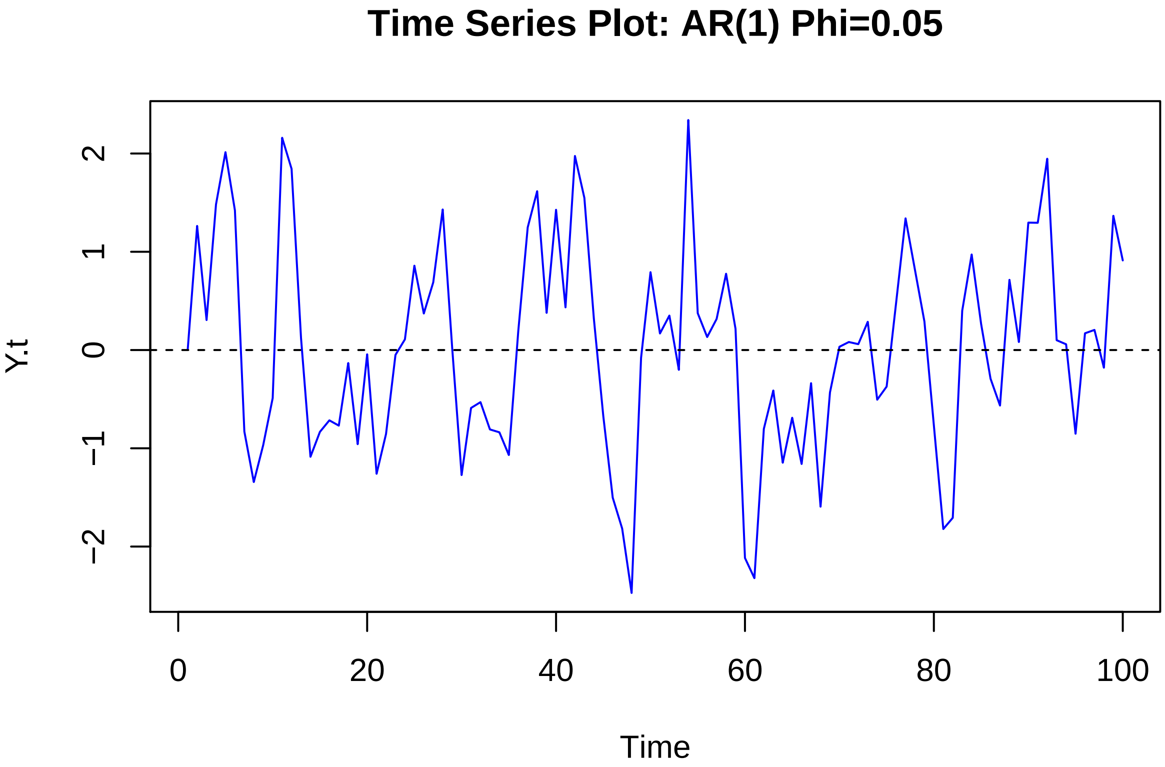
```
## (Intercept)  bound[, 2]
##   3.8294908   0.9993226
```

2) Test the **phi.hat** function on the following simulated AR(1) datasets **Y.t**.

AR(1) with phi=.5

```
set.seed(0)
Y.t <- NULL
Y.t[1] <- 0
phi <- .5
n <- 100
for (i in 2:n) {
  Y.t[i] <- phi*Y.t[i-1]+rnorm(1)

}
plot(1:100,Y.t,type="l",main="Time Series Plot: AR(1) Phi=0.05",col="blue",xlab="Time")
abline(h=0,lty=2)
```



```
# AR(1) estimated parameter
phi.hat(Y.t)
```

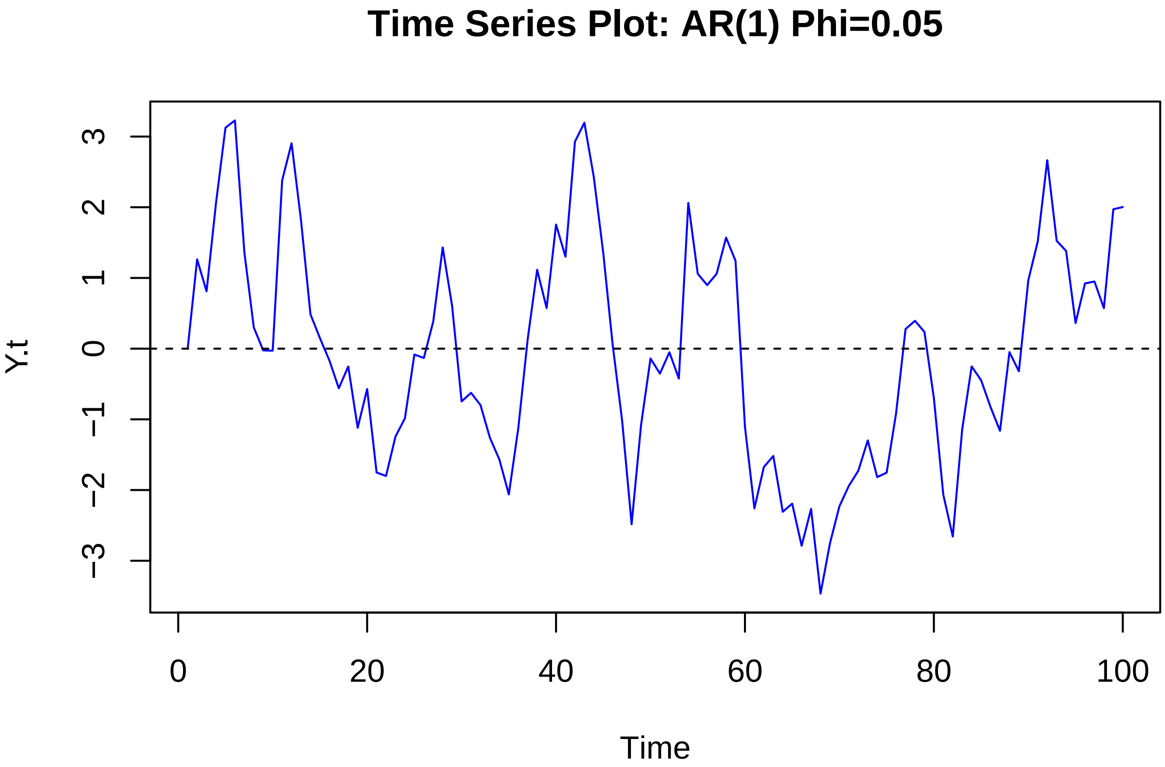
```
##  (Intercept)   bound[, 2]
## -0.001682094  0.542184108
```

AR(1) with phi=.90

```
set.seed(0)
Y.t <- NULL
Y.t[1] <- 0
phi <- .90
n <- 100
for (i in 2:n) {
  Y.t[i] <- phi*Y.t[i-1]+rnorm(1)
}

plot(1:100,Y.t,type="l",main="Time Series Plot: AR(1) Phi=0.05",col="blue",xlab="Time")
```

```
abline(h=0,lty=2)
```



```
# AR(1) estimated parameter
phi.hat(Y.t)
```

```
## (Intercept)  bound[, 2]
## -0.02991171  0.82015335
```

How well does the function **phi.hat** estimate the AR(1) parameter?

3) Test the **phi.hat** function on the following simulated random walk datasets **Y**.

Coin Flip Random Walk 1

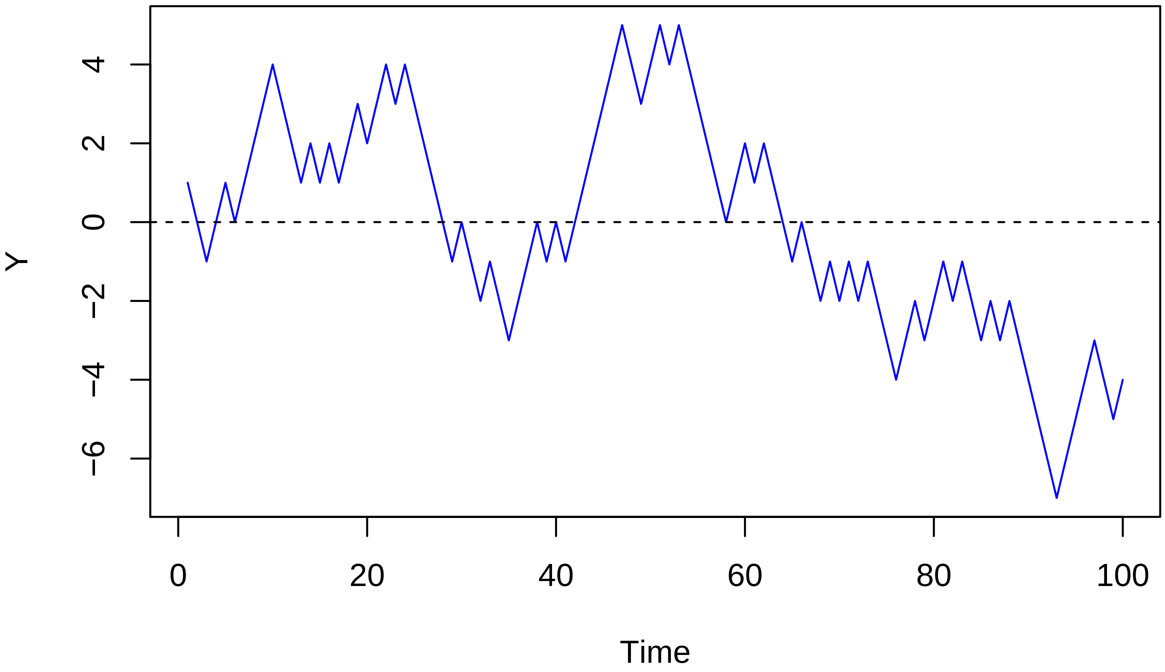
```
set.seed(0)
trials <- 100
coins <- ifelse(rbinom(n=trials,size=1,prob=.5)==1,1,-1)
coins
```

```
##   [1]  1 -1 -1  1  1 -1  1  1  1  1 -1 -1 -1  1 -1  1 -1  1  1 -1  1  1 -1
##  [24]  1 -1 -1 -1 -1 -1  1 -1 -1  1 -1 -1  1  1  1 -1  1 -1  1  1  1  1  1
##  [47]  1 -1 -1  1  1 -1  1 -1 -1 -1 -1 -1  1  1 -1  1 -1 -1 -1  1 -1 -1  1
##  [70] -1  1 -1  1 -1 -1 -1  1  1 -1  1  1 -1  1 -1 -1  1 -1  1 -1 -1 -1 -1
##  [93] -1  1  1  1  1 -1 -1  1
```

```
Y <- cumsum(coins)
Y
```

```
##   [1]  1  0 -1  0  1  0  1  2  3  4  3  2  1  2  1  2  1  2  3  2  3  4  3
##  [24]  4  3  2  1  0 -1  0 -1 -2 -1 -2 -3 -2 -1  0 -1  0 -1  0  1  2  3  4
##  [47]  5  4  3  4  5  4  5  4  3  2  1  0  1  2  1  2  1  0 -1  0 -1 -2 -1
##  [70] -2 -1 -2 -1 -2 -3 -4 -3 -2 -3 -2 -1 -2 -1 -2 -3 -2 -3 -2 -3 -4 -5 -6
##  [93] -7 -6 -5 -4 -3 -4 -5 -4
```

```r
plot(1:trials,Y,type="l",col="blue",xlab="Time")
abline(h=0,lty=2)
```



```r
# AR(1) estimated parameter
phi.hat(Y)
```

```
## (Intercept)  bound[, 2]
##  0.04216269  0.92491871
```

Coin Flip Random Walk 2

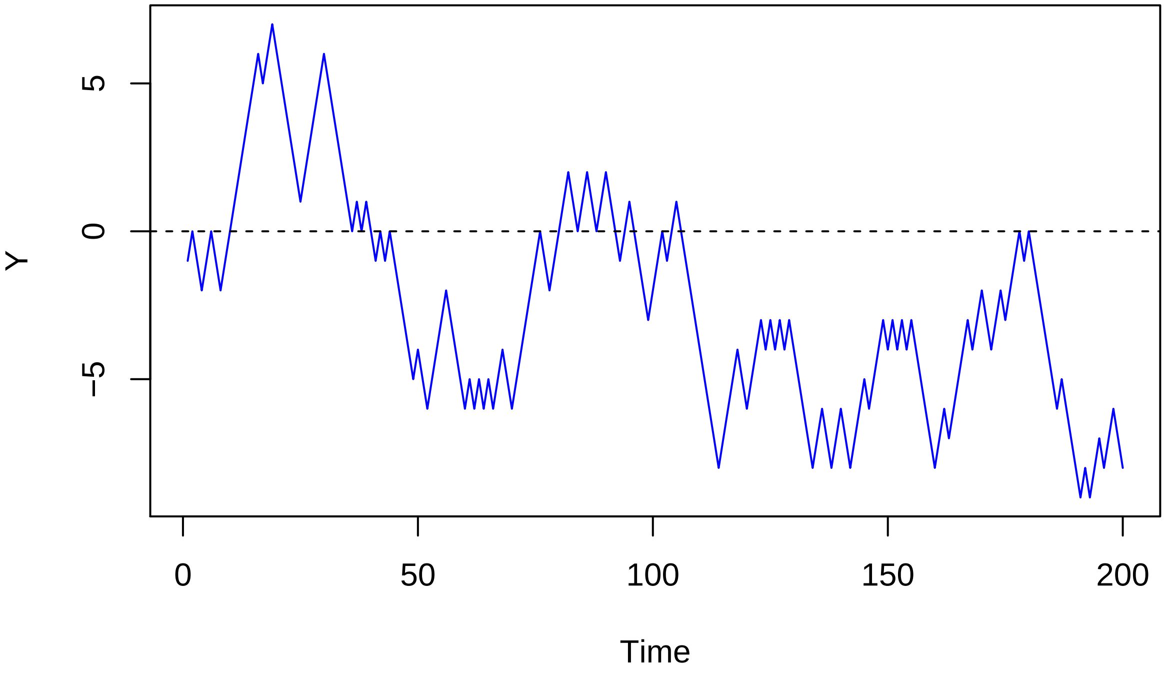
```r
set.seed(3)
trials <- 200
coins <- ifelse(rbinom(n=trials,size=1,prob=.5)==1,1,-1)
coins
```

```
##   [1] -1  1 -1 -1  1  1 -1 -1  1  1  1  1  1  1  1  1 -1  1  1 -1 -1 -1 -1
##  [24] -1 -1  1  1  1  1  1 -1 -1 -1 -1 -1 -1  1 -1  1 -1 -1  1 -1  1 -1 -1
##  [47] -1 -1 -1  1 -1 -1  1  1  1  1 -1 -1 -1 -1  1 -1  1 -1  1 -1  1  1 -1
##  [70] -1  1  1  1  1  1  1 -1 -1  1  1  1  1 -1 -1  1  1 -1 -1  1  1 -1 -1
##  [93] -1  1  1 -1 -1 -1 -1  1  1  1 -1  1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1
## [116]  1  1  1 -1 -1  1  1  1 -1  1 -1  1 -1  1 -1 -1 -1 -1 -1  1  1 -1 -1
## [139]  1  1 -1 -1  1  1  1 -1  1  1  1 -1  1 -1  1 -1  1 -1 -1 -1 -1 -1  1
## [162]  1 -1  1  1  1  1 -1  1  1 -1 -1  1  1 -1  1  1  1 -1  1 -1 -1 -1 -1
## [185] -1 -1  1 -1 -1 -1 -1  1 -1  1  1 -1  1  1 -1 -1
```

```r
Y <- cumsum(coins)
Y
```

```
##   [1] -1  0 -1 -2 -1  0 -1 -2 -1  0  1  2  3  4  5  6  5  6  7  6  5  4  3
##  [24]  2  1  2  3  4  5  6  5  4  3  2  1  0  1  0  1  0 -1  0 -1  0 -1 -2
##  [47] -3 -4 -5 -4 -5 -6 -5 -4 -3 -2 -3 -4 -5 -6 -5 -6 -5 -6 -5 -6 -5 -4 -5
##  [70] -6 -5 -4 -3 -2 -1  0 -1 -2 -1  0  1  2  1  0  1  2  1  0  1  2  1  0
##  [93] -1  0  1  0 -1 -2 -3 -2 -1  0 -1  0  1  0 -1 -2 -3 -4 -5 -6 -7 -8 -7
## [116] -6 -5 -4 -5 -6 -5 -4 -3 -4 -3 -4 -3 -4 -3 -4 -5 -6 -7 -8 -7 -6 -7 -8
## [139] -7 -6 -7 -8 -7 -6 -5 -6 -5 -4 -3 -4 -3 -4 -3 -4 -3 -4 -5 -6 -7 -8 -7
## [162] -6 -7 -6 -5 -4 -3 -4 -3 -2 -3 -4 -3 -2 -3 -2 -1  0 -1  0 -1 -2 -3 -4
## [185] -5 -6 -5 -6 -7 -8 -9 -8 -9 -8 -7 -8 -7 -6 -7 -8
```

```
plot(1:trials,Y,type="l",col="blue",xlab="Time")
abline(h=0,lty=2)
```



```
# AR(1) estimated parameter
phi.hat(Y)
```

```
## (Intercept)  bound[, 2]
## -0.07866302  0.95549324
```

Comment on the estimated AR(1) parameters, i.e., are they close to 1?

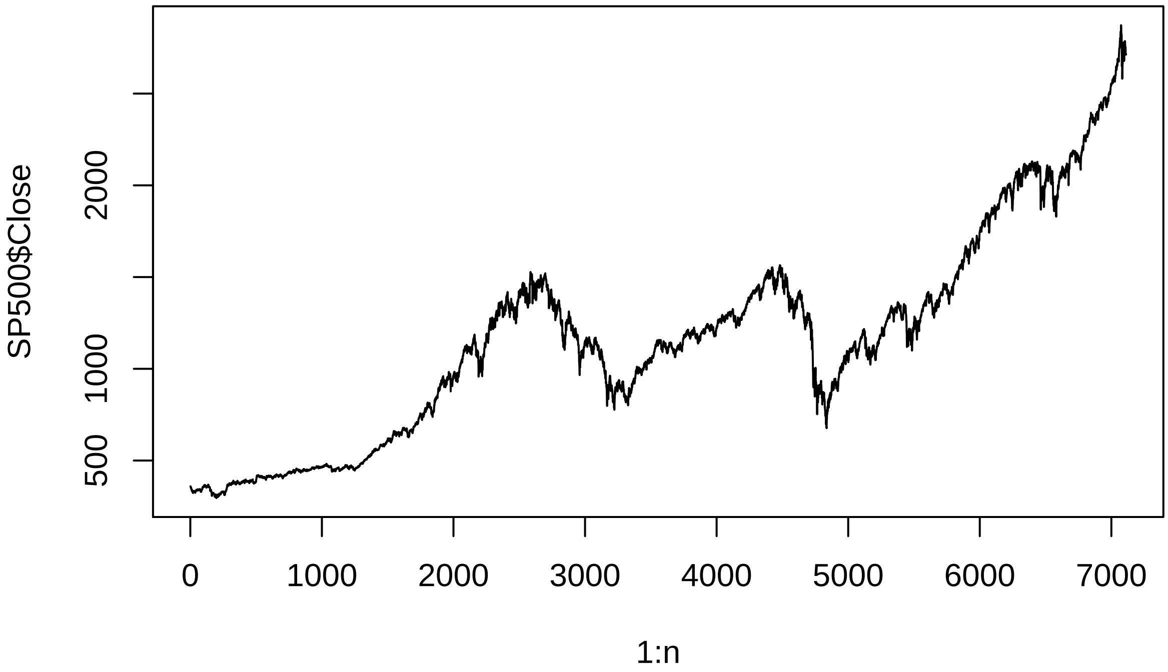
Both values are pretty close to 1. The first being 0.92491871 and the second 0.95549324 meaning the data is predictable.

4) Test the **phi.hat** function on the SP500 closing price using all 7111 time points.

```
SP500 <- read.csv("SP500Weekly.csv",as.is=T)
n <- nrow(SP500)
n
```

```
## [1] 7111
```

```
plot(1:n,SP500$Close,type="l")
```



```
phi.hat(SP500$Close)
```

```
## (Intercept)  bound[, 2]
##   0.4406806   0.9993393
```

Comment on the estimated AR(1) parameter. What does this say about the predictability of the time series?

The estimation is 0.9993393 meaning the time series is very predictable.

Split/Apply/Combine

5) Use the Split/Apply/Combine strategy to estimate the autoregressive parameter ϕ over the years 1990 through 2018 on the SP500 Closing prices. This will result in 29 values. intervals for each dataset. To summarize the results, plot the estimated parameters as a function of time. Also plot a horizontal line 1, which will give some insight on which years were more predictable. **Note** you might have to modify the function **phi.hat**.

```
# Solution
head(SP500)
```

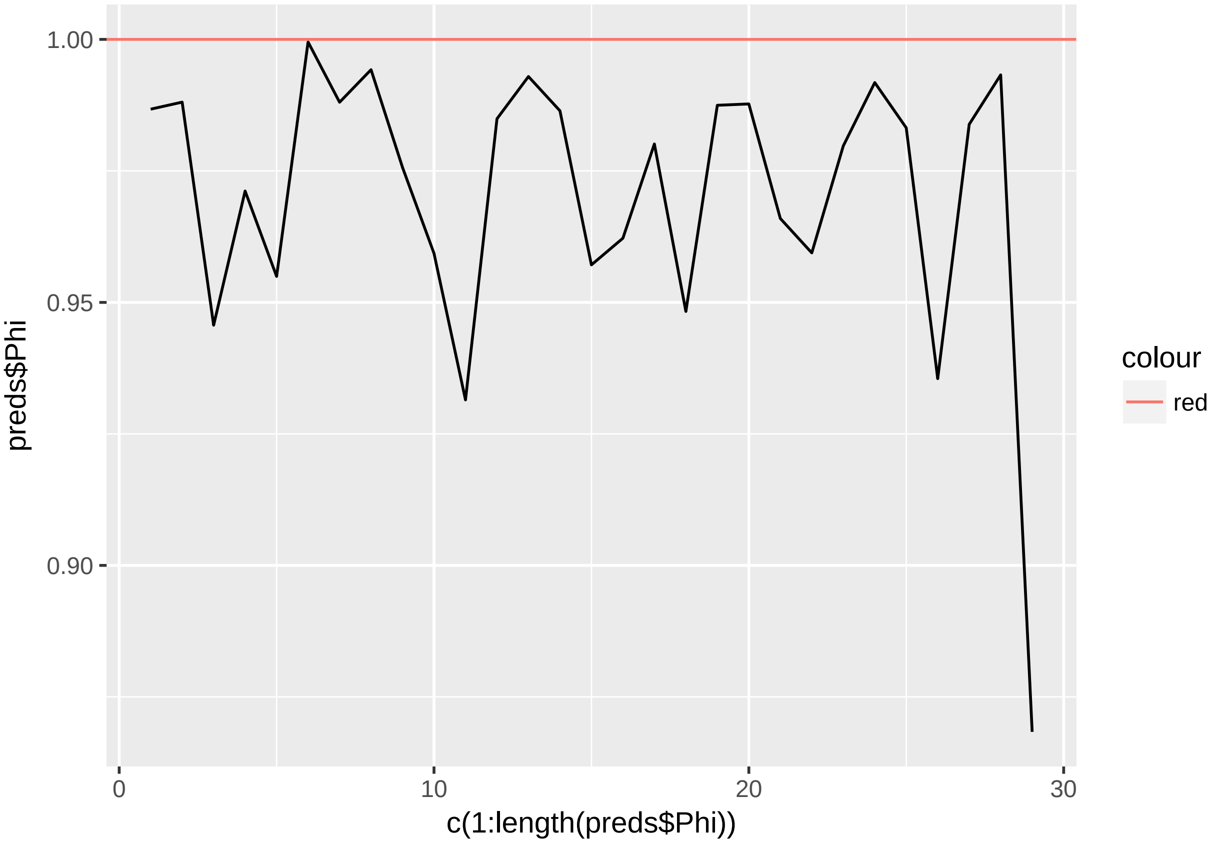
```
##         Date   Open   High    Low  Close Adj.Close    Volume
## 1 1990-01-02 353.40 359.69 351.98 359.69    359.69 162070000
## 2 1990-01-03 359.69 360.59 357.89 358.76    358.76 192330000
## 3 1990-01-04 358.76 358.76 352.89 355.67    355.67 177000000
## 4 1990-01-05 355.67 355.67 351.35 352.20    352.20 158530000
## 5 1990-01-08 352.20 354.24 350.54 353.79    353.79 140110000
## 6 1990-01-09 353.83 354.17 349.61 349.62    349.62 155210000
```

```
phi.hat <- function(dff) {
  len <- length(dff$Close)
  bound <- cbind(dff$Close[1:len-1],dff$Close[2:len])
  res <- coef(lm(bound[,1] ~ bound[,2]))
  return(res)
}
```

```r
SP500$Year <- matrix(unlist(strsplit(SP500$Date, "-")), nrow=nrow(SP500), byrow=T)[,1]
library(plyr)
preds <- ddply(SP500, .(Year), phi.hat)
colnames(preds) <- c("Year", "Intercept", "Phi")
preds
```

```
##    Year   Intercept       Phi
## 1  1990   4.5551348 0.9867332
## 2  1991   4.1289349 0.9880744
## 3  1992  22.5163407 0.9456651
## 4  1993  12.8959003 0.9711760
## 5  1994  20.7714911 0.9549369
## 6  1995  -0.3593164 0.9995103
## 7  1996   7.5406396 0.9880497
## 8  1997   4.1094835 0.9942381
## 9  1998  25.4307485 0.9756493
## 10 1999  53.0458044 0.9593240
## 11 2000  98.3433035 0.9314659
## 12 2001  18.5643060 0.9849081
## 13 2002   8.1076047 0.9929401
## 14 2003  12.3001822 0.9864224
## 15 2004  48.0606087 0.9571318
## 16 2005  45.4493231 0.9622005
## 17 2006  25.4565750 0.9801205
## 18 2007  76.2068210 0.9482791
## 19 2008  17.4286272 0.9874746
## 20 2009  10.9043980 0.9877286
## 21 2010  38.3163769 0.9659533
## 22 2011  51.5089470 0.9594105
## 23 2012  27.3478514 0.9797453
## 24 2013  11.9577094 0.9917938
## 25 2014  31.5806034 0.9831840
## 26 2015 132.9879686 0.9355041
## 27 2016  32.9320253 0.9838504
## 28 2017  14.8607360 0.9932551
## 29 2018 361.2247090 0.8683573
```

```r
plt <- ggplot()+geom_line(aes(x=c(1:length(preds$Phi)),y=preds$Phi))
plt + geom_hline(aes(yintercept=1, color="red"))
```



x=1 corresponds to Year = 1990 and so one

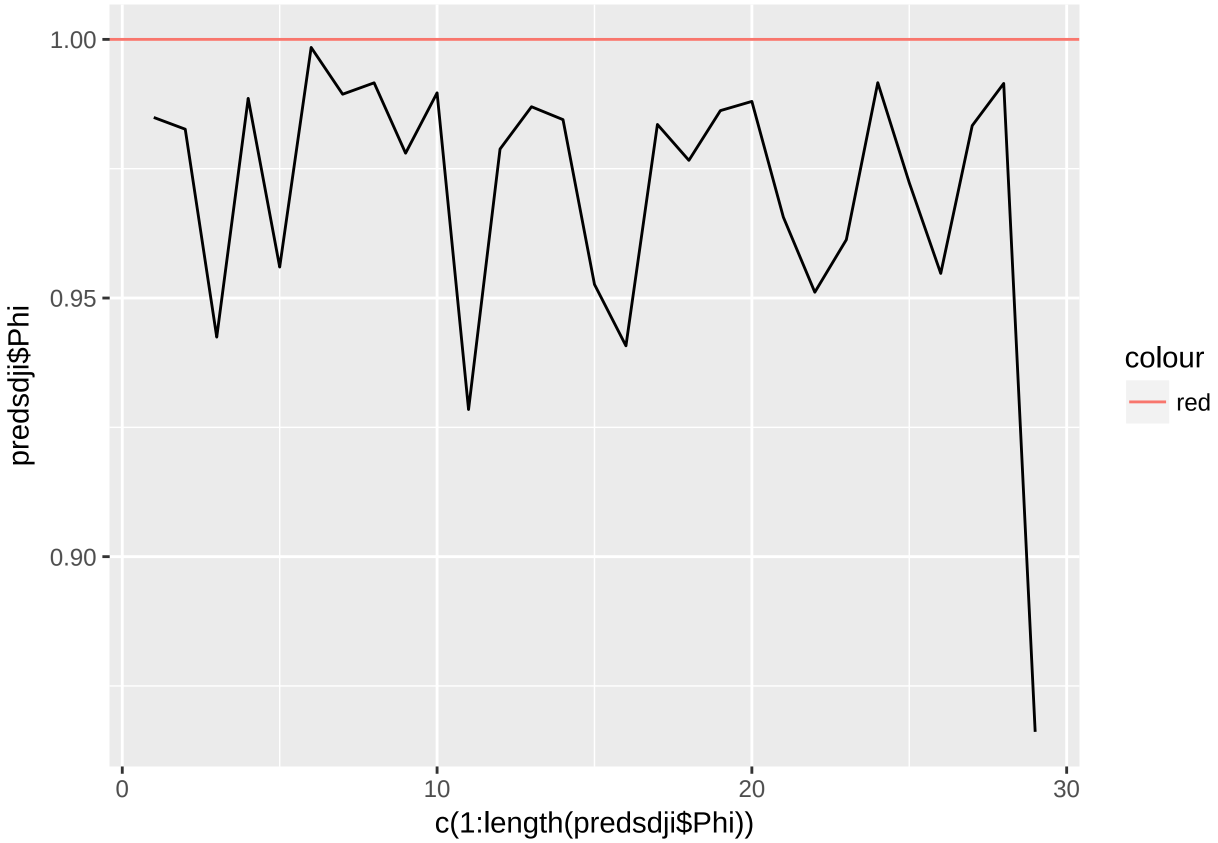
6) Use the Split/Apply/Combine strategy to estimate the autoregressive parameter ϕ over the years 1990 through 2018 using the Dow Jones daily closing price data. To summarize the results, plot the estimated parameters as a function of time. Also plot a horizontal line 1, which will give some insight on which years were more predictable.

```
# Solution
head(dji)
```

```
##         Date    Open    High     Low   Close Adj.Close   Volume
## 1 1990-01-02 2748.72 2811.65 2732.51 2810.15   2810.15 20680000
## 2 1990-01-03 2814.20 2834.04 2786.26 2809.73   2809.73 23620000
## 3 1990-01-04 2804.39 2821.46 2766.42 2796.08   2796.08 24370000
## 4 1990-01-05 2786.90 2810.15 2758.11 2773.25   2773.25 20290000
## 5 1990-01-08 2761.73 2803.97 2753.41 2794.37   2794.37 16610000
## 6 1990-01-09 2792.88 2810.79 2760.03 2766.00   2766.00 15800000
```

```
dji$Year <- matrix(unlist(strsplit(dji$Date, "-")), nrow=nrow(dji), byrow=T)[,1]
predsdji <- ddply(dji, .(Year), phi.hat)
colnames(predsdji) <- c("Year", "Intercept", "Phi")

plt <- ggplot()+geom_line(aes(x=c(1:length(predsdji$Phi)),y=predsdji$Phi))
plt + geom_hline(aes(yintercept=1, color="red"))
```


1.00
0.95
0.90
predsdji$Phi
0
10
20
30
c(1:length(predsdji$Phi))
colour
red