title: “Untitled” author: “Souradeep Das” date: “2022-11-16”

Problem 5: Computational Finance - Modelling Stock prices

Following piece of code download the prices of TCS since 2007

```
library(quantmod)
```

```
## Warning: package 'quantmod' was built under R version 4.2.2
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 4.2.2
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.2.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Warning: package 'TTR' was built under R version 4.2.2
```

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

```
getSymbols('TCS.NS')
```

```
## Warning: TCS.NS contains missing values. Some functions will not work if objects
## contain missing values in the middle of the series. Consider using na.omit(),
## na.approx(), na.fill(), etc to remove or replace them.
```

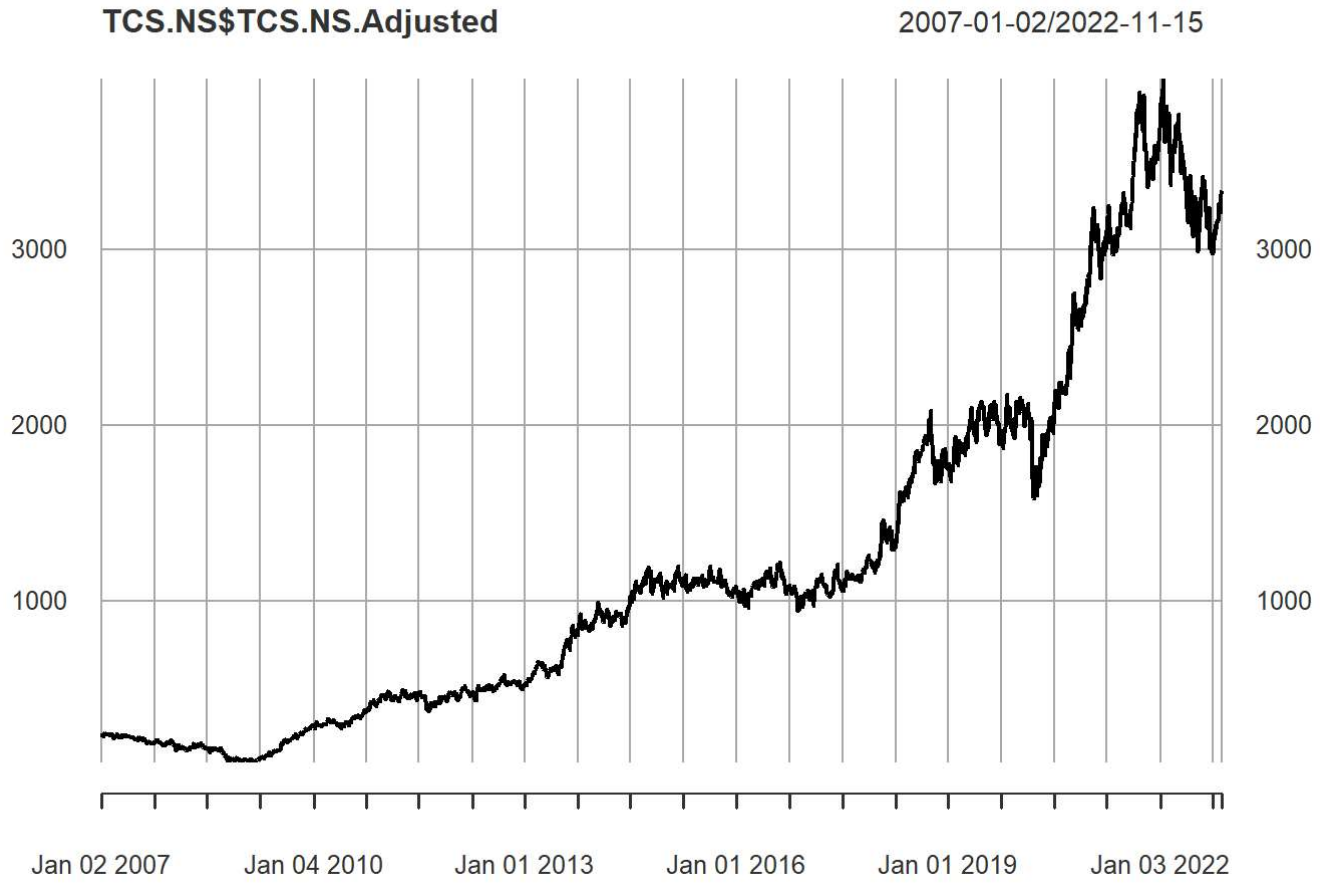
```
## [1] "TCS.NS"
```

```
tail(TCS.NS)
```

```
##            TCS.NS.Open TCS.NS.High TCS.NS.Low TCS.NS.Close TCS.NS.Volume
## 2022-11-07      3229.0     3242.80    3195.10      3233.70       1474498
## 2022-11-09      3249.8     3249.80    3201.65      3216.05       1162267
## 2022-11-10      3170.0     3225.00    3170.00      3205.65       1573092
## 2022-11-11      3269.6     3341.60    3255.05      3315.95       3265394
## 2022-11-14      3324.0     3349.00    3309.00      3335.50       1342074
## 2022-11-15      3321.0     3339.95    3292.00      3332.60       1400708
##            TCS.NS.Adjusted
## 2022-11-07         3233.70
## 2022-11-09         3216.05
## 2022-11-10         3205.65
## 2022-11-11         3315.95
## 2022-11-14         3335.50
## 2022-11-15         3332.60
```

Plot the adjusted close prices of TCS

```
plot(TCS.NS$TCS.NS.Adjusted)
```



Download the data of market index Nifty50. The Nifty 50 index indicates how the over all market has done over the similar period.

```
getSymbols('^NSEI')
```

```
## Warning: ^NSEI contains missing values. Some functions will not work if objects
## contain missing values in the middle of the series. Consider using na.omit(),
## na.approx(), na.fill(), etc to remove or replace them.
```

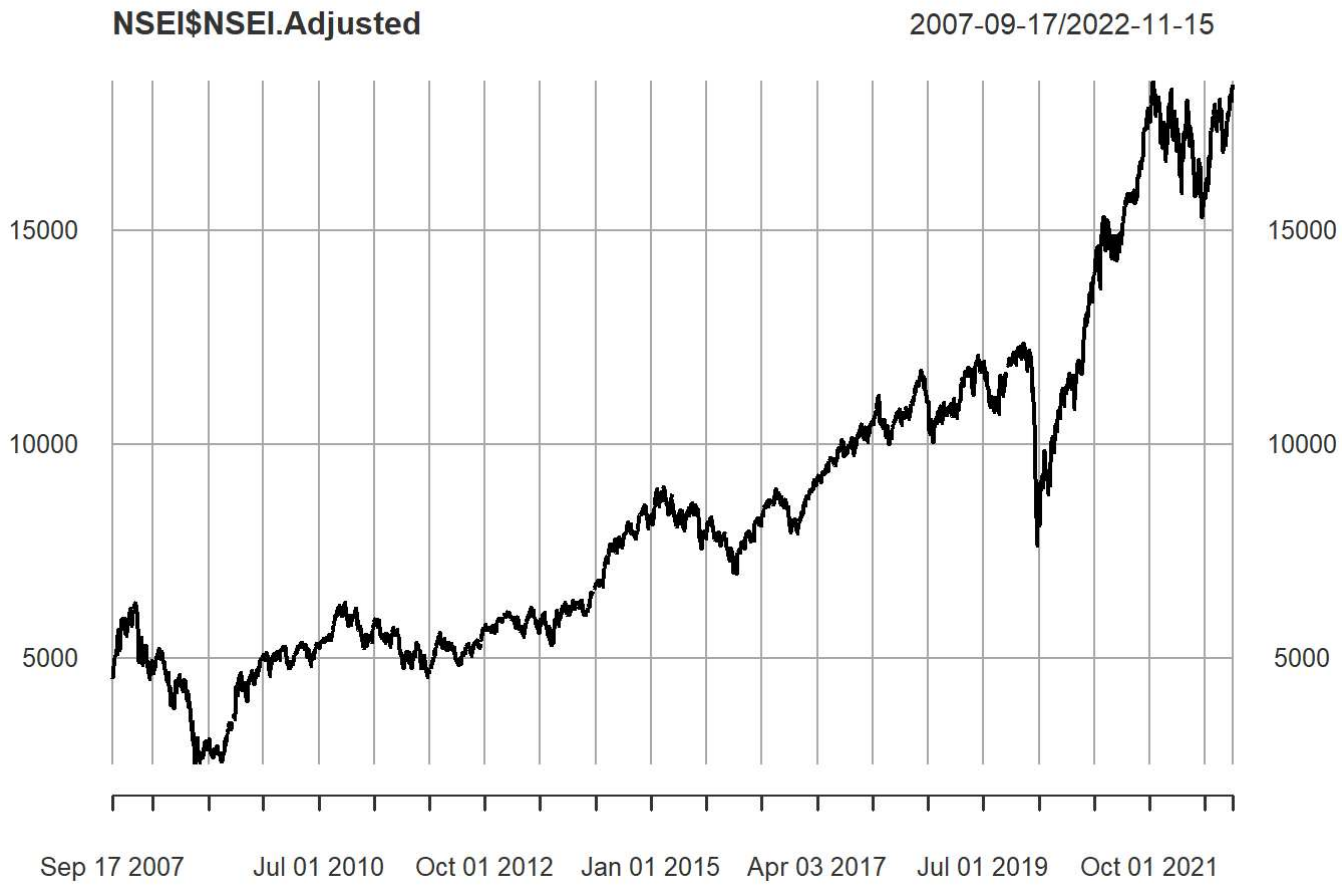
```
## [1] "^NSEI"
```

```
tail(NSEI)
```

```
##            NSEI.Open NSEI.High NSEI.Low NSEI.Close NSEI.Volume NSEI.Adjusted
## 2022-11-07  18211.75  18255.50 18064.75   18202.80      314800      18202.80
## 2022-11-09  18288.25  18296.40 18117.50   18157.00      307200      18157.00
## 2022-11-10  18044.35  18103.10 17969.40   18028.20      256500      18028.20
## 2022-11-11  18272.35  18362.30 18259.35   18349.70      378500      18349.70
## 2022-11-14  18376.40  18399.45 18311.40   18329.15      301400      18329.15
## 2022-11-15  18362.75  18427.95 18282.00   18403.40      250900      18403.40
```

Plot the adjusted close value of Nifty50

```
plot(NSEI$NSEI.Adjusted)
```



Log-Return

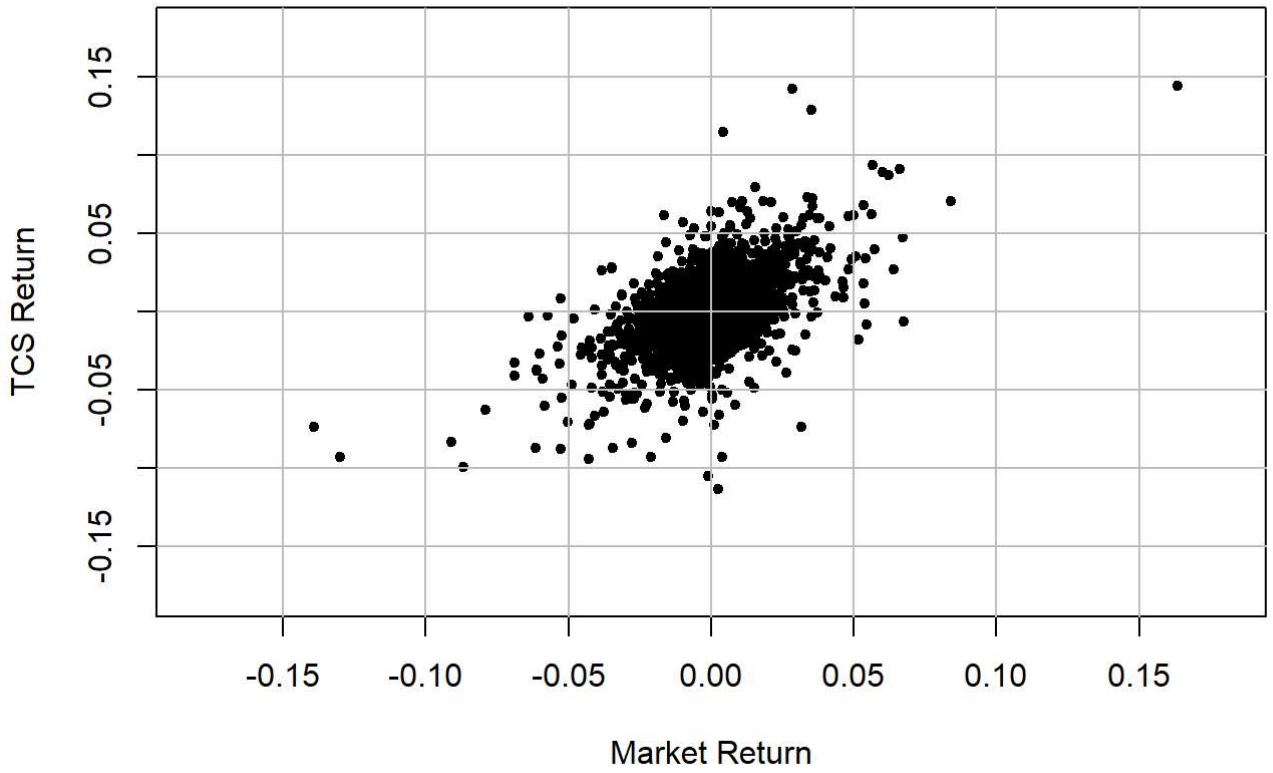
We calculate the daily log-return, where log-return is defined as

$$r_t = \log(P_t) - \log(P_{t-1}) = \Delta \log(P_t),$$

where P_t is the closing price of the stock on t^{th} day.

```
TCS_rt = diff(log(TCS.NS$TCS.NS.Adjusted))
Nifty_rt = diff(log(NSEI$NSEI.Adjusted))
retrn = cbind.xts(TCS_rt,Nifty_rt)
retrn = na.omit(data.frame(retrn))

plot(retrn$NSEI.Adjusted,retrn$TCS.NS.Adjusted
     ,pch=20
     ,xlab='Market Return'
     ,ylab='TCS Return'
     ,xlim=c(-0.18,0.18)
     ,ylim=c(-0.18,0.18))
grid(col='grey',lty=1)
```



- Consider the following model:

$$r_t^{TCS} = \alpha + \beta r_t^{Nifty} + \varepsilon,$$

where $\mathbb{E}(\varepsilon) = 0$ and $\mathbb{V}ar(\varepsilon) = \sigma^2$.

1. Estimate the parameters of the models $\theta = (\alpha, \beta, \sigma)$ using the method of moments type plug-in estimator discussed in the class.

```
library(tinytex)
```

```
## Warning: package 'tinytex' was built under R version 4.2.2
```

```
mt=TCS_mean=mean(retrn$TCS.NS.Adjusted)
varT=TCS_var=var(retrn$TCS.NS.Adjusted)
a=TCS_sd=sd(retrn$TCS.NS.Adjusted)
mn=NSEI_mean=mean(retrn$NSEI.Adjusted)
NSEI_var=var(retrn$NSEI.Adjusted)
b=NSEI_sd=sd(retrn$NSEI.Adjusted)
x=retrn$NSEI.Adjusted
y=retrn$TCS.NS.Adjusted
Cov=cov(x,y)
r=Cov/(a*b)
alpha_hat=mt-r*(a/b)*mn
beta_hat=r*(a/b)
sigma=sqrt(varT-Cov*beta_hat)
paste('The estimated value of alpha =',alpha_hat)
```

```
## [1] "The estimated value of alpha = 0.00046165283708186"
```

```
paste('The estimated value of beta =',beta_hat)
```

```
## [1] "The estimated value of beta = 0.743661476148057"
```

```
paste('The estimated value of sigma =',sigma)
```

```
## [1] "The estimated value of sigma = 0.016182626195474"
```

2. Estimate the parameters using the `lm` built-in function of `R`. Note that `lm` using the OLS method.

```r
a=model <- lm(y~x, data=retrn)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x, data = retrn)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.115339 -0.008756 -0.000086  0.008537  0.120641
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0004617  0.0002668    1.73   0.0836 .
## x           0.7436615  0.0191618   38.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01618 on 3681 degrees of freedom
## Multiple R-squared:  0.2904, Adjusted R-squared:  0.2902
## F-statistic:  1506 on 1 and 3681 DF,  p-value: < 2.2e-16
```

```
alphaols = model$coefficients[[1]]
betaols = model$coefficients[[2]]
retrn$r_tcs_predicted = model$fitted.values
retrn$error = retrn$r_tcs_predicted - retrn$TCS.NS.Adjusted
sigmaols = sd(retrn$error)
paste('The estimated value of alpha =',alphaols)
```

```
## [1] "The estimated value of alpha = 0.000461652837081857"
```

```
paste('The estimated value of beta =',betaols)
```

```
## [1] "The estimated value of beta = 0.743661476148059"
```

```
paste('The estimated value of sigma =',sigmaols)
```

```
## [1] "The estimated value of sigma = 0.016182626195474"
```

3. Fill-up the following table

Parameters	Method of Moments	OLS
α	0.00046165264727328	0.000461652647273278
β	0.743661766737062	0.743661766737066
σ	0.0161826159860443	0.0161826159860443

4. If the current value of Nifty is 18000 and it goes up to 18200. The current value of TCS is Rs. 3200/-. How much you can expect TCS price to go up?

```
nif1 = 18000
nif2 = 18200
tcs1 = 3200
nifr = (log(nif2) - log(nif1))
predicttcs = predict(model, data.frame(x=c(nifr)))
tcsvalue = round(exp(predicttcs) * tcs1)
paste('The value of TCS is expected to go up at:',tcsvalue)
```

```
## [1] "The value of TCS is expected to go up at: 3228"
```