Problem A

A) Price this geometric Asian call option.

In python code:

```
def gasian(r,T,S0,sig,N,K):
    sigma=sig*sqrt((2*N+1)/6/(N+1))
    rho=0.5*(r-0.5*sig**2+sigma**2)
    d1=1/sqrt(T)/sigma*(math.log(S0/K)+(rho+0.5*sigma**2)*T)
    d2=1/sqrt(T)/sigma*(math.log(S0/K)+(rho-0.5*sigma**2)*T)
    p=math.exp(-r*T)*(S0*math.exp(rho*T)*norm.cdf(d1)-K*norm.cdf(d2))
    return p
```

Output:

> gasian(0.03,5.,100,0.3,10000,100)

#the price of the Asian call option: 15.174849755456105

B) Implement a Monte Carlo scheme to price an arithmetic Asian call option. Record the answer, a confidence interval and the time it takes to obtain the result.

Python code:

```
for j in range(M):
    St=S
    sumSt=0
    productSt=1
    for i in range(N):
        e=gauss(0.,1.)
        St=St*math.exp(nudt+sigsdt*e)
        sumSt=sumSt+St
        productSt=productSt*St
        i=i+1
    A=sumSt/N
    G=productSt**(1/N)
    CT=max(0,A-K)-max(0,G-K)
    sum_CT=sum_CT+CT
    sum_CT2=sum_CT2+CT*CT
     j=j+1
portfolio_value=sum_CT/M*math.exp(-r*T)
call_value=gasian(0.03,5.,100,0.3,10000,100)+portfolio_value
```

Output:

#time: total time is 301.898767 seconds

#value: 15.208425837912918

#confidence interval (with a confidence=0.95):

Mean: 15.224452770533023

Interval: array([15.2237895]), array([15.22511605])

C) Implement a Monte Carlo scheme to price a geometric Asian Call option.

Python code:

```
for j in range(M):
```

```
        St=S
        sum_St=0
        for i in range(N):
                e=gauss(0.,1.)
                St=St*math.exp((nu-sig**2/2)*dt+sig*sqrt(dt)*e)
                sum_St=sum_St+St
                i=i+1
        CT=max(0,sum_St/(N+1)-K)
        sum_CT=sum_CT+CT
        sum_CT2=sum_CT2+CT*CT
        j=j+1
```

Output:

#call value: 17.217726912841904

D) • M replications for the arithmetic Asian Option price.

```
for j in range(M):
        St=S
        sumSt=0
        productSt=1
        for i in range(N):
                e=gauss(0.,1.)
                St=St*math.exp(nudt+sigsdt*e)
                sumSt=sumSt+St
                productSt=productSt*St
                i=i+1
        A=sumSt/N
        G=productSt**(1/N)
        CT=max(0,A-K)-max(0,G-K)
        sum_CT=sum_CT+CT
        sum_CT2=sum_CT2+CT*CT
        pv=sum_CT/(j+1)*math.exp(-r*T)
        callp=pv+gasian(0.03,5.,100,0.3,10000,100)
        j=j+1
```

Output:

> data.head(5)

Xi
15.20543963
15.21388654
15.20087428
15.23614352
15.22388477

• M replication for the geometric Asian Option price.

Output:

> data.tail(5)

Yi
17.8827965
17.8648599
17.84695923
17.85047701
17.83262653

- **Calculate:** # b^* : $\boxed{-0.349147025}$

E) Calculate the error of pricing for the geometric Asian.

```
> pgsim=17.217726912841904
> pg=17.8958145
> error=pg-pgsim
```

error: $\boxed{0.678087587158096}$

F) Calculate the modified arithmetic option price.

```
>pasim=15.208425837912918      #price of simulation
>beg=b*error
>pa=pasim-beg
```

#price: $\boxed{15.445178101658595}$

Compare with the results in (b).

Value in (b)	15.208425837912918
Value in (f)	15.445178101658595

The result in (f) is slightly higher than it in (b), caused by the negative b^*.

Vary the value of M in part (d). What do you observe.

#M=100
pasim=15.208425837912918
beg=b*error
pa=pasim-beg

Output: 14.676489793078446

#M=1000
pasim=15.208425837912918
beg=b*error
pa=pasim-beg

Output: 14.965913850735788

From M=100 to M=10000, the results getting higher and much closer to 15.

Problem B

1) Decide which of the models could be the source of each of the appropriate columns in the dataset. Use the AIC criterion.

```
#model 1
fx1<-expression(theta[1]*x)
gx1<-expression(theta[2]*x^theta[3])
m1<-fitsde(data=s5,drift=fx1,diffusion=gx1,start=list(theta1=1,theta2=1,theta3=1),pmle='euler')
...
```

```
AIC<-c(AIC(m1),AIC(m2),AIC(m3),AIC(m4),AIC(m5))
Test<-data.frame(AIC,row.names=c('1','2','3','4','5'))
Bestmode<-rownames(Test)[which.min(Test[,1])]
```

Output:

Stock1: 2
Stock2: 4
Stock3: 1
Stock4: 2
Stock5: 4

2) Implement Euler method, Ozaki method, Shoji-Ozaki method and Kessler method to estimate parameters for the model that you chose as the best model. Report the model and your parameter estimates in a nice table.

```
#euler
#1
fitmod <- fitsde(data = s5, drift = fx4, diffusion = gx4, start = list(theta1=1, theta2=1,theta3=1),pmle="euler")
fitmod
summary(fitmod)
confint(fitmod, level=0.95)
#   theta1          theta2          theta3          theta4
#3.504814e-05    2.214008e-05    7.536530e-03    3.903047e-01
#2
#   theta1          theta2          theta3
#0.007636416    0.006855019    0.537076888
#3
#   theta1          theta2          theta3
#0.003498813    -3.583787343    3.734064202
#4
#   theta1          theta2          theta3          theta4
#-9.421015e-05    9.618334e-06    1.471477e-02    4.269734e-01
#5
#   theta1          theta2          theta3
#0.003342042    0.009936535    0.530964394
```

```
#ozaki
#1
fitmod <- fitsde(data=s5,drift=fx4,diffusion=gx4,start = list(theta1=1,theta2=1,theta3=1),pmle="ozaki")
summary(fitmod)
fitmod
#   Estimate    Std. Error
#theta1    0.0547352626    5.141170e-04
#theta2    -0.0001325927    1.693968e-06
#theta3    0.0040184172    1.368547e-05
#theta4    0.5276808366    8.348900e-04
```

```
#    theta1              theta2              theta3              theta4
#0.0547352626    -0.0001325927    0.0040184172    0.5276808366
#2
#    theta1    theta2    theta3
#      1          1          1
#3
#    Estimate      Std. Error
#theta1    7.747055e-06    3.535981e-06
#theta2    -2.858406e-03    4.646396e-06
#theta3    8.356426e-01    5.486330e-04
#    theta1              theta2              theta3
#7.747055e-06    -2.858406e-03    8.356426e-01
#4
#    Estimate      Std. Error
#theta1    6.279268e-03 1.729448e-03
#theta2 -5.520604e-05 1.081078e-05
#theta3    2.751369e-02 4.288770e-04
#theta4    3.015314e-01 3.084769e-03
#    theta1              theta2              theta3              theta4
#6.279268e-03    -5.520604e-05    2.751369e-02    3.015314e-01
#5
#    theta1 theta2 theta3
#1          1          1
```

#Shoji-Ozaki

```
fitsde(data=s5,drift=fx4,diffusion=gx4,start =
list(theta1=1,theta2=1,theta3=1,theta4=1),pmle="shoji",lower=c(-3,0),upper=c(-1,1))
summary(fitmod)
#1
#    theta1 theta2      theta3
#    1            1          1
#2
#    theta1    theta2    theta3
#    -1          1          -1
#3
#    theta1    theta2    theta3
#      -1          1          -1
#4
#    theta1    theta2    theta3      theta4
#      -3          0          -1          0
#5
#    theta1    theta2    theta3    theta4
#    -1          1          -1          1
```

#Kessler

```
fitmod <- fitsde(data=s5,drift=fx4,diffusion=gx4,start =
```

```
list(theta1=1,theta2=1,theta3=1),pmle="kessler")
summary(fitmod)
fitmod
#1
#   theta1       theta2         theta3       theta4
#1.0000448   0.6862942   0.8824043   0.4091350
#2
#   Estimate    Std. Error
#theta1 0.004397859 0.0003881697
#theta2 0.007841422 0.0000599039
#theta3 0.513984058 0.0014465610
#   theta1            theta2            theta3
#0.004397859   0.007841422   0.513984058
#3
#   theta1       theta2         theta3
#0.5498652   0.9359193   0.6267437
#4
#   theta1         theta2         theta3          theta4
#1.0072354   0.4606634   0.6495840   0.1907163
#5
#   Estimate    Std. Error
#theta1 0.01128962 5.614897e-04
#theta2 0.01075011 9.730073e-05
#theta3 0.51575952 1.674067e-03
#   theta1            theta2           theta3
#0.01128962   0.01075011   0.51575952
```

			Theta1	Theta2	Theta3	Theta4
Euler method	stock 1	Model 2	3.5048e-05	2.2140e-05	7.5365e-03	3.9030e-01
	stock 2	Model 4	**0.00763641**	**0.00685501**	**0.5370768**	
	stock 3	Model 1	**0.00349881**	**-3.5837873**	**3.7340642**	
	stock 4	Model 2	-9.421e-05	9.6183e-06	1.4714e-02	4.2697e-01
	stock 5	Model 4	**0.00334204**	**0.00993653**	**0.53096439**	
Ozaki method	stock 1	Model 2	**0.05473526**	**-0.0001325**	**0.0040184**	**0.5276808**
	stock 2	Model 4	1	1	1	
	stock 3	Model 1	7.7470e-06	-2.858e-03	8.3564e-01	
	stock 4	Model 2	6.27926e-3	-5.520e-05	2.7513e-02	3.0153e-01
	stock 5	Model 4	1	1	1	
Shoji-Ozaki method	stock 1	Model 2	-3	0	-1	0
	stock 2	Model 4	-1	1	1	
	stock 3	Model 1	-1	1	-1	
	stock 4	Model 2	-3	0	-1	0
	stock 5	Model 4	-1	1	-1	1

Kessler method	stock 1	Model 2	**1.0000448**	**0.6862942**	**0.8824043**	**0.4091350**
	stock 2	Model 4	**0.00439785**	**0.00784142**	**0.5139840**	
	stock 3	Model 1	**0.5498652**	**0.9359193**	**0.6267437**	
	stock 4	Model 2	**1.0072354**	**0.4606634**	**0.6495840**	**0.1907163**
	stock 5	Model 4	**0.01128962**	**0.01075011**	**0.5157595**	

3) In your opinion which method gives you the best estimates?

From the results I have got, the Kessler method gives a relatively better estimation while solving the parameters. A boundary has been given for the Shoji-Ozaki method, thus from my perspective the estimates are a range or a tendency of changes of values of the real results.

Problem C

1. Download daily prices for the components of DJIA for the last 5 years.

```
stockData <- new.env()
lookup.symb=c('MMM','AXP','AAPL','BA','CAT','CVX','CSCO','KO','DWDP','XOM','GE','GS',
              'HD','INTC','IBM','JNJ','JPM','MCD','MRK','MSFT','NKE','PFE','PG',
              'TRV','UNH','UTX','VZ','V','WMT','DIS')
getSymbols(lookup.symb, from='2013-1-1',to='2017-12-31', env=stockData, src="yahoo")
ReturnMatrix=NULL
for(i in 1:length(lookup.symb))
{
  tmp <- get(lookup.symb[i], pos=stockData)    # get data from stockData environment
  ReturnMatrix=cbind(ReturnMatrix,    (Cl(tmp)-Op(tmp)) / Op(tmp)    )
  colnames(ReturnMatrix)[i]=lookup.symb[i]
}
```

```
> head(ReturnMatrix)
                   MMM          AXP         AAPL           BA          CAT          CVX
2013-01-02 0.006263903  0.007200377 -0.008649049  0.006792906  0.005917192  0.001088256
2013-01-03 0.003498007  0.003401378 -0.010549752  0.006234615  0.012658238 -0.001997467
2013-01-04 0.006118810  0.008970870 -0.018567149 -0.001285499  0.003700941  0.004636803
2013-01-07 0.004946338  0.009784109  0.003639879 -0.018184214  0.006767463 -0.002454072
2013-01-08 0.003467500  0.008713153 -0.007369385 -0.014752805 -0.007077195 -0.002009509
2013-01-09 0.005632638 -0.002648130 -0.010334960  0.024559584  0.003613220 -0.001640540
                    CSCO            KO         DWDP          XOM            GE
2013-01-02  0.0109343434  0.0164908345  0.001818642  0.010479531 -0.0079033008
2013-01-03 -0.0004886608  0.0040052602  0.005459539  0.001017454 -0.0176908752
2013-01-04  0.0034296913  0.0002655511  0.012951807  0.005197729 -0.0004713814
2013-01-07 -0.0044159963 -0.0058635661  0.012967431 -0.004303001 -0.0014178166
2013-01-08  0.0064419722 -0.0040332886 -0.011940358  0.007515407 -0.0094786730
2013-01-09  0.0019742843 -0.0045699461  0.005402161 -0.002941210 -0.0042775663
                     GS            HD         INTC           IBM          JNJ
2013-01-02  0.002741820 -0.0014157622  0.010874657  0.0116441344 -0.002534512
2013-01-03 -0.005770653 -0.0075258703 -0.003738318 -0.0020442275 -0.003381304
2013-01-04  0.027342872 -0.0004745926 -0.009363342 -0.0010299037  0.004351558
2013-01-07  0.001043841  0.0038338979 -0.001409821 -0.0013443382 -0.001398573
2013-01-08 -0.004936085  0.0074900558 -0.006594442 -0.0002591903  0.001402412
2013-01-09  0.006896649 -0.0053534878  0.010838926 -0.0059953950  0.001256337
                    JPM           MCD          MRK          MSFT          NKE
2013-01-02 -0.007114273  0.0080537023 -0.012422384  0.0135780183 -0.010687061
2013-01-03 -0.003799776  0.0035433397  0.010503772 -0.0137531312  0.008084658
2013-01-04  0.020472440 -0.0084970754 -0.011540226 -0.0194352769  0.007238133
2013-01-07  0.007096917  0.0126991984  0.001902902 -0.0029883825  0.003220307
2013-01-08  0.002202599  0.0044179478 -0.001656805 -0.0074766729 -0.006069803
2013-01-09 -0.001975417  0.0001100771  0.004718118 -0.0007484282 -0.002282671
                     PFE            PG          TRV          UNH           UTX
2013-01-02  0.0148844099  0.0107792714 -0.003146737 -0.006195335  0.0068320748
2013-01-03 -0.0050038491 -0.0070565234  0.008516414 -0.027133178  0.0066865433
2013-01-04  0.0042552805  0.0011591653  0.007619020 -0.002107299  0.0060376701
2013-01-07  0.0027016596 -0.0043528005 -0.008280169  0.004047783  0.0003548498
2013-01-08  0.0007692308 -0.0001458698  0.002877487 -0.011538423 -0.0099537387
2013-01-09  0.0095347067  0.0015995348  0.007082593  0.015119190  0.0072671073
                    VZ           V         WMT         DIS
2013-01-02 -0.004944976 0.007913856 0.004497287 0.005905492
```

Construct the corresponding matrix of standardized returns.

```
for(i in 1:length(lookup.symb))
{
  for(j in 1:length(ReturnMatrix$MMM))
  {
    tmp <- get(lookup.symb[i], pos=stockData)    # get data from stockData environment
    ReturnMatrix=cbind(ReturnMatrix,    (Cl(tmp)-Op(tmp)) / Op(tmp)    )
    mean<-cbind(mean,mean(ReturnMatrix[,i]))
    sig<-cbind(sig,(ReturnMatrix[j,i]-mean[,i])^2)
    sr<-cbind(sr,(ReturnMatrix[j,i]-mean[,i])/sig[,i])
    colnames(ReturnMatrix)[i]=lookup.symb[i]
  }
}
```

```
> head(data1)
        MMM       AXP      AAPL        BA       CAT        CVX      CSCO         KO
1 0.7234936 0.6889000 0.7958418 0.5681240 0.4752218 0.02686954 1.1066630 2.15365128
2 0.3568513 0.2796156 0.9558265 0.5151456 1.1019786 0.26684889 0.1414564 0.43928665
3 0.7042603 0.8796437 1.6306615 0.1984680 0.2691632 0.36464246 0.2866772 0.07420395
4 0.5488395 0.9672579 0.2385337 1.8020544 0.5542767 0.31031141 0.5705714 0.91577851
5 0.3528074 0.8518786 0.6881308 1.4764344 0.7329469 0.26799513 0.6158101 0.66446751
6 0.6398142 0.3721270 0.9377472 2.2540749 0.2610072 0.23287435 0.1276540 0.73815461
        DWDP        XOM        GE         GS        HD      INTC       IBM        JNJ
1 0.08359762 1.10292074 0.9001839 0.17419322 0.2347422 0.9631287 1.2645430 0.44503774
2 0.38415987 0.02411188 1.9118201 0.59178388 0.8803189 0.4346880 0.3325472 0.55785205
3 1.00265935 0.50072158 0.1320263 2.38786751 0.1353009 0.9727544 0.2142009 0.47236252
4 1.00394914 0.58249426 0.2298492 0.02140413 0.3199221 0.2119536 0.2508875 0.29370168
5 1.05223126 0.76496923 1.0630131 0.51668702 0.7062214 0.7078929 0.1242778 0.07946101
6 0.37942321 0.42723107 0.5254306 0.54805684 0.6507911 0.9597108 0.7935498 0.06000009
        JPM        MCD       MRK      MSFT       NKE         PFE        PG       TRV
1 0.7977352 0.96970722 1.2989610 1.2211806 1.0992328 1.559068854 1.3683182 0.5084694
2 0.4638957 0.36624507 0.9522774 1.3920475 0.6961583 0.643388943 1.0787198 0.9918573
3 1.9808268 1.24469761 1.2123374 1.9353376 0.6151938 0.381981898 0.0484574 0.8764182
4 0.6336304 1.59124956 0.1077132 0.3627918 0.2309152 0.209931520 0.7077740 1.1688248
5 0.1406697 0.48326291 0.2418329 0.7919331 0.6576226 0.004068776 0.1305910 0.2664759
6 0.2801444 0.09310703 0.3841539 0.1486217 0.2954084 0.966634137 0.1088754 0.8074132
        UNH        UTX        VZ         V       WMT       DIS
1 0.6597164 0.70346654 0.6741384 0.6800120 0.4290379 0.5523646
2 2.6326393 0.68647763 0.8872333 1.0574961 0.8489511 0.2521847
3 0.2745106 0.61073043 0.5975639 0.6422604 0.4643758 1.6203005
4 0.3054681 0.05266245 1.0711181 0.5020070 0.8197920 1.7405095
5 1.1631828 1.25605171 2.5704418 0.9606132 0.4340122 0.3427714
6 1.3487003 0.75425068 0.3433266 0.1540657 0.2460379 0.4063199
```

2. Calculate the sample correlation matrix.

```
y2=[]
for j in range(30):
    y=0
    for i in range(1258):
        for z in range(30):
            y=y+data3.iat[i,z]*data3.iat[i,j]
    print(y/1258)
```

```
> head(data1)
        MMM       AXP      AAPL        BA       CAT       CVX      CSCO        KO
1 1.0003724 0.6739886 0.6254081 0.6775964 0.6835027 0.6805575 0.6865727 0.6805162
2 0.6739886 1.0002470 0.6215815 0.6390482 0.6393937 0.6161189 0.6493662 0.6406562
3 0.6254081 0.6215815 1.0004003 0.6292106 0.6133323 0.6338934 0.6696885 0.6180964
4 0.6775964 0.6390482 0.6292106 1.0003768 0.6408405 0.6034144 0.6866050 0.6256681
5 0.6835027 0.6393937 0.6133323 0.6408405 1.0006967 0.6839641 0.6614754 0.6499022
6 0.6805575 0.6161189 0.6338934 0.6034144 0.6839641 1.0007612 0.6400042 0.6384488
       DWDP       XOM        GE        GS        HD      INTC       IBM       JNJ
1 0.6836252 0.7012336 0.6773382 0.6900006 0.6782581 0.6758835 0.6932072 0.7217449
2 0.6525208 0.6104317 0.6148239 0.7042990 0.6362978 0.6151173 0.6644371 0.6374350
3 0.6278236 0.6238913 0.6085489 0.6367086 0.6522542 0.6669880 0.6566974 0.6406260
4 0.6556440 0.6155669 0.6323678 0.6620556 0.6290586 0.6495346 0.6520044 0.6474325
5 0.6541996 0.6748650 0.6221509 0.6594441 0.6296041 0.6363042 0.6795763 0.6200650
6 0.6792556 0.8277533 0.6163886 0.6380781 0.6474603 0.6370943 0.6536648 0.6455525
        JPM       MCD       MRK      MSFT       NKE       PFE        PG       TRV
1 0.7011557 0.6484429 0.6374175 0.6940629 0.6569853 0.6652167 0.6970468 0.7103686
2 0.6912813 0.6192125 0.6050799 0.6508959 0.6477193 0.6194142 0.6098470 0.6434908
3 0.6459015 0.6186717 0.6056737 0.7017000 0.6317310 0.6207185 0.6366852 0.6307402
4 0.6472508 0.6113768 0.5857813 0.6513677 0.6435868 0.6100264 0.6267558 0.6376712
5 0.6466365 0.6205683 0.5931349 0.6193180 0.6469693 0.6164908 0.6320901 0.6217068
6 0.6504434 0.6301942 0.6220489 0.6551352 0.6375840 0.6503867 0.6313823 0.6278094
        UNH       UTX        VZ         V       WMT       DIS
1 0.6769293 0.7381390 0.6614434 0.6747320 0.6255814 0.6914079
2 0.6190844 0.6646365 0.6469450 0.6692124 0.5872408 0.6632217
3 0.6327599 0.6591312 0.6452380 0.6626335 0.5784242 0.6490351
4 0.6474469 0.6804328 0.6230225 0.6350590 0.6154688 0.6702493
5 0.6213879 0.6813565 0.6378271 0.6169353 0.6012449 0.6188032
6 0.6230686 0.6500856 0.6376804 0.6284602 0.6129099 0.6329955
```

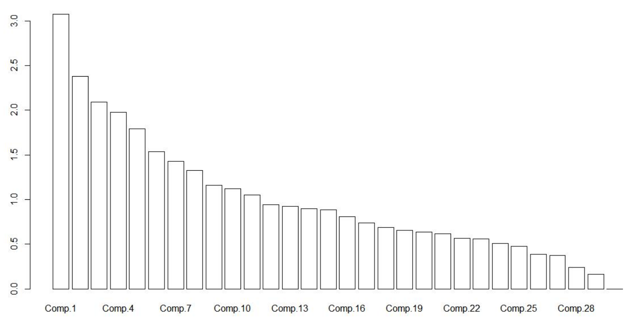
3. Calculate the eigenvalues and eigenvectors.

```
pc1<-princomp(data2,cor=TRUE,scor=TRUE)
summary(pc1)
evec = pc1$rotation[]
eval1 = eigen(data2, symmetric=TRUE)
eval = pc1$sd^2
barplot(eval,col=colors)
```

```
eigen() decomposition
$values
 [1] 19.7481936  0.6256150  0.5613282  0.5454473  0.5177285  0.4695562  0.4580157
 [8]  0.4359166  0.4066354  0.3943283  0.3880594  0.3640348  0.3580971  0.3565372
[15]  0.3469905  0.3405226  0.3352657  0.3143832  0.3043505  0.3008763  0.2908941
[22]  0.2840899  0.2814808  0.2741031  0.2600669  0.2511645  0.2307260  0.2193700
[29]  0.1846239  0.1610267
```

```
$vectors
           [,1]         [,2]         [,3]         [,4]        [,5]         [,6]
 [1,] -0.1916621 -0.049145066 -0.074334636 -0.016399602  0.12379935  0.221332872
 [2,] -0.1808168  0.182506570 -0.154544377 -0.251543214  0.17514086 -0.141433860
 [3,] -0.1797568  0.131087955  0.092052412 -0.029301643 -0.41325587  0.062218988
 [4,] -0.1806002  0.156535270 -0.182511862 -0.034616599 -0.13326383  0.088356783
 [5,] -0.1805134 -0.248243670 -0.223848033 -0.153183022 -0.02711652  0.076229247
 [6,] -0.1824424 -0.550867608  0.067826764 -0.214467871 -0.12154830 -0.043522161
 [7,] -0.1869406  0.150189185 -0.033857599 -0.054091945 -0.21894928  0.125991819
 [8,] -0.1821477 -0.067267001 -0.122619415  0.324480657  0.19931991  0.133024228
 [9,] -0.1829220 -0.098800711 -0.071532027 -0.197789227  0.03082556 -0.086628254
[10,] -0.1821372 -0.557012159  0.035527790 -0.203734230 -0.05917349 -0.018368095
[11,] -0.1775229  0.052662983 -0.080145106  0.044145693  0.14019276  0.038876755
[12,] -0.1872628  0.156660713 -0.151838427 -0.254275657  0.29584201 -0.104115489
[13,] -0.1837041  0.041568083 -0.012322764  0.150010777 -0.09576812 -0.438530365
[14,] -0.1819534  0.066148475  0.025302073  0.054584830 -0.37309463  0.136266211
[15,] -0.1855408 -0.006285863 -0.106679004 -0.037613478 -0.06476403  0.252134559
[16,] -0.1862692  0.021110035  0.265764075  0.116413173  0.18535374  0.137494795
[17,] -0.1873969  0.131679516 -0.069089184 -0.276449772  0.31286275 -0.031843994
[18,] -0.1787236 -0.029453420 -0.042139476  0.287369526 -0.01524325 -0.338856711
[19,] -0.1738943  0.003666977  0.561623246 -0.037163588  0.11902769  0.107869897
[20,] -0.1851035  0.155726937  0.098683173 -0.041829092 -0.35420959  0.193608577
[21,] -0.1826308  0.034665609  0.044889829 -0.004415151 -0.05955369 -0.435172223
[22,] -0.1808823 -0.041570328  0.472382138  0.038807609  0.17150662 -0.003476908
[23,] -0.1834057 -0.050441014  0.009034868  0.385032133  0.19925710  0.228020316
[24,] -0.1827111  0.065537797 -0.219648340  0.143502602  0.12523849  0.174190560
[25,] -0.1812572  0.054428980  0.170978698  0.122272635  0.04499653 -0.014395693
[26,] -0.1876275  0.062135967 -0.193845562 -0.078697193 -0.05044829  0.199208431
[27,] -0.1808041 -0.038243669 -0.088914848  0.032195233  0.08984873 -0.047283840
[28,] -0.1826268  0.227694664  0.108458658 -0.097155954 -0.10796033 -0.157320818
[29,] -0.1711339 -0.169562496 -0.197186752  0.439319458 -0.11353680 -0.193958975
[30,] -0.1855039  0.188083580  0.093268680 -0.109161758 -0.02117109 -0.152744069
```

Graph the eigenvalues.



What percent of the trace is explained by summing the first 5 eigenvalues?

```
> summary(pc1)
Importance of components:
                          Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6
Standard deviation     1.7528775 1.54189837 1.44591994 1.40644559 1.33921779 1.23931755
Proportion of Variance 0.1024193 0.07924835 0.06968948 0.06593631 0.05978348 0.05119693
Cumulative Proportion  0.1024193 0.18166767 0.25135715 0.31729346 0.37707694 0.42827387
                           Comp.7     Comp.8     Comp.9    Comp.10    Comp.11    Comp.12
Standard deviation     1.19574963 1.15088273 1.07782893 1.05783125 1.02679558 0.97029310
Proportion of Variance 0.04766057 0.04415103 0.03872384 0.03730023 0.03514364 0.03138229
Cumulative Proportion  0.47593444 0.52008548 0.55880932 0.59610955 0.63125319 0.66263548
                          Comp.13    Comp.14    Comp.15    Comp.16    Comp.17    Comp.18
Standard deviation     0.96142434 0.94648063 0.94103991 0.89985639 0.85952546 0.82888560
Proportion of Variance 0.03081123 0.02986085 0.02951854 0.02699138 0.02462613 0.02290171
Cumulative Proportion  0.69344670 0.72330755 0.75282609 0.77981748 0.80444361 0.82734532
                          Comp.19   Comp.20    Comp.21   Comp.22    Comp.23    Comp.24
Standard deviation     0.80998758 0.7964384 0.78521203 0.7522725 0.74735226 0.71178913
Proportion of Variance 0.02186933 0.0211438 0.02055193 0.0188638 0.01861785 0.01688813
Cumulative Proportion  0.84921465 0.8703585 0.89091039 0.9097742 0.92839203 0.94528016
                          Comp.25    Comp.26   Comp.27     Comp.28     Comp.29
Standard deviation     0.69011796 0.62215000 0.6147561 0.488624486 0.401973728
Proportion of Variance 0.01587543 0.01290235 0.0125975 0.007958463 0.005386096
Cumulative Proportion  0.96115558 0.97405794 0.9866554 0.994613904 1.000000000
                            Comp.30
Standard deviation     2.239110e-08
Proportion of Variance 1.671204e-17
Cumulative Proportion  1.000000e+00
```

The first five eigenvalues explain 37.7% of the trace.

4. **Define the factor** F_t**. Calculate the sample mean and sample standard deviation of the factor F.**

> head (ft)

Ft
-2.216847808
0.98033572
-1.742837236
-0.16383466
1.121384779

Mean
-0.221865151
Standard deviation
3.22953647

5. Perform a linear regression of the returns of F with the standardized returns of DIA.

```
lookup.symb=c('DIA')
getSymbols(lookup.symb, from='2013-1-1',to='2017-12-31', env=stockData, src="yahoo")
ReturnMatrix=NULL
for(i in 1:length(lookup.symb))
{
  tmp <- get(lookup.symb[i], pos=stockData)    # get data from stockData environment
  ReturnMatrix=cbind(ReturnMatrix,    (Cl(tmp)-Op(tmp)) / Op(tmp)    )
  colnames(ReturnMatrix)[i]=lookup.symb[i]
}
```

```
>head(DIA)
0.005865095
-7.48E-05
0.002692603
-0.001346451
-0.001950262
```

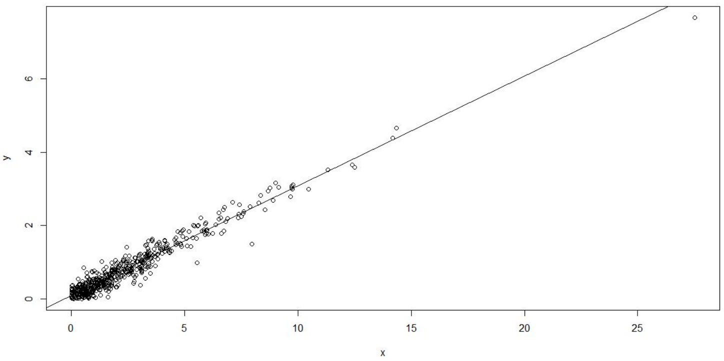
```
y<-as.numeric(data3[,1])
x<-as.numeric(data3[,2])
plot(y~x)
lm.ab<-lm(y ~ x)
lm.ab
Xm<-mean(x);Xm
Ym<-mean(y);Ym
b <- sum((x-Xm)*(y-Ym)) / sum((x-Xm)^2) ;b
a <- Ym - b * Xm;a
abline(lm.ab)
```

```
> lm.ab

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
     0.0854       0.3000
```

```
> Xm<-mean(x);Xm
[1] 2.376973
> Ym<-mean(y);Ym
[1] 0.7985658
> b <- sum((x-Xm)*(y-Ym)) / sum((x-Xm)^2) ;b
[1] 0.3000325
> a <- Ym - b * Xm;a
[1] 0.08539663
```



Calculate the R-squared of the regression and discuss whether F and the capitalization-weighted market portfolio are good proxies for each other. Discuss the result and argue why F and the particular market index might be related.

eruption.lm = lm(y ~ 1+x, data=faithful)

summary(eruption.lm)$r.squared   #r-square

#R-square: $\boxed{0.9428803}$

From the regression figure, F and the standard returns of DIA are well linearly regressive. Since F is calculated based on all the underlying assets, we can include that F shows some properties of the portfolio.

In this case, while calculating F, the first component of PCA results are chosen, which could explain most of the data.

6. A)Estimate the mean μs and standard deviation σ s for return of each equity Rs.

> head(mean)
0.000805972
0.000282479
7.72E-05
0.000739275
0.000338447

>head(stdev)
0.007546853
0.00927096
0.011862867
0.010542052
0.010749545

Standardize the returns.

```
>head(data)
> head(data4)
    MMM    AXP   AAPL     BA    CAT     CVX   CSCO      KO   DWDP     XOM      GE      GS
1 0.723  0.746 -0.736  0.574  0.519  0.0776  1.120  2.2000  0.130  1.1600 -0.8050  0.2150
2 0.357  0.336 -0.896  0.521  1.150 -0.2160 -0.130  0.4890  0.431  0.0837 -1.8200 -0.5510
3 0.704  0.937 -1.570 -0.192  0.313  0.4160  0.298 -0.0251  1.050  0.5610 -0.0333  2.4300
4 0.549  1.020  0.300 -1.800  0.598 -0.2600 -0.559 -0.8670  1.050 -0.5240 -0.1320  0.0624
5 0.353  0.909 -0.628 -1.470 -0.690 -0.2170  0.627 -0.6160 -1.010  0.8260 -0.9690 -0.4760
6 0.640 -0.316 -0.878  2.260  0.305 -0.1820  0.139 -0.6900  0.426 -0.3680 -0.4280  0.5890
       HD   INTC     IBM    JNJ    JPM     MCD    MRK   MSFT    NKE     PFE      PG
1 -0.1900  0.949  1.3300 -0.392 -0.759  1.0100 -1.230  1.220 -1.060  1.6400  1.4500
2 -0.8360 -0.448 -0.2670 -0.505 -0.425  0.4050  1.030 -1.400  0.739 -0.5680 -1.0000
3 -0.0901 -0.986 -0.1490  0.527  2.020 -1.2100 -1.140 -1.940  0.658  0.4600  0.1250
4  0.3650 -0.225 -0.1860 -0.240  0.673  1.6300  0.180 -0.367  0.273  0.2880 -0.6330
5  0.7520 -0.721 -0.0588  0.133  0.180  0.5220 -0.170 -0.796 -0.616  0.0731 -0.0543
6 -0.6060  0.946 -0.7290  0.114 -0.241 -0.0542  0.457 -0.153 -0.253  1.0500  0.1860
     TRV    UNH     UTX     VZ      V    WMT    DIS
1 -0.456 -0.650  0.7880 -0.602  0.746  0.467  0.621
2  1.050 -2.620  0.7710 -0.816 -0.994 -0.812  0.320
3  0.931 -0.265  0.6950  0.672  0.709  0.502  1.690
4 -1.120  0.315  0.0299  1.150  0.568 -0.783 -1.680
5  0.320 -1.150 -1.1800 -2.500  1.030  0.472 -0.276
6  0.862  1.360  0.8390  0.418  0.219 -0.209 -0.340
```

2) Run a regression with the 5 factors and obtain the parameters β sk.

```
lm.ab<-lm(y ~ x1+x2+x3+x4+x5)
lm.ab
```

Results:

>head

Stock	Beta1	Beta2	Beta3	Beta4	Beta5
MMM	-0.22699	-0.02787	-0.07007	0.00602	0.05990
AXP	-0.17508	0.17640	-0.09083	-0.15193	0.15586
AAPL	-0.16644	0.10390	0.06186	-0.02504	-0.32225
BA	-0.17221	0.10793	-0.15768	-0.02944	-0.02982
CAT	-0.16673	-0.20605	-0.16168	-0.17382	-0.06215

Generate realizations for Rs for the next 10 days for all the index components.

```
def student_t(nu): # nu equals number of degrees of freedom
    x = random.gauss(0.0, 1.0)
    y = 2.0*random.gammavariate(0.5*nu, 2.0)
    return x / (math.sqrt(y/nu))

for i in range(30):
    for j in range(10):
        f1=student_t(3.5)
        f2=student_t(3.5)
        f3=student_t(3.5)
        f4=student_t(3.5)
        f5=student_t(3.5)
        g=student_t(3.5)
        R=data1.iat[i,0]+data1.iat[i,1]*(data1.iat[i,2]*f1+data1.iat[i,3]*f2+data1.iat[i,4]*f3+data1.iat[i,5]*f4+data1.iat[i,6]*f5)+data1.iat[i,1]*data1.iat[i,7]*g
        print(R)
```

Output:

```
          MMM          AXP         AAPL           BA          CAT          CVX
1 -0.009872060  0.005567393 -0.012128444  0.008879117 -0.009681579  0.003875956
2  0.006437382  0.014143701  0.002533408 -0.005811765  0.011033932  0.005177635
3 -0.000637923  0.015010864  0.000654041 -0.006905980 -0.000537946 -0.006180682
4 -0.013044761  0.005774173 -0.005303592  0.001142762 -0.005690289  0.011885409
5  0.006396585 -0.005610263  0.011605855 -0.003004574  0.003922358 -0.000239803
6  0.003072506  0.014368911  0.011912401  0.015393177 -0.000312075  0.001065590
         CSCO           KO         DWDP          XOM           GE           GS
1  0.012151262 -0.024642846  0.009574500 -0.010868713 -0.003968454  0.014271275
2  0.010880483 -0.031439895  0.006497319  0.000981946  0.000423234  0.008956890
3 -0.010746385 -0.002868255  0.011051516 -0.011350750  0.005144127 -0.010264179
4 -0.002330994 -0.003468691 -0.003347728  0.016766935 -0.008313722  0.005897147
5  0.013861433 -0.000524964 -0.005303483  0.011979802 -0.008694859 -0.005188954
6 -0.002347634  0.007382393  0.013020450  0.010408935 -0.000908299 -0.000457126
           HD         INTC          IBM          JNJ          JPM          MCD
1  0.003361212 -0.015849638  0.000935782  0.004734152  0.007006969 -0.003982920
2  0.000286363 -0.010355811 -0.018246930  0.001988754  0.005354679  0.005949425
3  0.024455218  0.003803971  0.010792830  0.006534811  0.007975776  0.040861708
4  0.006797553 -0.000457325  0.037344036 -0.001837859  0.002894326  0.000368532
5 -0.009461229  0.012912131  0.004532769  0.003606485  0.027356393 -0.017163128
6 -0.007096618 -0.013890413 -0.005561776  0.010429315 -0.025872580  0.002971796
          MRK         MSFT          NKE          PFE           PG         TRV
1  0.001918651  0.013693172 -0.021961303 -0.011166118  0.007188333 0.007945748
2  0.012114632 -0.003042009 -0.005486833 -0.011700201 -0.003973493 0.001134682
3  0.006195289  0.018672750  0.001257363 -0.005890831  0.003468504 0.008935485
4 -0.006078633  0.023781216 -0.008167751 -0.006730820  0.003578280 0.013984860
5  0.017161315  0.009288696  0.001395778 -0.006873975 -0.007892119 0.000346561
6  0.002620634  0.014544518 -0.010549655  0.007574440  0.016085972 0.001025173
          UNH          UTX           VZ            V          WMT         DIS
1 -0.006711837 -0.010108115  0.000870613 -0.011892156 -0.013594880 0.001758849
2  0.010070679 -0.007576787  0.008153332 -0.008712654  0.007139806 0.000304629
3  0.026081751  0.002666776 -0.003563157 -0.010206977  0.012161915 0.021750098
4  0.001465623 -0.004067102 -0.001410236 -0.010751805  0.000707192 0.000730926
5 -0.004636628 -0.004600915  0.007353025 -0.007122755 -0.000797379 0.007786459
6  0.008178878 -0.001390710 -0.006505049 -0.000291911  0.005432064 0.003781423
```

Calculate the return of a sample portfolio equally weighted in its components.

```
for j in range(100000):
    R=0
    for i in range(30):
        f1=student_t(3.5)
        f2=student_t(3.5)
        f3=student_t(3.5)
        f4=student_t(3.5)
        f5=student_t(3.5)
        g=student_t(3.5)
        R=R+(data1.iat[i,0]+data1.iat[i,1]*(data1.iat[i,2]*f1+data1.iat[i,3]*f2+data1.iat[i,4]*f3+
                data1.iat[i,5]*f4+data1.iat[i,6]*f5)+data1.iat[i,1]*data1.iat[i,7]*g)/30
    print(R)
```

```
>head(data)
0.000322897971623
0.00148725194129
0.000835025079128
0.00109671029008
0.000303957947664
```

Calculate 99% one-week VAR.

```
#VAR
1.-0.002972272
2.-0.003164219
3.-0.003061704
4.-0.002988024
5.-0.003248583
```

Appendix

1. The sample correlation matrix.

MMM	AXP	AAPL	BA	CAT
1.00037241	0.673988616	0.625408085	0.677596395	0.683502677
0.673988616	1.000247029	0.621581502	0.639048173	0.639393683
0.625408085	0.621581502	1.000400261	0.629210611	0.613332301
0.677596395	0.639048173	0.629210611	1.00037684	0.640840475
0.683502677	0.639393683	0.613332301	0.640840475	1.000696742
0.680557463	0.616118932	0.633893432	0.603414416	0.683964132
0.686572662	0.649366235	0.669688504	0.686604976	0.661475434
0.680516213	0.64065624	0.618096387	0.625668103	0.649902198
0.683625203	0.652520807	0.627823602	0.655643985	0.654199644
0.701233582	0.610431725	0.623891326	0.615566915	0.674865013
0.677338211	0.614823928	0.608548879	0.632367814	0.622150933
0.690000586	0.704299048	0.636708565	0.662055636	0.659444088
0.678258072	0.636297793	0.652254227	0.629058595	0.629604051
0.675883499	0.615117324	0.666988049	0.649534619	0.636304196
0.693207226	0.664437085	0.656697436	0.652004386	0.679576309
0.721744877	0.63743501	0.640626042	0.647432455	0.620065012
0.701155651	0.691281308	0.645901485	0.647250826	0.646636471
0.648442856	0.619212496	0.61867168	0.611376769	0.620568319
0.637417479	0.605079905	0.605673688	0.585781308	0.593134896
0.694062854	0.650895927	0.701700045	0.651367664	0.619318046
0.656985288	0.647719346	0.63173097	0.643586827	0.646969291
0.665216738	0.619414219	0.620718474	0.610026415	0.616490792
0.697046766	0.609846954	0.636685215	0.626755821	0.632090076
0.710368591	0.64349075	0.63074017	0.637671163	0.621706814
0.67692929	0.619084443	0.632759878	0.647446897	0.621387877
0.738138954	0.66463655	0.659131205	0.680432804	0.68135649
0.661443411	0.646944975	0.645237971	0.623022529	0.637827073
0.674731972	0.669212413	0.662633498	0.635058996	0.616935336
0.625581417	0.587240832	0.578424224	0.615468842	0.601244904
0.691407933	0.663221653	0.649035079	0.670249347	0.618803196

CVX	CSCO	KO	DWDP	XOM
0.680557463	0.686572662	0.680516213	0.683625203	0.701233582
0.616118932	0.649366235	0.64065624	0.652520807	0.610431725
0.633893432	0.669688504	0.618096387	0.627823602	0.623891326
0.603414416	0.686604976	0.625668103	0.655643985	0.615566915
0.683964132	0.661475434	0.649902198	0.654199644	0.674865013

1.000761246	0.640004153	0.638448777	0.679255576	0.827753337
0.640004153	1.000667103	0.645417054	0.656474662	0.632605386
0.638448777	0.645417054	1.00064783	0.624844529	0.638151685
0.679255576	0.656474662	0.624844529	1.00035397	0.664121714
0.827753337	0.632605386	0.638151685	0.664121714	1.00052702
0.616388637	0.621382298	0.635717617	0.645239954	0.605840919
0.638078077	0.679099193	0.653112355	0.667094252	0.645000404
0.647460291	0.654782244	0.641031289	0.654344279	0.629132397
0.637094312	0.676006647	0.644782795	0.633588405	0.634572194
0.65366478	0.70217026	0.653394087	0.653925366	0.655835558
0.645552491	0.667721958	0.665852307	0.663465273	0.650064088
0.65044339	0.672649923	0.647581683	0.671788501	0.657867052
0.630194229	0.635639027	0.653422209	0.626508581	0.624025536
0.622048909	0.635223474	0.586792911	0.606228504	0.605116772
0.655135217	0.704750371	0.638650989	0.643216733	0.633902034
0.637584042	0.680862301	0.634626454	0.635917148	0.651357849
0.650386664	0.639913839	0.63479424	0.636683855	0.646944909
0.631382335	0.654476385	0.72126452	0.630320171	0.640789166
0.627809383	0.665455363	0.664904416	0.635827217	0.629615386
0.623068556	0.657172319	0.648011979	0.632067293	0.635934556
0.650085567	0.695509501	0.650887479	0.672856263	0.651670137
0.637680351	0.666573365	0.673656376	0.653554662	0.650072336
0.628460241	0.663919737	0.63181248	0.642948346	0.613230642
0.612909938	0.612928334	0.614015492	0.604810118	0.610004422
0.632995505	0.691434522	0.642777504	0.670640026	0.637259211

GE	GS	HD	INTC	IBM
0.677338211	0.690000586	0.678258072	0.675883499	0.693207226
0.614823928	0.704299048	0.636297793	0.615117324	0.664437085
0.608548879	0.636708565	0.652254227	0.666988049	0.656697436
0.632367814	0.662055636	0.629058595	0.649534619	0.652004386
0.622150933	0.659444088	0.629604051	0.636304196	0.679576309
0.616388637	0.638078077	0.647460291	0.637094312	0.65366478
0.621382298	0.679099193	0.654782244	0.676006647	0.70217026
0.635717617	0.653112355	0.641031289	0.644782795	0.653394087
0.645239954	0.667094252	0.654344279	0.633588405	0.653925366
0.605840919	0.645000404	0.629132397	0.634572194	0.655835558
1.000009536	0.629807295	0.634005334	0.6166227	0.626304862
0.629807295	1.000624595	0.665021722	0.656969295	0.678937667
0.634005334	0.665021722	1.00034129	0.648428217	0.644652725
0.6166227	0.656969295	0.648428217	1.000768987	0.654072559
0.626304862	0.678937667	0.644652725	0.654072559	1.000446933
0.645392056	0.667431948	0.655815473	0.663244867	0.663929188

0.64945726	0.80454391	0.651616428	0.642295649	0.671720542
0.617025046	0.644284187	0.6713242	0.628078018	0.646069718
0.600873007	0.615975393	0.601265904	0.610584014	0.639214926
0.624979891	0.651857164	0.651654668	0.7067459	0.678573078
0.633529289	0.674201915	0.699183974	0.631144444	0.652394173
0.60714982	0.647306135	0.643754344	0.632649588	0.625138048
0.635713274	0.656266028	0.658885259	0.617649841	0.671374615
0.635297451	0.678464391	0.643837225	0.64153448	0.673687707
0.61790738	0.654253926	0.641376996	0.633409975	0.647474263
0.671882213	0.682815097	0.670834523	0.661010531	0.695418059
0.625872053	0.660413204	0.652673638	0.62889879	0.633836431
0.633023844	0.659272793	0.681490623	0.654926257	0.641063202
0.578242	0.596244439	0.634183009	0.627739339	0.61136855
0.64357189	0.67734772	0.663526619	0.651760213	0.637558288

JNJ	JPM	MCD	MRK	MSFT
0.721744877	0.701155651	0.648442856	0.637417479	0.694062854
0.63743501	0.691281308	0.619212496	0.605079905	0.650895927
0.640626042	0.645901485	0.61867168	0.605673688	0.701700045
0.647432455	0.647250826	0.611376769	0.585781308	0.651367664
0.620065012	0.646636471	0.620568319	0.593134896	0.619318046
0.645552491	0.65044339	0.630194229	0.622048909	0.655135217
0.667721958	0.672649923	0.635639027	0.635223474	0.704750371
0.665852307	0.647581683	0.653422209	0.586792911	0.638650989
0.663465273	0.671788501	0.626508581	0.606228504	0.643216733
0.650064088	0.657867052	0.624025536	0.605116772	0.633902034
0.645392056	0.64945726	0.617025046	0.600873007	0.624979891
0.667431948	0.80454391	0.644284187	0.615975393	0.651857164
0.655815473	0.651616428	0.6713242	0.601265904	0.651654668
0.663244867	0.642295649	0.628078018	0.610584014	0.7067459
0.663929188	0.671720542	0.646069718	0.639214926	0.678573078
0.999455504	0.686225364	0.638225169	0.673479506	0.653525498
0.686225364	0.999738566	0.630243192	0.629612592	0.669037374
0.638225169	0.630243192	1.000319906	0.588572882	0.636915249
0.673479506	0.629612592	0.588572882	0.999923927	0.626243252
0.653525498	0.669037374	0.636915249	0.626243252	1.000769854
0.64948248	0.651945206	0.644518902	0.630636274	0.64545465
0.703284877	0.654663152	0.624625471	0.687199053	0.643466778
0.695216952	0.660227561	0.648121715	0.624120463	0.666977276
0.643957927	0.670153764	0.643336745	0.585262617	0.647371931
0.664583456	0.652654109	0.636862405	0.62420327	0.649928756
0.668406113	0.678952458	0.627138507	0.609411761	0.674144112
0.650426243	0.660753209	0.624863255	0.593570335	0.637311618

0.668353911	0.673317543	0.631832448	0.606970801	0.684179008
0.618585168	0.597141052	0.621388085	0.557810199	0.588013564
0.668628684	0.684314572	0.646337161	0.635940282	0.680130818

NKE	PFE	PG	TRV	UNH
0.656985288	0.665216738	0.697046766	0.710368591	0.67692929
0.647719346	0.619414219	0.609846954	0.64349075	0.619084443
0.63173097	0.620718474	0.636685215	0.63074017	0.632759878
0.643586827	0.610026415	0.626755821	0.637671163	0.647446897
0.646969291	0.616490792	0.632090076	0.621706814	0.621387877
0.637584042	0.650386664	0.631382335	0.627809383	0.623068556
0.680862301	0.639913839	0.654476385	0.665455363	0.657172319
0.634626454	0.63479424	0.72126452	0.664904416	0.648011979
0.635917148	0.636683855	0.630320171	0.635827217	0.632067293
0.651357849	0.646944909	0.640789166	0.629615386	0.635934556
0.633529289	0.60714982	0.635713274	0.635297451	0.61790738
0.674201915	0.647306135	0.656266028	0.678464391	0.654253926
0.699183974	0.643754344	0.658885259	0.643837225	0.641376996
0.631144444	0.632649588	0.617649841	0.64153448	0.633409975
0.652394173	0.625138048	0.671374615	0.673687707	0.647474263
0.64948248	0.703284877	0.695216952	0.643957927	0.664583456
0.651945206	0.654663152	0.660227561	0.670153764	0.652654109
0.644518902	0.624625471	0.648121715	0.643336745	0.636862405
0.630636274	0.687199053	0.624120463	0.585262617	0.62420327
0.64545465	0.643466778	0.666977276	0.647371931	0.649928756
0.999875034	0.638238123	0.628616192	0.635099534	0.648570107
0.638238123	1.000783501	0.652974936	0.624527576	0.671556623
0.628616192	0.652974936	1.000463137	0.674353646	0.643614847
0.635099534	0.624527576	0.674353646	1.000624463	0.650837096
0.648570107	0.671556623	0.643614847	0.650837096	0.998898988
0.642491761	0.644598117	0.652988093	0.689723141	0.64122144
0.632160172	0.627555021	0.65889905	0.642246416	0.619892604
0.655670596	0.64698335	0.640800895	0.62937865	0.648182547
0.616123438	0.584017578	0.636893296	0.617872799	0.604044988
0.675758398	0.665135459	0.628869703	0.659361423	0.661499066

UTX	VZ	V	WMT	DIS
0.738138954	0.661443411	0.674731972	0.625581417	0.691407933
0.66463655	0.646944975	0.669212413	0.587240832	0.663221653
0.659131205	0.645237971	0.662633498	0.578424224	0.649035079
0.680432804	0.623022529	0.635058996	0.615468842	0.670249347
0.68135649	0.637827073	0.616935336	0.601244904	0.618803196
0.650085567	0.637680351	0.628460241	0.612909938	0.632995505

0.695509501	0.666573365	0.663919737	0.612928334	0.691434522
0.650887479	0.673656376	0.63181248	0.614015492	0.642777504
0.672856263	0.653554662	0.642948346	0.604810118	0.670640026
0.651670137	0.650072336	0.613230642	0.610004422	0.637259211
0.671882213	0.625872053	0.633023844	0.578242	0.64357189
0.682815097	0.660413204	0.659272793	0.596244439	0.67734772
0.670834523	0.652673638	0.681490623	0.634183009	0.663526619
0.661010531	0.62889879	0.654926257	0.627739339	0.651760213
0.695418059	0.633836431	0.641063202	0.61136855	0.637558288
0.668406113	0.650426243	0.668353911	0.618585168	0.668628684
0.678952458	0.660753209	0.673317543	0.597141052	0.684314572
0.627138507	0.624863255	0.631832448	0.621388085	0.646337161
0.609411761	0.593570335	0.606970801	0.557810199	0.635940282
0.674144112	0.637311618	0.684179008	0.588013564	0.680130818
0.642491761	0.632160172	0.655670596	0.616123438	0.675758398
0.644598117	0.627555021	0.64698335	0.584017578	0.665135459
0.652988093	0.65889905	0.640800895	0.636893296	0.628869703
0.689723141	0.642246416	0.62937865	0.617872799	0.659361423
0.64122144	0.619892604	0.648182547	0.604044988	0.661499066
1.000315754	0.639165446	0.672081366	0.628604488	0.682302711
0.639165446	0.999900242	0.62998166	0.592248028	0.648884677
0.672081366	0.62998166	1.000421124	0.567880436	0.680857921
0.628604488	0.592248028	0.567880436	1.000215539	0.593599837
0.682302711	0.648884677	0.680857921	0.593599837	1.000480777

2. Run a regression with the 5 factors and obtain the parameters β sk.

Stock	Beta1	Beta2	Beta3	Beta4	Beta5
MMM	-0.22699	-0.02787	-0.07007	0.00602	0.05990
AXP	-0.17508	0.17640	-0.09083	-0.15193	0.15586
AAPL	-0.16644	0.10390	0.06186	-0.02504	-0.32225
BA	-0.17221	0.10793	-0.15768	-0.02944	-0.02982
CAT	-0.16673	-0.20605	-0.16168	-0.17382	-0.06215
CVX	-0.18489	-0.44166	0.03149	-0.14671	-0.06677
CSCO	-0.200419	0.096230	0.004097	-0.044199	-0.193292
KO	-0.17711	-0.07897	-0.09864	0.28056	0.08743
DWDP	-0.178950	-0.053710	-0.046254	-0.162089	0.009626
XOM	-0.18826	-0.44773	0.03276	-0.12775	-0.02873
GE	-0.186856	-0.001456	-0.073220	-0.024511	0.096689
GS	-0.17553	0.15098	-0.09869	-0.23502	0.20222
HG	-0.19086	0.07824	-0.03822	0.10068	-0.06976
INTC	-0.18717	0.03594	0.03116	-0.02182	-0.27799
IBM	-0.18972	-0.03837	-0.07082	-0.03017	-0.07833
JNJ	-0.20637	-0.02389	0.20663	0.10844	0.11101

JPM	-0.19026	0.11293	-0.07015	-0.20785	0.21664
MCD	-0.175378	-0.003837	-0.036195	0.206470	-0.013799
MRK	-0.16414	-0.01428	0.41075	-0.04511	0.11552
MSFT	-0.19472	0.10283	0.05637	-0.00157	-0.27741
NKE	-0.16443	0.12098	0.04781	0.03754	-0.04629
PFE	-0.178779	0.004865	0.380417	0.016251	0.129579
PG	-0.192451	-0.090280	0.003169	0.275773	0.102399
TRV	-0.18425	0.05884	-0.17944	0.10311	0.15558
UNH	-0.16256	0.09584	0.13652	0.03772	0.07615
YTX	-0.20264	0.04369	-0.13376	-0.03843	-0.02119
VZ	-0.15988	-0.09807	-0.07155	0.08054	0.07232
V	-0.18989	0.17129	0.09570	-0.03055	-0.07830
WMT	-0.16080	-0.07478	-0.12140	0.32818	-0.02552
DIS	-0.1830484	0.1229401	0.0375484	-0.0551090	-0.0001604