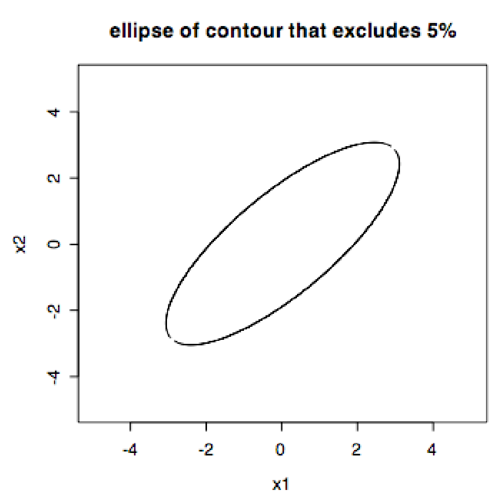
Today:

1. Computing those Bivariate Contour Ellipses
2. More about principal components:

1. Computing those Bivariate Contour Ellipses

One way: Start with a (discretized circle)
Expand using eigenvalues of sample covariance matrix
Rotate according to angle of first eigenvalue

```
contour
function (x=mvrnorm(n=100,mu=c(0,0),Sigma=matrix(c(1,0.5,0.5,1),ncol=2)))
{
        eigen=eigen(cov(x))
        z1=(-1000:1000)/400   #fewer points if gap OK
        z2=(5.99-z1^2)^.5     #circle
        z1=z1*eigen$values[1] # angle is for 1st PC
        z2=z2*eigen$values[2]
        angle=pi/2+atan(eigen$vectors[,1][2]/eigen$vectors[,1][1])
        x1=mean(x[,1])+z1*cos(angle)+z2*sin(angle) # see text p 35
        x2=mean(x[,2])-z1*sin(angle)+z2*cos(angle)
        plot(x1,x2,ylim=c(mean(x[,2])-5,mean(x[,2])+5),xlim=c(mean(x[,1])-
5,mean(x[,1])+5),type="l",main="ellipse of contour that excludes 5%" )
        z2=-z2            # to get rest of ellipse
        x1=mean(x[,1])+z1*cos(angle)+z2*sin(angle)
        x2=mean(x[,2])-z1*sin(angle)+z2*cos(angle)
        lines(x1,x2,type="l")
}
```



2. More about principal components:

Data Matrix x (n x p)

eigenvalues of cor(x) (or cov(x)) $\lambda_1 > \lambda_2 > ... > \lambda_p$ (actually all estimates)

ith eigenvector of x, e_i corresponding to λ is p x 1 (column vector)

Suppose jth row of x is x_j' which is 1 by p (row vector)

Compute ith Principal Component using $x_j' e_i$ (a scalar)

Compute vector of principal components for row x_j' by $x_j' e$

where e is the p x p matrix of column vectors that are eigenvectors.

x e is the n x p matrix of PCs – each row is a p-tuple of PCs describing one case.

Note: $x_j' e_i$ and $e_i' x_j$ are the same scalar quantity. (book uses e x)

An example:

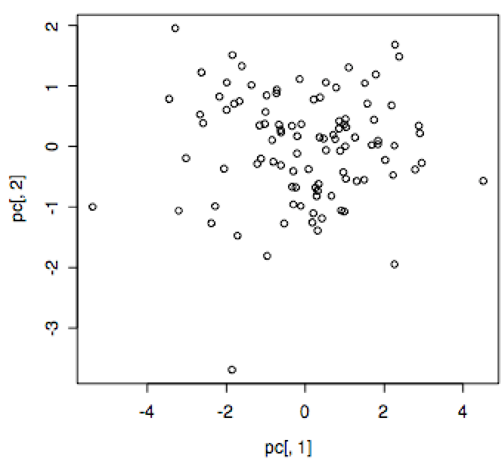
Weekly rates of return for 5 US stocks: Example 8.5 p 447

```
> T8.5.df=read.table("T8.4.txt")
> cor=cor(T8.5.df)
> eigen=eigen(cor)
> eigen
$values
[1] 2.8564869 0.8091185 0.5400440 0.4513468 0.3430038

$vectors
           [,1]       [,2]       [,3]       [,4]
[1,] -0.4635405  0.2408499  0.6133570 -0.3813727
[2,] -0.4570764  0.5090997 -0.1778996 -0.2113068
[3,] -0.4699804  0.2605774 -0.3370355  0.6640985
[4,] -0.4216770 -0.5252647 -0.5390181 -0.4728036
[5,] -0.4213291 -0.5822416  0.4336029  0.3812273
           [,5]
[1,] -0.4532876
[2,]  0.6749814
[3,] -0.3957247
[4,] -0.1794482
[5,]  0.3874672

> z=scale(T8.5.df)
```

```
> dim(z)
[1] 100   5
> 2.856/5
[1] 0.5712
> .809/5
[1] 0.1618
> (2.856+.809)/5
[1] 0.733
> mode(z)
[1] "numeric"
> pc=z%*%as.matrix(eigen$vectors)
> plot(pc[,1],pc[,2])
```



Can the dimensions be described? pc1 is like mean. pc2 is chem – oil.

Actually sequence may be of interest here. How can we label points?

sequence, size, colour?

```
week=as.character(rep(1:10,each=10))
> week
  [1] "1"  "1"  "1"  "1"  "1"  "1"  "1"  "1"  "1"  "1"
 [11] "2"  "2"  "2"  "2"  "2"  "2"  "2"  "2"  "2"  "2"
 [21] "3"  "3"  "3"  "3"  "3"  "3"  "3"  "3"  "3"  "3"
 [31] "4"  "4"  "4"  "4"  "4"  "4"  "4"  "4"  "4"  "4"
 [41] "5"  "5"  "5"  "5"  "5"  "5"  "5"  "5"  "5"  "5"
 [51] "6"  "6"  "6"  "6"  "6"  "6"  "6"  "6"  "6"  "6"
 [61] "7"  "7"  "7"  "7"  "7"  "7"  "7"  "7"  "7"  "7"
 [71] "8"  "8"  "8"  "8"  "8"  "8"  "8"  "8"  "8"  "8"
 [81] "9"  "9"  "9"  "9"  "9"  "9"  "9"  "9"  "9"  "9"
 [91] "10" "10" "10" "10" "10" "10" "10" "10" "10" "10"
```

```
> plot(pc[,1],pc[,2],pch=week)
```