PRINCIPAL COMPONENTS ANALYSIS

Iris Data

Let's find Principal Components using the iris dataset. This is a well known dataset, often used to demonstrate the effect of clustering algorithms. It contains measurements for 150 iris flowers on 4 features:

1. Sepal.Length
2. Sepal.Width
3. Petal.Length
4. Petal.Width

The fifth variable in the dataset tells us what species the flower is. There are 3 species:

5. Species
    1. Setosa
    2. Versicolor
    3. Virginica

Let's first take a look at the scatterplot matrix:

```
> pairs(~Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
+     data = iris, col = c("red", "green3", "blue")[iris$Species])
```

It is apparent that some of our variables are correlated. We can confirm this by computing the correlation matrix (**cor** function). We can also check out the individual variances of the variables and the covariances between variables by examining the covariance matrix (**cov** function). Remember - when looking at covariances, we can really only interpret the sign of the number and not the magnitude as we can with the correlations.

```
> cor(iris[1:4])
```

```
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

```
> cov(iris[1:4])
```

```
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    0.6856935  -0.0424340    1.2743154   0.5162707
Sepal.Width    -0.0424340   0.1899794   -0.3296564  -0.1216394
Petal.Length    1.2743154  -0.3296564    3.1162779   1.2956094
Petal.Width     0.5162707  -0.1216394    1.2956094   0.5810063
```

We have relatively strong positive correlation between Petal Length, Petal Width and Sepal Length. It is also clear that Petal Length has more than 3 times the variance of the other 3 variables. How will this effect our analysis?

The scatter plots and correlation matrix provide useful information, but they don't give us a true sense for how the data looks when all 4 attributes are considered simultaneously.

We will compute the principal components, using both the covariance matrix and the correlation matrix, and see what we can learn about the data. Let's start with the covariance matrix which is the default setting in R.

Iris Data: PCA on the Covariance Matrix

Principal Components, Loadings, and Variance Explained

```
> covM = cov(iris[1:4])
> eig=eigen(covM,symmetric=TRUE,only.values=FALSE)
> c=colnames(iris[1:4])
> eig$values
```

```
[1] 4.22824171 0.24267075 0.07820950 0.02383509
```

```
> rownames(eig$vectors)=c(colnames(iris[1:4]))
> eig$vectors
```

```
                    [,1]        [,2]        [,3]       [,4]
Sepal.Length  0.36138659 -0.65658877 -0.58202985  0.3154872
Sepal.Width  -0.08452251 -0.73016143  0.59791083 -0.3197231
Petal.Length  0.85667061  0.17337266  0.07623608 -0.4798390
Petal.Width   0.35828920  0.07548102  0.54583143  0.7536574
```

The eigenvalues tell us how much of the total variance in the data is directed along each eigenvector. Thus, the amount of variance along $\mathbf{v}_1$ is λ_1 and the *proportion* of variance explained by the first principal component is

$$\frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4}$$

```
> eig$values[1]/sum(eig$values)
```

```
[1] 0.9246187
```

Thus 92% of the variation in the Iris data is explained by the first component alone. What if we consider the first and second principal component directions? Using this two dimensional representation (approximation/projection) we can capture the following proportion of variance:

$$\frac{\lambda_1 + \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4}$$

```
> sum(eig$values[1:2])/sum(eig$values)
```

```
[1] 0.9776852
```

With two dimensions, we explain 97.8% of the variance in these 4 variables! The entries in each eigenvector are called the **loadings** of the variables on the component. The loadings give us an idea how important each variable is to each component. For example, it seems that the third variable in our dataset (Petal Length) is dominating the first principal component. This should not come as too much of a shock - that variable had (by far) the largest amount of variation of the four. In order to capture the most amount of variance in a single dimension, we should certainly be considering this variable strongly. The variable with the next largest variance, Sepal Length, dominates the second principal component.

Note: *Had Petal Length and Sepal Length been correlated, they would not have dominated separate principal components, they would have shared one. These two variables are not correlated and thus their variation cannot be captured along the same direction.*

The PCA Projection i.e. Observation Scores

Lets plot the *projection* of the four-dimensional iris data onto the two dimensional space spanned by the first 2 principal components. To do this, we need coordinates. These coordinates are commonly called **scores** in statistical texts. We can find the coordinates of the data on the principal components by solving the system

$$\mathbf{X} = \mathbf{A}\mathbf{V}^T$$

where $\mathbf{X}$ is our original iris data **(centered to have mean = 0)** and $\mathbf{A}$ is a matrix of coordinates in the new principal component space, spanned by the eigenvectors in $\mathbf{V}$.

Solving this system is simple enough - since $\mathbf{V}$ is an orthogonal matrix. Let's confirm this:

```
> eig$vectors %*% t(eig$vectors)
```

```
              Sepal.Length  Sepal.Width  Petal.Length   Petal.Width
Sepal.Length  1.000000e+00 4.163336e-17 -2.775558e-17 -2.775558e-17
Sepal.Width   4.163336e-17 1.000000e+00  1.665335e-16  1.942890e-16
Petal.Length -2.775558e-17 1.665335e-16  1.000000e+00 -2.220446e-16
Petal.Width  -2.775558e-17 1.942890e-16 -2.220446e-16  1.000000e+00
```

```
> t(eig$vectors) %*% eig$vectors
```

```
              [,1]          [,2]         [,3]          [,4]
[1,]  1.000000e+00 -2.289835e-16 0.000000e+00 -1.110223e-16
[2,] -2.289835e-16  1.000000e+00 2.775558e-17 -1.318390e-16
[3,]  0.000000e+00  2.775558e-17 1.000000e+00  1.110223e-16
[4,] -1.110223e-16 -1.318390e-16 1.110223e-16  1.000000e+00
```

We'll have to settle for precision at 15 decimal places. Close enough!

So to find the loadings, we simply subtract the means from our original variables to create the data matrix $\mathbf{X}$ and compute

$$\mathbf{A} = \mathbf{XV}$$

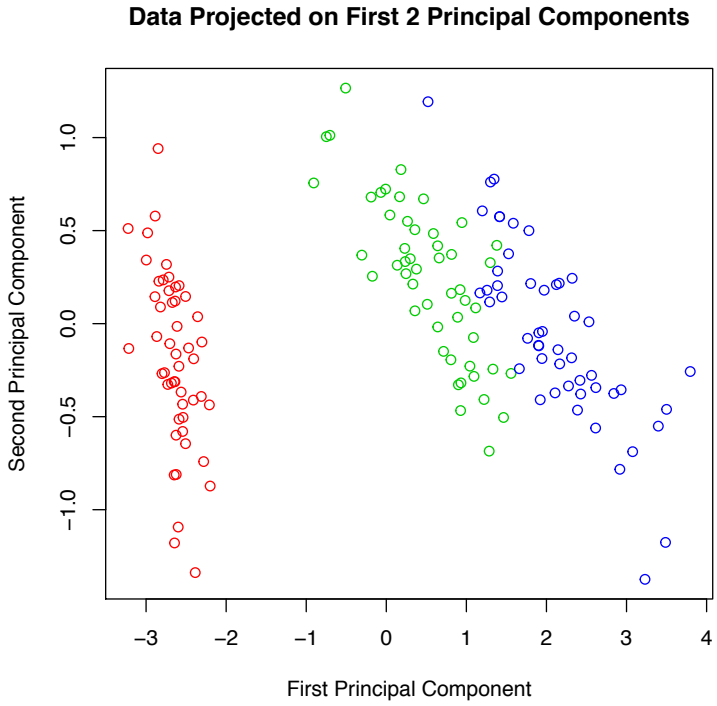
```
> X = scale(iris[1:4], center = TRUE, scale = FALSE)
> scores = data.frame(X %*% eig$vectors)
> colnames(scores) = c("Prin1", "Prin2", "Prin3", "Prin4")
> scores[1:10, ]
```

```
       Prin1       Prin2       Prin3        Prin4
1  -2.684126 -0.31939725 -0.02791483  0.002262437
2  -2.714142  0.17700123 -0.21046427  0.099026550
3  -2.888991  0.14494943  0.01790026  0.019968390
4  -2.745343  0.31829898  0.03155937 -0.075575817
5  -2.728717 -0.32675451  0.09007924 -0.061258593
6  -2.280860 -0.74133045  0.16867766 -0.024200858
7  -2.820538  0.08946138  0.25789216 -0.048143106
8  -2.626145 -0.16338496 -0.02187932 -0.045297871
9  -2.886383  0.57831175  0.02075957 -0.026744736
10 -2.672756  0.11377425 -0.19763272 -0.056295401
```

To this point, we have simply computed coordinates (scores) on a new set of axis (principal components, eigenvectors, loadings). These axis are orthogonal and are aligned with the directions of maximal variance in the data. When we consider only a subset of principal components (like 2 components accounting for 97% of the variance), then we are projecting the data onto a lower dimensional space. Generally, this is one of the primary goals of PCA: Project the data down into a lower dimensional space (*onto the span of the principal components*) while keeping the maximum amount of information (i.e. variance).

Thus, we know that almost 98% of the data's variance can be seen in two-dimensions using the first two principal components. Let's go ahead and see what this looks like:

```
> plot(scores$Prin1, scores$Prin2, main = "Data Projected on First 2 Principal Components",
+     xlab = "First Principal Component", ylab = "Second Principal Component",
+     col = c("red", "green3", "blue")[iris$Species])
```



Principal Components in R

```
> irispca=princomp(iris[1:4])
> # Variance Explained
> summary(irispca)
```

```
Importance of components:
                          Comp.1     Comp.2     Comp.3      Comp.4
Standard deviation     2.0494032 0.49097143 0.27872586 0.153870700
Proportion of Variance 0.9246187 0.05306648 0.01710261 0.005212184
Cumulative Proportion  0.9246187 0.97768521 0.99478782 1.000000000
```

```
> # Eigenvectors:
> irispca$loadings
```

```
Loadings:
             Comp.1 Comp.2 Comp.3 Comp.4
Sepal.Length  0.361 -0.657 -0.582  0.315
Sepal.Width         -0.730  0.598 -0.320
Petal.Length  0.857  0.173        -0.480
Petal.Width   0.358         0.546  0.754

               Comp.1 Comp.2 Comp.3 Comp.4
SS loadings      1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00
```

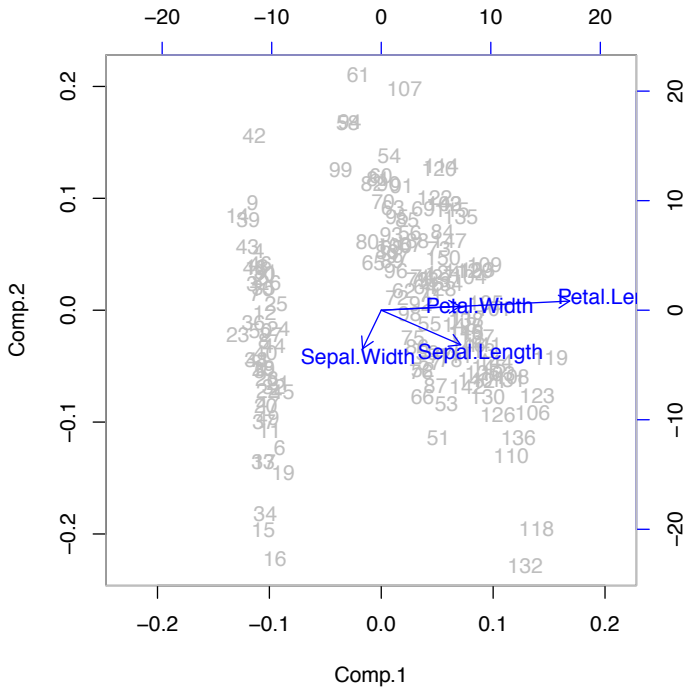
```
> # Coordinates of data along PCs:
> irispca$scores[1:10, ]
```

```
         Comp.1      Comp.2      Comp.3       Comp.4
 [1,] -2.684126 -0.31939725 -0.02791483  0.002262437
 [2,] -2.714142  0.17700123 -0.21046427  0.099026550
 [3,] -2.888991  0.14494943  0.01790026  0.019968390
 [4,] -2.745343  0.31829898  0.03155937 -0.075575817
 [5,] -2.728717 -0.32675451  0.09007924 -0.061258593
 [6,] -2.280860 -0.74133045  0.16867766 -0.024200858
 [7,] -2.820538  0.08946138  0.25789216 -0.048143106
 [8,] -2.626145 -0.16338496 -0.02187932 -0.045297871
 [9,] -2.886383  0.57831175  0.02075957 -0.026744736
[10,] -2.672756  0.11377425 -0.19763272 -0.056295401
```

```
> # You'll notice this is different from SAS output when the option cor=T is used on the princom
> # sqrt((n-1)/n) because when someone wrote this function they standardized data using populati
> # standard deviation.
```

All of the information we just computed is correct. One additional feature that R users have created is the **biplot**. The PCA biplot allows us to see where our original variables fall in the space of the principal components. Highly correlated variables will fall along the same direction (or exactly opposite directions) as a change in one of these variables correlates to a change in the other. Uncorrelated variables will appear further apart.

```
> biplot(irispca, col = c("gray", "blue"))
```



We can examine some of the outlying observations to see how they align with these projected variable directions. It helps to compare them to the quartiles of the data. Also keep in mind the direction of the arrows in the plot. If the arrow points down then the positive direction is down - indicating observations which are greater than the mean. Let's pick out observations 42 and 132 and see what the actual data points look like in comparison to the rest of the sample population.

```
> summary(iris[1:4])
```

```
 Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
Median :5.800   Median :3.000   Median :4.350   Median :1.300
Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

```
> # Consider orientation of outlying observations:
> iris[42, ]
```

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
42          4.5         2.3          1.3         0.3  setosa
```

```
> iris[132, ]
```

```
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
132          7.9         3.8          6.4           2 virginica
```

Variable Clustering with PCA

The direction arrows on the biplot are merely the coefficients of the original variables when combined to make principal components. Don't forget that principal components are simply linear combinations of the original variables.

For example, here we have the first principal component (the first column of $\mathbf{V}$), $\mathbf{v}_1$ as:

```
> eig$vectors[,1]
```

```
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
  0.36138659  -0.08452251   0.85667061   0.35828920
```

This means that

$$comp_1 = 0.36 Sepal.Length - 0.08 Sepal.Width + 0.85 Petal.Length + 0.35 Petal.Width$$

the same equation could be written for each of the principal components, $comp_1, \ldots, comp_4$.

Essentially, we have a system of equations telling us that the rows of $\mathbf{V}^T$ (i.e. the columns of $\mathbf{V}$) give us the weights of each variable for each principal component:

$$\begin{bmatrix} comp_1 \\ comp_2 \\ comp_3 \\ comp_4 \end{bmatrix} = \mathbf{V}^T \begin{bmatrix} Sepal.Length \\ Sepal.Width \\ Petal.Length \\ Petal.Width \end{bmatrix}$$

Thus, if want the coordinates of our original variables in terms of Principal Components (so that we can plot them as we do in the biplot) we need to look no further than the rows of the matrix $\mathbf{V}$ as

$$\begin{bmatrix} Sepal.Length \\ Sepal.Width \\ Petal.Length \\ Petal.Width \end{bmatrix} = \mathbf{V} \begin{bmatrix} comp_1 \\ comp_2 \\ comp_3 \\ comp_4 \end{bmatrix}$$

means that the rows of $\mathbf{V}$ give us the coordinates of our original variables in the PCA space.

```
> #First entry in each eigenvectors give coefficients for Variable 1:
> eig$vectors[1,]
```

```
[1]  0.3613866 -0.6565888 -0.5820299  0.3154872
```

$$Sepal.Length = 0.361comp_1 - 0.657comp_2 - 0.582comp_3 + 0.315comp_4$$

You can see this on the biplot. The vector shown for Sepal.Length is (0.361, -0.656), which is the two dimensional projection formed by throwing out components 3 and 4.

Variables which lie upon similar directions in the PCA space tend to change in a similar fashion. We'd consider Petal.Width and Petal.Length as a cluster of variables. It does not appear that we need both in our model.

Comparison with PCA on the Correlation Matrix

We can complete the same analysis using the correlation matrix. I'll leave it as an exercise to compute the Principal Component loadings and scores and variance explained directly from eigenvectors and eigenvalues. You should do this and compare your results to the R output. *(Beware: you must transform your data before solving for the scores. With the covariance version, this meant centering - for the correlation version, this means standardization as well)*

```
> irispca2 = princomp(iris[1:4], cor = TRUE)
> summary(irispca2)
```

```
Importance of components:
                          Comp.1    Comp.2     Comp.3      Comp.4
Standard deviation     1.7083611 0.9560494 0.38308860 0.143926497
Proportion of Variance 0.7296245 0.2285076 0.03668922 0.005178709
Cumulative Proportion  0.7296245 0.9581321 0.99482129 1.000000000
```

```
> irispca2$loadings
```

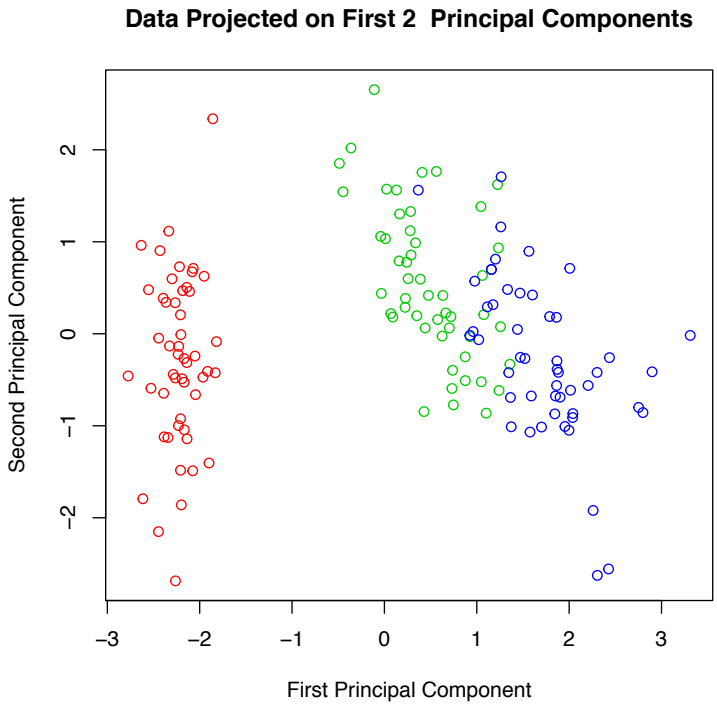
```
Loadings:
             Comp.1 Comp.2 Comp.3 Comp.4
Sepal.Length  0.521 -0.377  0.720  0.261
Sepal.Width  -0.269 -0.923 -0.244 -0.124
Petal.Length  0.580        -0.142 -0.801
Petal.Width   0.565        -0.634  0.524

               Comp.1 Comp.2 Comp.3 Comp.4
SS loadings      1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00
```

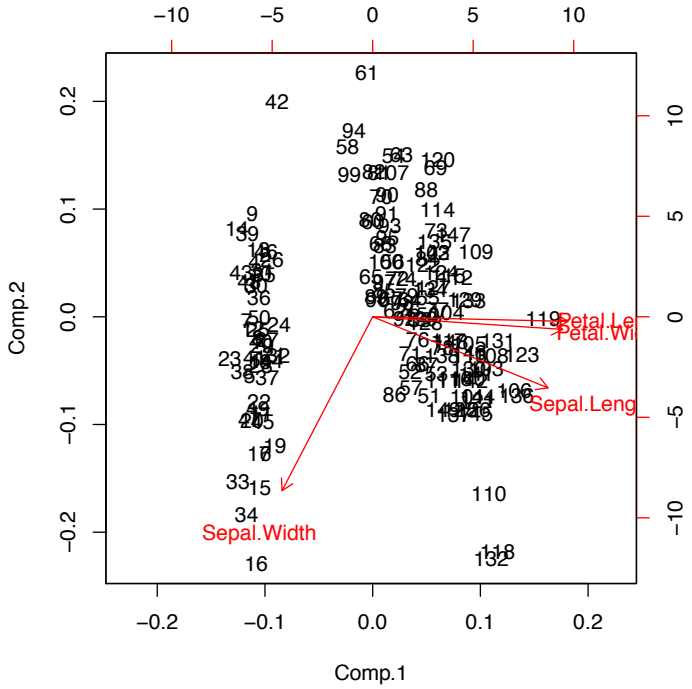
```
> irispca2$scores[1:10, ]
```

```
        Comp.1     Comp.2      Comp.3      Comp.4
 [1,] -2.264703 -0.4800266  0.12770602  0.02416820
 [2,] -2.080961  0.6741336  0.23460885  0.10300677
 [3,] -2.364229  0.3419080 -0.04420148  0.02837705
 [4,] -2.299384  0.5973945 -0.09129011 -0.06595556
 [5,] -2.389842 -0.6468354 -0.01573820 -0.03592281
 [6,] -2.075631 -1.4891775 -0.02696829  0.00660818
 [7,] -2.444029 -0.0476442 -0.33547040 -0.03677556
 [8,] -2.232847 -0.2231481  0.08869550 -0.02461210
 [9,] -2.334640  1.1153277 -0.14507686 -0.02685922
[10,] -2.184328  0.4690136  0.25376557 -0.03989929
```

```
> plot(irispca2$scores[, 1], irispca2$scores[, 2], main = "Data Projected on First 2  Principal
+     xlab = "First Principal Component", ylab = "Second Principal Component",
+     col = c("red", "green3", "blue")[iris$Species])
```




```
> biplot(irispca2)
```



Here you can see the direction vectors of the original variables are relatively uniform in length in the PCA space. This is due to the standardization in the correlation matrix. However, the general message is the same: Petal.Width and Petal.Length Cluster together, and many of the same observations appear "on the fray" on the PCA space - although not all of them!

Which Projection is Better?

What do you think? It depends on the task, for this data. The results in terms of variable clustering are pretty much the same. For clustering/-classifying the 3 species of flowers, we can see better separation in the Covariance version.

Beware of biplots

Be careful not to draw improper conclusions from biplots. Particularly, be careful about situations where the first two principal components do

not summarize the majority of the variance. If a large amount of variance is captured by the 3rd or 4th (or higher) principal components, then we must keep in mind that the variable projections on the first two principal components are flattened out versions of a higher dimensional picture. If a variable vector appears short in the 2-dimensional projection, it means one of two things:

- That variable has small variance
- That variable appears to have small variance when depicted in the space of the first two principal components, but truly has a larger variance which is represented by 3rd or higher principal components.

Let's take a look at an example of this. We'll generate 500 rows of data on 4 nearly independent normal random variables. Since these variables are uncorrelated, we might expect that the 4 orthogonal principal components will line up relatively close to the original variables. If this doesn't happen, then at the very least we can expect the biplot to show little to no correlation between the variables. We'll give variables 2 and 3 the largest variance. Multiple runs of this code will generate different results with similar implications.

```
> means=c(2,4,1,3)
> sigmas=c(7,9,10,8)
> sample.size=500
> data=mapply(function(mu,sig){rnorm(mu,sig, n=sample.size)},mu=means,sig=sigmas)
> cor(data)
```

```
            [,1]        [,2]        [,3]       [,4]
[1,]  1.00000000 -0.06572818 -0.02442179 0.10800380
[2,] -0.06572818  1.00000000  0.04127398 0.02682223
[3,] -0.02442179  0.04127398  1.00000000 0.04358849
[4,]  0.10800380  0.02682223  0.04358849 1.00000000
```

```
> pc=princomp(data,scale=TRUE)
> summary(pc)
```

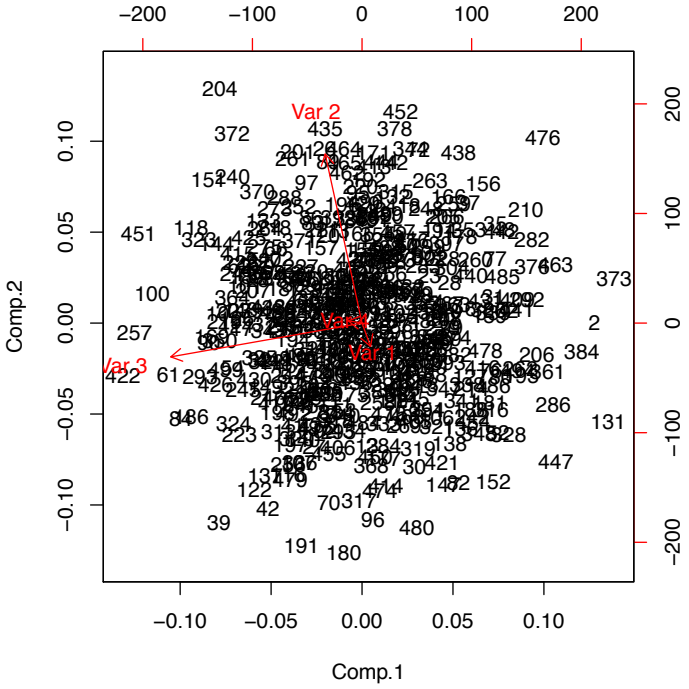
```
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4
Standard deviation     9.9687318 8.8742441 7.6430119 6.7925399
Proportion of Variance 0.3515455 0.2785893 0.2066478 0.1632173
Cumulative Proportion  0.3515455 0.6301349 0.8367827 1.0000000
```

```
> pc$loadings
```

```
Loadings:
     Comp.1 Comp.2 Comp.3 Comp.4
[1,]        -0.134 -0.612  0.779
[2,] -0.188  0.972         0.126
[3,] -0.979 -0.194
[4,]               -0.787 -0.612

               Comp.1 Comp.2 Comp.3 Comp.4
SS loadings      1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00
```

```
> biplot(pc)
```



Obviously, the wrong conclusion to make from this biplot is that Variables 1 and 4 are correlated. Variables 1 and 4 do not load highly on the first two principal components - in the *whole* 4-dimensional principal component

space they are nearly orthogonal to each other and to variables 1 and 2. Thus, their orthogonal projections appear near the origin of this 2-dimensional subspace.

The moral of the story: Always corroborate your results using the variable loadings and the amount of variation explained by each variable.

PCA as an SVD

Let's demonstrate the fact that PCA and SVD are equivalent by computing the SVD of the centered iris data:

```
> X=scale(iris[,1:4],center=TRUE,scale=FALSE)
> irisSVD=svd(X)
> u=irisSVD$u
> d=diag(irisSVD$d)
> SVDpcs=irisSVD$v
> SVDscores=u%*%d
> irisPCA=princomp(iris[,1:4])
> irisPCA$scores[1:8,]
```

```
       Comp.1      Comp.2      Comp.3       Comp.4
[1,] -2.684126 -0.31939725 -0.02791483  0.002262437
[2,] -2.714142  0.17700123 -0.21046427  0.099026550
[3,] -2.888991  0.14494943  0.01790026  0.019968390
[4,] -2.745343  0.31829898  0.03155937 -0.075575817
[5,] -2.728717 -0.32675451  0.09007924 -0.061258593
[6,] -2.280860 -0.74133045  0.16867766 -0.024200858
[7,] -2.820538  0.08946138  0.25789216 -0.048143106
[8,] -2.626145 -0.16338496 -0.02187932 -0.045297871
```

```
> SVDscores[1:8,]
```

```
          [,1]        [,2]        [,3]         [,4]
[1,] -2.684126 -0.31939725  0.02791483  0.002262437
[2,] -2.714142  0.17700123  0.21046427  0.099026550
[3,] -2.888991  0.14494943 -0.01790026  0.019968390
[4,] -2.745343  0.31829898 -0.03155937 -0.075575817
[5,] -2.728717 -0.32675451 -0.09007924 -0.061258593
[6,] -2.280860 -0.74133045 -0.16867766 -0.024200858
[7,] -2.820538  0.08946138 -0.25789216 -0.048143106
[8,] -2.626145 -0.16338496  0.02187932 -0.045297871
```

```
> irisPCA$loadings
```

```
Loadings:
             Comp.1 Comp.2 Comp.3 Comp.4
Sepal.Length  0.361 -0.657 -0.582  0.315
Sepal.Width         -0.730  0.598 -0.320
Petal.Length  0.857  0.173        -0.480
Petal.Width   0.358         0.546  0.754

               Comp.1 Comp.2 Comp.3 Comp.4
SS loadings      1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00
```

```
> SVDpcs
```

```
            [,1]        [,2]        [,3]       [,4]
[1,]  0.36138659 -0.65658877  0.58202985  0.3154872
[2,] -0.08452251 -0.73016143 -0.59791083 -0.3197231
[3,]  0.85667061  0.17337266 -0.07623608 -0.4798390
[4,]  0.35828920  0.07548102 -0.54583143  0.7536574
```