Birkbeck
UNIVERSITY OF LONDON

Big Data Analytics

Session 10

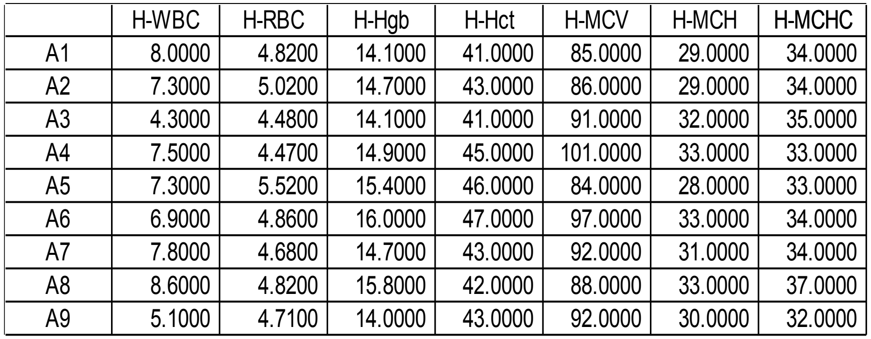
Dimension Reduction

Unsupervised Learning

- Mainly used to support the selection of appropriate statistical learning tools and techniques
  - Clustering
    - K Means Clustering
    - Hierarchical Clustering
  - Principal Component Analysis

Data Presentation

- Example: 7 Blood and urine measurements (wet chemistry) from 9 people (4 alcoholics, 5 non-alcoholics).
- How to present the data visually?



	H-WBC	H-RBC	H-Hgb	H-Hct	H-MCV	H-MCH	H-MCHC
A1	8.0000	4.8200	14.1000	41.0000	85.0000	29.0000	34.0000
A2	7.3000	5.0200	14.7000	43.0000	86.0000	29.0000	34.0000
A3	4.3000	4.4800	14.1000	41.0000	91.0000	32.0000	35.0000
A4	7.5000	4.4700	14.9000	45.0000	101.0000	33.0000	33.0000
A5	7.3000	5.5200	15.4000	46.0000	84.0000	28.0000	33.0000
A6	6.9000	4.8600	16.0000	47.0000	97.0000	33.0000	34.0000
A7	7.8000	4.6800	14.7000	43.0000	92.0000	31.0000	34.0000
A8	8.6000	4.8200	15.8000	42.0000	88.0000	33.0000	37.0000
A9	5.1000	4.7100	14.0000	43.0000	92.0000	30.0000	32.0000

- We could plot the two-dimensional scatterplots of the data, each of which contains the 9 observations' results on two of the measurements.
- How many scatterplots are there?
- What problems do you see here?
  1. If the number of features is high, → not possible to look at them all.
  2. Most likely none of them are informative → each containing a small fraction of information in the data set.

PCA

- **Principal components analysis (PCA)** is a technique that can be used to simplify a dataset
  - The principal components are computed
  - Use those components to understand the data

- PCA is an unsupervised approach
  - Involves only a set of features X_1, X_2, …, X_p , and no associated response Y.

- PCA can be used for
  - Reducing dimensionality (the number of features)
    - to produce derived variables for use in supervised learning
  - A tool for data visualisation

PCA Toy Example

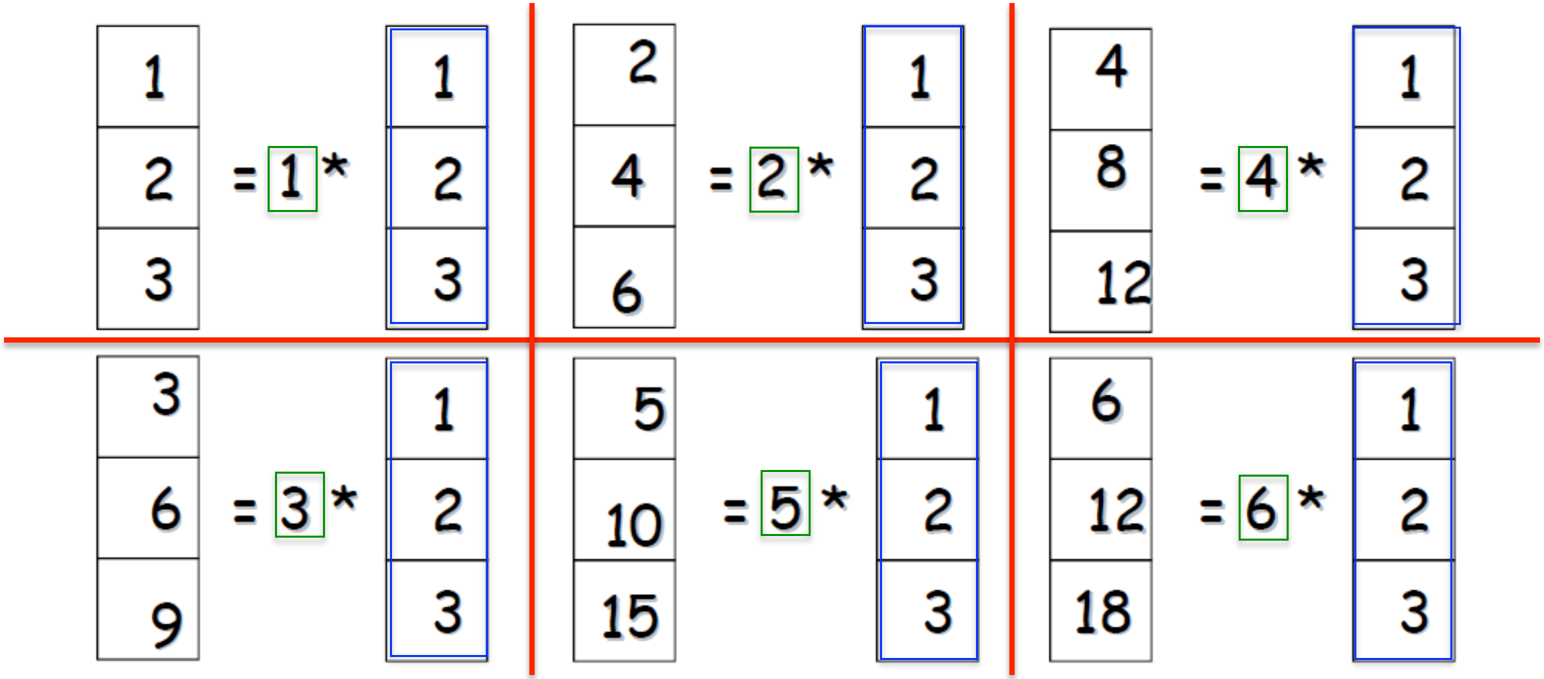
- Consider the following six 3D points

1	2	4	3	5	6
2	4	8	6	10	12
3	6	12	9	15	18

- If each component is stored in a byte, we need 18=3*6 bytes

PCA Toy Example

- Looking closer, we can see that all the points are related geometrically: they are all the same point, scaled by a factor:



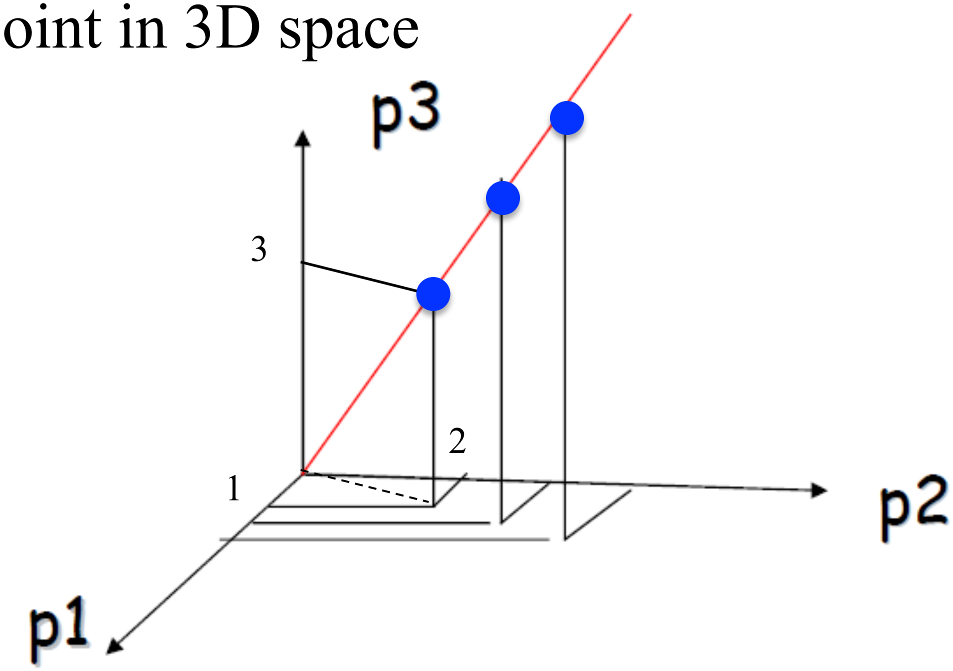
$$\begin{bmatrix}1\\2\\3\end{bmatrix} = \boxed{1} * \begin{bmatrix}1\\2\\3\end{bmatrix} \qquad \begin{bmatrix}2\\4\\6\end{bmatrix} = \boxed{2} * \begin{bmatrix}1\\2\\3\end{bmatrix} \qquad \begin{bmatrix}4\\8\\12\end{bmatrix} = \boxed{4} * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

$$\begin{bmatrix}3\\6\\9\end{bmatrix} = \boxed{3} * \begin{bmatrix}1\\2\\3\end{bmatrix} \qquad \begin{bmatrix}5\\10\\15\end{bmatrix} = \boxed{5} * \begin{bmatrix}1\\2\\3\end{bmatrix} \qquad \begin{bmatrix}6\\12\\18\end{bmatrix} = \boxed{6} * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

- They can be stored using only 9 bytes (50% savings!)
  - Store one point (3 bytes) + the multiplying constants (6 bytes)

Geometrical Interpretation

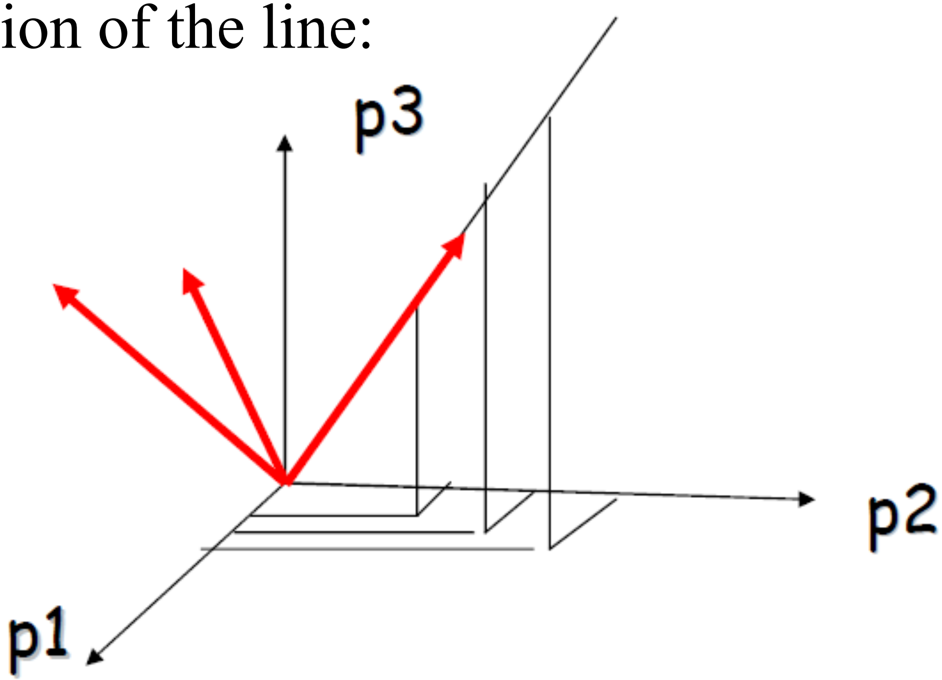
- View each point in 3D space



- But in this example, all the points happen to belong to a line
  - A 1D subspace of the original 3D space

Geometrical Interpretation

- Consider a new coordinate system where one of the axes is along the direction of the line:



- In this coordinate system, every point has only one non-zero coordinate: we only need to store
    - the direction of the line (a 3 byte image) and
    - the non-zero coordinate for each of the points (6 bytes)

Principal Component Analysis (PCA)

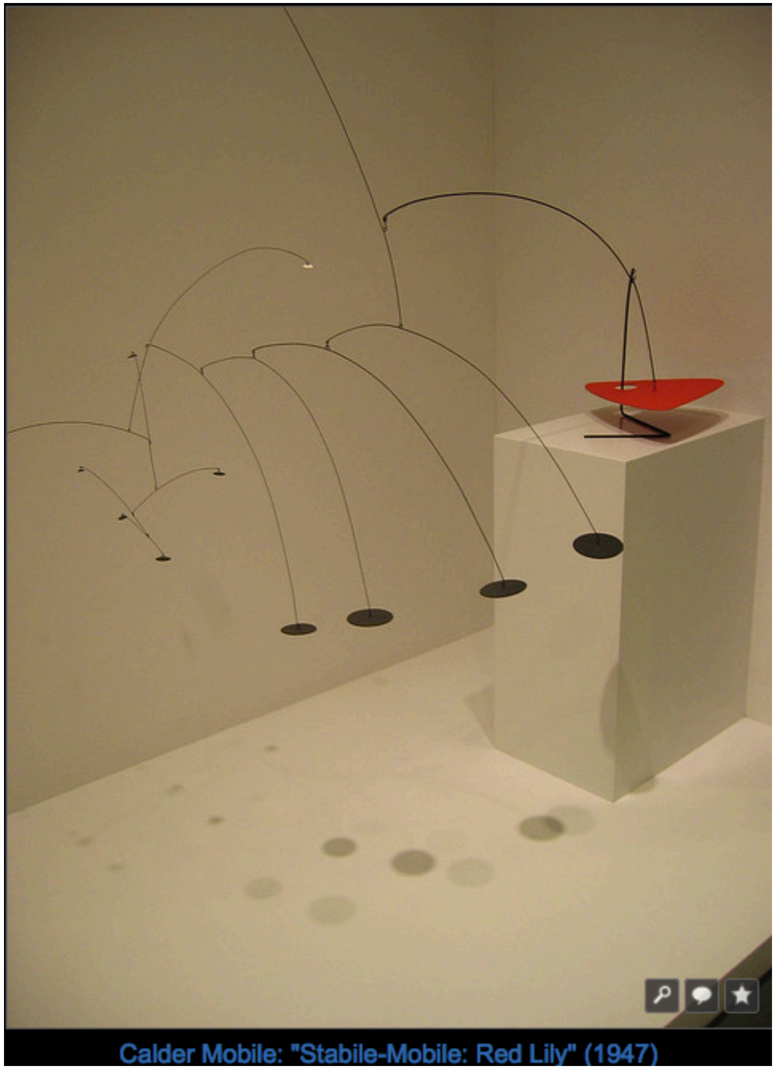
- Given a set of points, how do we know if they can be compressed like in the previous example?
  - The answer is to look into the covariance between the points
  - The tool for doing this is called PCA

  - PCA aims to fit straight lines to the data points. We call these straight lines "principal components".
  - There are as many principal components as there are variables.
  - The first principal component is the best straight line you can fit to the data.
  - The second principal component is the best straight line you can fit to the errors from the first principal component.
  - The third principal component is the best straight line you can fit to the errors from the first and second principal components, etc.

Some Intuitive Explanations

- Projection of a Calder Mobile

- Projection of a book



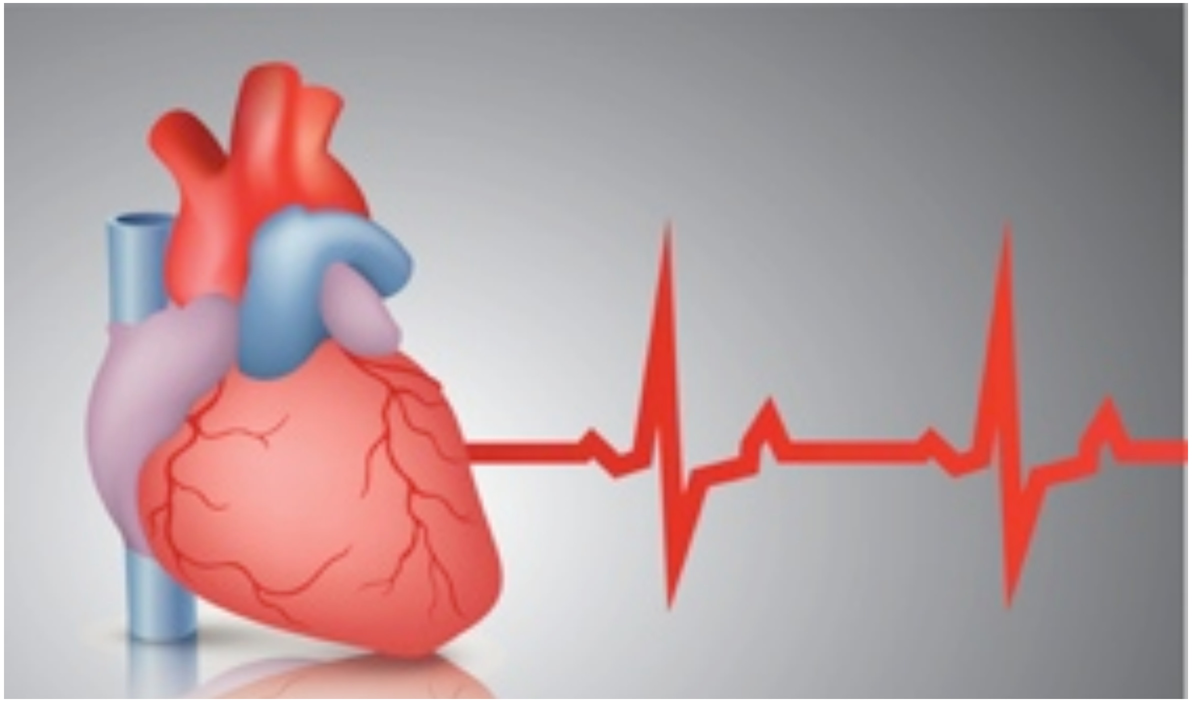
Calder Mobile: "Stabile-Mobile: Red Lily" (1947)

Some Intuitive Explanations

- Cider shop

Some Intuitive Explanations

- Heart attack risk factors (height and weight or ?)



$$\text{BMI} = \frac{(\text{weight in kilograms})}{(\text{height in centimeters})^2}$$

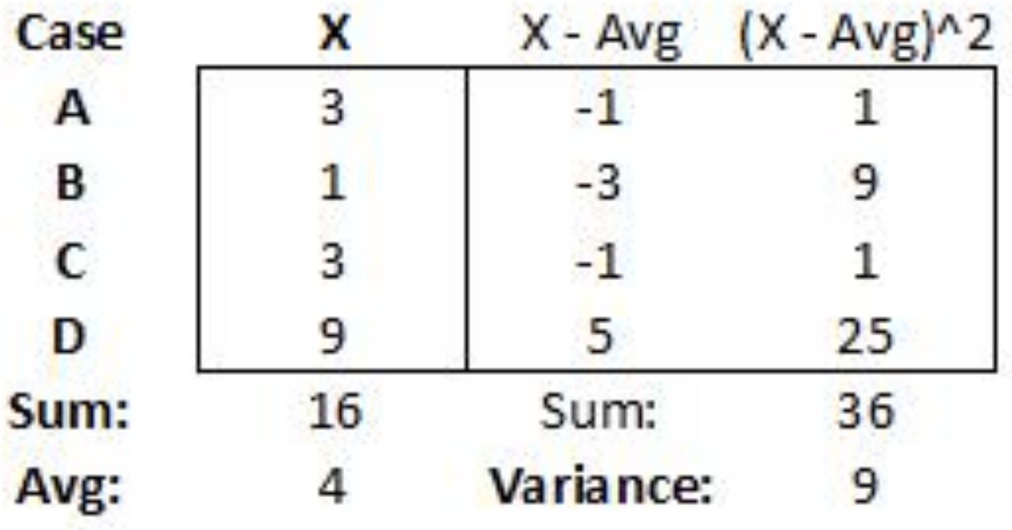
Some Background Mathematics

- Standard Deviation, Variance and Covariance

- Vectors and Matrices
- Covariance Matrix
- Eigenvectors and Eigenvalues

Variance and Standard Deviation

- First recall: variance of one variable



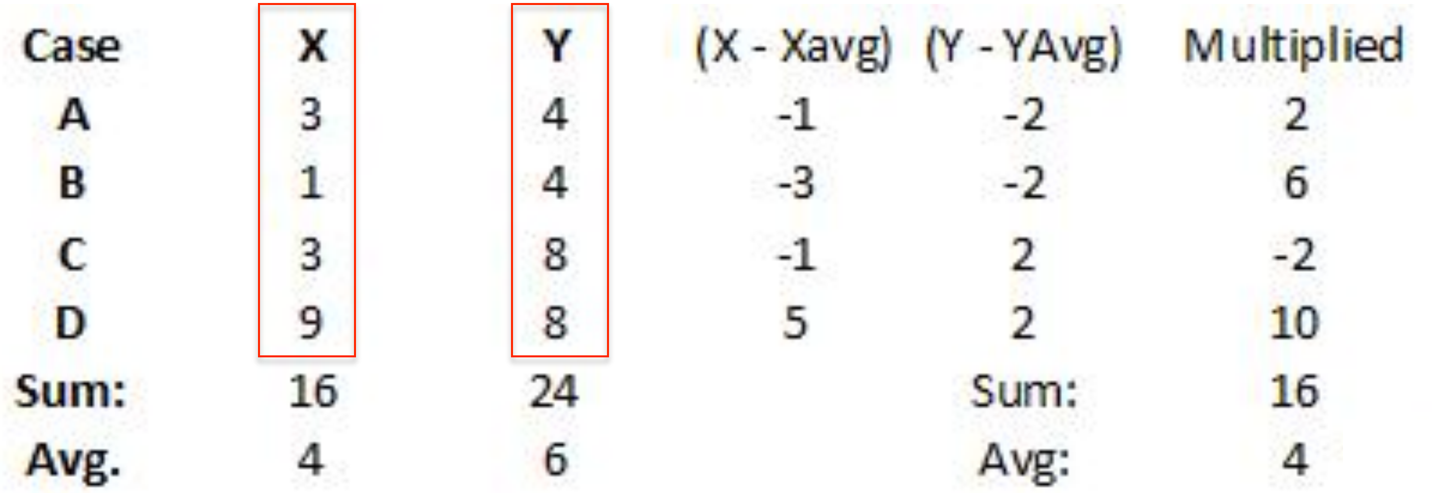
Case	X	X - Avg	(X - Avg)^2
A	3	-1	1
B	1	-3	9
C	3	-1	1
D	9	5	25
Sum:	16	Sum:	36
Avg:	4	Variance:	9

Variance = Σ (x-Avg)^2 / N =36/4 =9

- Standard deviation is the square root of variance

Covariance

- Variance of one variable
- Covariance of two variables



Case	X	Y	(X - Xavg)	(Y - YAvg)	Multiplied
A	3	4	-1	-2	2
B	1	4	-3	-2	6
C	3	8	-1	2	-2
D	9	8	5	2	10
Sum:	16	24		Sum:	16
Avg.	4	6		Avg:	4

Covariance = $\Sigma\ (X_i - Xavg)(Y_i - Yavg)\ /\ N = (2+6-2+10)/4 = 4$

covariance coefficient

Vectors and Matrices

- Vectors: an ordered collection of numbers written in a column (n × 1)

$$\begin{pmatrix} 1 \\ -2 \end{pmatrix}, \quad \begin{pmatrix} .6 \\ .4 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 3 \\ -4 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ -1 \\ 2 \\ 4 \end{pmatrix}$$

- Matrices: a rectangular array of numbers $(m \times n)$

$$(1, 2, 3), \quad \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \quad \begin{pmatrix} 1 & -1 \\ -2 & 2 \end{pmatrix}; \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \quad \begin{pmatrix} 1 & 7 & -8 & 9 & 10 \\ 3 & -1 & 14 & 2 & -6 \\ 0 & 3 & -5 & 7 & 0 \end{pmatrix}$$

(1 × 3)   (3 × 1)   (2 × 2)   (4 × 4)   (3 × 5)

- Matrices multiplication

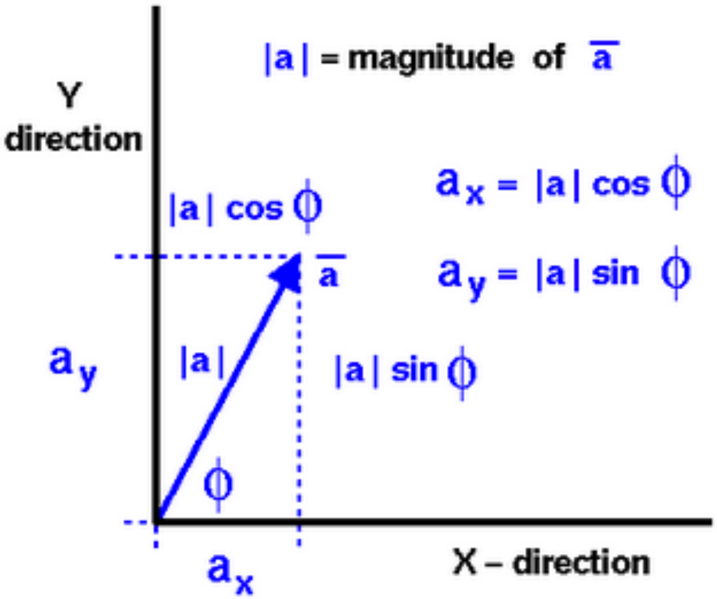
(m × n) × (n × k) = m × k,
e.g. (below), (2 × 2) × (2 × 1) = 2 × 1

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 11 \\ 5 \end{pmatrix}$$

$$11 = 2 \times 1 + 3 \times 3$$
$$5 = 2 \times 1 + 1 \times 3$$

Vector Direction and Magnitude

- A vector quantity has both magnitude (length) and direction.



- $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$ and $\begin{pmatrix} 6 \\ 4 \end{pmatrix}$ have the same direction, but different magnitude (or lengths).

Covariance Matrix

- Recall that covariance is always measured between 2 dimensions (variables). If we have a dataset with more than 2 dimensions, there is more than one covariance measurement that can be calculated.

- For example, for a 3-dimensional dataset (x,y,z), we could calculate cov(x,y), cov(x,z) and cov(y,z).

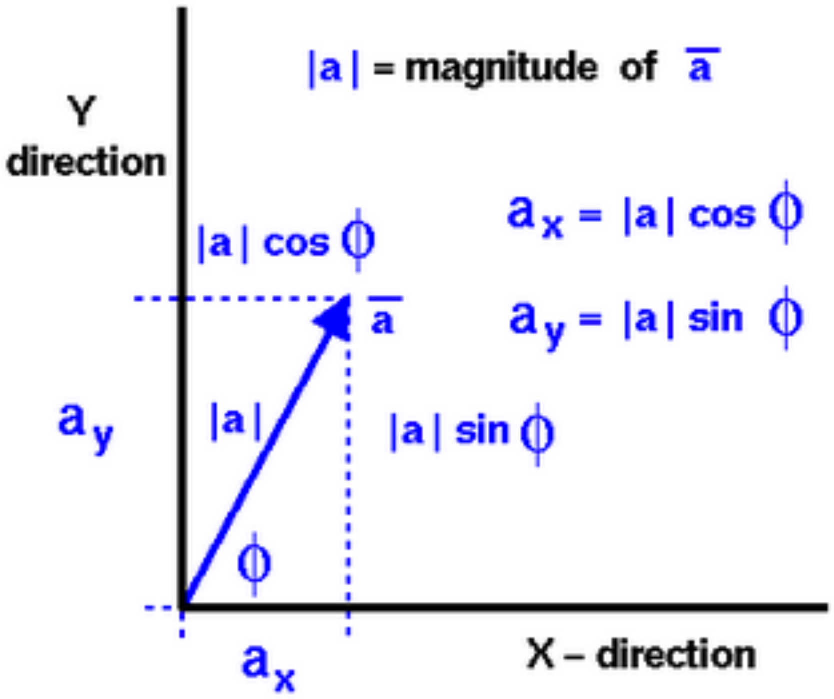
- We use a covariance matrix like this:

$$C = \begin{pmatrix} \cancel{cov(x,x)} & \boxed{cov(x,y)} & \boxed{cov(x,z)} \\ \boxed{cov(y,x)} & \cancel{cov(y,y)} & \boxed{cov(y,z)} \\ \boxed{cov(z,x)} & \boxed{cov(z,y)} & \cancel{cov(z,z)} \end{pmatrix}$$

  - Note that cov(a,b)=cov(b,a), for any a and b.
  - So the matrix C is **symmetrical** about main diagonal.

Vector Direction and Magnitude

- A vector quantity has both magnitude (length) and direction.



$\begin{pmatrix} 3 \\ 2 \end{pmatrix}$ length is $\sqrt{(3^2+2^2)} = \sqrt{13}$

$\begin{pmatrix} 6 \\ 4 \end{pmatrix}$ length is ?

- $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$ and $\begin{pmatrix} 6 \\ 4 \end{pmatrix}$ have the same direction, but different lengths.

Eigenvectors and Eigenvalues

- Given a matrix A, and a vector x, when A is used to transform x, the result is $y = Ax$, e.g.,

$$y = Ax = \boxed{\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}} \times \boxed{\begin{pmatrix} 1 \\ 3 \end{pmatrix}} = \boxed{\begin{pmatrix} 11 \\ 5 \end{pmatrix}}$$

- Now an interesting question
  - Are there any vectors x which does not change its direction under this transformation? However allow the vector magnitude to vary by scalar λ.
  - Such a question is of the form $Ax=\lambda x$.

$$y = Ax = \boxed{\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}} \times \boxed{\begin{pmatrix} 3 \\ 2 \end{pmatrix}} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = \boxed{4} \times \boxed{\begin{pmatrix} 3 \\ 2 \end{pmatrix}} = \lambda x$$

  - Such special x are called eigenvectors and λ are called eigenvalues

Eigenvectors

- All the eigenvectors of a matrix are perpendicular, i.e., at right angles to each other, no matter how many dimensions there are.
  - In maths talk, another word for perpendicular is orthogonal.

- Eigenvectors determine the directions.

- The length of a vector does not affect whether it's an eigenvector or not.

- Since the length of an eigenvector does not affect its direction, we can scale (or standardise) eigenvectors so that they have the same length of 1.

$\begin{pmatrix} 3 \\ 2 \end{pmatrix} \div \sqrt{13} = \begin{pmatrix} 3/\sqrt{13} \\ 2/\sqrt{13} \end{pmatrix}$, and the length of $\begin{pmatrix} 3/\sqrt{13} \\ 2/\sqrt{13} \end{pmatrix}$ is 1.

Eigenvalues

- Eigenvalues are closely related to eigenvectors
  - They always come in pairs.
  - Recall $Ax=\lambda x$

$$\boxed{\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}} \times \boxed{\begin{pmatrix} 3 \\ 2 \end{pmatrix}} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = \boxed{4} \times \boxed{\begin{pmatrix} 3 \\ 2 \end{pmatrix}}$$

  - 4 is the eigenvalue associated with the eigenvector

- If we scale the eigenvector, then the eigenvalue stays the same

$$2 \times \boxed{\begin{pmatrix} 3 \\ 2 \end{pmatrix}} = \boxed{\begin{pmatrix} 6 \\ 4 \end{pmatrix}} \qquad 2\boldsymbol{x} = z$$

$$\boxed{\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}} \times \boxed{\begin{pmatrix} 6 \\ 4 \end{pmatrix}} = \begin{pmatrix} 24 \\ 16 \end{pmatrix} = \boxed{4} \times \boxed{\begin{pmatrix} 6 \\ 4 \end{pmatrix}} \qquad Az = \lambda z$$

Eigenvectors and Eigenvalues

- Eigenvectors can only be found for square matrices.
  - Matrix $A_{p\times p}$ ✔
  - Matrix $A_{p\times q}$ *(p≠q)* ✖
- For a matrix $A_{p\times p}$, there are at most *p* pairs of (eigenvector, eigenvalue)

How PCA Works

- Given a dataset DS with *n* observations and *p* features, we can build a covariance matrix $C_{p \times p}$
  - Recall: There are at most *p* eigenvectors
- Compute a set of pairs of (eigenvector, eigenvalue)
  - (e_1, 0.382), (e_2, 2.618), (e_3, 1.439)
- Sort them by eigenvalues in a descending order
  - (e_2, 2.618), (e_3, 1.439), (e_1, 0.382)
- The first eigenvector is the first princile component (PC), the second eigenvector is the second PC, and so on.
  - The first PC: e_2
  - The second PC: e_3
  - The third PC: e_1

Intuitively, the larger the eigenvalue is, the more important the direction is

PCA Example – Step 1

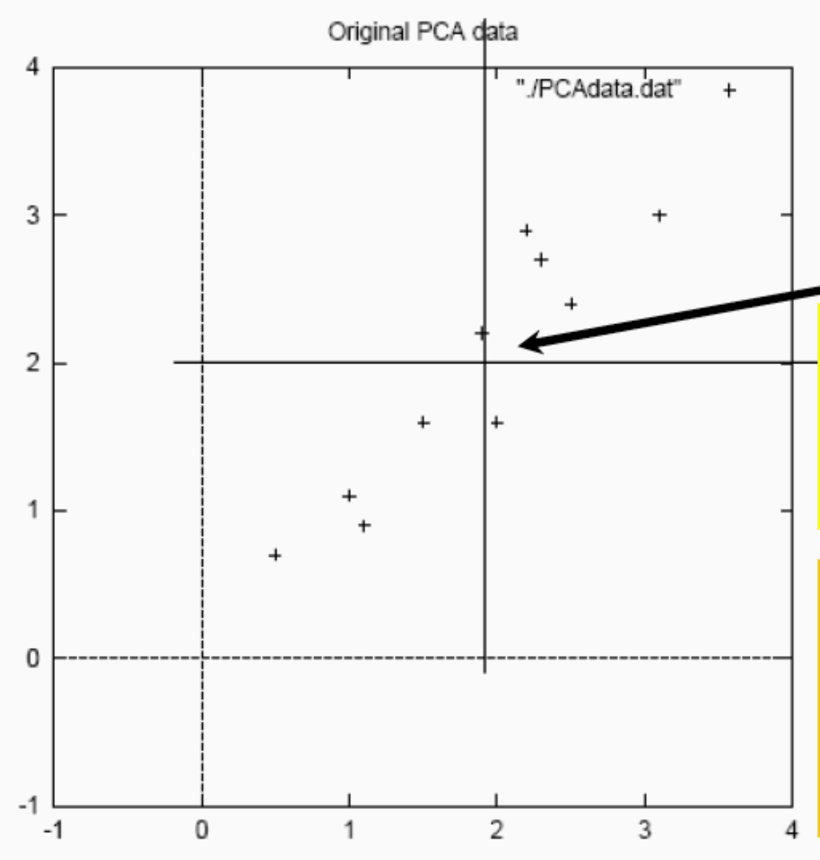
Original data

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

Data with the means subtracted

x-x	y-y
0.69	0.49
-1.31	-1.21
0.39	0.99
0.09	0.29
1.29	1.09
0.49	0.79
0.19	-0.31
-0.81	-0.81
-0.31	-0.31
-0.71	-1.01

Mean: $\bar{x}$=1.81, $\bar{y}$=1.91



Mean:
(1.81, 1.91)

This becomes the new origin of the data from now on

Before PCA is performed, the variables should be centered to have mean 0.

```
> x=c(2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2, 1, 1.5, 1.1)
> mean(x)
[1] 1.81
> x-1.81
 [1]  0.69 -1.31  0.39  0.09  1.29  0.49  0.19 …
```

PCA Example – Step 2

- Calculate the covariance matrix

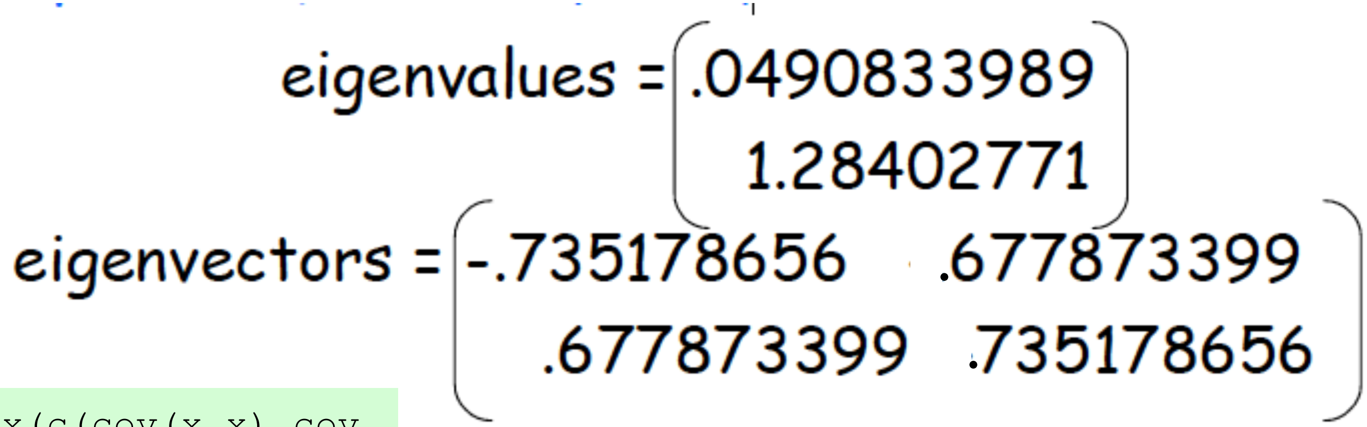
$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

```
> cov(x,y)
[1] 0.6154444
> cov(x,x)
[1] 0.6165556
> cov(y,y)
[1] 0.7165556
```

- Since the non-diagonal elements (cov(x,y)) in this covariance matrix are positive, we should expect that both the x and y variable increase together.

PCA Example – Step 3

- Calculate the eigenvectors and eigenvalues of the covariance matrix



$$\text{eigenvalues} = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$\text{eigenvectors} = \begin{pmatrix} -.735178656 & .677873399 \\ .677873399 & .735178656 \end{pmatrix}$$

```
>CoMatrix=matrix(c(cov(x,x),cov
(x,y),cov(y,x),cov(y,y)),2,2)
> CoMatrix   #covariance matrix
          [,1]      [,2]
[1,] 0.6165556 0.6154444
[2,] 0.6154444 0.7165556
> eigen(CoMatrix)
$values      #The eigenvalues
[1] 1.2840277 0.0490834
$vectors      #The eigenvectors
          [,1]       [,2]
[1,] 0.6778734 -0.7351787
[2,] 0.7351787  0.6778734
```

PCA Example – Step 4

- Select the Principal Components (PC)
  - Comparing the eigenvalues:

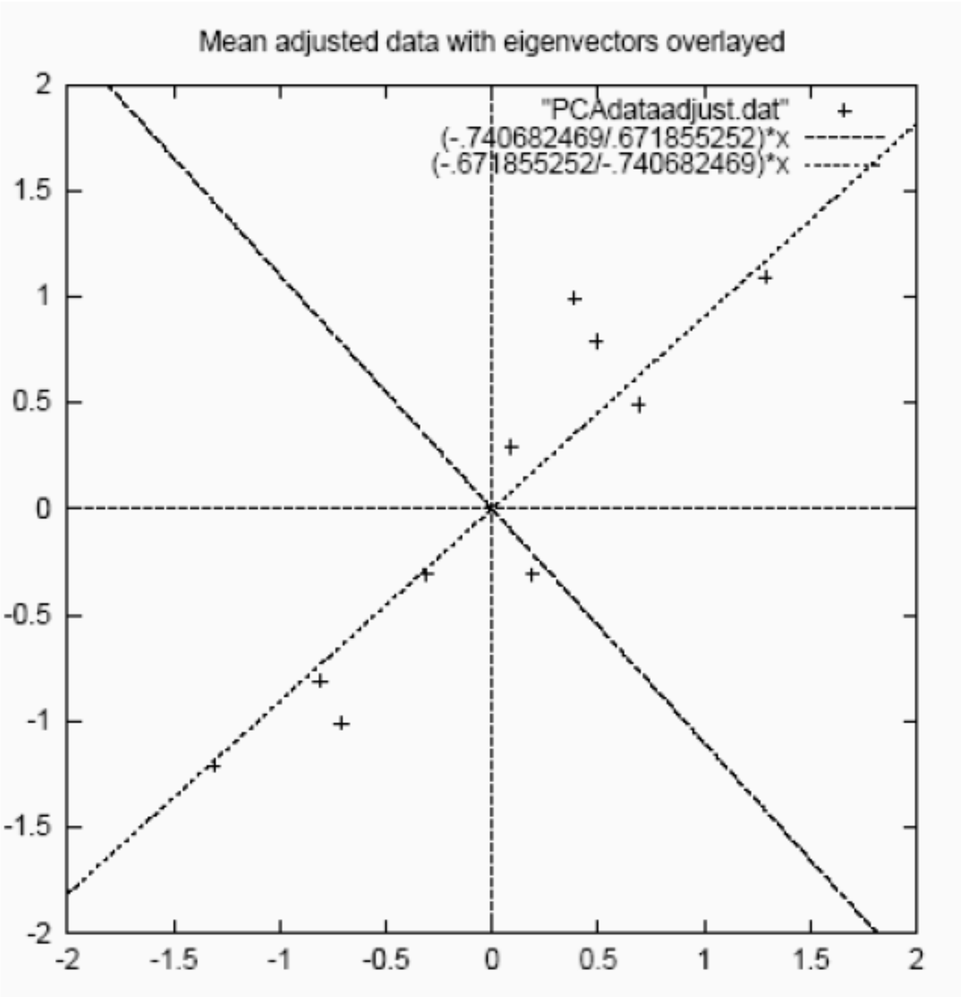
```
1.2840277 > 0.0490834
```

  - Then the first PC is

```
     [,1]
[1,] 0.6778734
[2,] 0.7351787
```

  - And the second PC is

```
     [,2]
[1,] -0.7351787
[2,] 0.6778734
```



Messy? Leave This Hassle to R

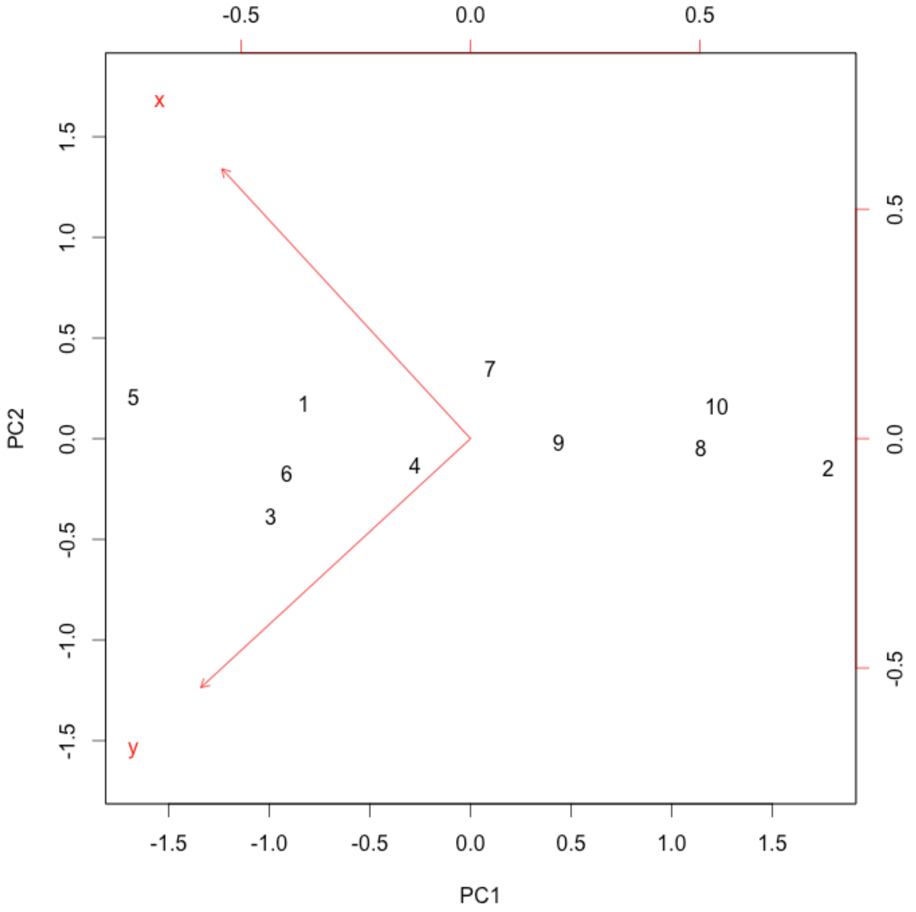
- Compute the Principal Components in R, hassle-free ☺
  - No need to construct a covariance matrix
  - No need to explicitly compute eigenvalues or eigenvectors
  - No need to compare and sort the eigenvalues/eigenvectors

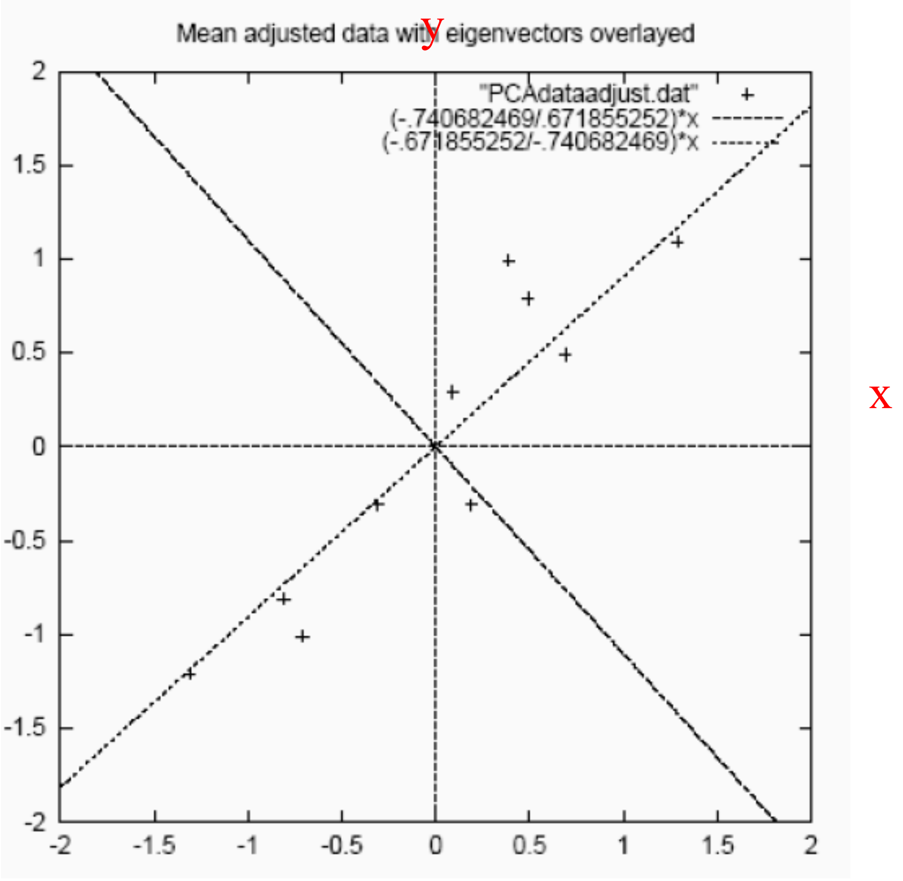
```
> DF.xy<-data.frame(x=c(2.5, 0.5,
2.2, 1.9, 3.1, 2.3, 2, 1, 1.5, 1.1),
y=c(2.4, 0.7, 2.9, 2.2, 3.0, 2.7,
1.6, 1.1, 1.6, 0.9))

> pr.xy=prcomp(DF.xy, scale = FALSE)

> pr.xy$rotation
         PC1         PC2
x -0.6778734  0.7351787
y -0.7351787 -0.6778734
```

```
> pr.xy$center  #  the centre of the new coordinates
x y 1.81 1.91   #  is the mean value

> pr.xy$scale   #  no scale
[1] FALSE

> pr.xy$sdev    #  the standard deviation
[1] 1.1331495 0.2215477

> pr.xy$sdev^2  #  the squared sdev is variance
[1] 1.2840277 0.0490834  #  eigenvalues!!

> biplot(pr.xy,scale=FALSE)
```

Biplot

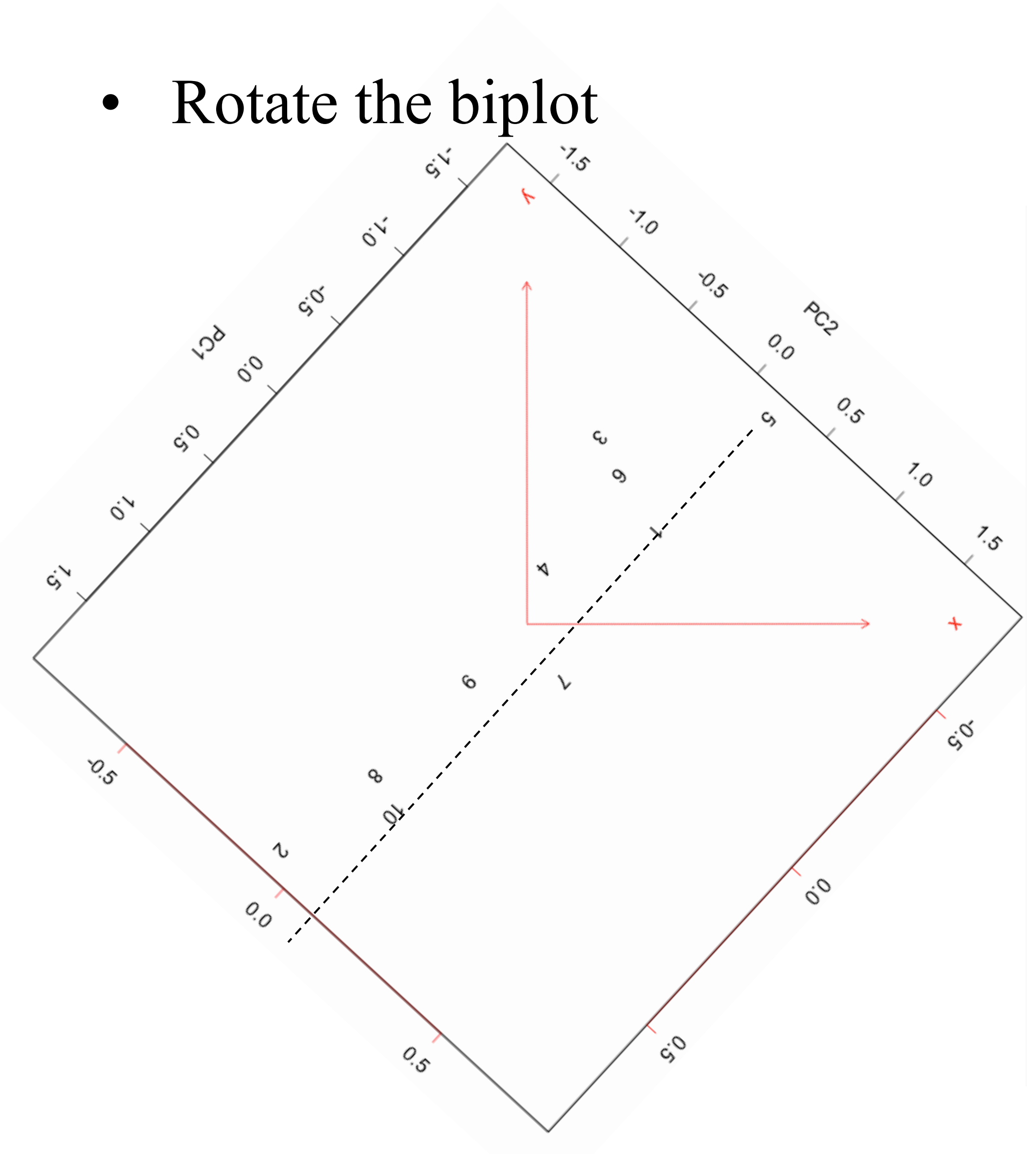
- Biplot: display both the principal component scores and the principal component loadings (directions)

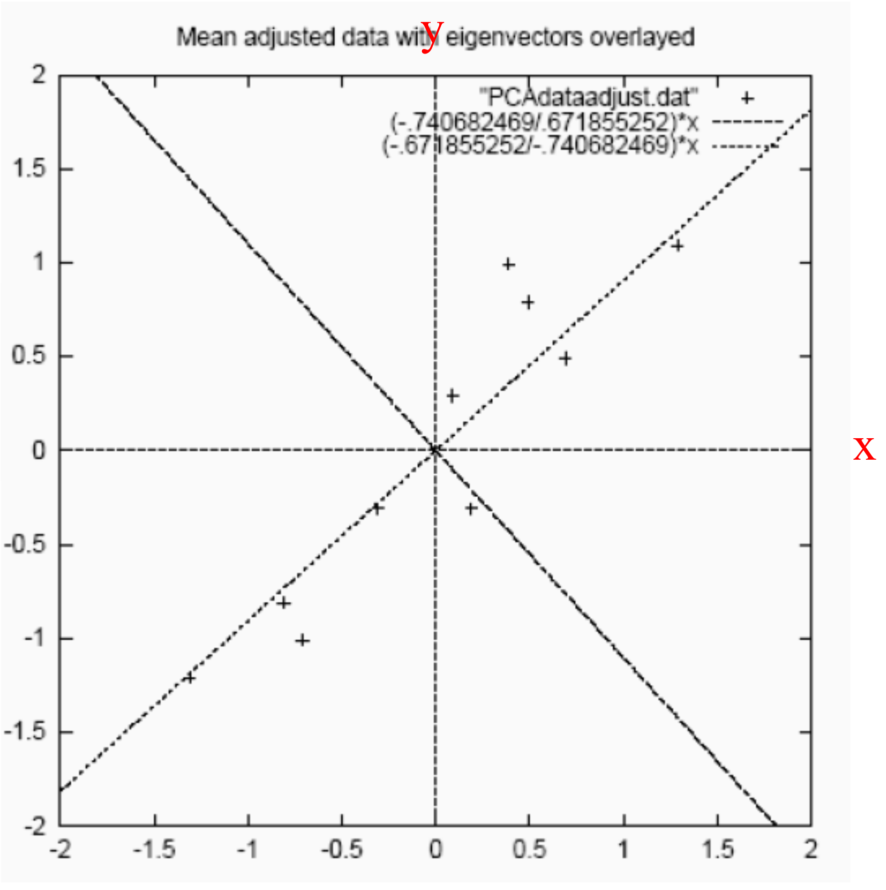




Biplots

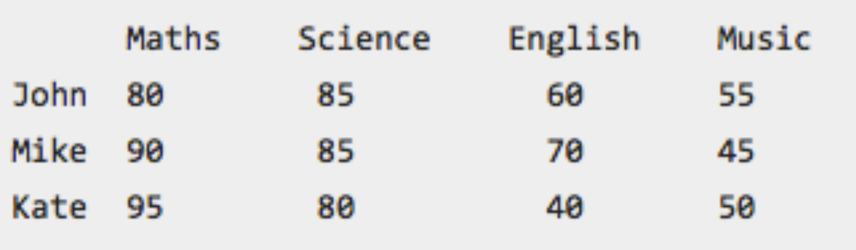
- Rotate the biplot





Example: Marks

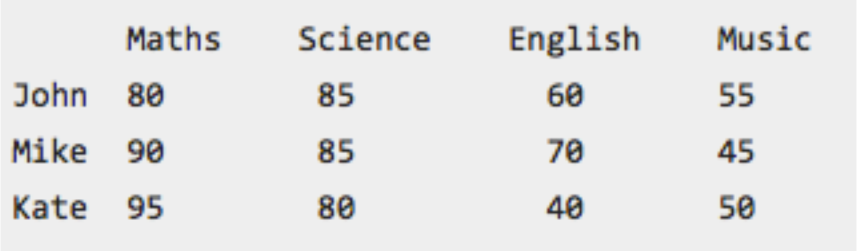
- John, Mike and Kate get the following percentages for exams in Maths, Science, English and Music as follows:



```
      Maths    Science    English    Music
John  80        85         60        55
Mike  90        85         70        45
Kate  95        80         40        50
```

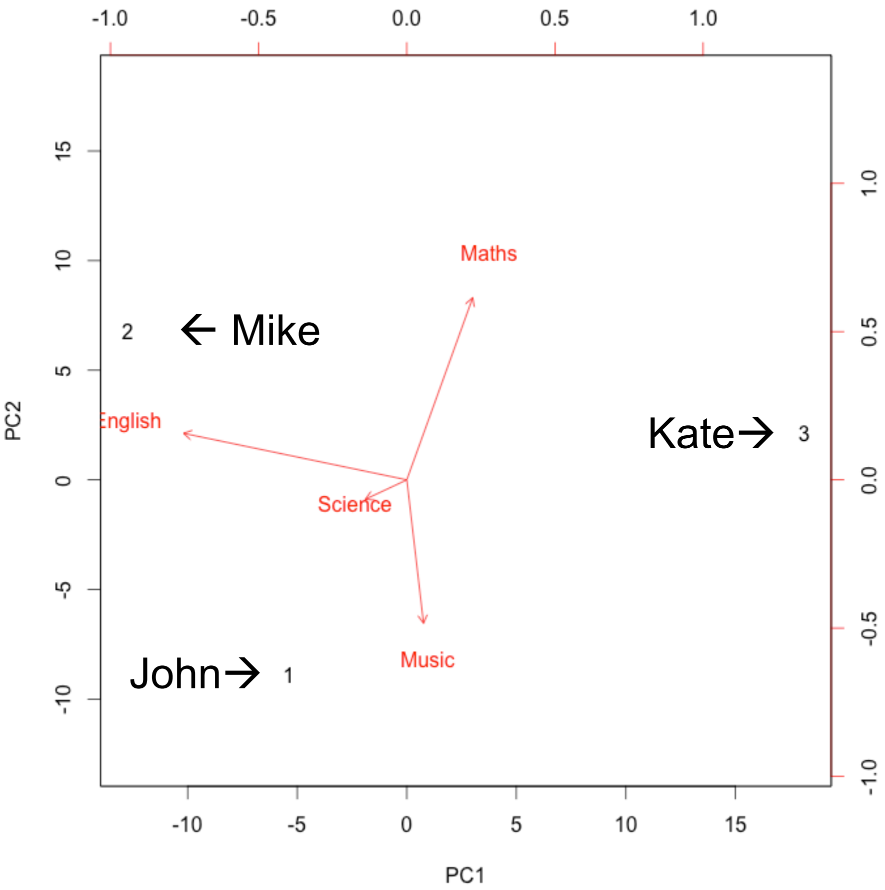
- We can’t visualise the dataset above, so we use PCA to reduce its dimensions by retaining maximal amount of information about the variables
  - For example, we could look at the types of subjects each student is more suited to.

Example: Marks



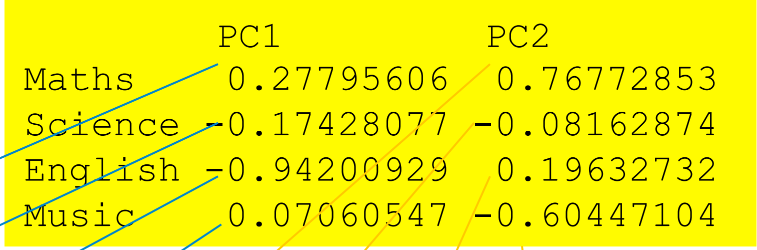
	Maths	Science	English	Music
John	80	85	60	55
Mike	90	85	70	45
Kate	95	80	40	50

```
> DF = data.frame(Maths=c(80, 90,
95), Science=c(85, 85, 80),
English=c(60, 70, 40), Music=c(55,
45, 50))
> pr.marks=prcomp(DF, scale = FALSE)
> pr.marks$rotation #omit PC3
              PC1         PC2
Maths    0.27795606  0.76772853
Science -0.17428077 -0.08162874
English -0.94200929  0.19632732
Music    0.07060547 -0.60447104
> biplot(pr.marks, scale=FALSE)
```



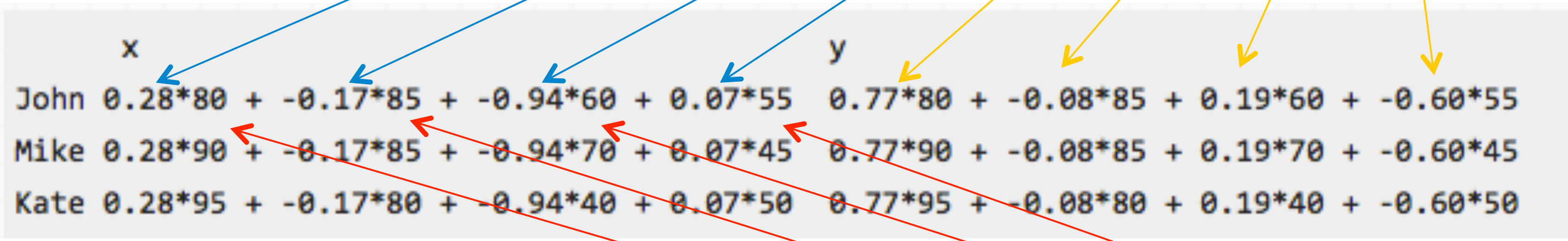
Kate is not good at English or Science
Mike is good at English, but rubbish at Music
John is not good at Maths, but good at Music
…

Example: Marks



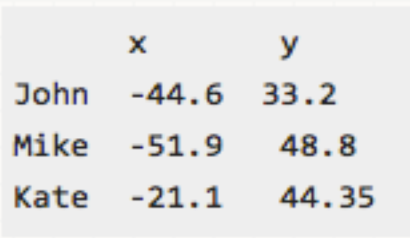
```
                PC1         PC2
Maths     0.27795606  0.76772853
Science  -0.17428077 -0.08162874
English  -0.94200929  0.19632732
Music     0.07060547 -0.60447104
```

- The output from R means we can now plot each person's score across all subjects in a 2D graph as follows:

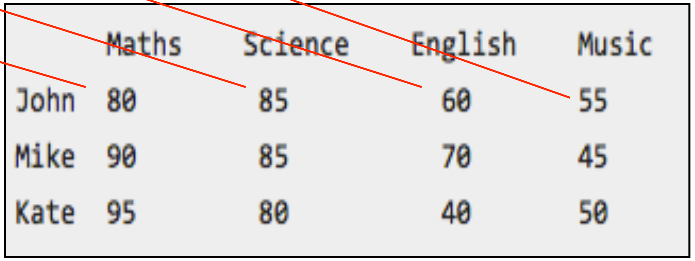


```
         x                                       y
John 0.28*80 + -0.17*85 + -0.94*60 + 0.07*55   0.77*80 + -0.08*85 + 0.19*60 + -0.60*55
Mike 0.28*90 + -0.17*85 + -0.94*70 + 0.07*45   0.77*90 + -0.08*85 + 0.19*70 + -0.60*45
Kate 0.28*95 + -0.17*80 + -0.94*40 + 0.07*50   0.77*95 + -0.08*80 + 0.19*40 + -0.60*50
```

- Which simplifies to



```
         x       y
John   -44.6   33.2
Mike   -51.9    48.8
Kate   -21.1    44.35
```



```
      Maths   Science   English   Music
John  80      85        60        55
Mike  90      85        70        45
Kate  95      80        40        50
```

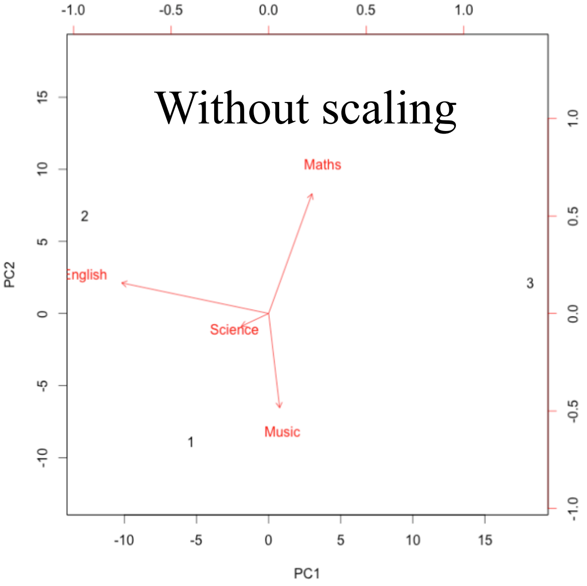
- You can now plot the scores in a 2D graph to get a sense of the type of subjects each student is perhaps more suited to.
  - However, the `biplot` function implicitly scales the variables and observations (type `?biplot.princomp` for more details)

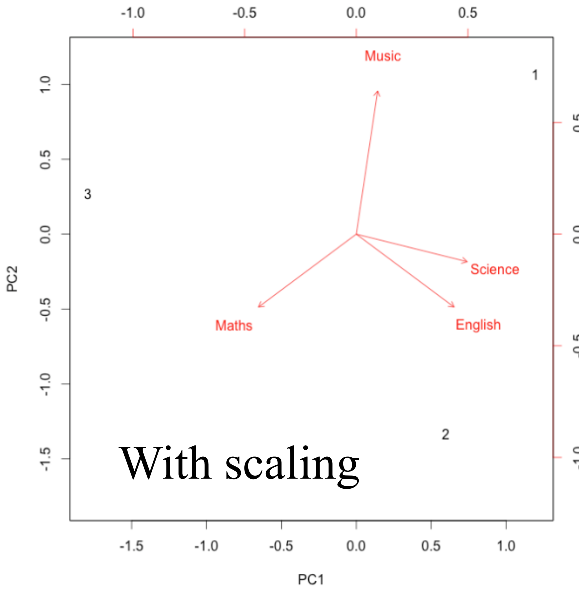
Why Scaling?

- Variables may have different units, and different variance:

```
> apply(DF,2,var)                          #The apply function allows us to apply a
Maths     Science   English    Music       #function var to each row (1)
58.333333 8.333333 233.333333 25.000000    #or column (2) of the dataset
```

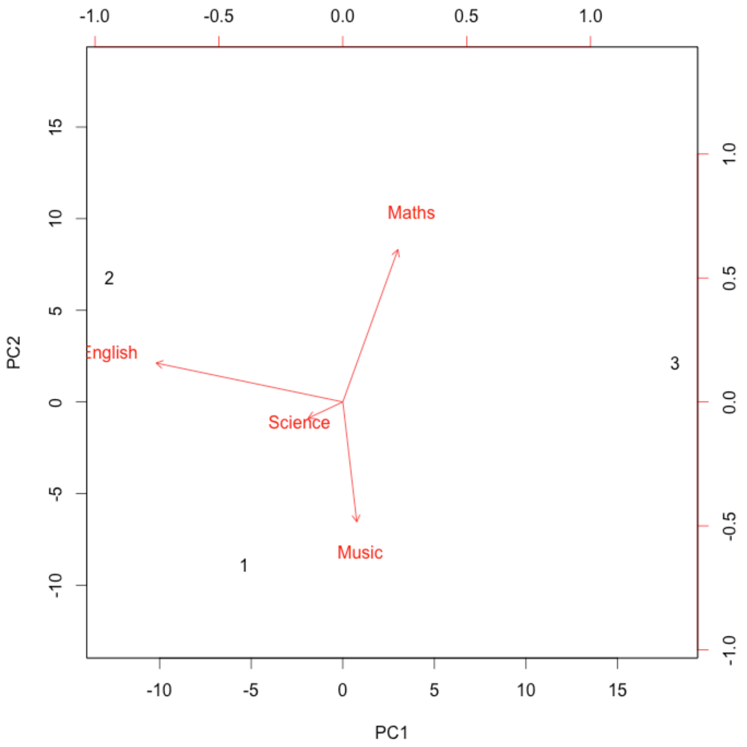
- If performing PCA on the unscaled data, then the first PC will place almost all its weighting on `English`, but little on `Science`.
- We usually scale the variables to have standard deviation one before we perform PCA.
- In certain settings, we choose not to scale the variables, i.e., variables corresponding to expression levels of *p* genes.





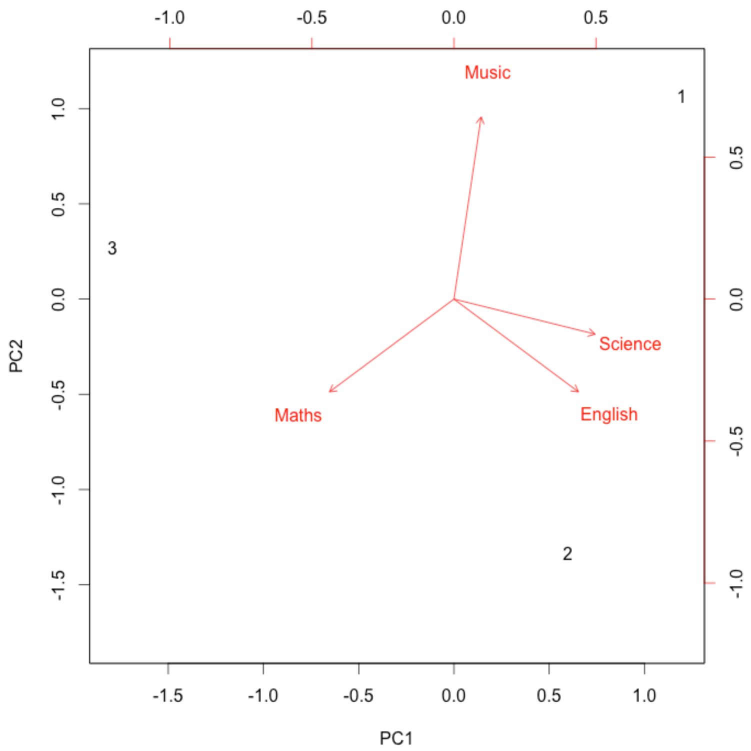
Scaling the Variables

- Without scaling



```
> pr.marks=prcomp(DF, scale = FALSE)
> biplot(pr.marks, scale=FALSE)
```

- With scaling



```
> pr.marks.s=prcomp(DF, scale = TRUE)
> biplot(pr.marks.s, scale=0)
```

The Proportion of Variance Explained

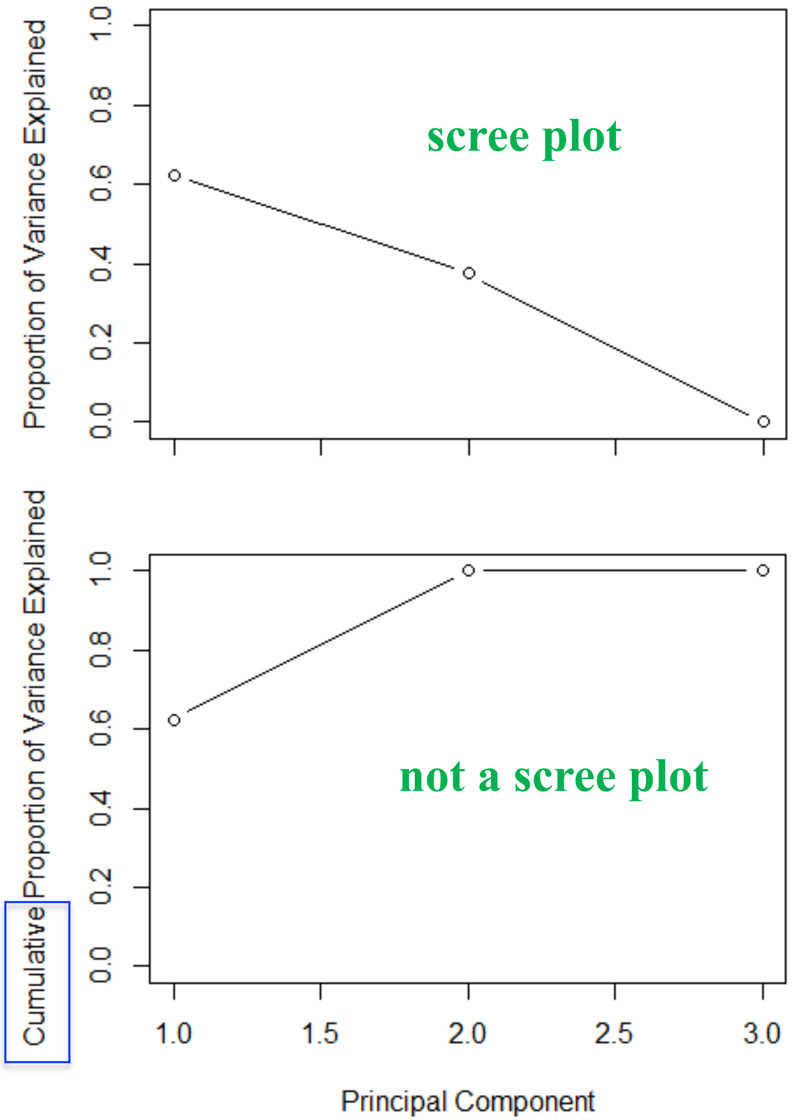
- How much of the information in a given dataset is lost by projecting the observations onto the first few PCs?
  - The Proportion of Variance Explained (PVE) by each component
  - **Σ Each PC's variance / Total variance**

```
> pr.marks.var=pr.marks.s$sdev^2
> pve=pr.marks.var/sum(pr.marks.var)
> pve
[1] 6.250000e-01 3.750000e-01 7.183785e-31
> plot(pve,xlab="Principal Component",
   ylab="Proportion of Variance
   Explained", type="b",ylim=c(0,1))
```

→ a scree plot

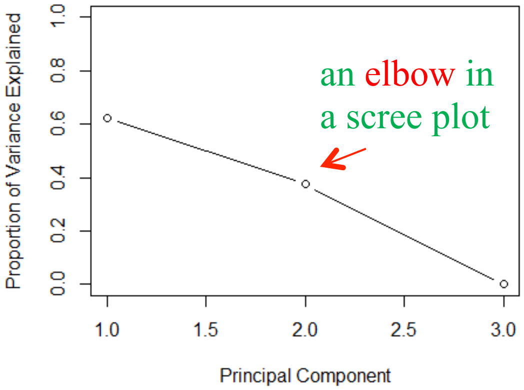
  - Cumulate each PVE

```
> plot(cumsum(pve),xlab="Principal
   Component", ylab="Cumulative Proportion
   of Variance Explained",
   type="b",ylim=c(0,1))
```



Deciding How Many PCs to Use

- Goal: Use smallest number of PCs to get a good understanding of the data
  - How many PCs are needed? No single answer ☹
- Eyeballing scree plot
  - Looking for a point at which the proportion of variance explained by each subsequent principal component drops off. → an elbow
  - This type of visual analysis is *ad hoc*
- In practice, we look at first few PCs to find interesting patterns.
  - If no interesting patterns are found
    → further PCs are unlikely to be interesting
  - If first few PCs are interesting
    •Continue to look at subsequent PCs
    •Until no further interesting patterns are found



- This is a subjective approach, generally used as a tool for exploratory data analysis.

Another Example: USArrest

- Read Section 10.4 Lab 1 in the textbook for more details