Introduction to Multivariate Analysis

Dr Joram M. Posma

Lecturer in Cancer Informatics

Section of Bioinformatics, Division of Systems Medicine, Department of Metabolism, Digestion and Reproduction, Imperial College London, UK

Rutherford Fellow

Health Data Research (HDR) UK, HDR-London, UK

HDRUK Health Data Research UK

MRC Medical Research Council

On the (pre-lunch) menu for today

- Data analysis of metabolic profiling data
  - Data curation
  - Outlier detection
  - Variable selection
  - Predictive models
- Multivariate methods
  - Data reduction and unsupervised analysis
  - Principal Component Analysis (PCA)
  - Visualization
  - Validation strategies

Shout-outs

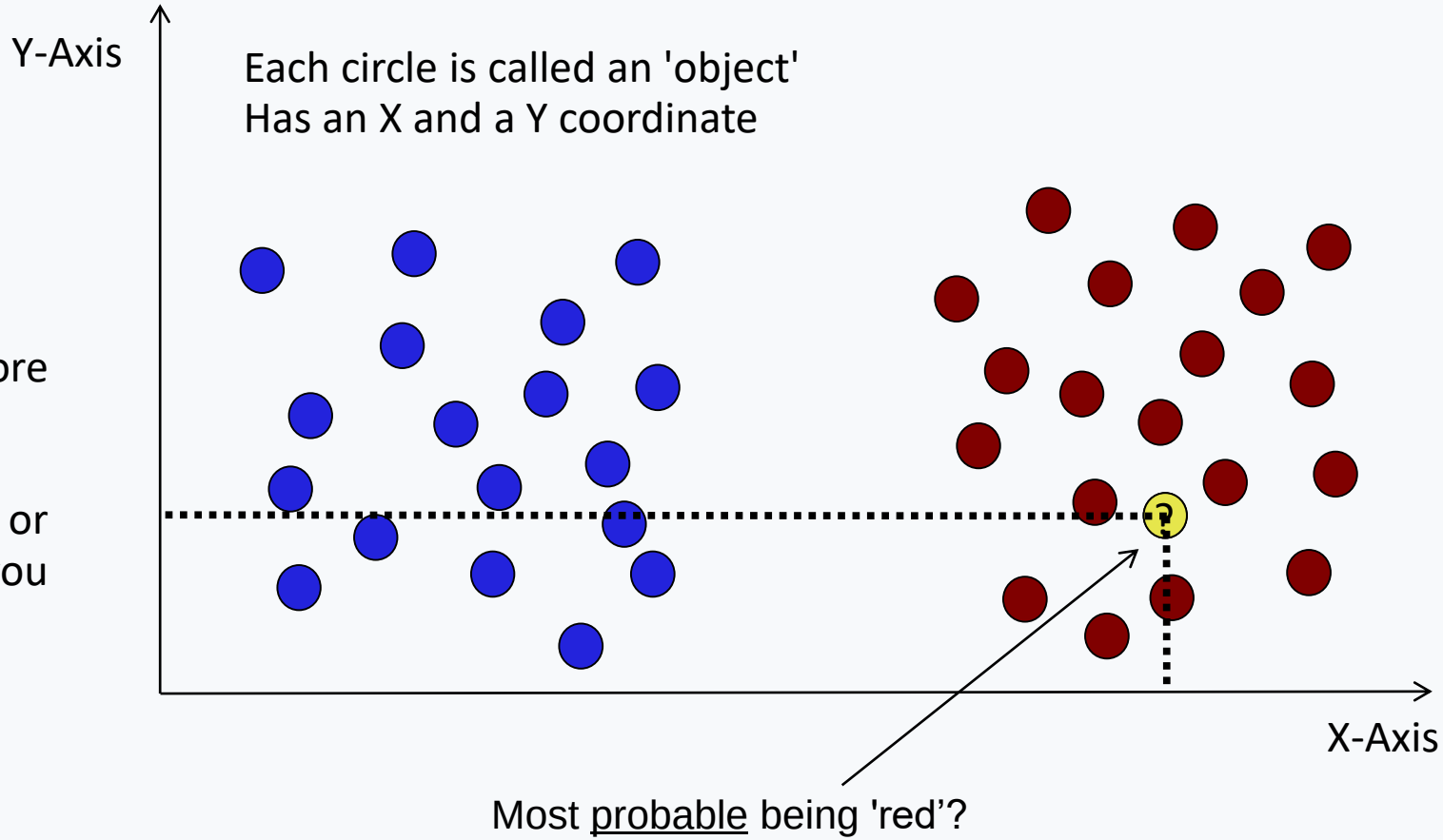
By the end of this session, you will be better able to

- Describe some ways in which data reduction methods can be used to evaluate data quality
- Explain the basis behind Principal Component Analysis
- Choose between different types of scaling
- Distinguish between visualizing similarity of samples and similarity of variables
- Detail the difference between unsupervised and supervised analyses
- Discuss the concept of cross-validation

Motivation: classification (example)

Visualizing:
2 axes is easy
3 axes still easy
4 axes getting more difficult

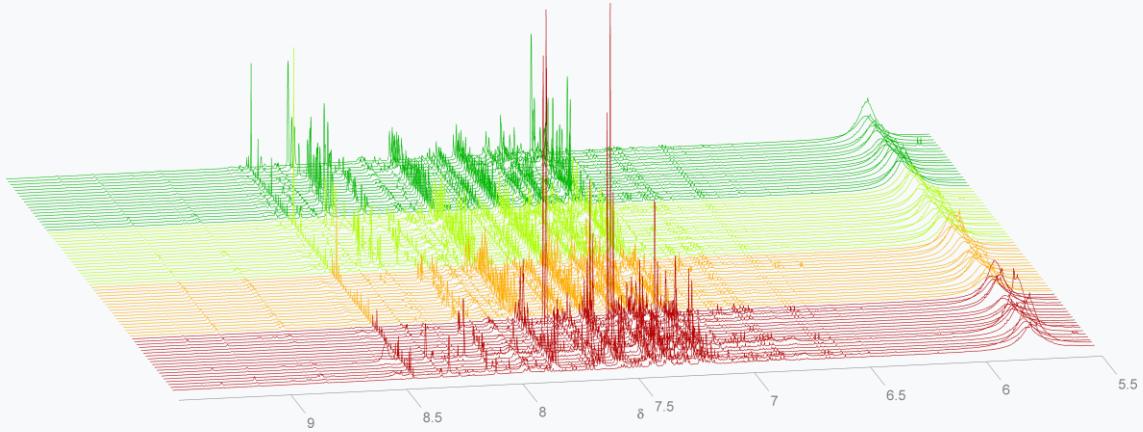
Hundred, thousand or more axes...can you <u>visualize</u> that?

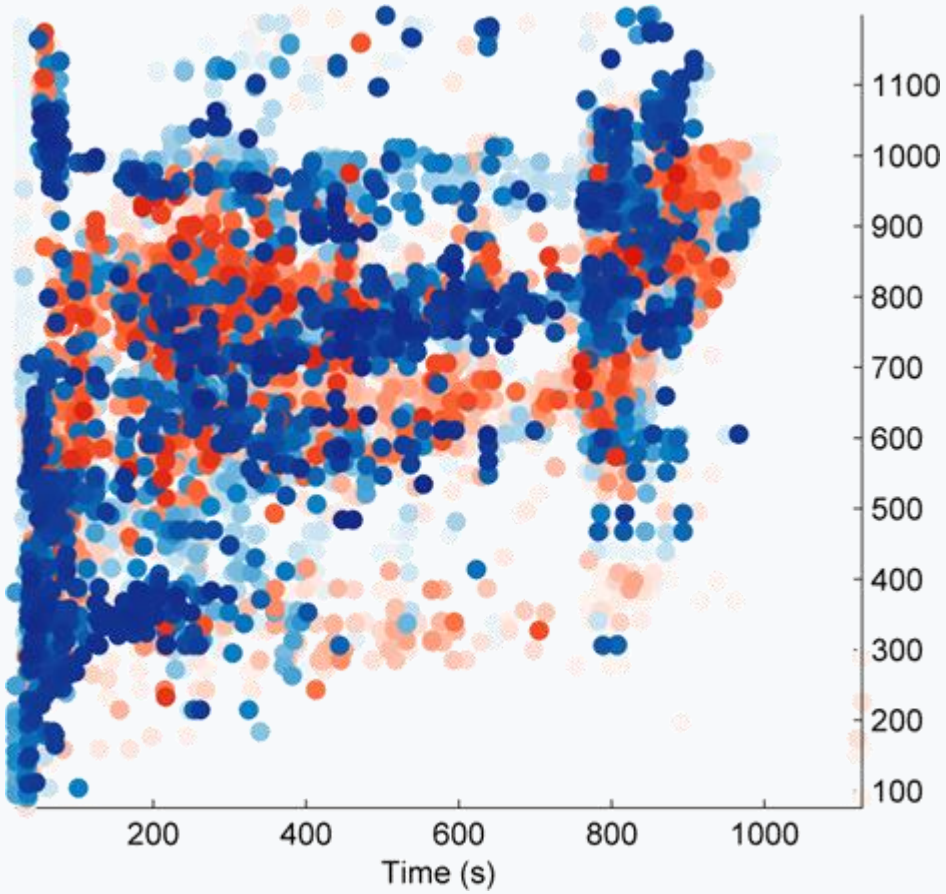


Class 1 or 'blue'

Class 2 or 'red'

What *multi*variate data looks like: NMR and MS



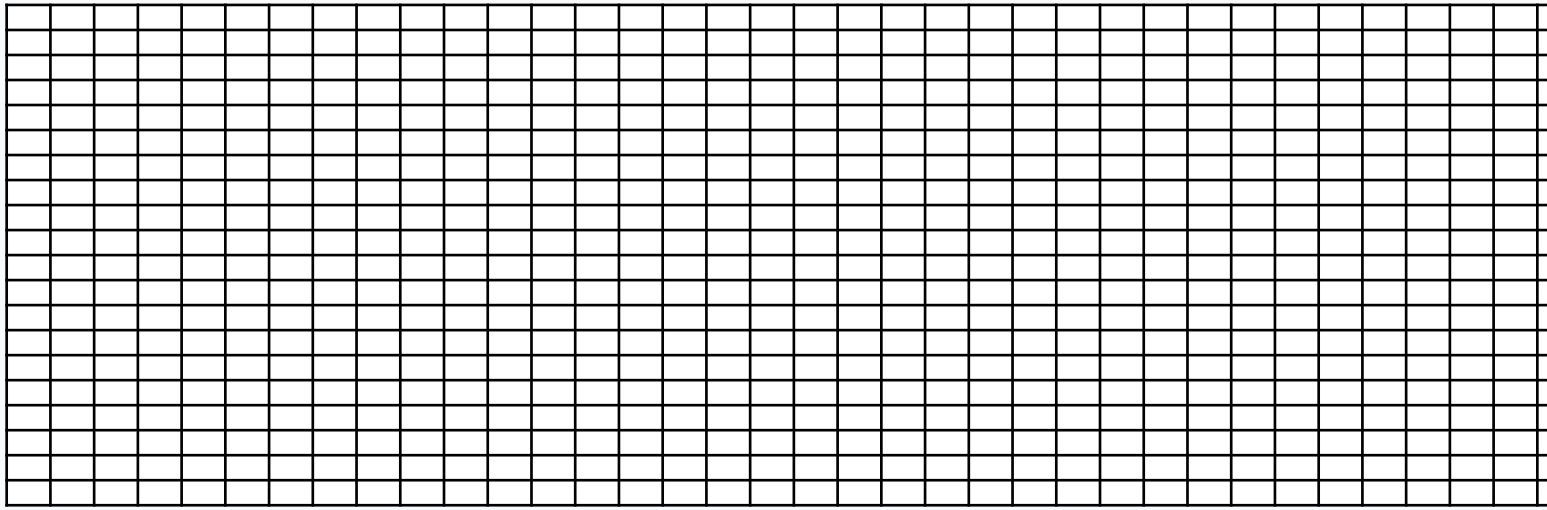


What *multi*variate data looks like: data matrices

Variables (only 17...)

Objects (only 33 samples...)

26	96	30	26	68	56	42	68	15	56	49	36	56	78	17	75	37
51	0	74	80	4	88	9	64	59	93	85	6	40	29	26	12	9
70	77	19	3	7	67	27	95	26	70	87	52	6	60	40	53	64
89	82	69	93	52	19	15	21	4	58	27	34	78	96	7	33	18
96	87	18	73	10	37	28	71	75	82	21	18	34	43	68	55	5
55	8	37	49	82	46	44	24	24	88	56	21	61	69	40	40	72
14	40	63	58	82	98	53	12	44	99	64	91	74	76	98	42	35
15	26	78	24	72	16	46	61	69	0	42	68	10	43	40	18	66
26	80	8	46	15	86	88	45	36	87	21	47	13	66	62	26	38
84	43	93	96	66	64	52	46	74	61	95	91	55	11	15	2	63
25	91	78	55	52	38	94	66	39	99	8	10	49	93	38	92	2
81	18	49	52	97	19	64	77	68	53	11	75	89	19	16	65	91
24	26	44	23	65	43	96	35	70	48	14	74	80	27	76	93	80
93	15	45	49	80	48	24	66	44	80	17	56	73	80	87	16	75
35	14	31	62	45	12	68	42	2	23	62	18	5	49	35	92	81
20	87	51	68	43	59	29	84	33	50	57	60	7	77	69	79	38
25	58	51	40	83	23	67	83	42	90	5	30	9	40	29	58	62
62	55	82	37	8	38	70	26	27	57	93	13	80	27	53	44	58
47	14	79	99	13	58	7	61	20	85	73	21	94	4	83	26	53
35	85	64	4	17	25	25	58	82	74	74	89	68	67	60	75	28
83	62	38	89	39	29	22	54	43	59	6	7	13	43	34	23	25
59	35	81	91	83	62	67	87	89	25	86	24	72	45	30	6	45
55	51	53	80	80	27	84	26	39	67	93	5	11	61	45	77	23
92	40	35	10	6	82	34	32	77	8	98	44	12	6	42	67	80
29	8	94	26	40	98	78	12	40	63	86	1	64	32	36	72	99
76	24	88	34	53	73	68	94	81	66	79	90	33	77	56	64	3
75	12	55	68	42	34	1	65	76	73	51	20	65	70	74	42	54
38	18	62	14	66	58	60	48	38	89	18	9	75	13	42	39	9
57	24	59	72	63	11	39	64	22	98	40	31	58	13	43	82	80
8	42	21	11	29	91	92	54	79	77	13	46	74	9	12	32	99
5	5	30	65	43	88	0	65	95	58	3	10	23	1	2	81	7
53	90	47	49	2	82	46	54	33	93	94	100	73	42	29	79	94
78	94	23	78	98	26	42	72	67	58	30	33	97	66	32	85	2



Data reduction

- Describing the original data (X) in another way

$$X = U\Sigma V^T = (U\Sigma)V^T = TP^T$$

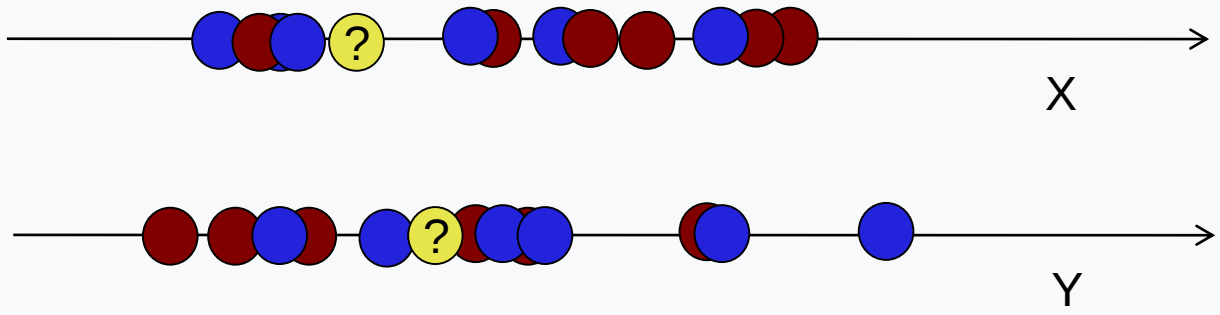
- Goal: have less variables to do data analysis on

- Unsupervised analysis: do not assume any prior relationship between samples
- Supervised analysis: algorithm uses relationships between samples (e.g. classification)

(We'll get back to the equation later and what it means)

Principal Component Analysis

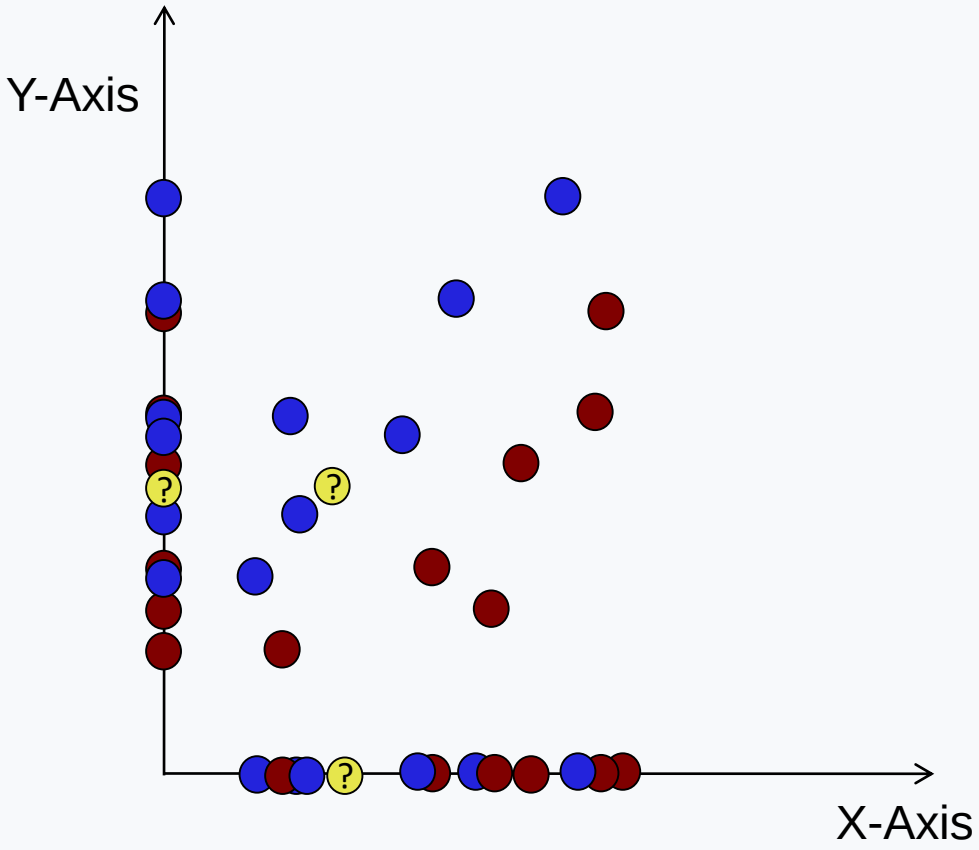
- PCA is a data reduction method
- A new way of looking at the data
- (X,Y)-example (2 variables):



- How would you classify the yellow object?

Principal Component Analysis

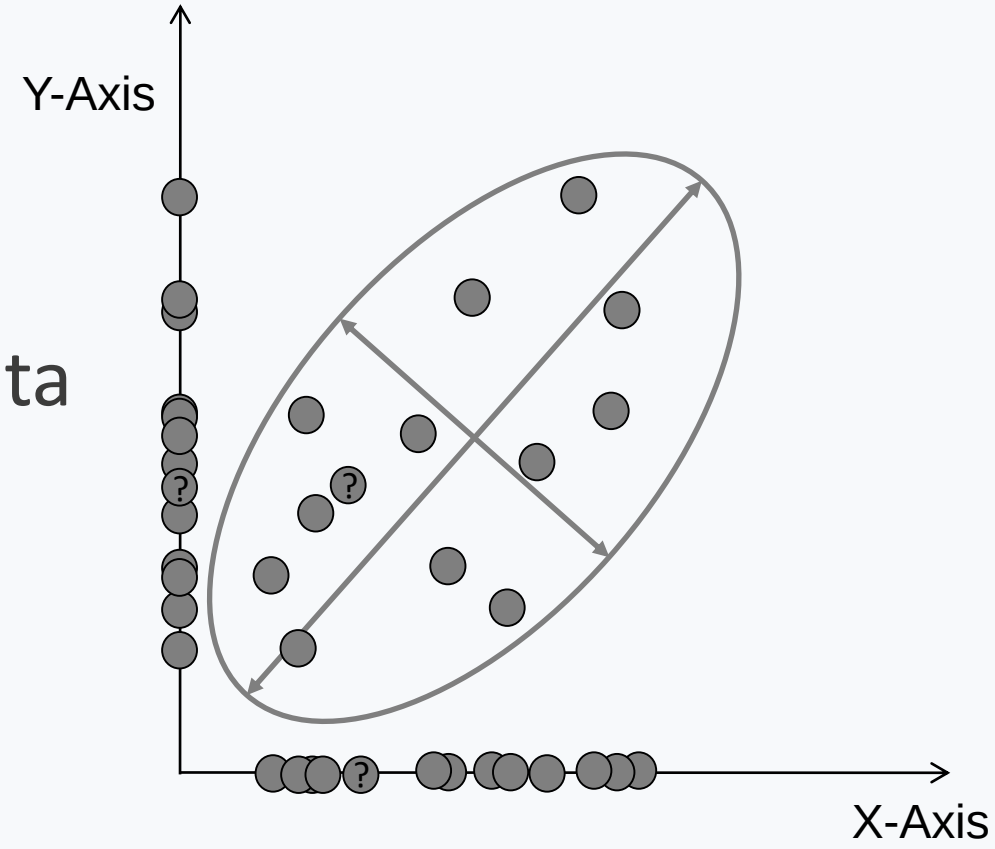
- Let’s make use of the two coordinates and use them together…



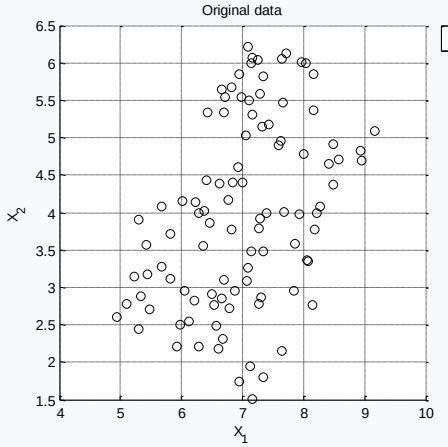
Principal Component Analysis

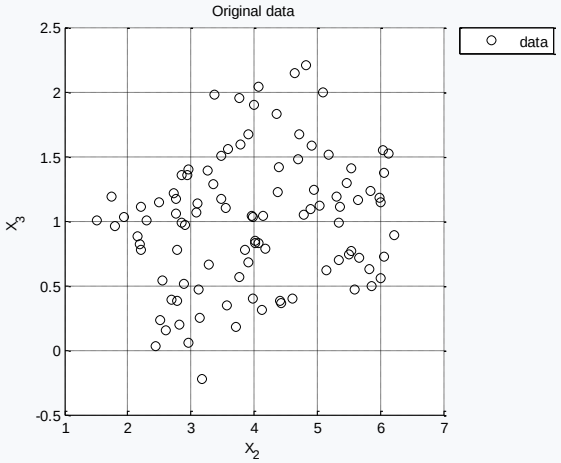
- PCA is not classification

- Interest:
- Spread of the data



Unsupervised analysis
Example data (n = 100, p = 3)

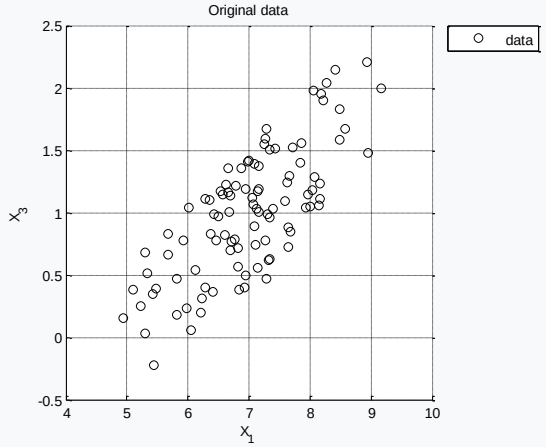


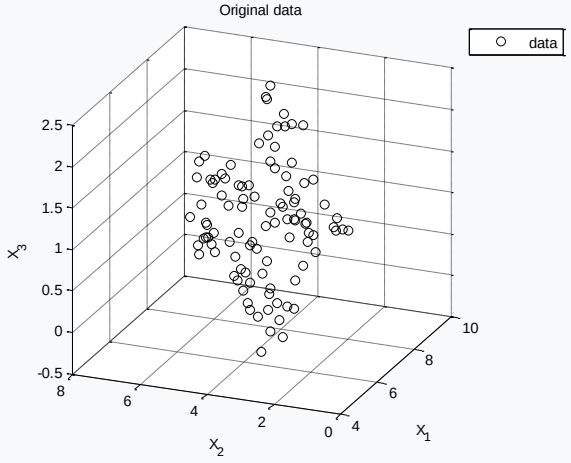


p →

n ↓

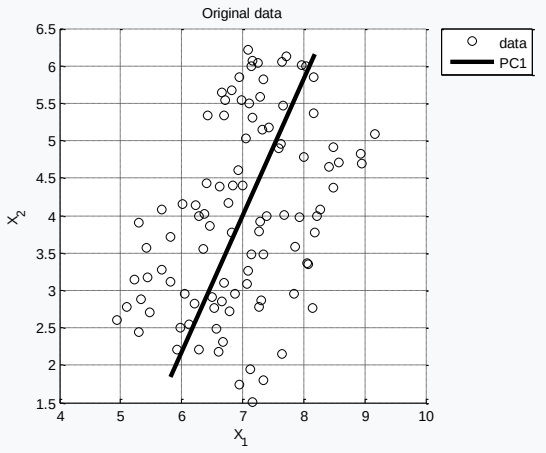
X_1	X_2	X_3
26	96	30
51	0	74
70	77	19
89	82	69
96	87	18
55	8	37
14	40	63
15	26	78
26	80	8
84	43	93
25	91	78
81	18	49
24	26	44
93	15	45
35	14	31
20	87	51
25	58	51
62	55	82
47	14	79
35	85	64
83	62	38
59	35	81
55	51	53
92	40	35
29	8	94
76	24	88
75	12	55
38	18	62
57	24	59
8	42	21
5	5	30
53	90	47
78	94	23

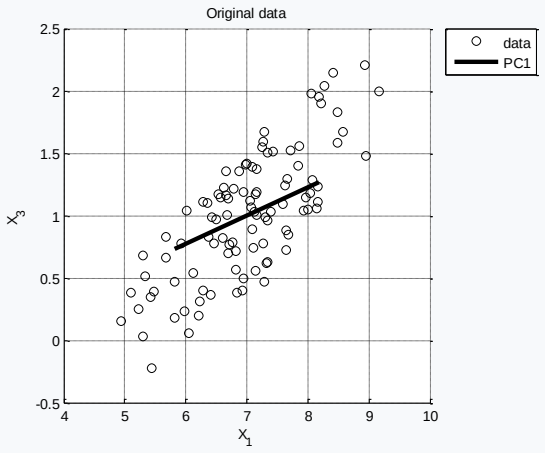


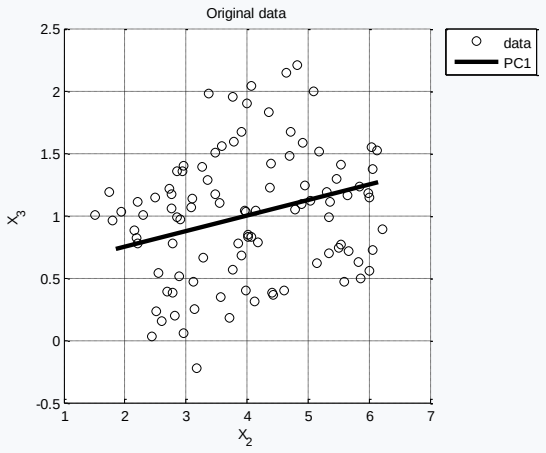


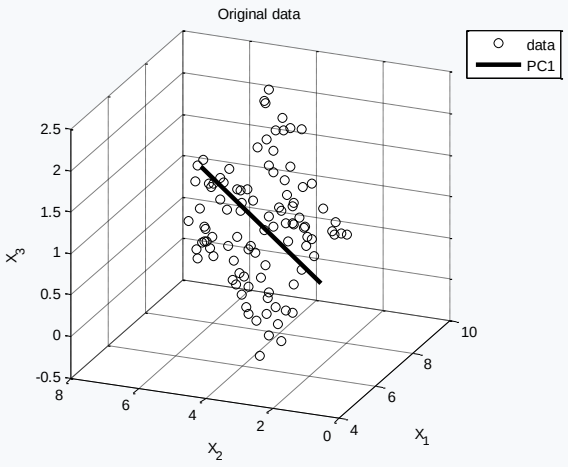
(Data matrix is not the same data as in the plots)

Principal Component Analysis (PC 1)

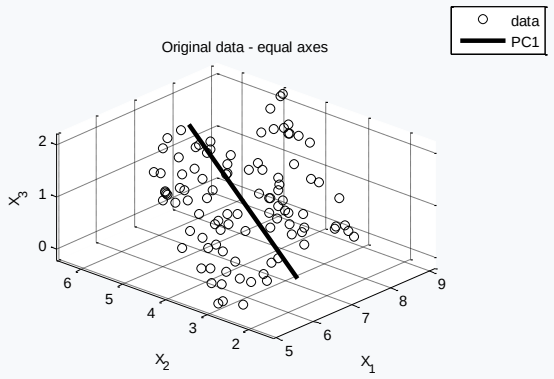




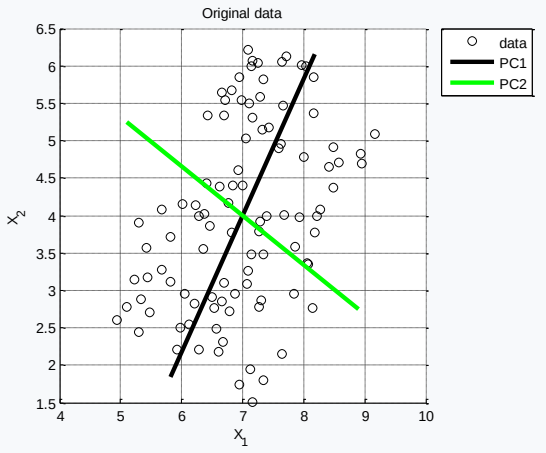


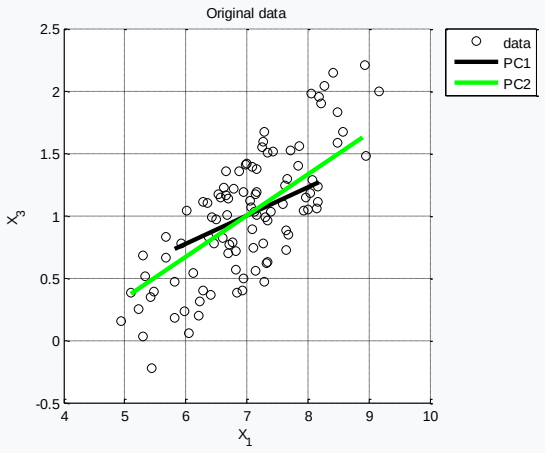


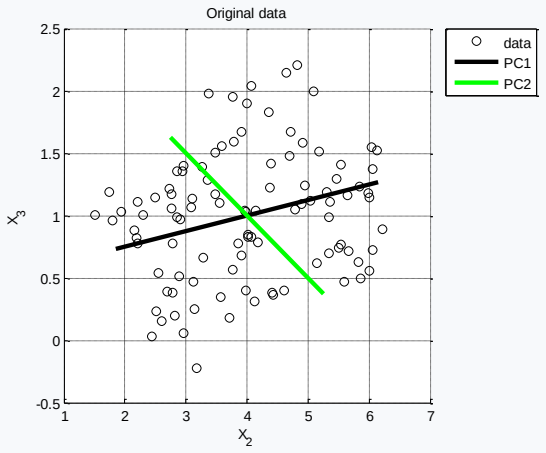
- Identify biggest spread in the data
- Variance is ‘interesting’

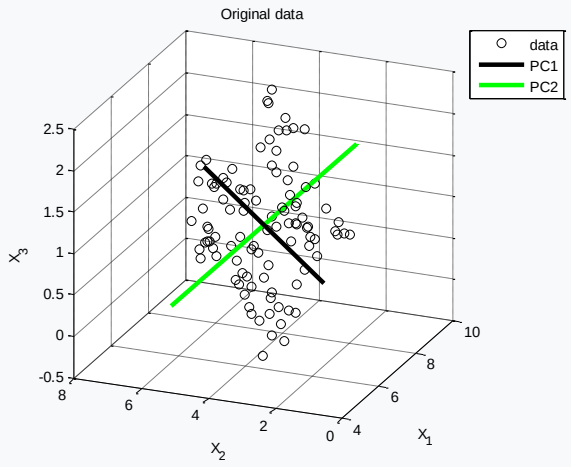


Principal Component Analysis (PC 2)

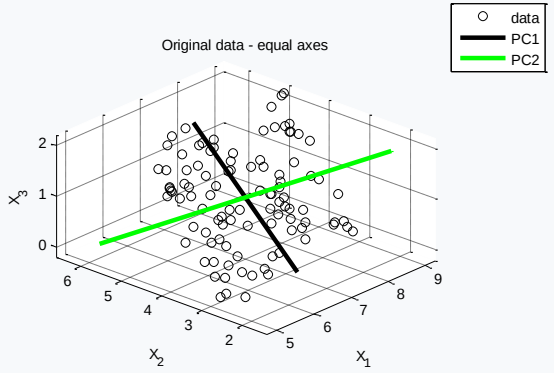






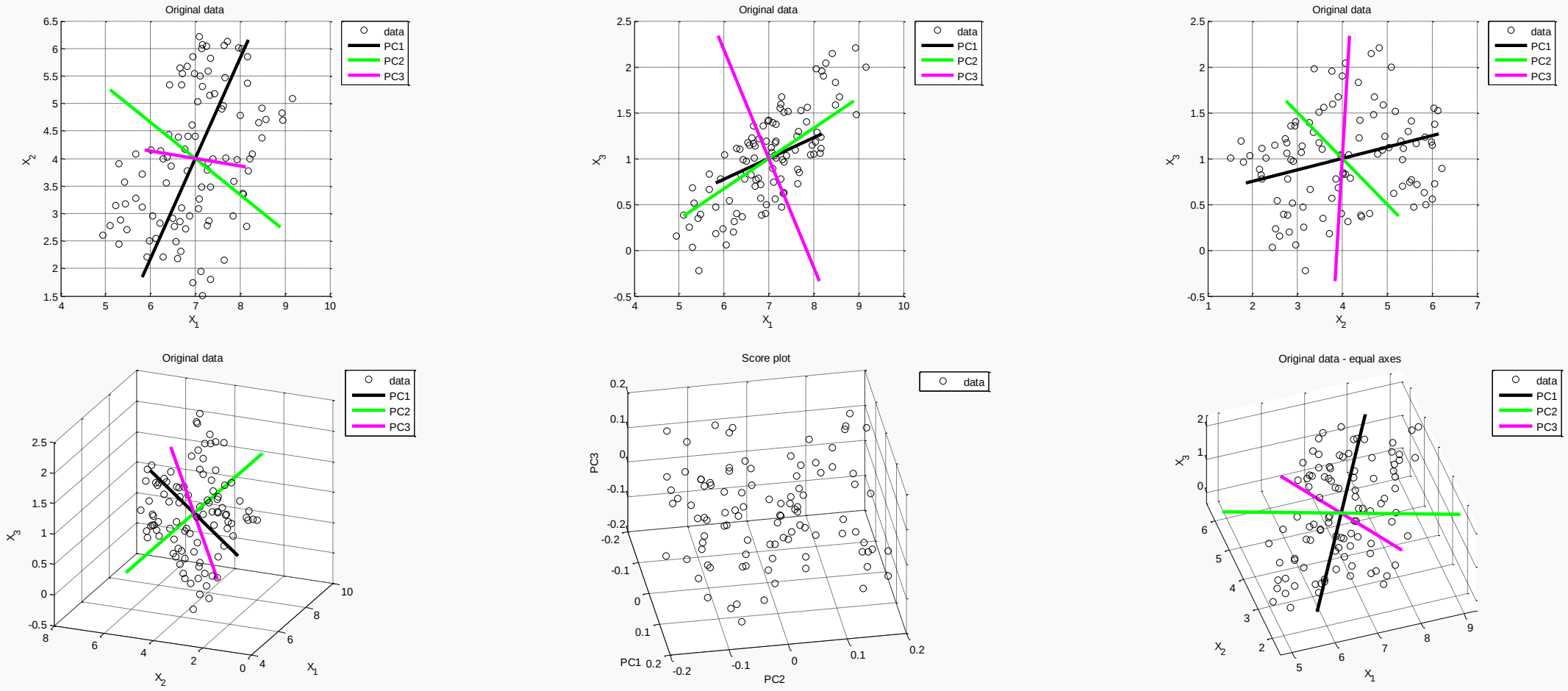


- Notice how PC2 is orthogonal to PC1



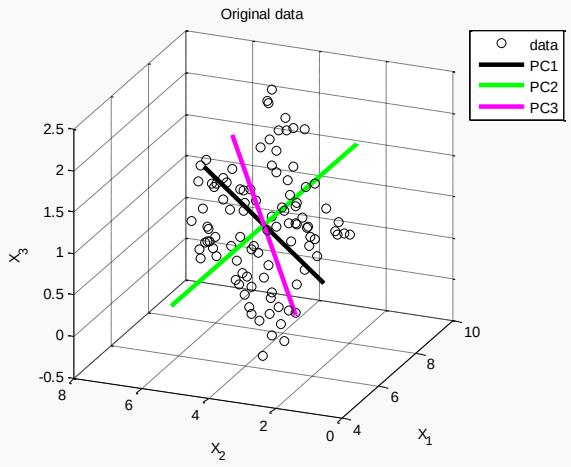
Again orthogonal

Principal Component Analysis (PC 3)



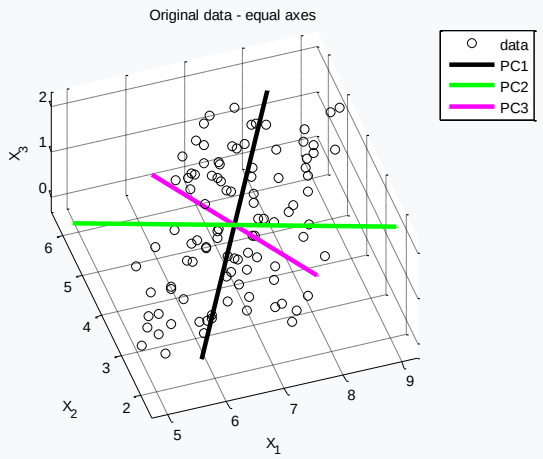
Centering and scaling

- Subtract average of each variable
- Mean of 0
- PCA finds direction of most variance



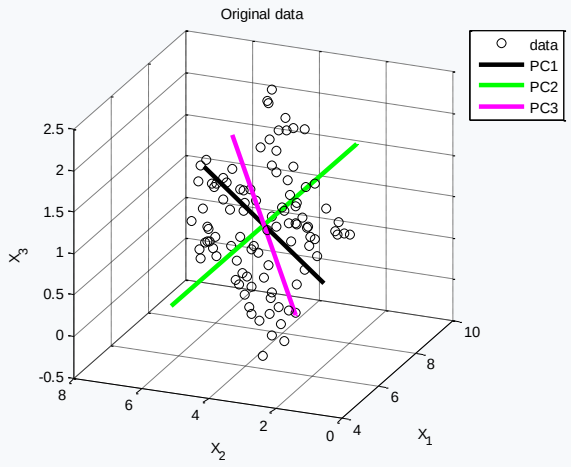
- The maths behind PCA assume the variables to be centered around 0
- Variance is spread around mean
- Mean-centering

- Difference in variance of variables
- Higher variance: more interesting for PCA



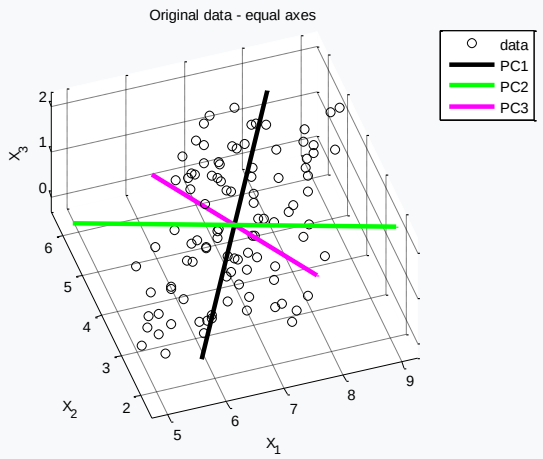
Centering and scaling

- Divide each mean-centered variable by standard deviation
- Variance of 1



- Making all variances equal
- Every variable has potential to be important as others
- ‘Unit variance’
- Combined with mean-centering = auto-scaling

- Difference in variance of variables
- Higher variance: more interesting for PCA



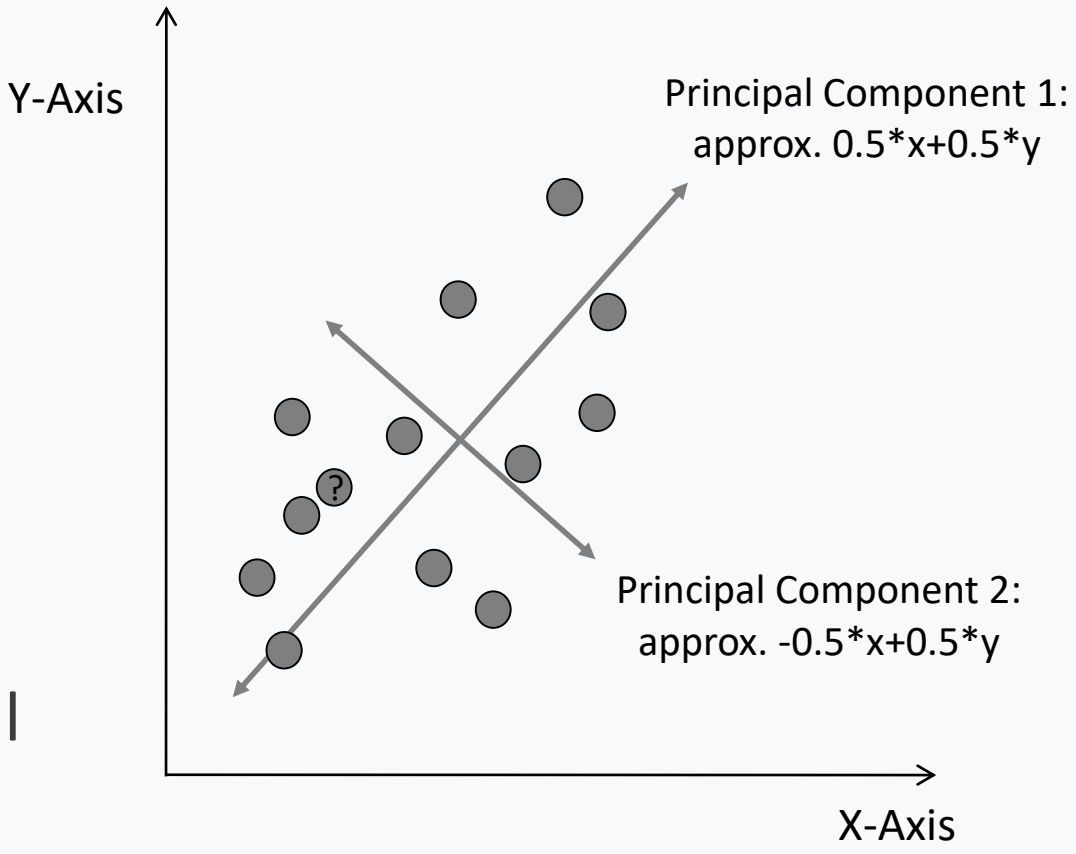
(more about scaling in a bit)

Principal Component Analysis

- Step 1: mean-center the data
- (Optional step 1b: scale the data)

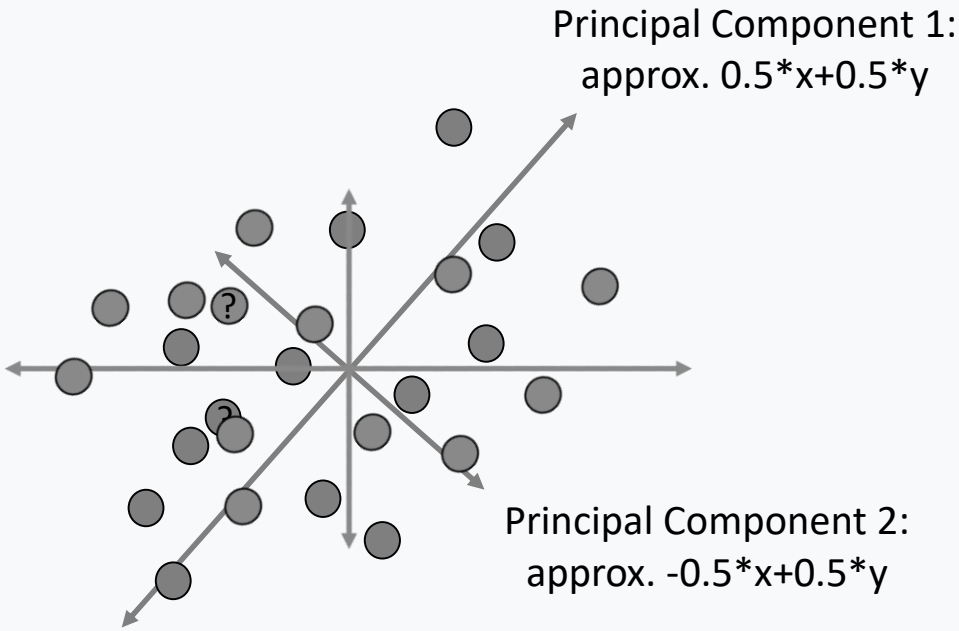
- Step 2: find direction with most variance (principal component 1)

- Step 3: find next direction with most (remaining) variance (orthogonal to PC1)
- (Potential steps 4-n): find next direction with most variance remaining (orthogonal to PC1, PC2, …)



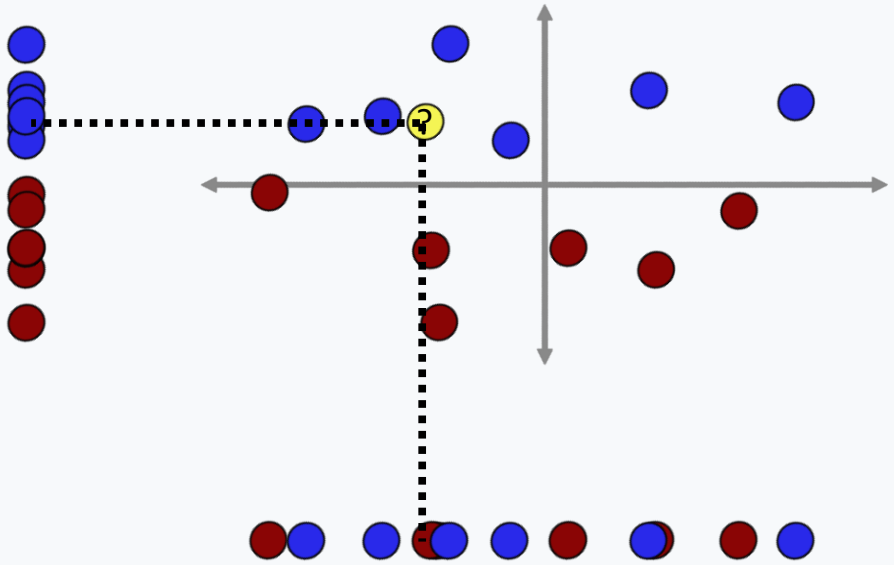
Principal Component Analysis

- PCs are ‘new’ variables
- Linear combinations of original variables = ‘latent’ variables
- Use PCs to define new axes (turn the data space)
- Same interpretation



Principal Component Analysis

- Do we need all axes in this classification example?



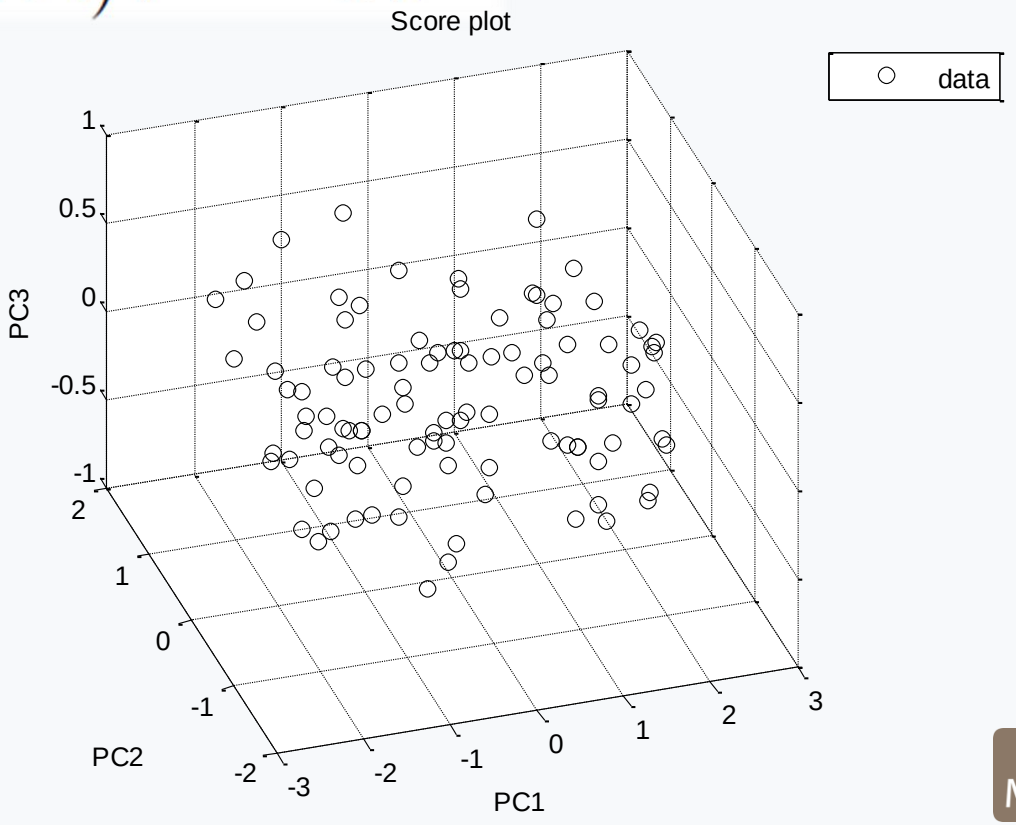
- Need less variables than before:
  - Data reduction

Data reduction

- Describing the original data (X) in another way

$$X = U\Sigma V^T = (U\Sigma)V^T = TP^T$$

- Score plot: similarity of samples
- Scores: ***T*** (3 components)
- Variance explained:
  - PC1: 68.36%
  - PC2: 28.35%
  - PC3: 3.29%

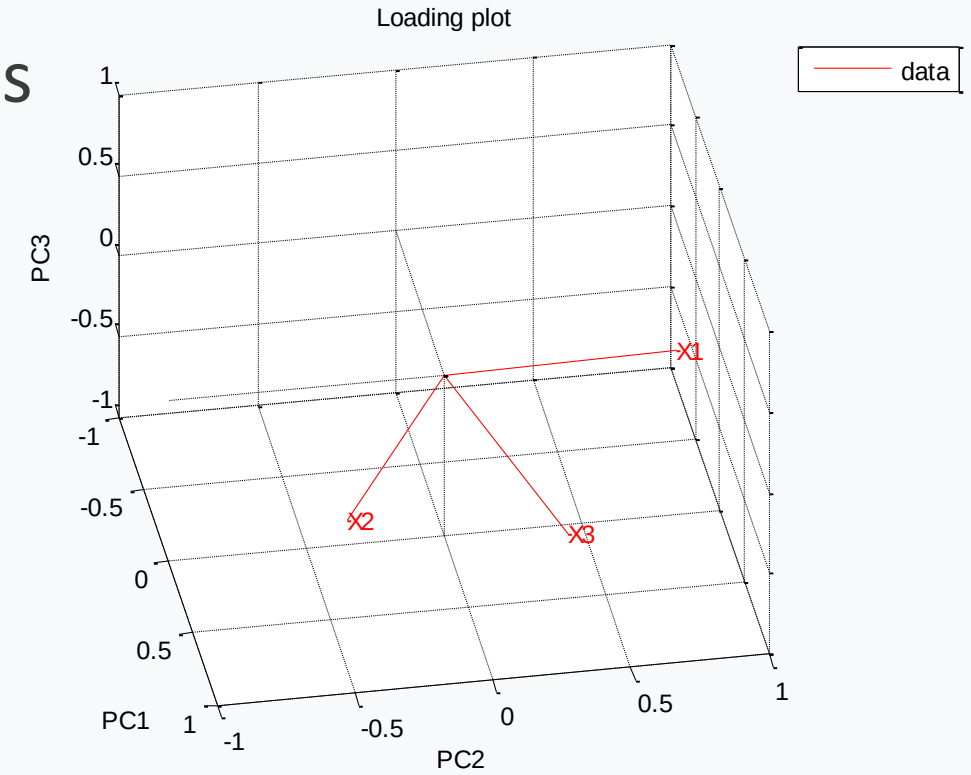


Data reduction

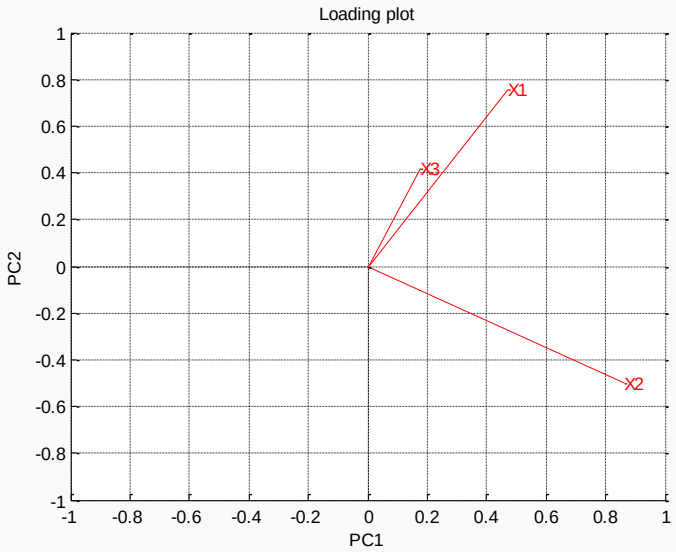
- Describing the original data (X) in another way

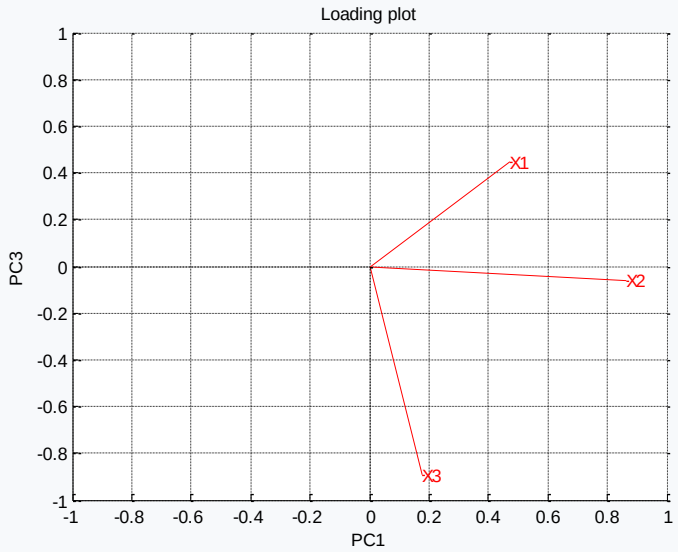
$$X = U\Sigma V^T = (U\Sigma)V^T = TP^T$$

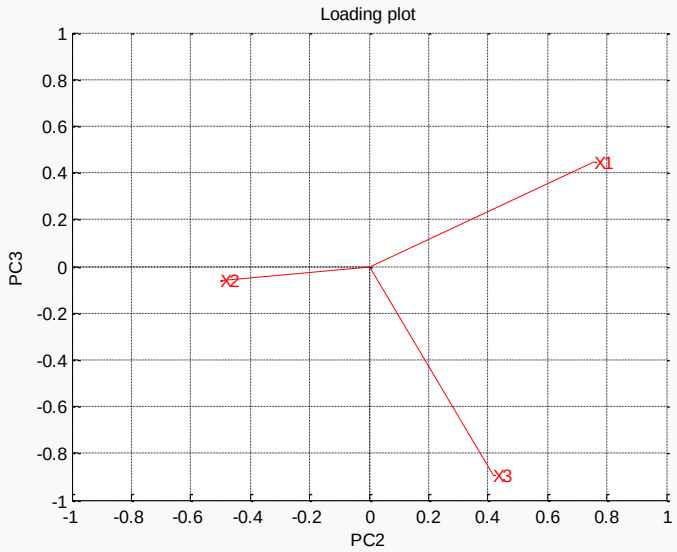
- Loading plot: similarity of variables
- Loadings: ***P*** (3 components)
- Variance explained:
  - PC1: 68.36%
  - PC2: 28.35%
  - PC3: 3.29%



Loading plot (2D)

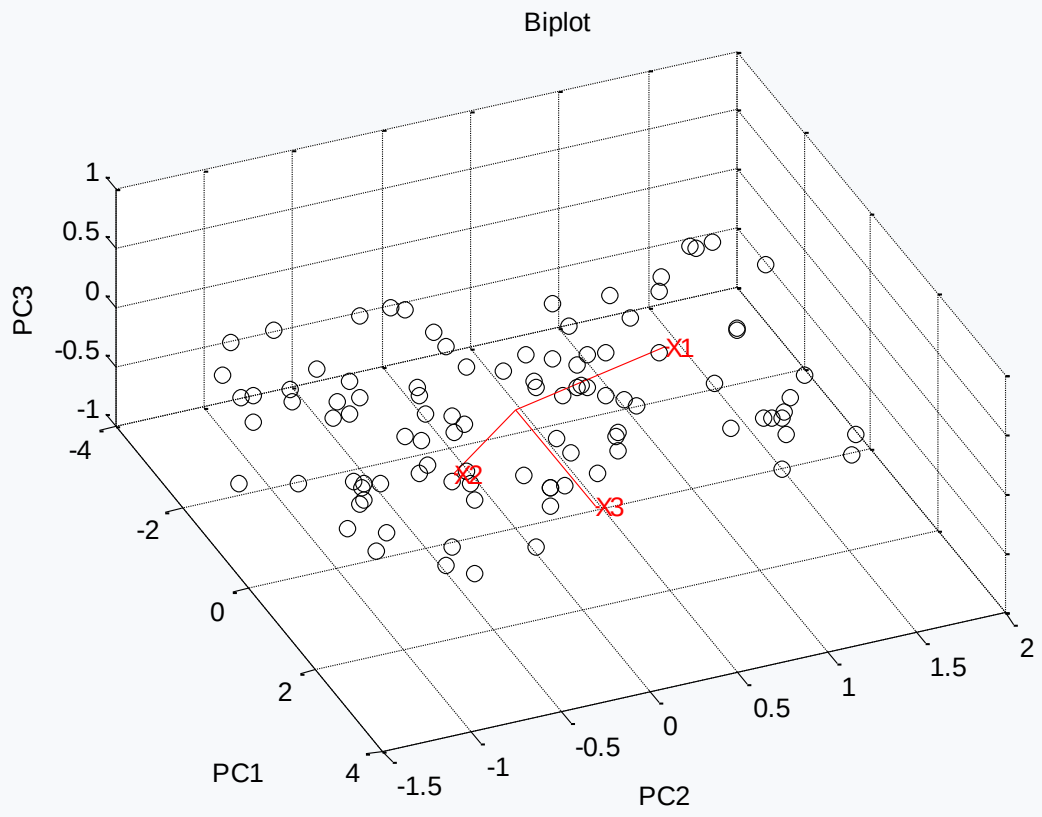


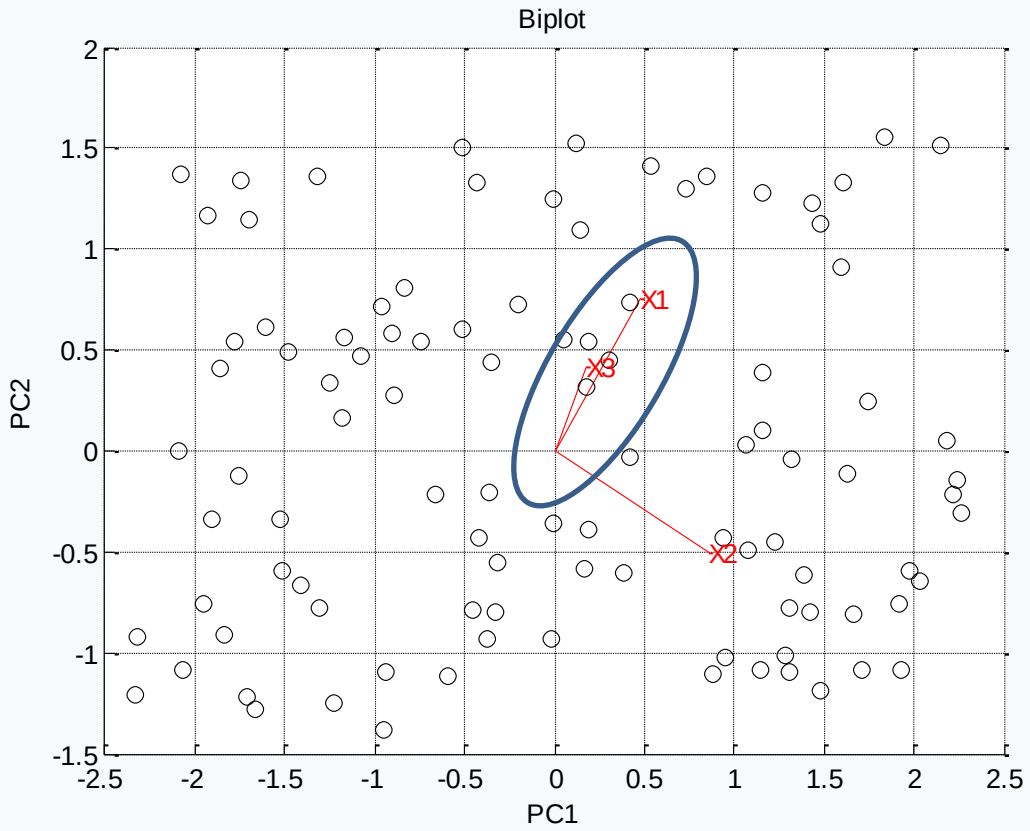




PC1+PC2 combined variance: 96.71%

Biplot (combining scores and loadings)

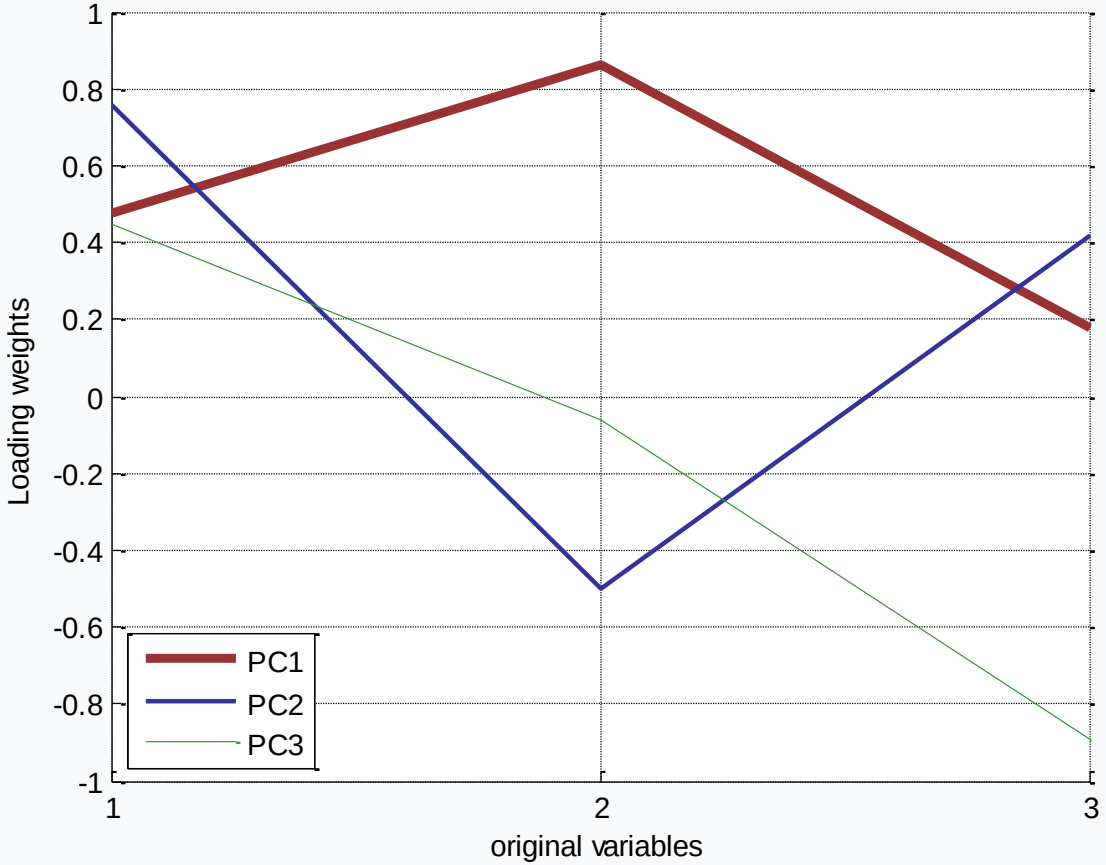




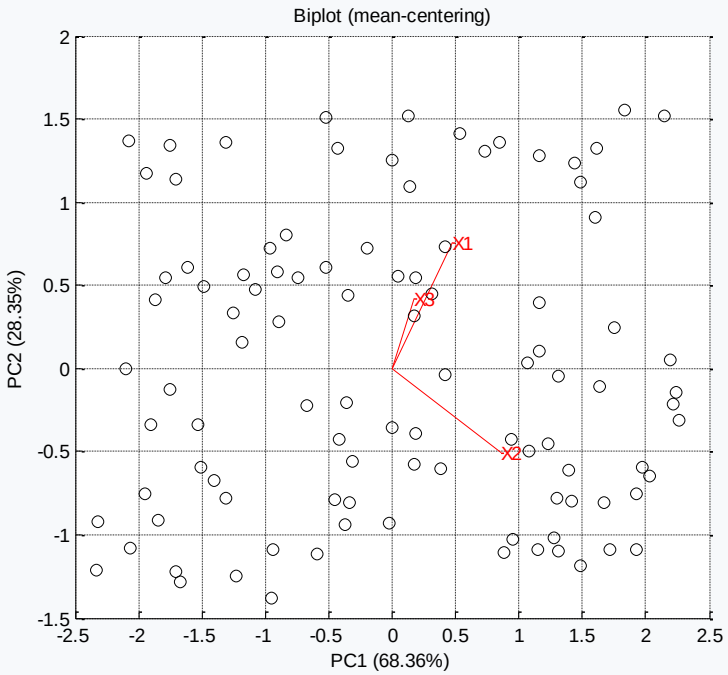
How many and which variables do we need?

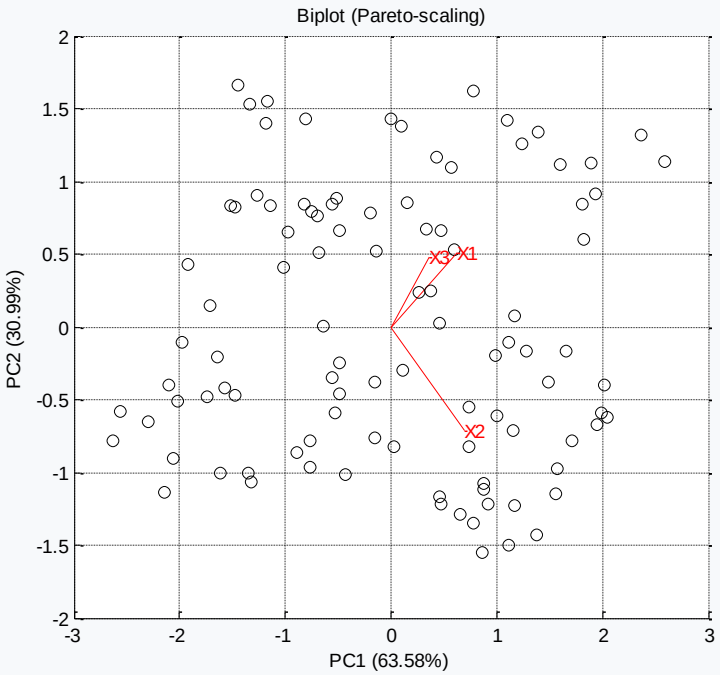
- PC1+PC2 combined variance: 96.71%
- Most of the data explained by 2 latent variables
- We had 3 variables, now just 2: data reduction

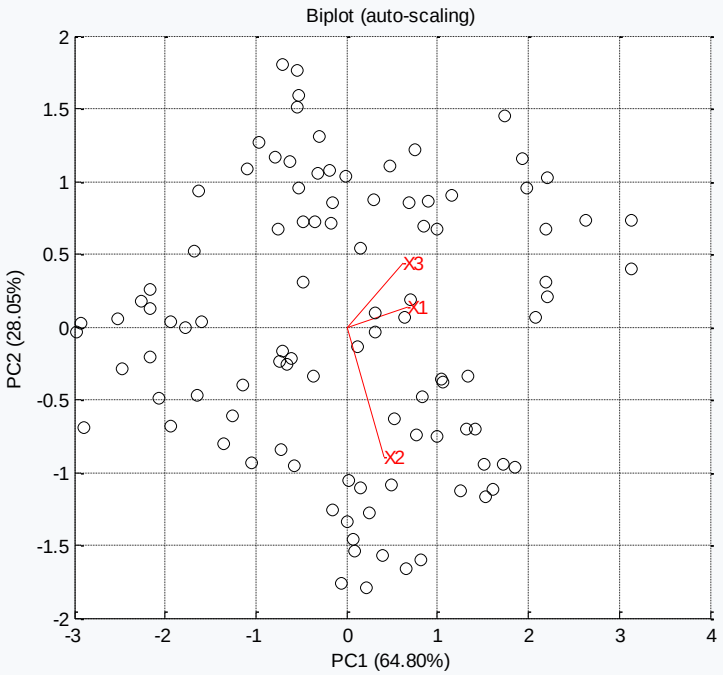
- Same extends to 1,000s of variables: PCs combine these into less variables



Scaling

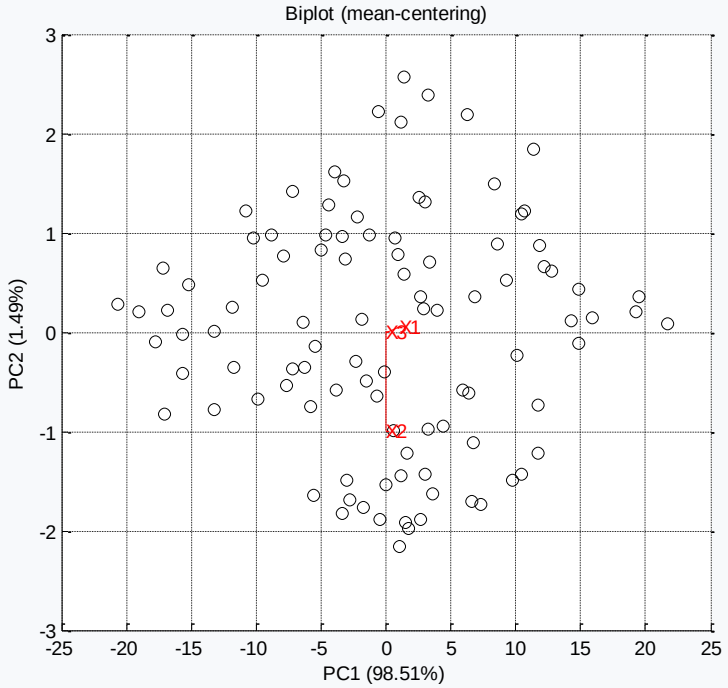


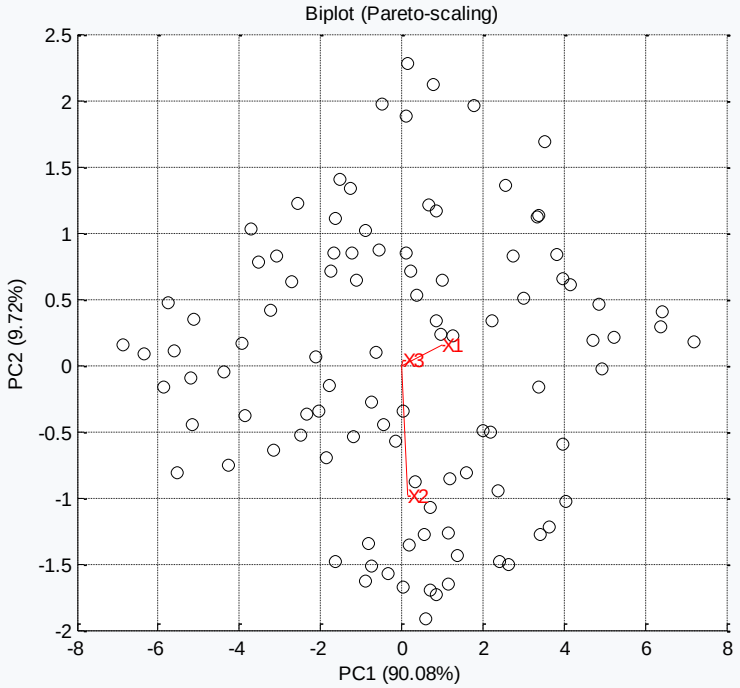


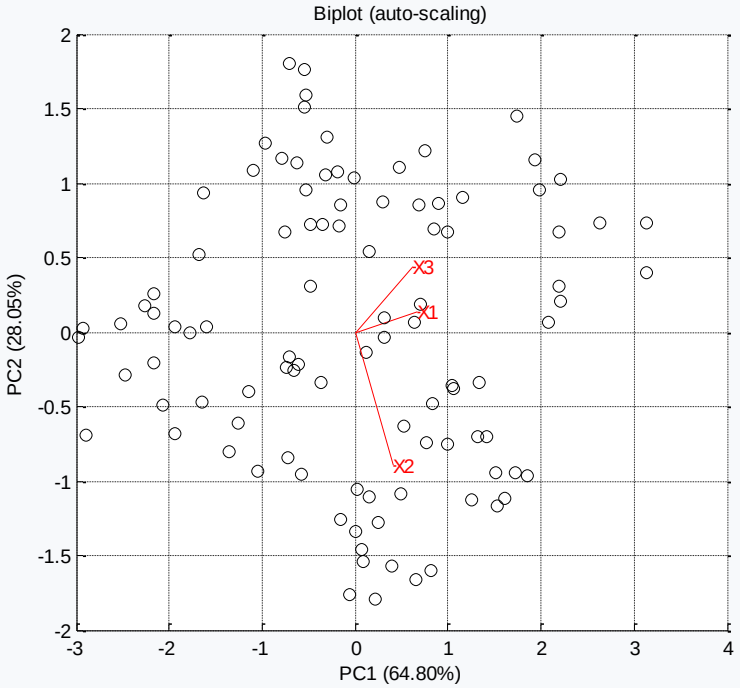


Means were different (X1 = 7, X2 = 4, X3 = 1), but standard deviations were relatively similar (X1 = 0.9, X2 = 1.2, X3 = 0.5)

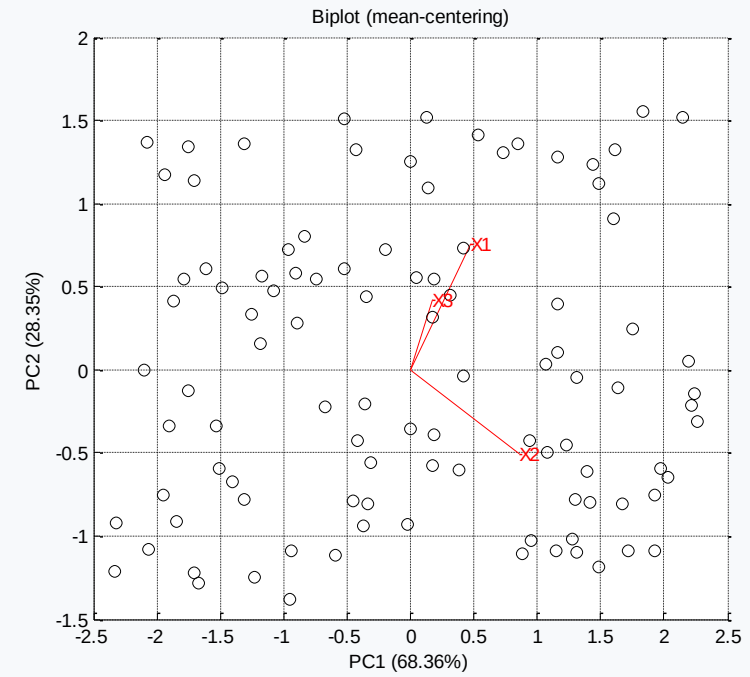
Scaling

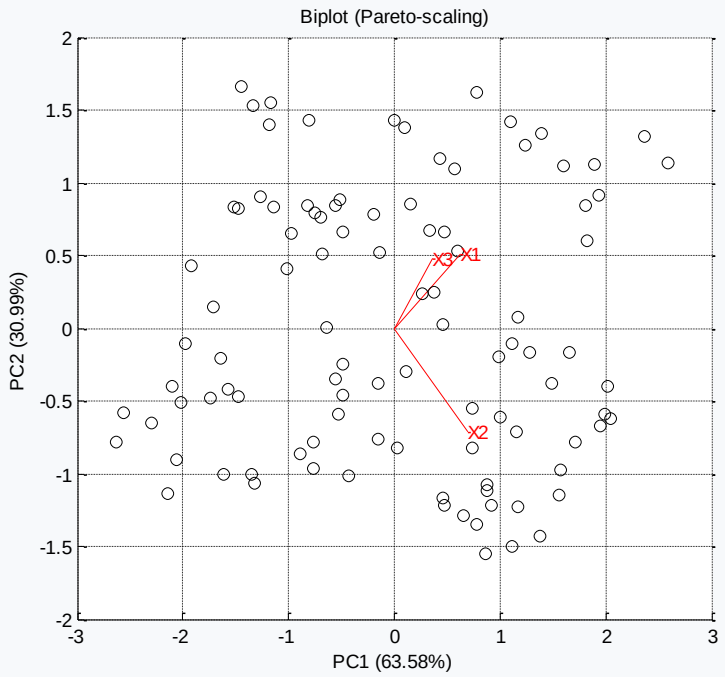


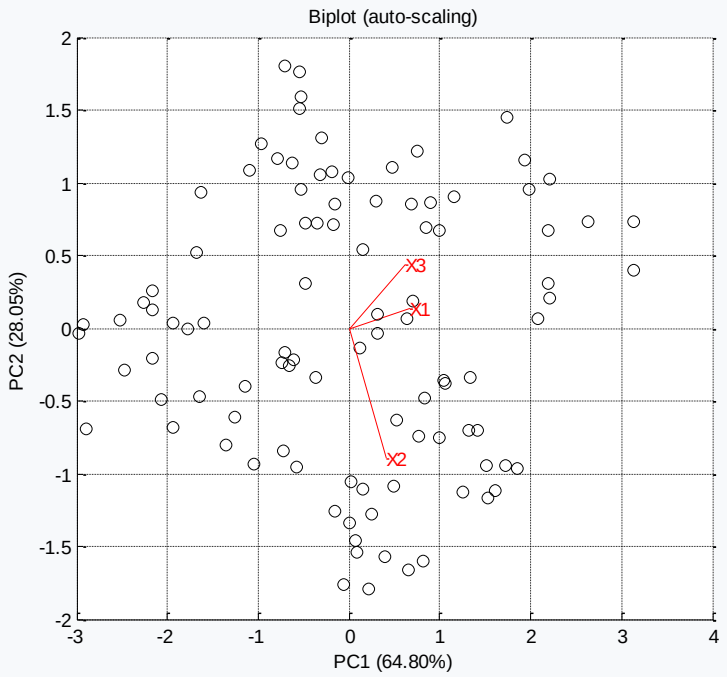


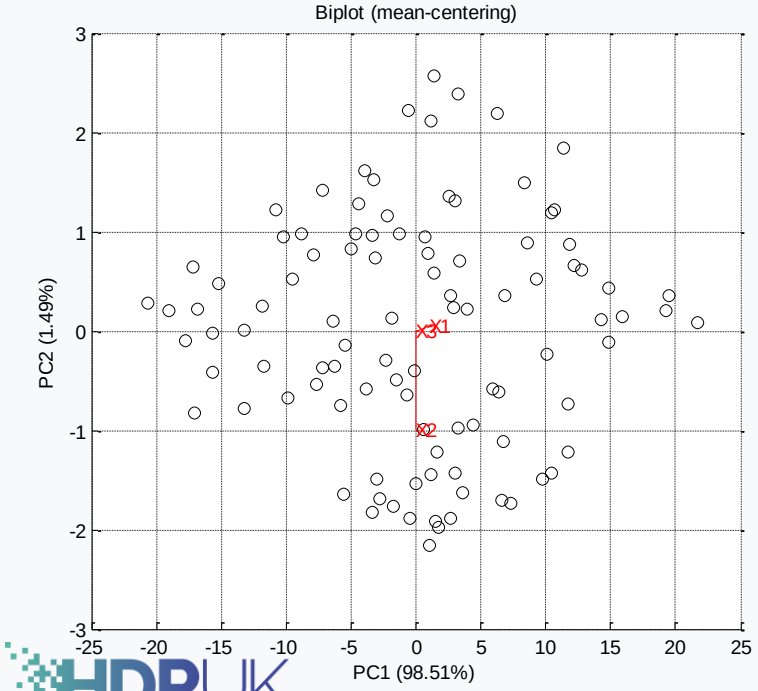


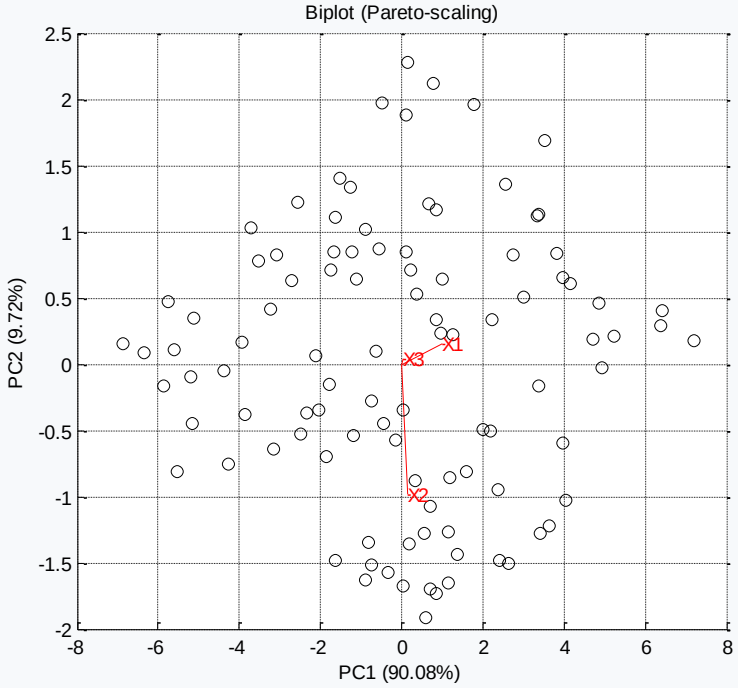
Means were different (X1 = 7, X2 = 4, X3 = 1), but variances were *very different* (X1 = 9.3, X2 = 1.2, X3 = 0.05)

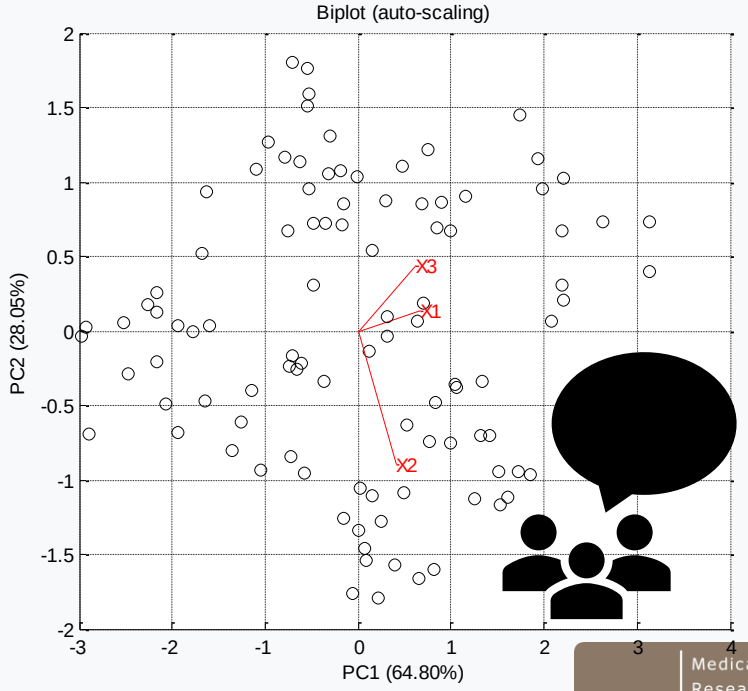












HDRUK
Health Data Research UK

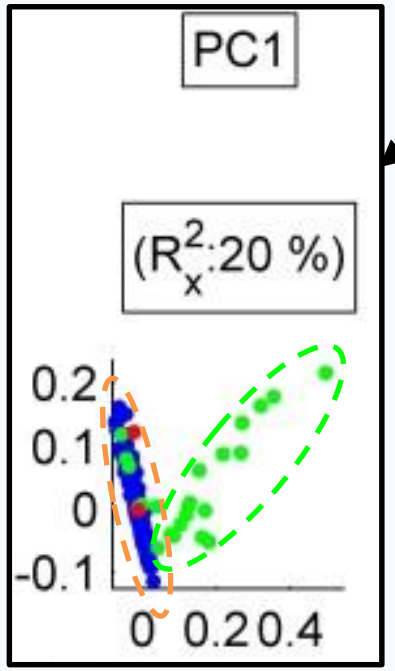
MRC Medical Research Council

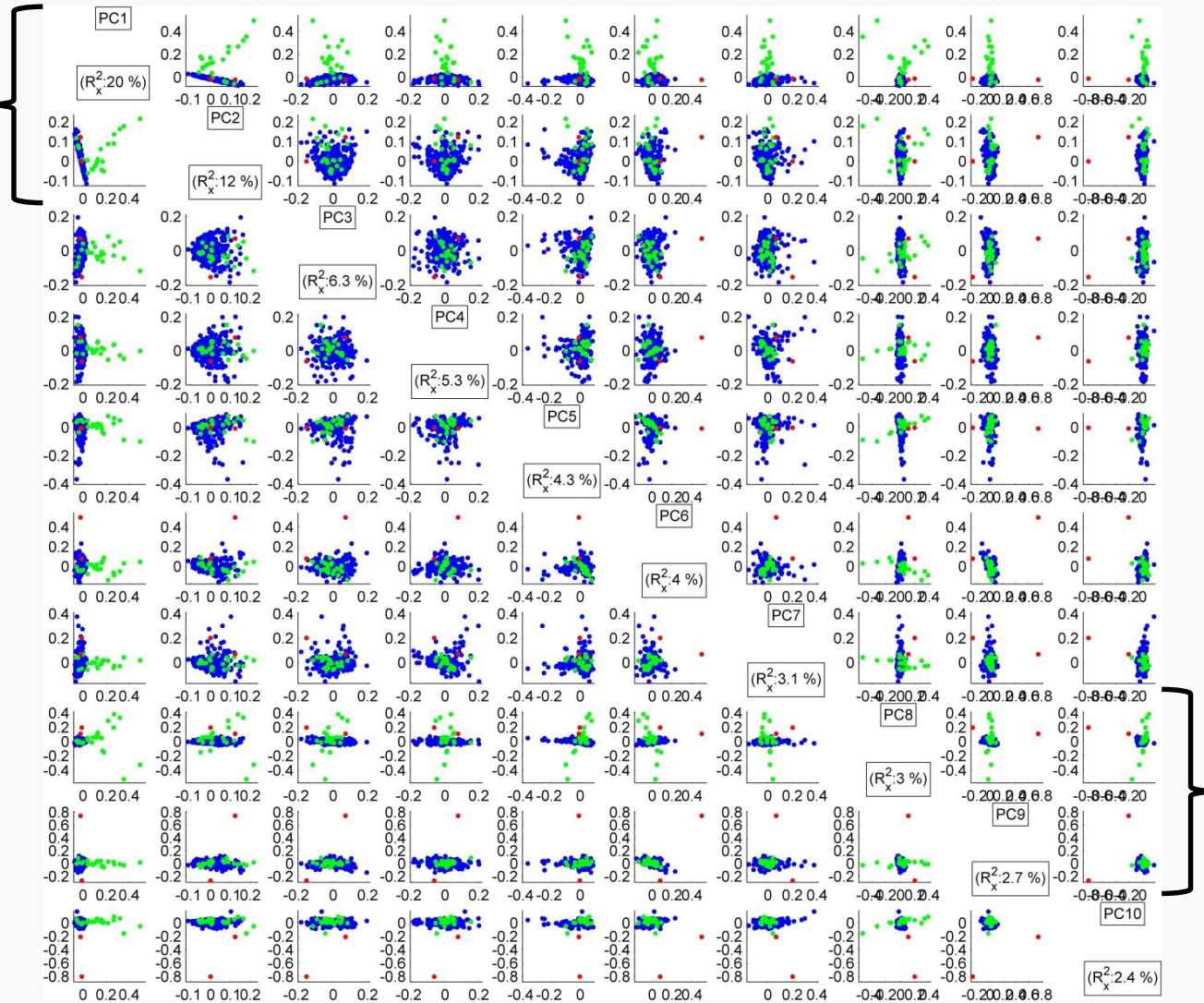
Scaling

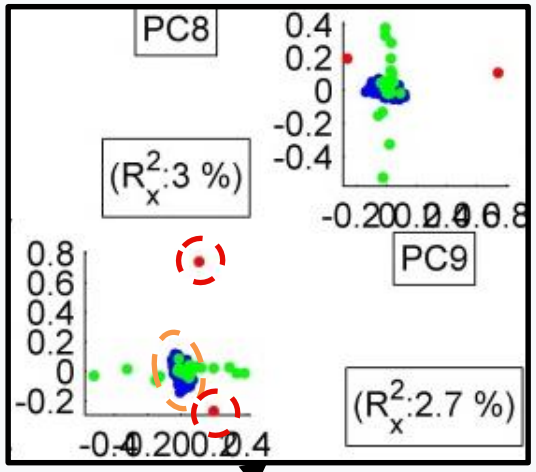
- Mean-centering: variables with high variance will dominate the model
- Auto-scaling: all variables are equally important (including noise variables)
- Pareto-scaling: intermediate between the above two (divide by square root of standard deviation)

- Other types exist:
  - Range scaling (divide by difference of highest and lowest value of each variable)
  - Log scaling (take the log of all values, be aware of values <1 and especially <0…)

Spotting outliers (faecal water NMR data)
PCA pairs plot





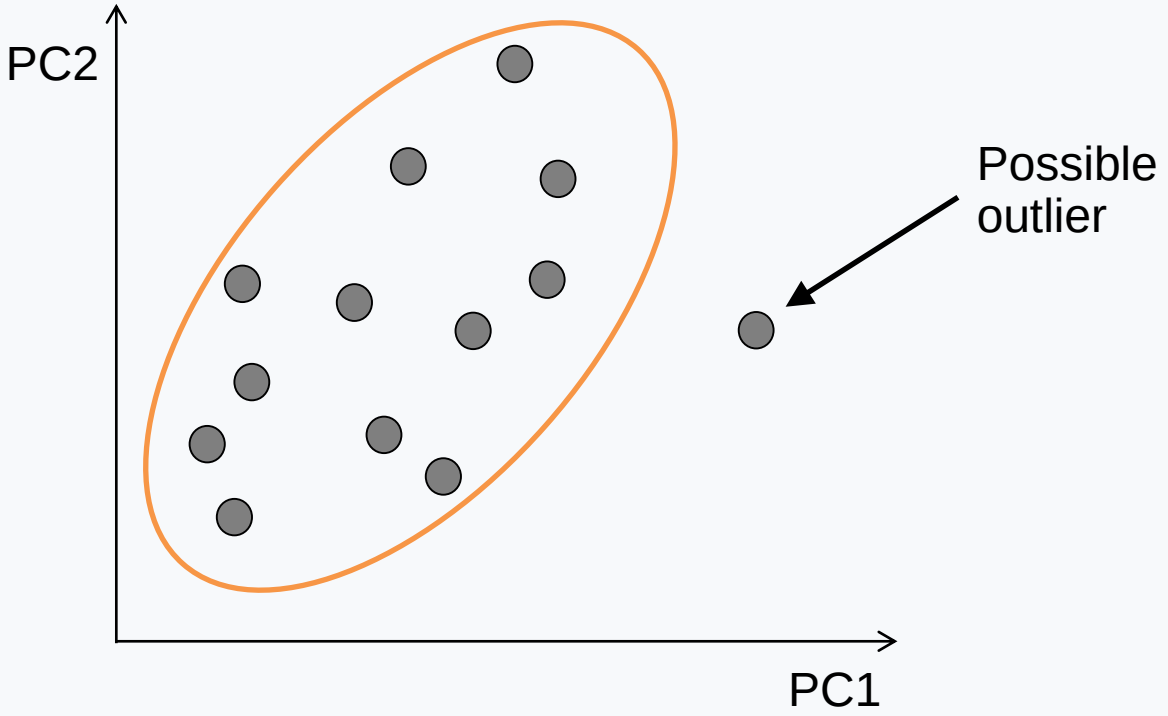


HDRUK Health Data Research UK

MRC Medical Research Council

Spotting outliers

- Hotelling's T^2 statistic
- Scores plot with two components: an ellipse
- Scores plot with 3 or more components: a (multidimensional) ellipsoid



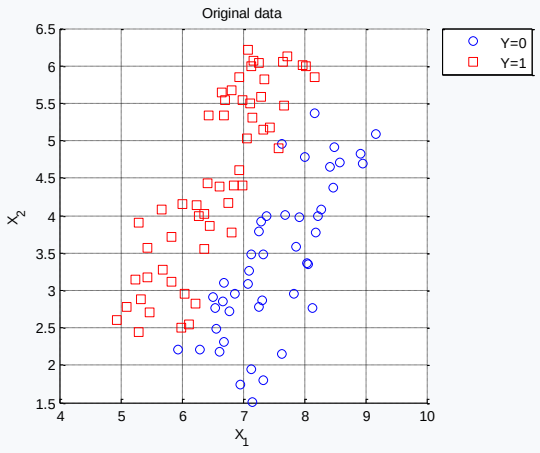
- Anything outside ellipse:
  - Potential outlier

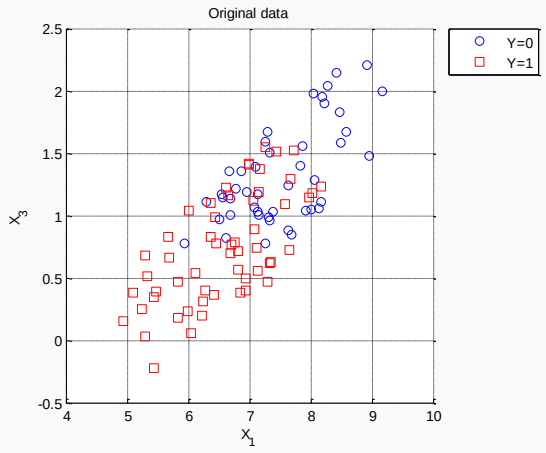
Unsupervised vs supervised

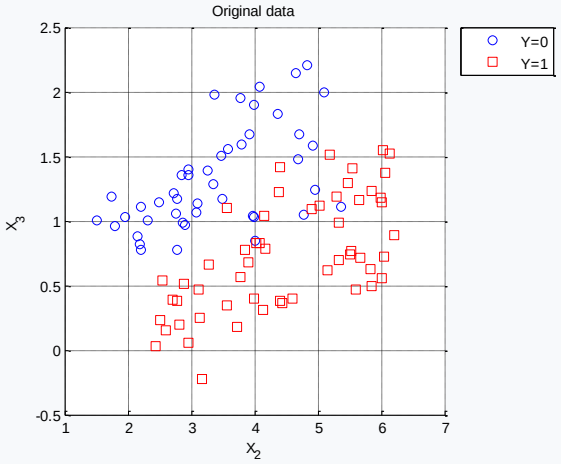
- PCA is unsupervised (a model that *projects*)
- But PCs can be used for supervised analysis

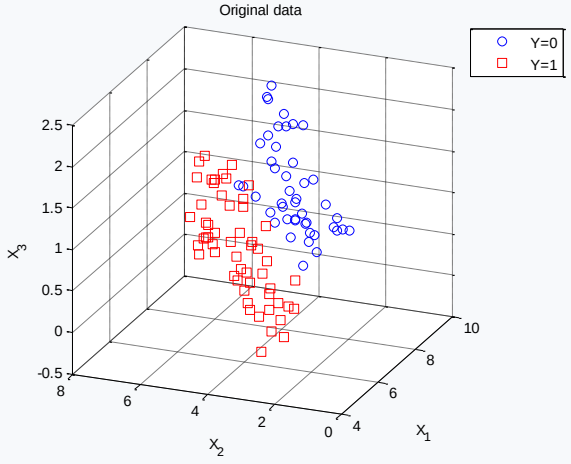
- For supervised analysis we use the class (outcome, ***Y***) information as well
- Principal Component Regression (PCR) and Principal Component Discriminant Analysis (PCDA) are models that try to *predict*

Supervised analysis
Example data (n = 100, p = 3)









Multiple Linear Regression (MLR)
Principal Component Regression (PCR)

Derivation 1 Multiple Linear Regression

INPUT $\boldsymbol{X}, \boldsymbol{Y}$

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \epsilon$$
$$e = \boldsymbol{Y} - \boldsymbol{X\beta}$$
$$e^T e = (\boldsymbol{Y} - \boldsymbol{X\beta})^T (\boldsymbol{Y} - \boldsymbol{X\beta})$$
$$e^T e = \boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{Y} - \boldsymbol{\beta}\boldsymbol{X}\boldsymbol{Y}^T + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta}$$
$$e^T e = \boldsymbol{Y}^T\boldsymbol{Y} - 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{Y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta}$$
$$\frac{d(e^T e)}{d\boldsymbol{\beta}} = -2\boldsymbol{X}^T\boldsymbol{Y} + 2\boldsymbol{X}^T\boldsymbol{X\beta} = 0$$
$$2\boldsymbol{X}^T\boldsymbol{X\beta} = 2\boldsymbol{X}^T\boldsymbol{Y} \longrightarrow \boldsymbol{X}^T\boldsymbol{X\beta} = \boldsymbol{X}^T\boldsymbol{Y}$$
$$\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

OUTPUT $\boldsymbol{\beta}$

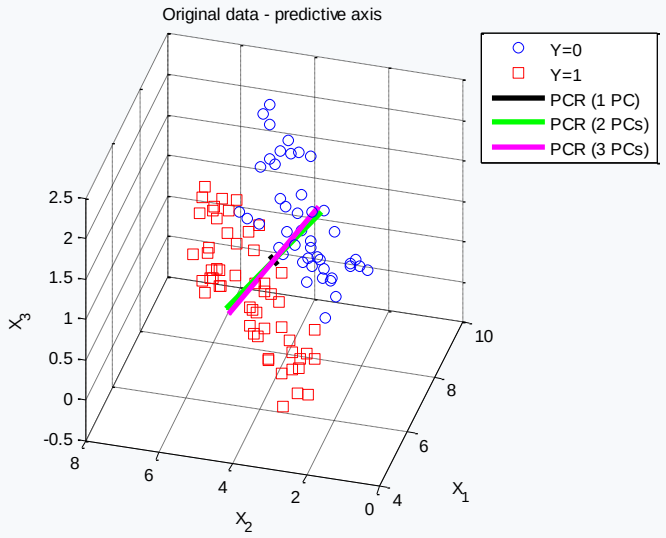
Derivation 2 Principal Component Regression

INPUT $\boldsymbol{T}, \boldsymbol{P}, \boldsymbol{Y}$

CONDITIONS $\text{rank } T = \text{rank } P = c \leq n - 1$

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{P}^T\boldsymbol{\beta} + \epsilon$$
$$\boldsymbol{P}^T\boldsymbol{\beta} = (\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{T}^T\boldsymbol{Y}$$
$$\boldsymbol{\beta} = \boldsymbol{P}(\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{T}^T\boldsymbol{Y}$$
$$\boldsymbol{\beta} = \boldsymbol{V}((\boldsymbol{U\Sigma})^T(\boldsymbol{U\Sigma}))^{-1}(\boldsymbol{U\Sigma})^T\boldsymbol{Y}$$
$$\boldsymbol{\beta} = \boldsymbol{V}(\boldsymbol{\Sigma}\boldsymbol{U}^T\boldsymbol{U\Sigma})^{-1}(\boldsymbol{U\Sigma})^T\boldsymbol{Y} = \boldsymbol{V}(\boldsymbol{\Sigma I\Sigma})^{-1}(\boldsymbol{U\Sigma})^T\boldsymbol{Y}$$
$$\boldsymbol{\beta} = \boldsymbol{V\Sigma}^{-2}\boldsymbol{\Sigma}\boldsymbol{U}^T\boldsymbol{Y} = \boldsymbol{V\Sigma}^{-1}\boldsymbol{U}^T\boldsymbol{Y}$$

OUTPUT $\boldsymbol{\beta}$



$$\boxed{\boldsymbol{X} = \boldsymbol{U\Sigma V}^T = (\boldsymbol{U\Sigma})\boldsymbol{V}^T = \boldsymbol{TP}^T}$$

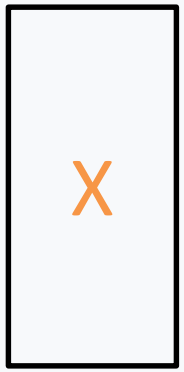
The problem with Multiple Linear Regression (Ordinary Least Squares)

$$\boldsymbol{\beta} = \boxed{(\boldsymbol{X}^T\boldsymbol{X})^{-1}}\boldsymbol{X}^T\boldsymbol{Y}$$

Only possible
for n > p

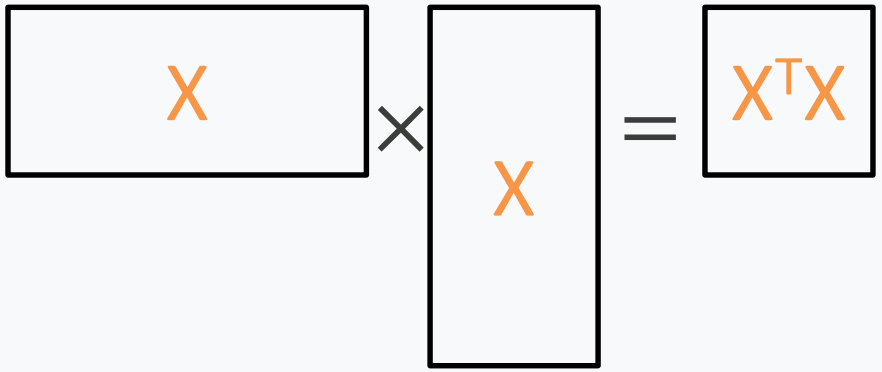
The problem with Multiple Linear Regression

- Suppose we have X with n > p



- Transpose of this matrix looks like

- And $X^T \times X =$



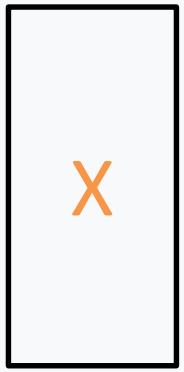
No problem:
smaller sized matrix than before
(lower dimension), we can calculate this!

The problem with Multiple Linear Regression

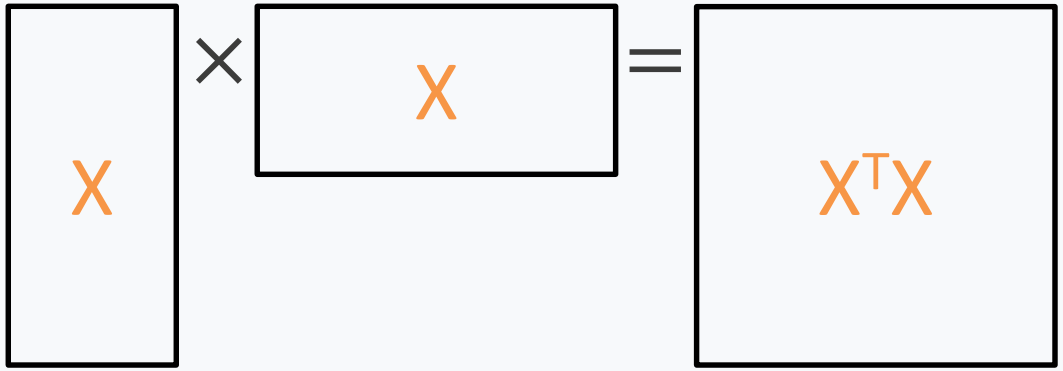
- Suppose we have X with n < p

  (like MS and NMR data…)

- Transpose of this matrix looks like



- And $X^T \times X =$



Problem!
Bigger sized matrix than before, we can not calculate this!
Too much uncertainty, bigger dimension: ***singular matrix***

Multiple Linear Regression and Principal Component Regression

Derivation 1 Multiple Linear Regression

INPUT $\boldsymbol{X}, \boldsymbol{Y}$

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \epsilon$$
$$e = \boldsymbol{Y} - \boldsymbol{X\beta}$$
$$e^T e = (\boldsymbol{Y} - \boldsymbol{X\beta})^T (\boldsymbol{Y} - \boldsymbol{X\beta})$$
$$e^T e = \boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{Y} - \boldsymbol{\beta}\boldsymbol{X}\boldsymbol{Y}^T + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta}$$
$$e^T e = \boldsymbol{Y}^T\boldsymbol{Y} - 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{Y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta}$$
$$\frac{d(e^T e)}{d\boldsymbol{\beta}} = -2\boldsymbol{X}^T\boldsymbol{Y} + 2\boldsymbol{X}^T\boldsymbol{X\beta} = 0$$
$$2\boldsymbol{X}^T\boldsymbol{X\beta} = 2\boldsymbol{X}^T\boldsymbol{Y} \longrightarrow \boldsymbol{X}^T\boldsymbol{X\beta} = \boldsymbol{X}^T\boldsymbol{Y}$$
$$\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

OUTPUT $\boldsymbol{\beta}$

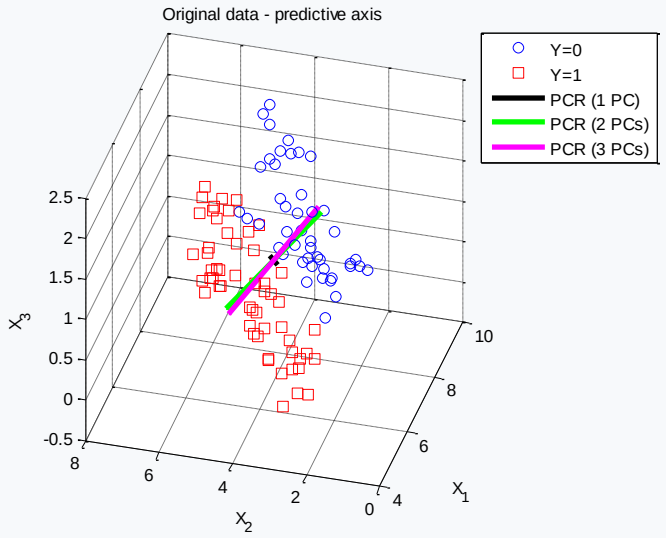
Derivation 2 Principal Component Regression

INPUT $\boldsymbol{T}, \boldsymbol{P}, \boldsymbol{Y}$

CONDITIONS $\text{rank } T = \text{rank } P = c \leq n - 1$

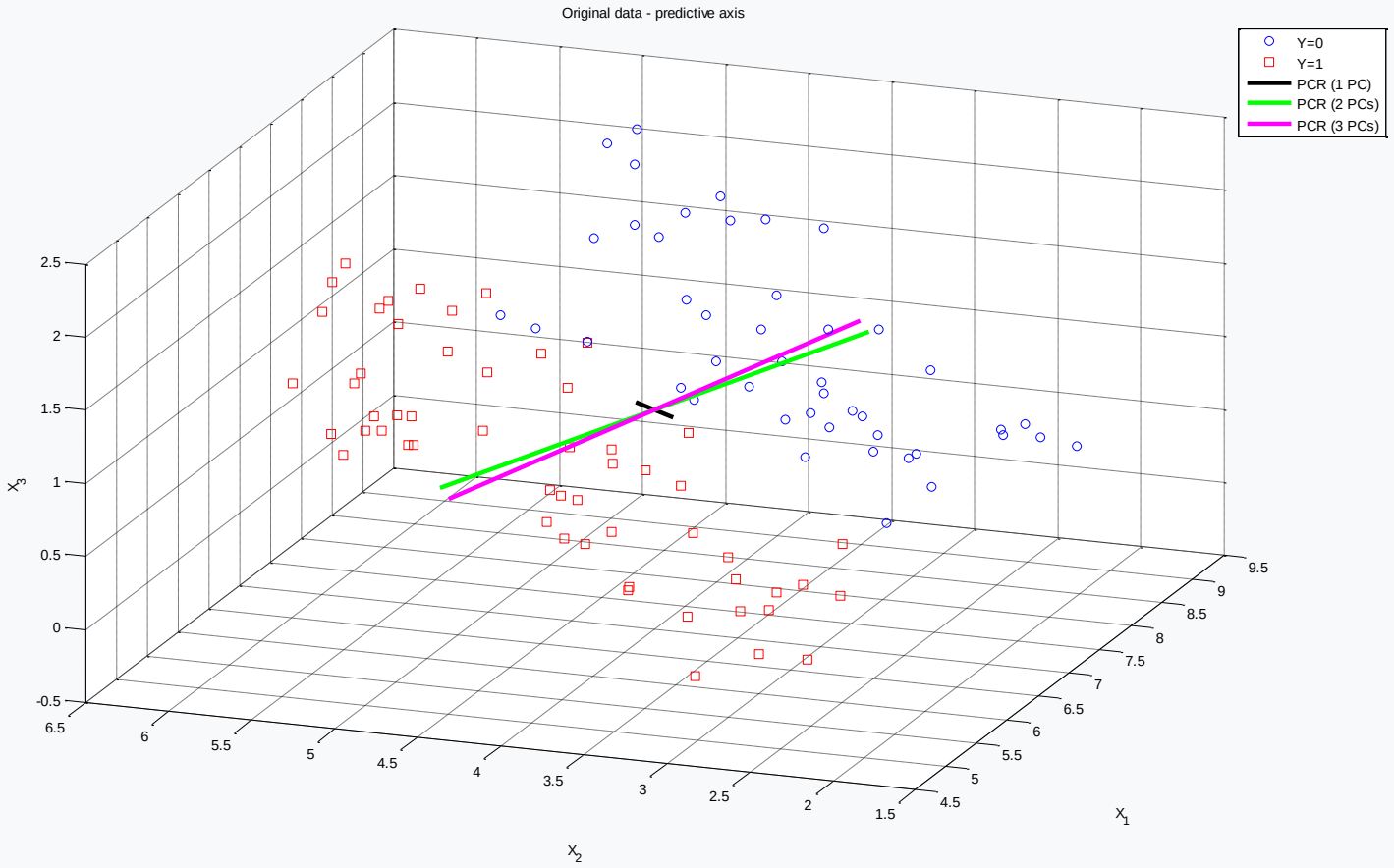
$$\boldsymbol{Y} = \boldsymbol{TP}^T\boldsymbol{\beta} + \epsilon$$
$$\boldsymbol{P}^T\boldsymbol{\beta} = (\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{T}^T\boldsymbol{Y}$$
$$\boldsymbol{\beta} = \boldsymbol{P}(\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{T}^T\boldsymbol{Y}$$
$$\boldsymbol{\beta} = \boldsymbol{V}((\boldsymbol{U\Sigma})^T(\boldsymbol{U\Sigma}))^{-1}(\boldsymbol{U\Sigma})^T\boldsymbol{Y}$$
$$\boldsymbol{\beta} = \boldsymbol{V}(\boldsymbol{\Sigma}\boldsymbol{U}^T\boldsymbol{U\Sigma})^{-1}(\boldsymbol{U\Sigma})^T\boldsymbol{Y} = \boldsymbol{V}(\boldsymbol{\Sigma I\Sigma})^{-1}(\boldsymbol{U\Sigma})^T\boldsymbol{Y}$$
$$\boldsymbol{\beta} = \boldsymbol{V\Sigma}^{-2}\boldsymbol{\Sigma}\boldsymbol{U}^T\boldsymbol{Y} = \boldsymbol{V\Sigma}^{-1}\boldsymbol{U}^T\boldsymbol{Y}$$

OUTPUT $\boldsymbol{\beta}$

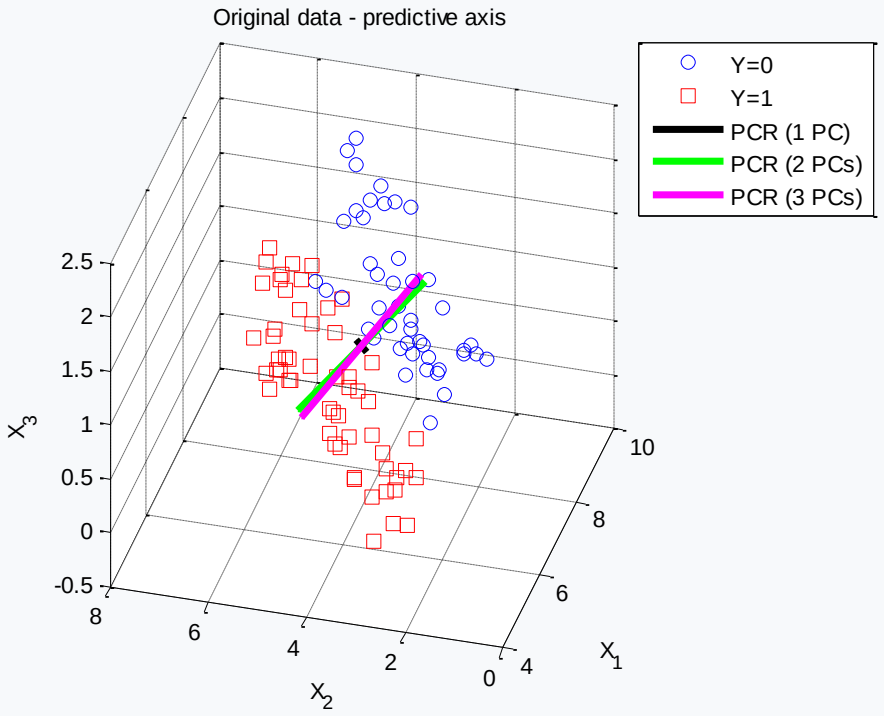


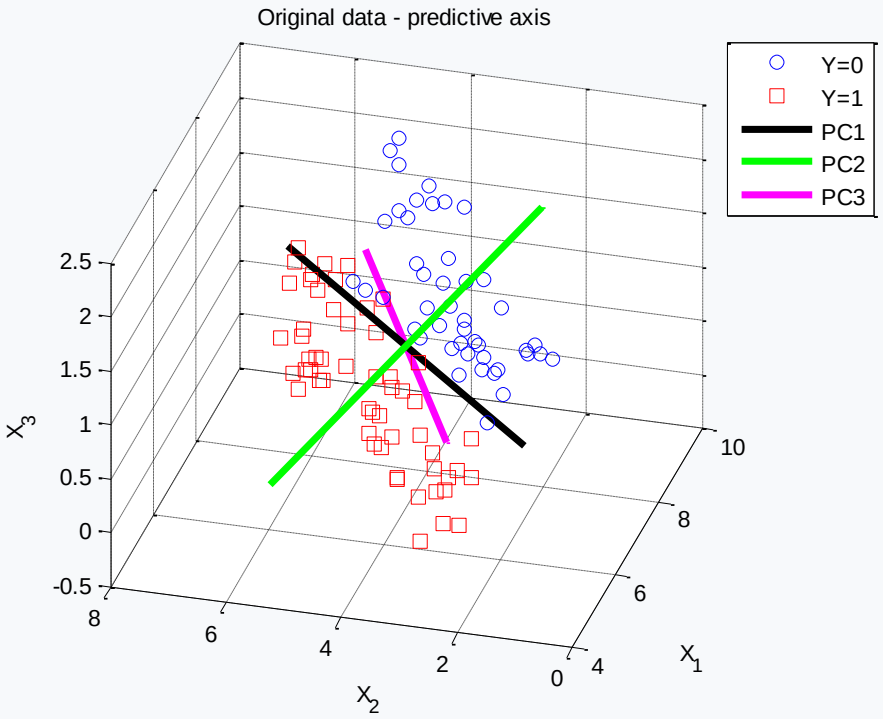
$$\boxed{\boldsymbol{X} = \boldsymbol{U\Sigma V}^T = (\boldsymbol{U\Sigma})\boldsymbol{V}^T = \boldsymbol{TP}^T}$$

Principal Component Regression



Principal Component Regression





Comparing the two methods

- MLR uses all 3 variables
  - 100% variance explained
- Goodness of fit: 0.79 (0.7897 exact)

$$R^2{}_Y = 1 - \frac{\sum_{i=1}^{n} \left(\hat{\boldsymbol{Y}}_i - \boldsymbol{Y}_i\right)^2}{\sum_{i=1}^{n} \left(\boldsymbol{Y}_i - \bar{y}\right)^2}$$

- PCR uses 1, 2 or 3 components
  - PC1 only: 68.36%
  - PC1 and 2: 96.71%
  - All 3 PCs: 100% of variance
- Goodness of fit:
  - PC1 only: 0.02 (bad model)
  - PC1 and 2: 0.79 (0.7876 exact)
  - All 3 PCs: 0.79 (0.7897 exact)

$R^2{}_Y$ = Goodness of fit of the model (to itself)

Validation

- Goodness of fit is self-fulfilling prophecy
- The model is fit to the data, so *that* data will always be predicted ($\hat{Y}$) as good as it can be: it is *trained* this way

$$R^2{}_Y = 1 - \frac{\sum_{i=1}^{n} (\hat{\boldsymbol{Y}}_i - \boldsymbol{Y}_i)^2}{\sum_{i=1}^{n} (\boldsymbol{Y}_i - \bar{y})^2}$$

- How well can it predict 'in the real world'?

- Require independent data set to be predicted ($\hat{Y}^*$) as *test*
- Evaluate goodness of prediction ($Q^2{}_Y$)

$$Q^2{}_Y = 1 - \frac{\sum_{i=1}^{n} \left(\hat{\boldsymbol{Y}}_i{}^* - \boldsymbol{Y}_i{}^*\right)^2}{\sum_{i=1}^{n} (\boldsymbol{Y}_i{}^* - \bar{y})^2}$$

Option 1: completely independent data set

- Data acquired using same technology as training data
- Data curated in the same way as training data
- Data processed in the same way as training data
- Data from similar population sample as target population
- Data cannot be the training data

- Pros: completely independent, best option
- Cons: studies often not designed with this in mind, expensive, do not always know how many samples are needed, difficult to obtain otherwise

•Option 2: split available data in two (hold out)

- All data is acquired in the same way
- A proportion of data is set aside (randomly) = test set
- Remainder is training set
- Model evaluated as before using this test set

- Pros: independent, same experimental design
- Cons: are enough samples left in training set, how random is the random split

Option 3: leave-one-out cross-validation

- All data is acquired in the same way
- One sample is the test set ('leave one (sample) out' of training set)
- Remainder is training set
- Model evaluated as before using this test set

- Pros: unbiased, same experimental design, big(ger) training set, good for small datasets, no random split
- Cons: training sets are related (overlap), predictions have high variability, for large datasets not very computationally efficient

Option 4: k-fold cross-validation

- Same as option 2, except the splitting is repeated
- All data is split randomly in k ways (e.g. k = 7)
- This creates 7 partitions, each is test set once
- Each partition is part of 6 training models
- Total of 7 models

- Pros: same experimental design, all samples are used in training and test sets, not relying on one model/split
- Cons: need to combine 7 models, how random is the random split

Option 5: do cross-validation many times

- Choose a number of times to run a model (say 100 or 1,000)
- Each time split the data randomly into training and test set
- Q^2_Y is evaluated across the test sets, samples are predicted in multiple models
- Monte Carlo cross-validation

- Pros: same as option 4, the more models means the less we need to worry about how random the random split is
- Cons: takes more time to calculate

Related: repeated cross-validation is repeating a full cross validation a number of times

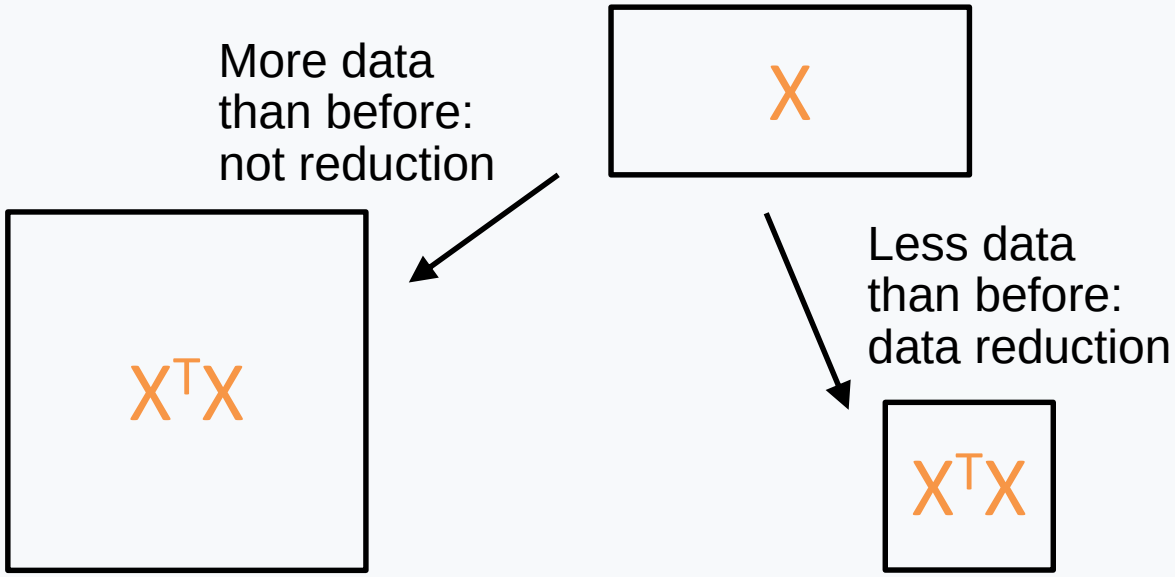
Evaluating how good a model is

- High R^2_Y is not everything
- High Q^2_Y is more important
- How high is high?
- The closer Q^2_Y is to R^2_Y the better

- High R^2_Y and low Q^2_Y? Overfitting (too much like training, not general enough)
- Low R^2_Y and low Q^2_Y? Underfitting (have not captured the data structure)

- One strategy to decide on a cut-off: randomly scramble your outcome (again: many times) and calculate the models in the same way
- How many times are the random models better than the actual model? Lower is better (empirical *P*-value)

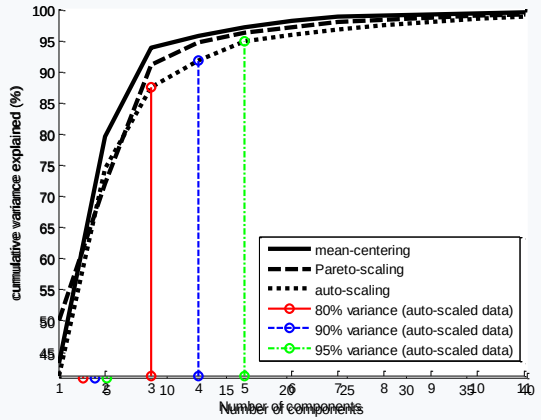
How many components do we need?

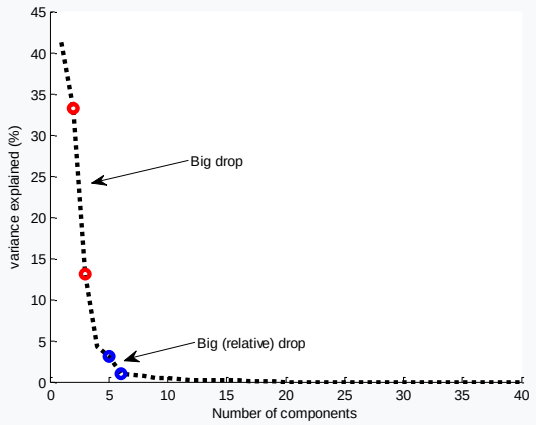


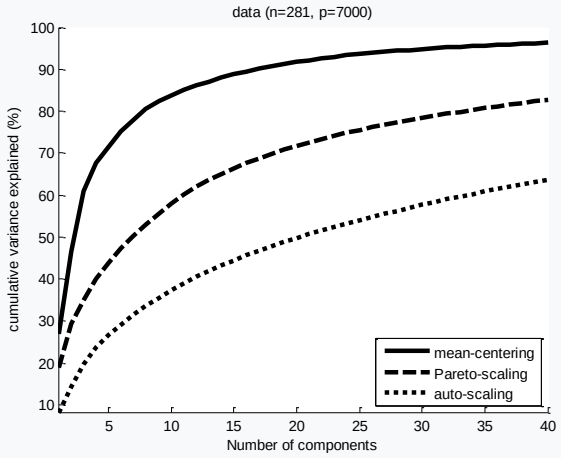
- If data is 'wide' (n samples and p variables, with n<p), there are a maximum of 'n' PCs we can calculate (that are orthogonal)

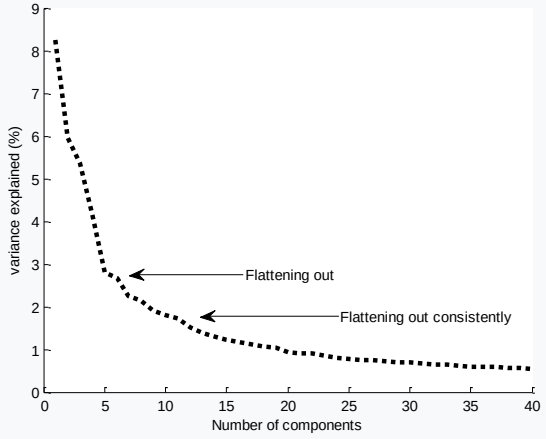
- No need for all n PCs
- Two simple (and one tricky) approaches:
- Keep PCs that explain 80, 90 or 95% of the total variance (arbitrary)
- Elbow plot of variance explained
- Leave-one-out cross-validation and calculate the reconstruction error of the left out sample

Selecting number of components









- Reconstruction of each sample
- Calculate PCA on all data except one (training = X, test = x): $X=TP^T$
- Calculate score: $t=x^TP$
- Calculate projected data: $tP^T = x^TPP^T$
- Error = $x - x^TPP^T$
- Do this for all samples, for different numbers of components

- Pro: pick the number of PCs with lowest error
- Con: takes a long time…

Summary

- Unsupervised: make no assumptions about groupings
- Visualize data: score plots, loading plots and biplots
- Data reduction: select number of PCs to use
- Outlier detection: Hotelling's T^2 on reduced data

- Scaling changes the data:
  - mean-centering – variables with high variance most important in first few PCs
  - auto-scaling – all variables equally important (including noise)

- Great way to inspect your data before doing further analyses