MONASH University

ETC3250/5250: Introduction to Machine Learning

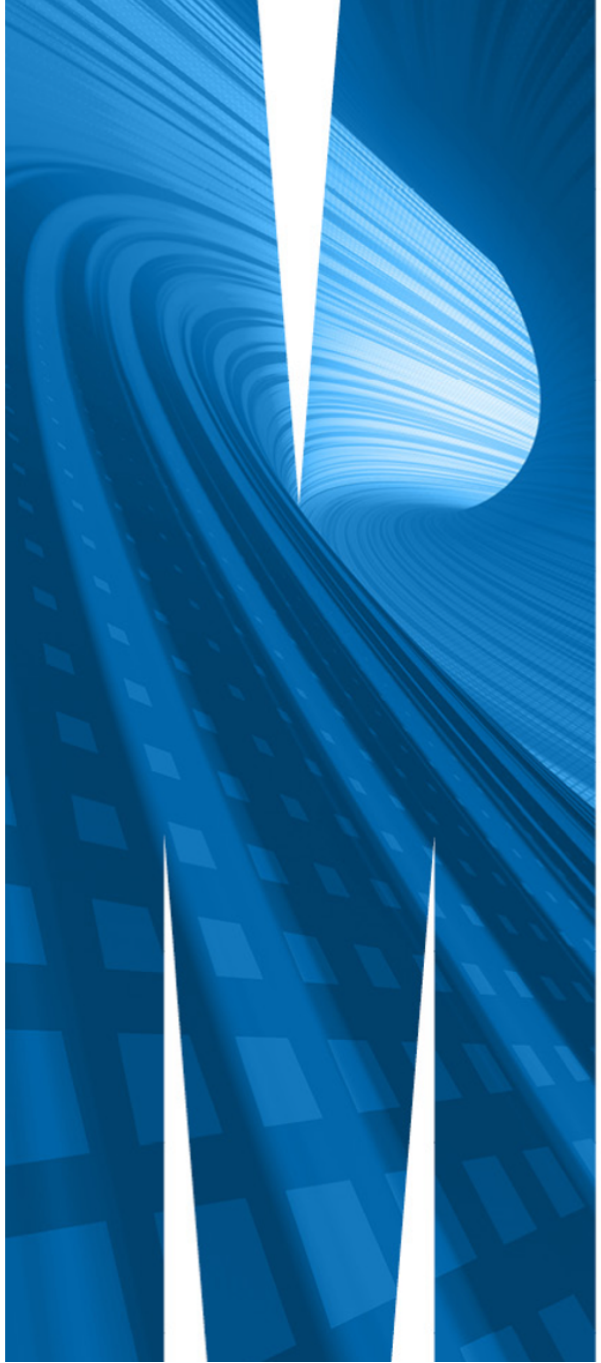
Dimension reduction

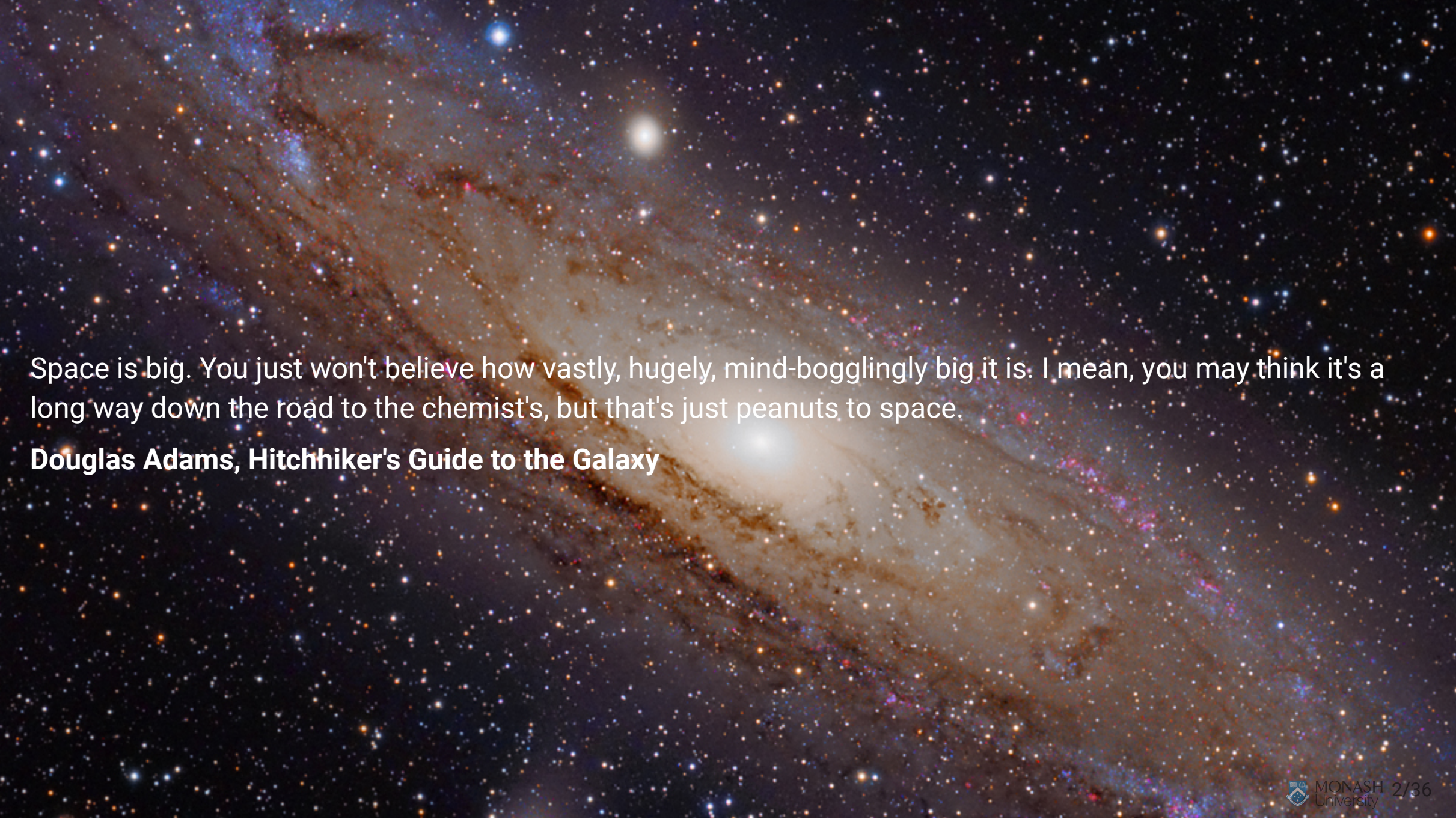
Lecturer: *Professor Di Cook*

Department of Econometrics and Business Statistics

ETC3250.Clayton-x@monash.edu

Week 4b





Space is big. You just won't believe how vastly, hugely, mind-bogglingly big it is. I mean, you may think it's a long way down the road to the chemist's, but that's just peanuts to space.

Douglas Adams, Hitchhiker's Guide to the Galaxy

High Dimensional Data

Remember, our data can be denoted as:

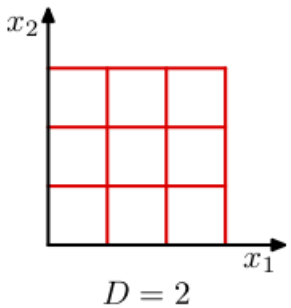
$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}, \quad \text{where } x_i = (x_{i1}, \ldots, x_{ip})^T$$

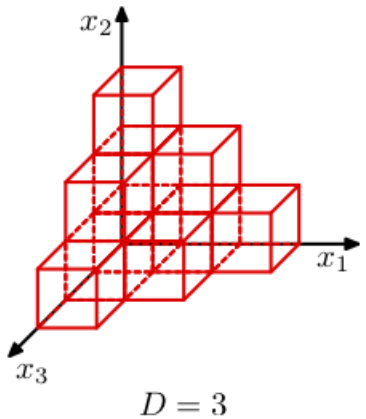
then

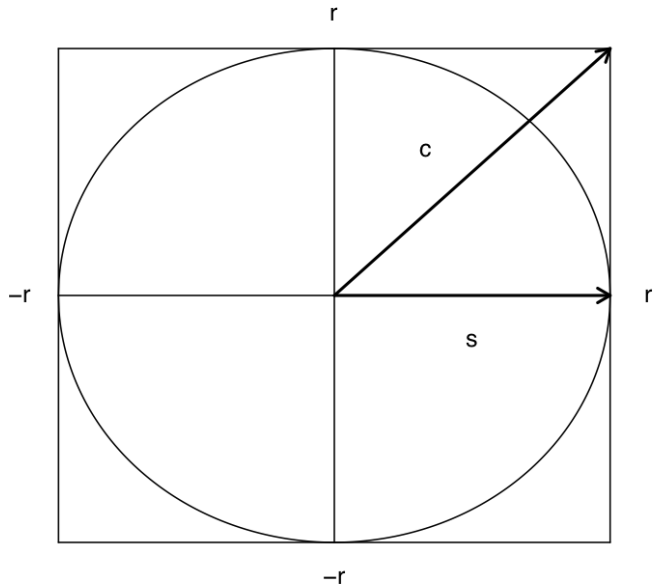
Dimension of the data is *p*, the number of variables.

Cubes and Spheres

Space expands exponentially with dimension:







As dimension increases the volume of a sphere of same radius as cube side length becomes much smaller than the volume of the cube.

Multivariate data

Mostly though, we're working on problems where $n >> p$, and $p > 1$. This would more commonly be referred to as multivariate data.

Sub-spaces

Data will often be confined to a region of the space having lower intrinsic dimensionality. The data lives in a low-dimensional subspace.

Analyse the data by, reducing dimensionality, to the subspace containing the data.

Principal Component Analysis (PCA)

Principal component analysis (PCA) produces a low-dimensional representation of a dataset. It finds a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated. It is an unsupervised learning method.

Why use PCA?

- We may have too many predictors for a regression. Instead, we can use the first few principal components.
- Understanding relationships between variables.
- Data visualisation. We can plot a small number of variables more easily than a large number of variables.

First principal component

The first principal component of a set of variables $x_1, x_2, \ldots, x_p$ is the linear combination

$$z_1 = \phi_{11} x_1 + \phi_{21} x_2 + \cdots + \phi_{p1} x_p$$

that has the largest variance such that

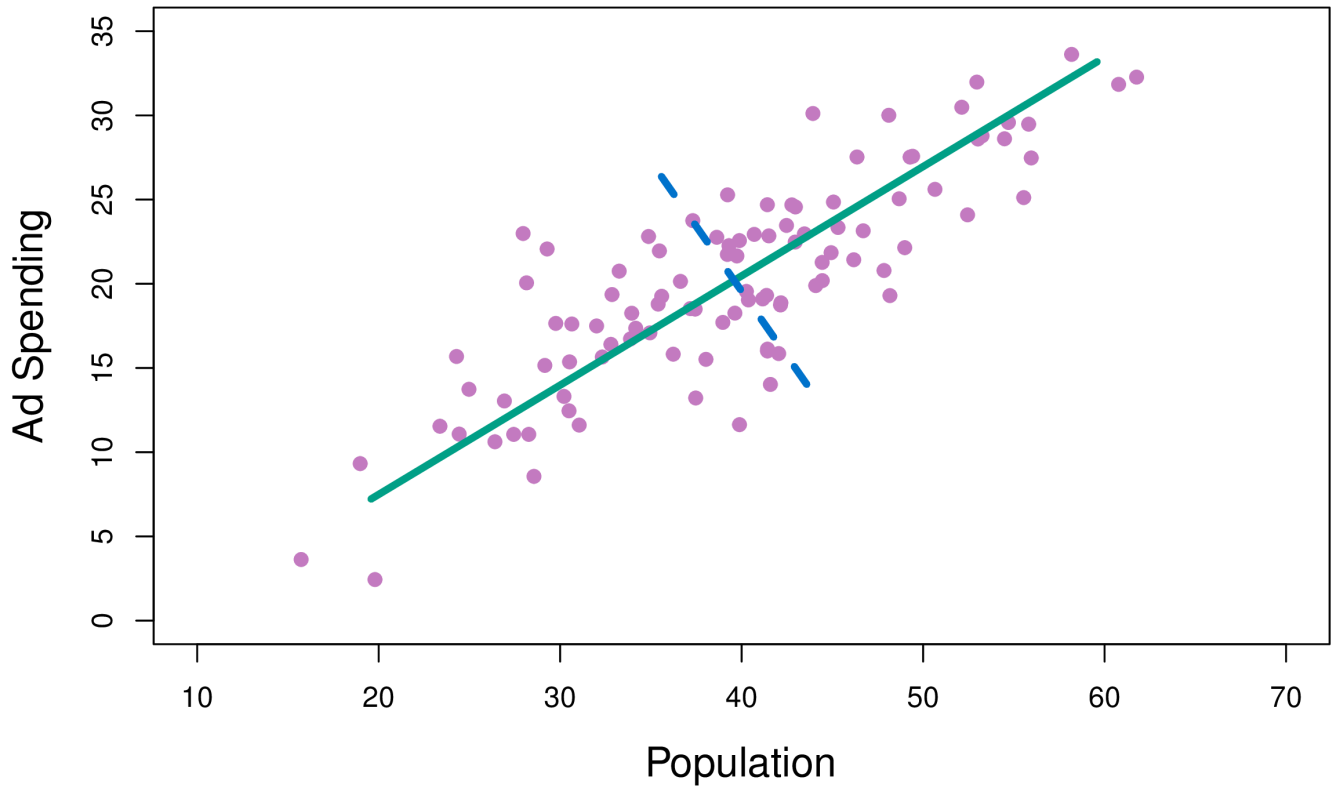
$$\sum_{j=1}^{p} \phi_{j1}^2 = 1$$

The elements $\phi_{11}, \ldots, \phi_{p1}$ are the loadings of the first principal component.

Geometry

- The loading vector $\phi_1 = [\phi_{11}, \ldots, \phi_{p1}]^T$ defines direction in feature space along which data vary most.
- If we project the n data points $x_1, \ldots, x_n$ onto this direction, the projected values are the principal component scores $z_{11}, \ldots, z_{n1}$.
- The second principal component is the linear combination $z_{i2} = \phi_{12}x_{i1} + \phi_{22}x_{i2} + \cdots + \phi_{p2}x_{ip}$ that has maximal variance among all linear combinations that are *uncorrelated* with z_1.
- Equivalent to constraining ϕ_2 to be orthogonal (perpendicular) to ϕ_1. And so on.
- There are at most $\min(n-1, p)$ PCs.

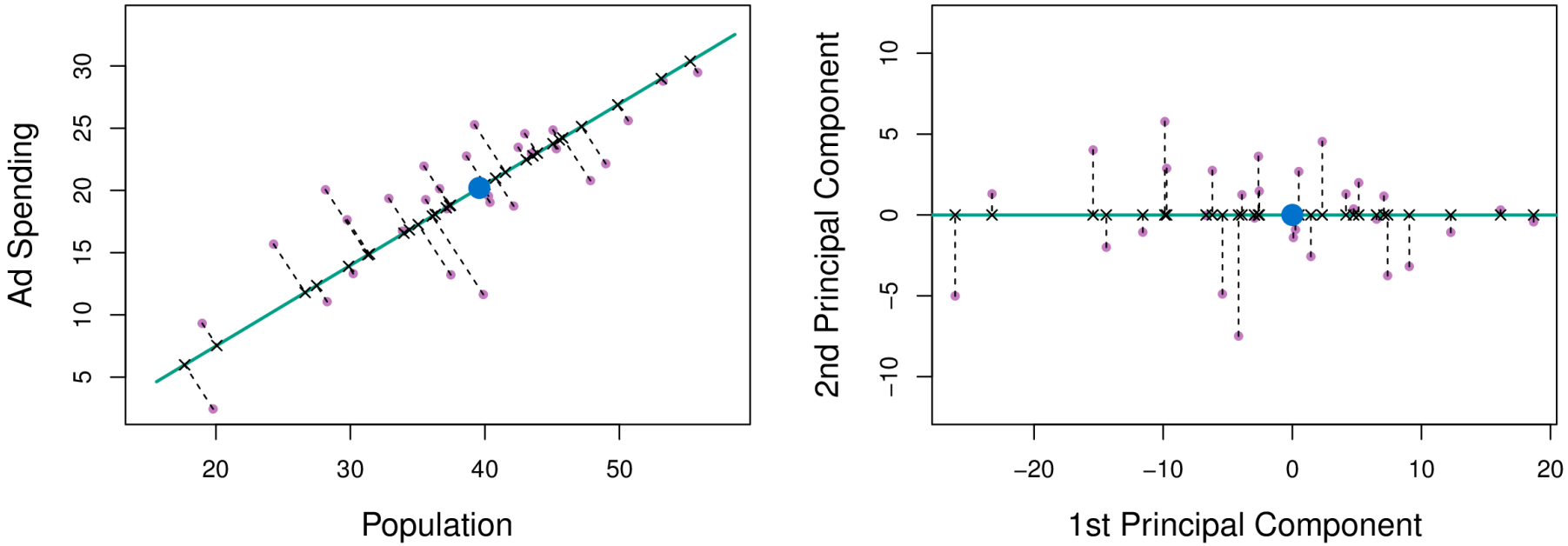
Example



First PC; second PC

(Chapter6/6.14.pdf)

Example



If you think of the first few PCs like a linear model fit, and the others as the error, it is like regression, except that errors are orthogonal to model.

(Chapter6/6.15.pdf)

Computation

PCA can be thought of as fitting an n-dimensional ellipsoid to the data, where each axis of the ellipsoid represents a principal component. The new variables produced by principal components correspond to rotating and scaling the ellipse into a circle.

Computation

Suppose we have a $n \times p$ data set $X = [x_{ij}]$.

1. Centre each of the variables to have mean zero (i.e., the column means of X are zero).
2. Let $z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \cdots + \phi_{p1}x_{ip}$
3. Compute sample variance of z_{i1} is $\frac{1}{n}\sum_{i=1}^{n} z_{i1}^2$.
4. Estimate ϕ_{j1}

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 \text{ subject to } \sum_{j=1}^{p}\phi_{j1}^2 = 1$$

Repeat optimisation to estimate ϕ_{jk}, with additional constraint that $\sum_{j=1,k<k'}^{p}\phi_{jk}\phi_{jk'} = 0$ (next vector is orthogonal to previous eigenvector).

Eigen-decomposition

1. Compute the covariance matrix (after centering the columns of X)

$$S = X^T X$$

2. Find eigenvalues (diagonal elements of D) and eigenvectors (V):

$$S = VDV^T$$

where columns of V are orthonormal (i.e., $V^T V = I$)

Singular Value Decomposition

$$X = U\Lambda V^T$$

- X is an $n \times p$ matrix
- U is $n \times r$ matrix with orthonormal columns ($U^T U = I$)
- Λ is $r \times r$ diagonal matrix with non-negative elements. (Square root of the eigenvalues.)
- V is $p \times r$ matrix with orthonormal columns (These are the eigenvectors, and $V^T V = I$).

It is always possible to uniquely decompose a matrix in this way.

Total variance

Total variance in data (assuming variables centered at 0):

$$\text{TV} = \sum_{j=1}^{p} \text{Var}(x_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$$

If variables are standardised, TV=number of variables!

Variance explained by *m*'th PC: $V_m = \text{Var}(z_m) = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2$

$$\text{TV} = \sum_{m=1}^{M} V_m \text{ where } M = \min(n-1, p).$$

How to choose k?

PCA is a useful dimension reduction technique for large datasets, but deciding on how many dimensions to keep isn't often clear. 🤔

How do we know how many principal components to choose?

How to choose k?

Proportion of variance explained:

$$\text{PVE}_m = \frac{V_m}{TV}$$

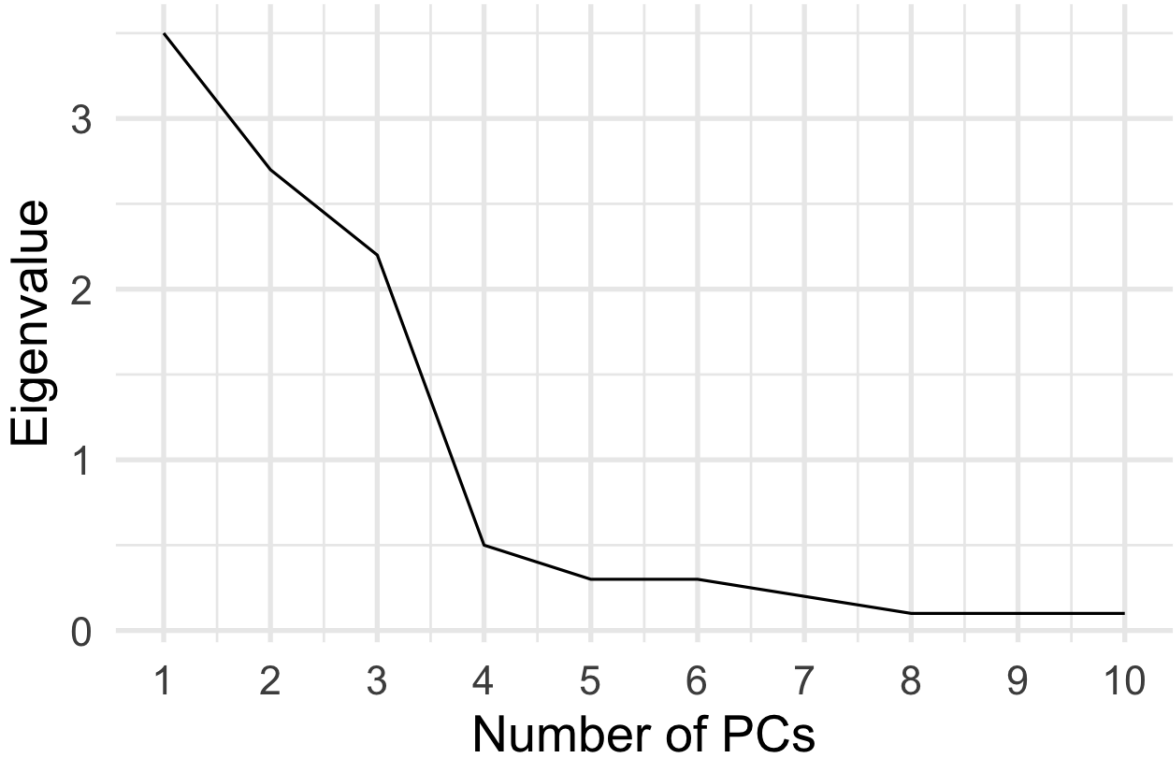
Choosing the number of PCs that adequately summarises the variation in X, is achieved by examining the cumulative proportion of variance explained.

Cumulative proportion of variance explained:

$$\text{CPVE}_k = \sum_{m=1}^{k} \frac{V_m}{TV}$$

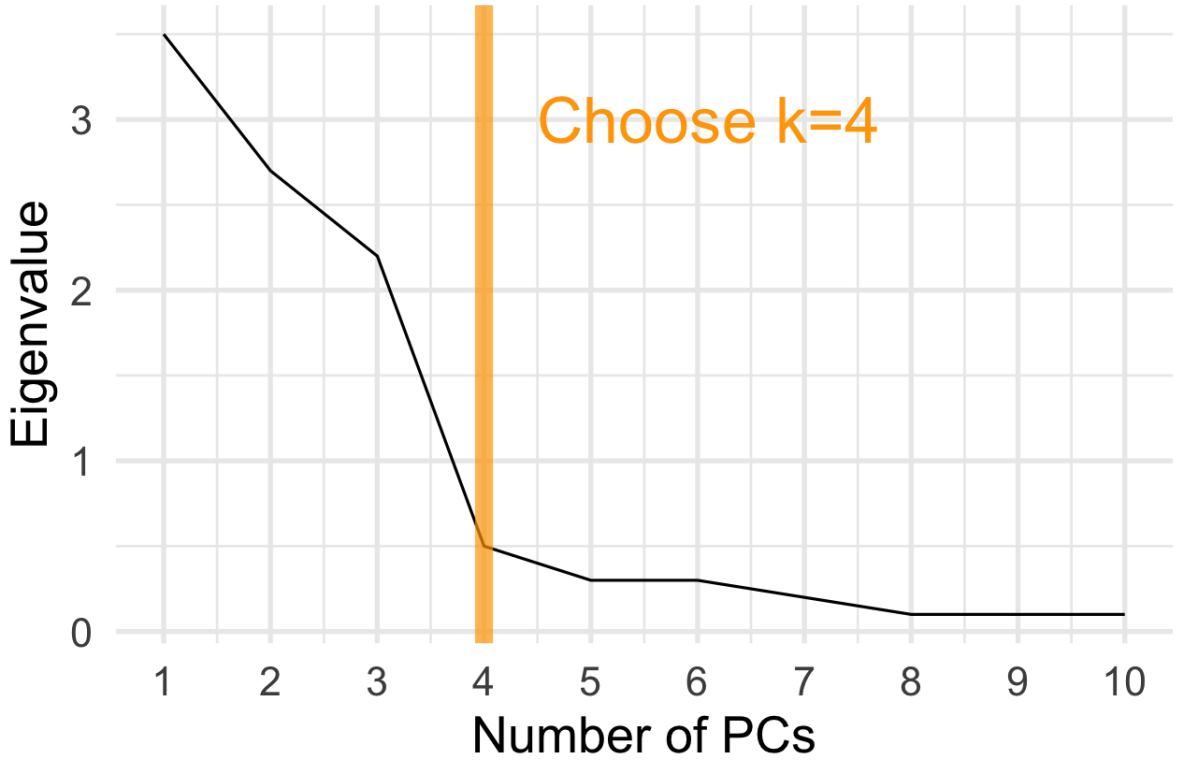
How to choose k?

Scree plot: Plot of variance explained by each component vs number of component.



How to choose k?

Scree plot: Plot of variance explained by each component vs number of component.



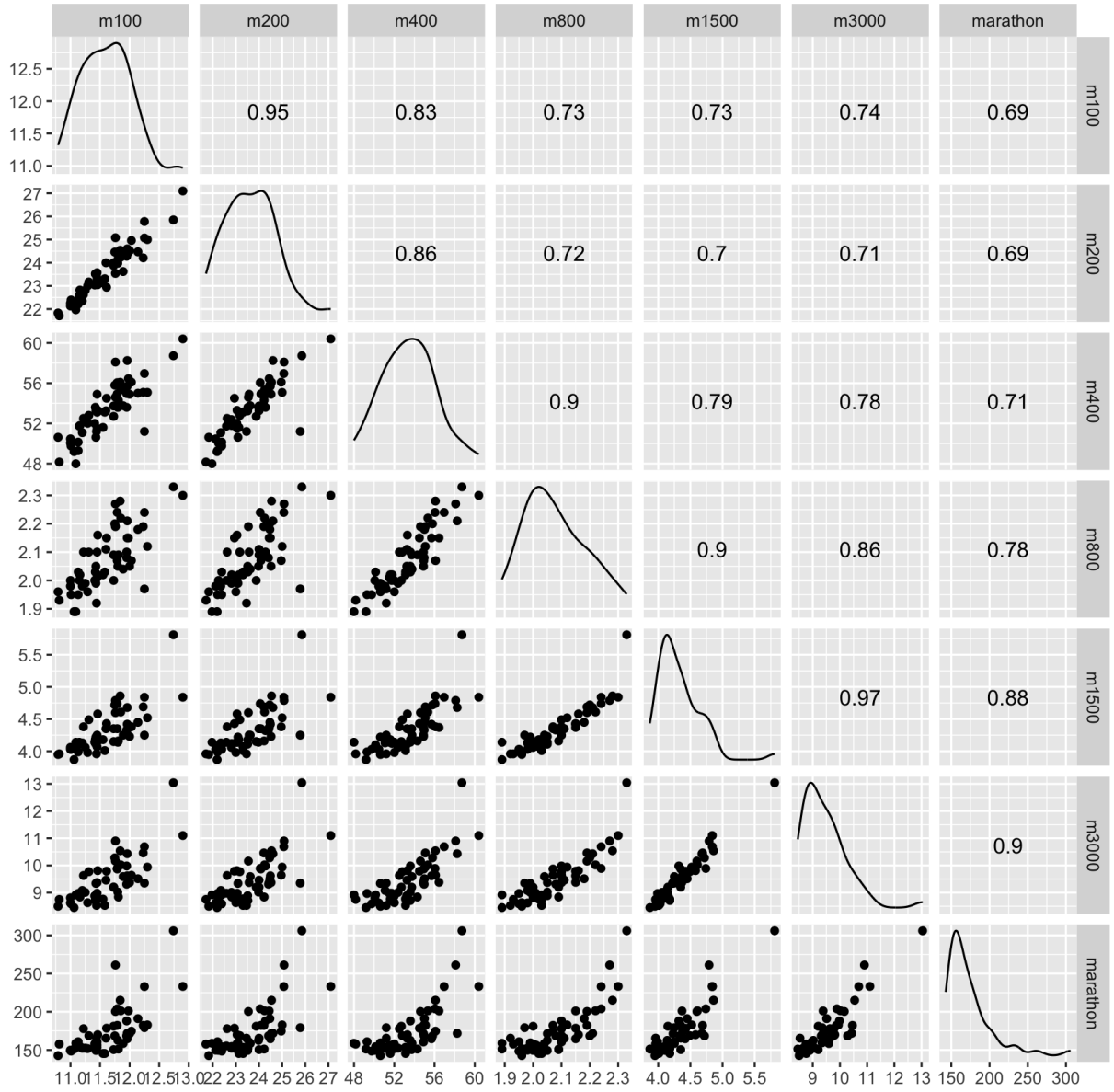
Example - track records

The data on national track records for women (as at 1984).

```
## Rows: 55
## Columns: 8
## $ m100     <dbl> 11.61, 11.20, 11.43, 11.41, 11.46, 11.31, 12.14, 11.00, 12.0
## $ m200     <dbl> 22.94, 22.35, 23.09, 23.04, 23.05, 23.17, 24.47, 22.25, 24.5
## $ m400     <dbl> 54.50, 51.08, 50.62, 52.00, 53.30, 52.80, 55.00, 50.06, 54.9
## $ m800     <dbl> 2.15, 1.98, 1.99, 2.00, 2.16, 2.10, 2.18, 2.00, 2.05, 2.08,
## $ m1500    <dbl> 4.43, 4.13, 4.22, 4.14, 4.58, 4.49, 4.45, 4.06, 4.23, 4.33,
## $ m3000    <dbl> 9.79, 9.08, 9.34, 8.88, 9.81, 9.77, 9.51, 8.81, 9.37, 9.31,
## $ marathon <dbl> 178.52, 152.37, 159.37, 157.85, 169.98, 168.75, 191.02, 149.
## $ country  <chr> "argentin", "australi", "austria", "belgium", "bermuda", "br
```

Source: Johnson and Wichern, Applied multivariate analysis

Explore the data



Compute PCA

```
track_pca <- prcomp(track[,1:7], center=TRUE, scale=TRUE)
track_pca

## Standard deviations (1, .., p=7):
## [1] 2.41 0.81 0.55 0.35 0.23 0.20 0.15
##
## Rotation (n x k) = (7 x 7):
##             PC1   PC2    PC3    PC4    PC5     PC6    PC7
## m100      0.37  0.49 -0.286  0.319  0.231  0.6198  0.052
## m200      0.37  0.54 -0.230 -0.083  0.041 -0.7108 -0.109
## m400      0.38  0.25  0.515 -0.347 -0.572  0.1909  0.208
## m800      0.38 -0.16  0.585 -0.042  0.620 -0.0191 -0.315
## m1500     0.39 -0.36  0.013  0.430  0.030 -0.2312  0.693
## m3000     0.39 -0.35 -0.153  0.363 -0.463  0.0093 -0.598
## marathon 0.37 -0.37 -0.484 -0.672  0.131  0.1423  0.070
```

Assess

Summary of the principal components:

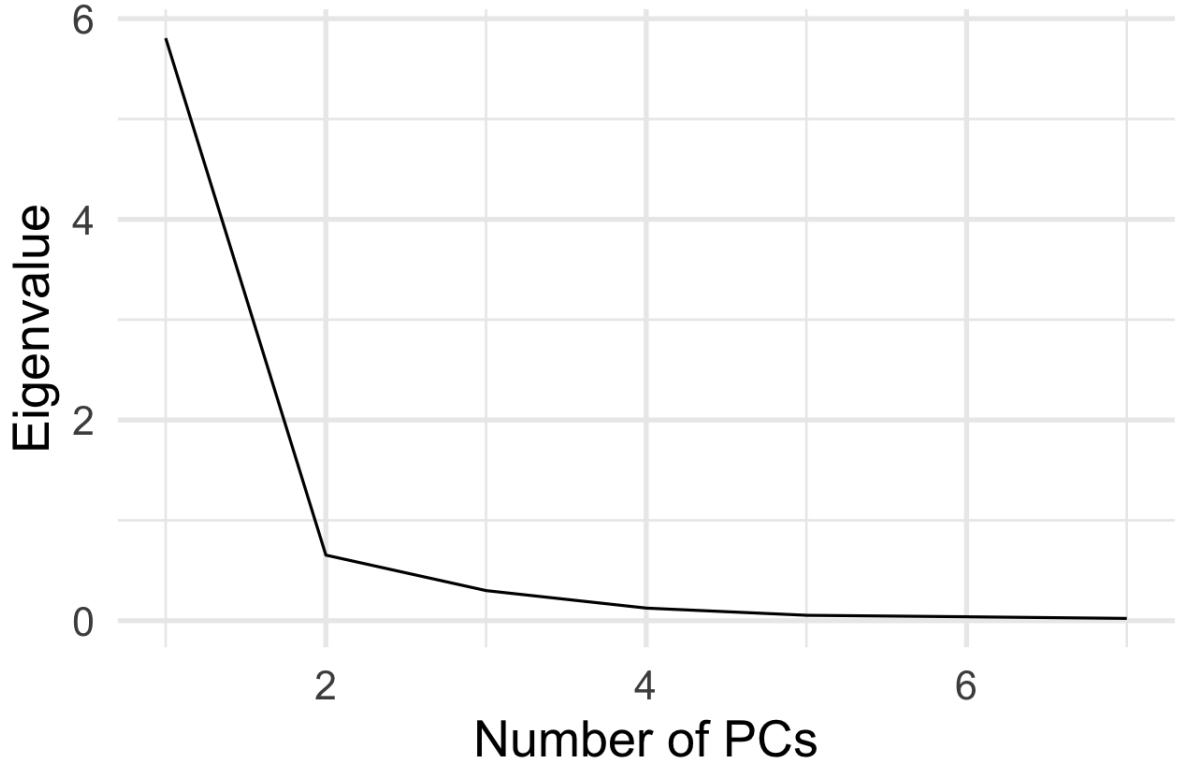
	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Variance	5.81	0.65	0.30	0.13	0.05	0.04	0.02
Proportion	0.83	0.09	0.04	0.02	0.01	0.01	0.00
Cum. prop	0.83	0.92	0.97	0.98	0.99	1.00	1.00

Increase in variance explained large until $k = 3$ PCs, and then tapers off. A choice of 3 PCs would explain 97% of the total variance.

Assess

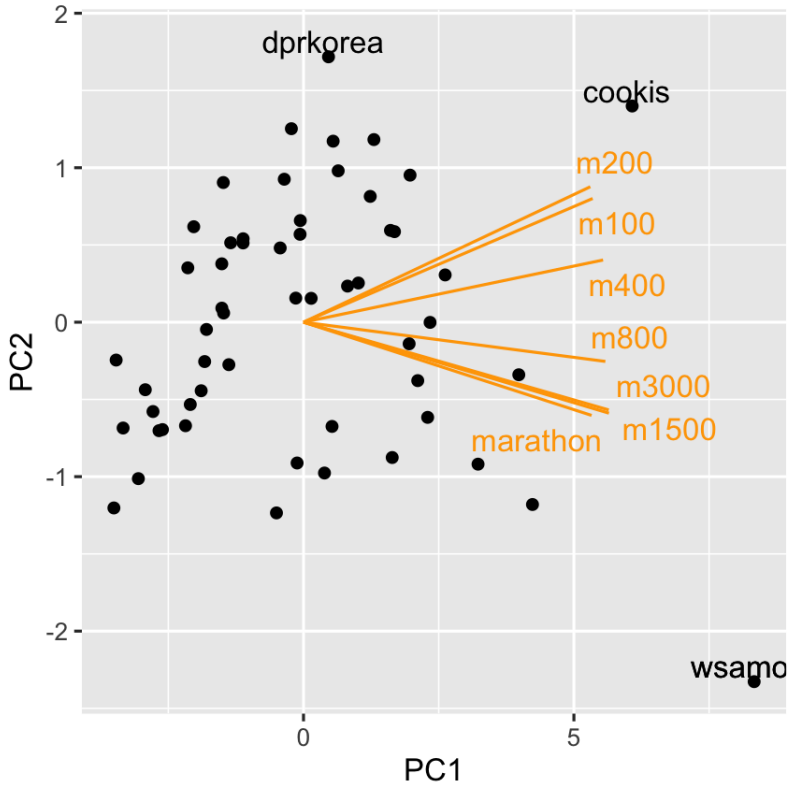
Scree plot: Where is the elbow?

At $k = 2$, thus the scree plot suggests 2 PCs would be sufficient to explain the variability.



Assess

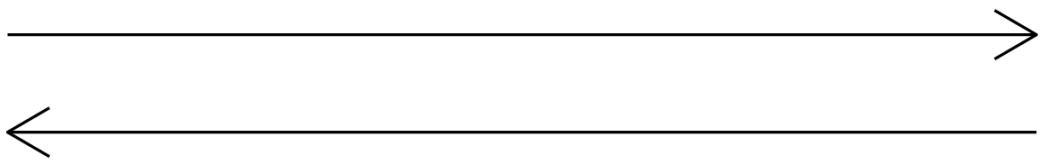
Visualise model using a biplot: Plot the principal component scores, and also the contribution of the original variables to the principal component.

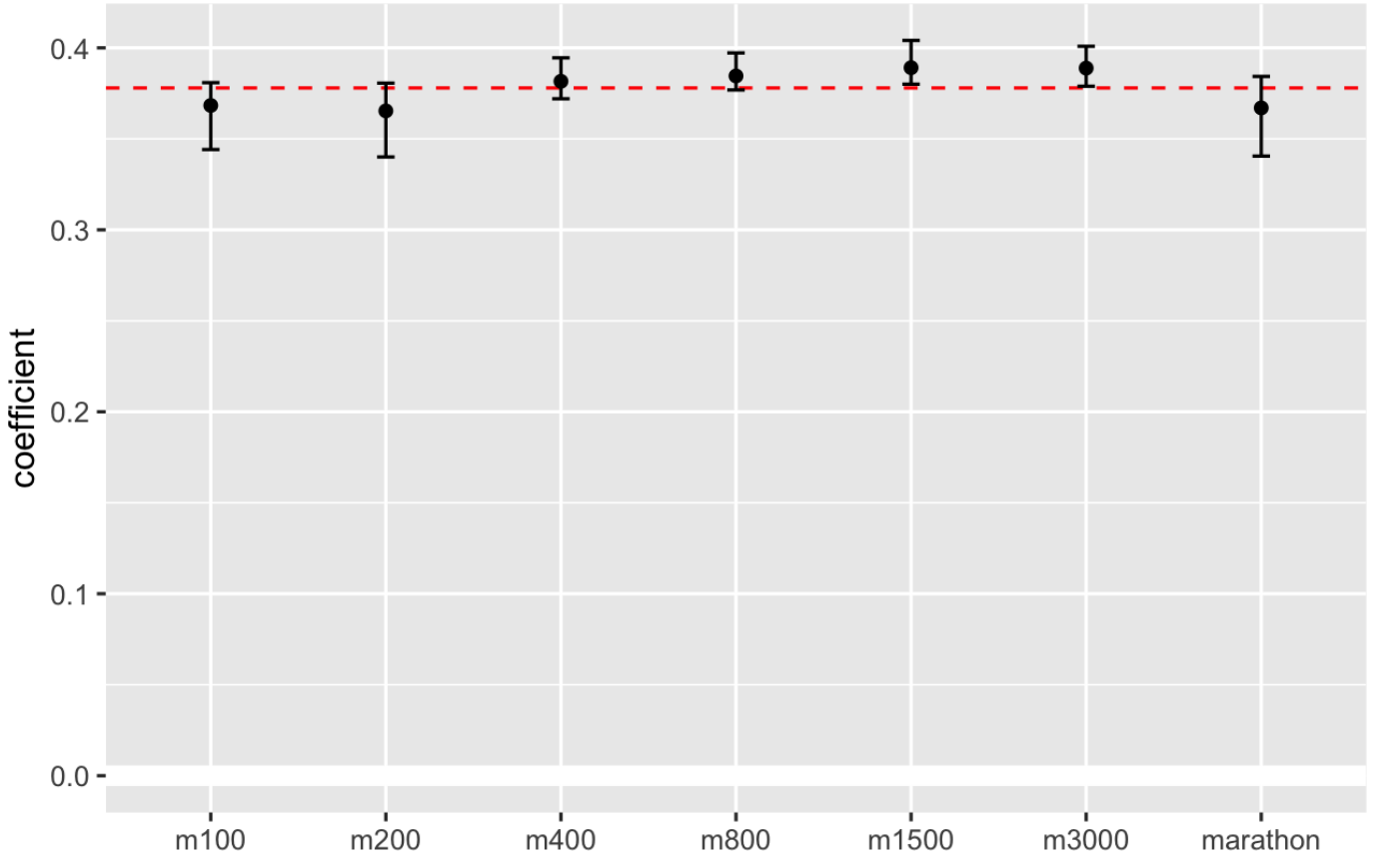


Significance of loadings

Bootstrap can be used to assess whether the coefficients of a PC are significantly different from 0. The 95% bootstrap confidence intervals can be computed by:

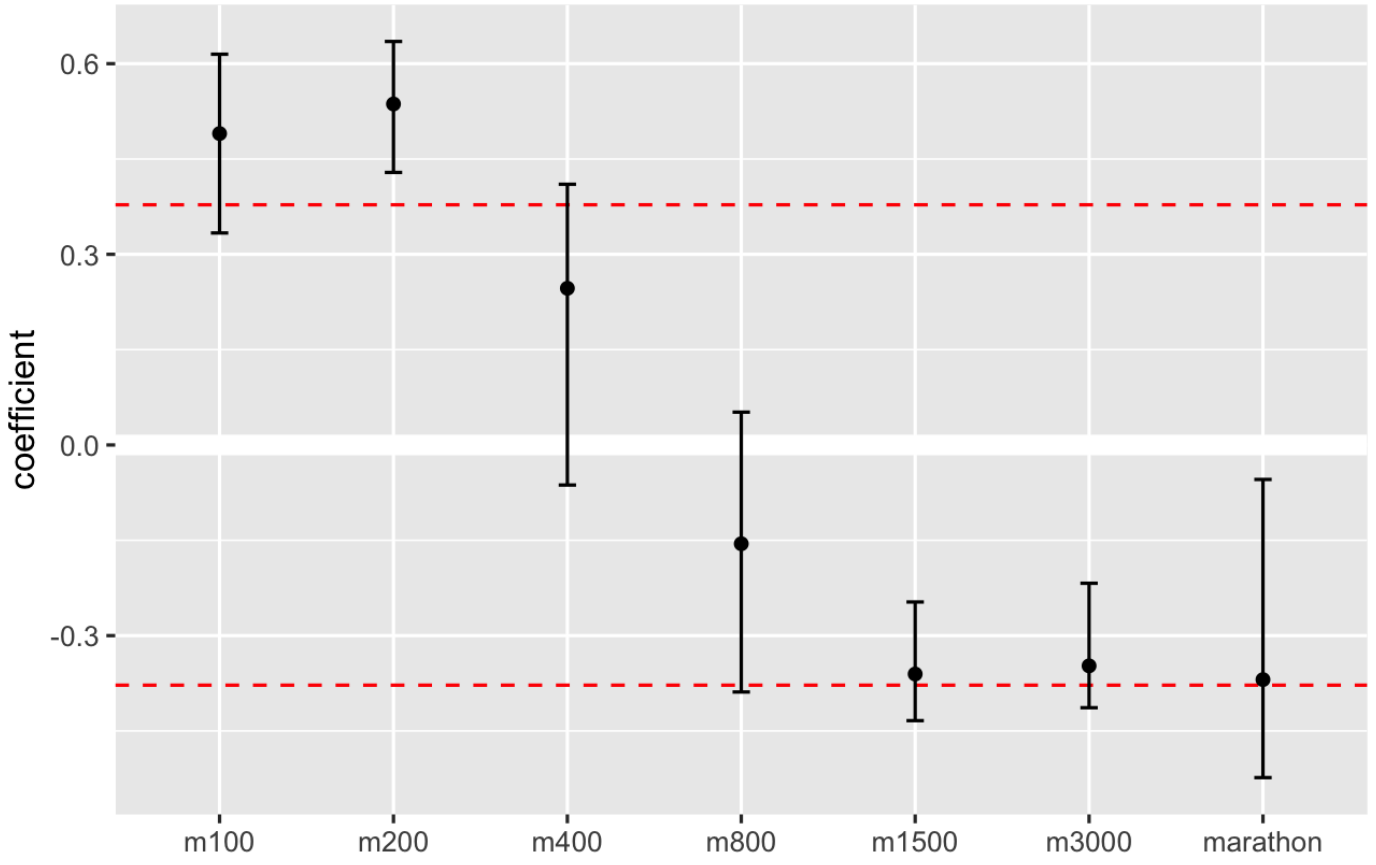
1. Generating B bootstrap samples of the data
2. Compute PCA, record the loadings
3. Re-orient the loadings, by choosing one variable with large coefficient to be the direction base
4. If B=1000, 25th and 975th sorted values yields the lower and upper bounds for confidence interval for each PC.





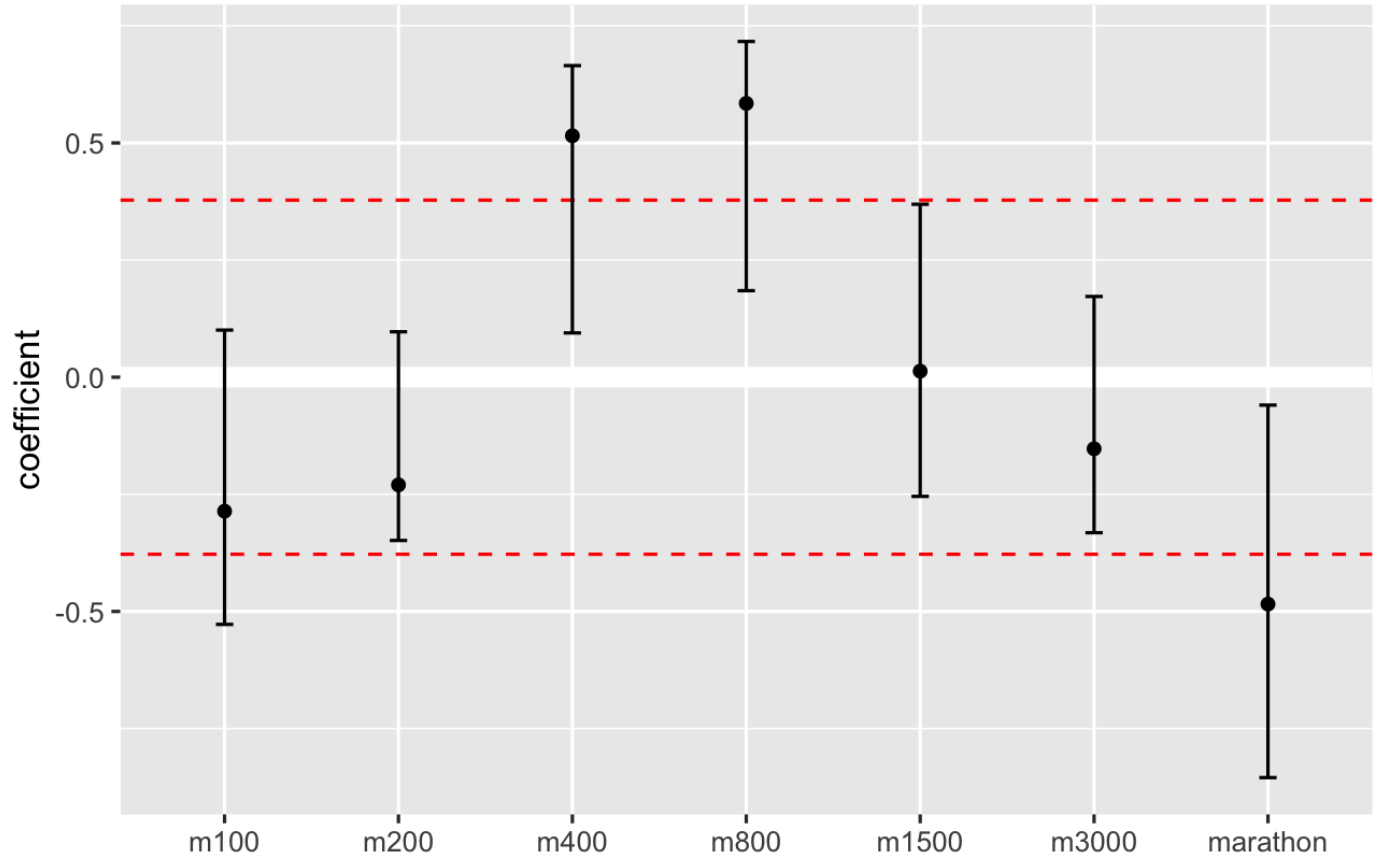
All of the coefficients on PC1 are significantly different from 0, and positive, approximately equal, not significantly different from being equal.

Loadings for PC2



On PC2 m100 and m200 contrast m1500 and m3000 (and possibly marathon). These are significantly different from 0.

Loadings for PC3



On PC3 m400 and m800 (and possibly marathon) are significantly different from 0.

Interpretation

- PC1 measures overall magnitude, the strength of the athletics program. High positive values indicate poor programs with generally slow times across events.
- PC2 measures the contrast in the program between short and long distance events. Some countries have relatively stronger long distance atheletes, while others have relatively stronger short distance athletes.
- There are several outliers visible in this plot, `wsamoa`, `cookis`, `dpkorea`. PCA, because it is computed using the variance in the data, can be affected by outliers. It may be better to remove these countries, and re-run the PCA.
- PC3, may or may not be useful to keep. The interpretation would that this variable summarises countries with different middle distance performance.

Other techniques

Projection pursuit (PP) generalises PCA

PCA:

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

PP:

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \; f\left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right) \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

MDS generalises PCA

Multidimensional scaling (MDS) finds a low-dimensional layout of points that minimises the difference between distances computed in the *p*-dimensional space, and those computed in the low-dimensional space.

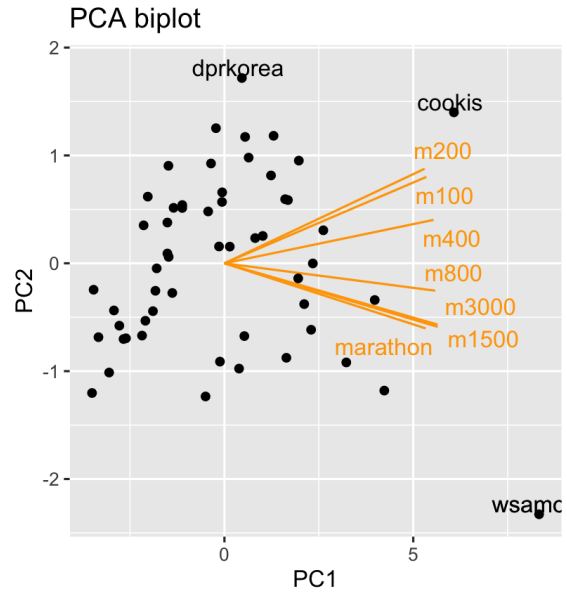
$$\text{Stress}_D(x_1, \ldots, x_N) = \left(\sum_{i,j=1; i \neq j}^{N} (d_{ij} - d_k(i,j))^2 \right)^{1/2}$$

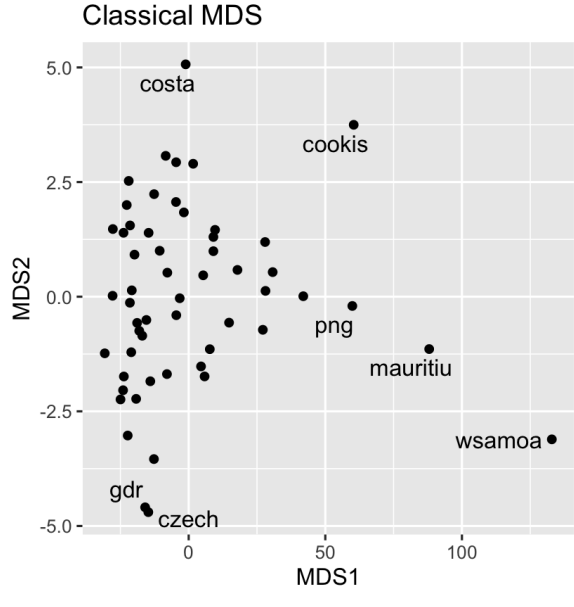
where D is an $N \times N$ matrix of distances (d_{ij}) between all pairs of points, and $d_k(i,j)$ is the distance between the points in the low-dimensional space.

MDS can do nonlinear dimension reduction

- Classical MDS similar results to PCA
- Metric MDS incorporates power transformations on the distances, d_{ij}^r.
- Non-metric MDS incorporates a monotonic transformation of the distances, e.g. rank

```
track <- read_csv(here::here("data
track_mds <-
  cmdscale(dist(track[,1:7])) %>%
  as_tibble() %>%
  mutate(country = track$country)
```





Challenge

For each of these distance matrices, find a layout in 1 or 2D that accurately reflects the full distances.

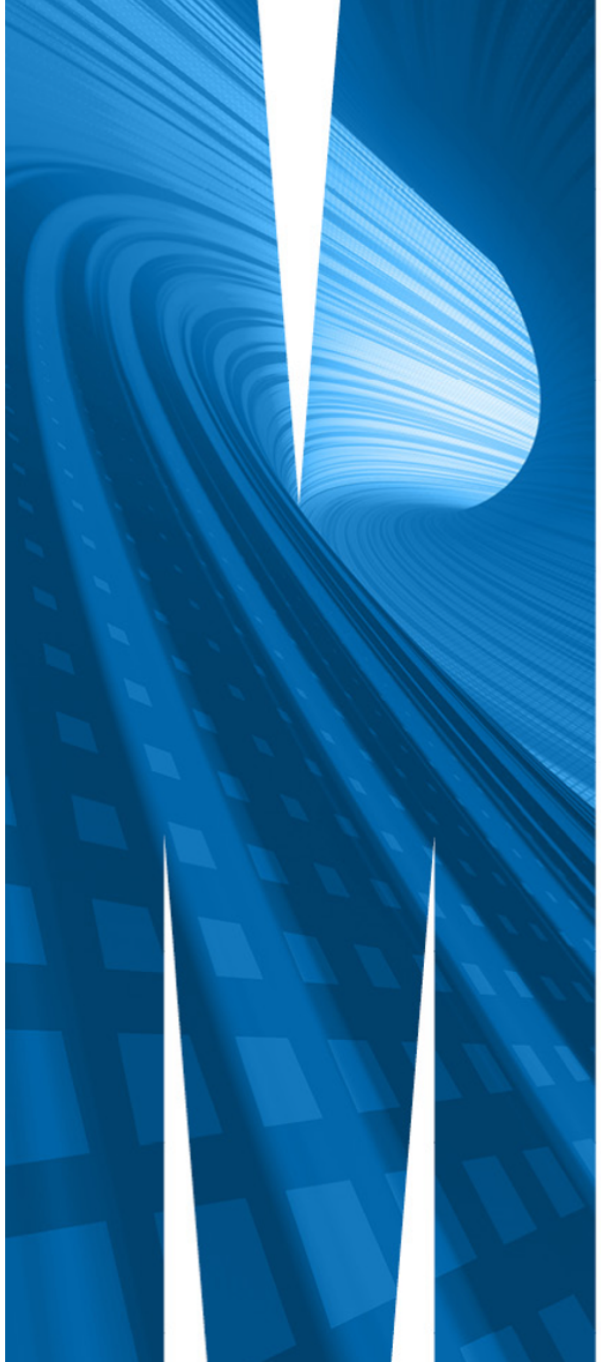
```
## # A tibble: 3 x 4
##   name      A     B     C
##   <chr> <dbl> <dbl> <dbl>
## 1 A       0.1   3.2   3.9
## 2 B       3.2  -0.1   5.1
## 3 C       3.9   5.1   0

## # A tibble: 4 x 5
##   name      A     B     C     D
##   <chr> <dbl> <dbl> <dbl> <dbl>
## 1 A       0.1   0.9   2.1   3
## 2 B       0.9   0     1.1   1.9
## 3 C       2.1   1.1   0.1   1.1
## 4 D       3     1.9   1.1  -0.1
```

Non-linear dimension reduction

- T-distributed Stochastic Neighbor Embedding (t-SNE): similar to MDS, except emphasis is placed on grouping observations into clusters. Observations within a cluster are placed close in the low-dimensional representation, but clusters themselves are placed far apart.
- Local linear embedding (LLE): Finds nearest neighbours of points, defines interpoint distances relative to neighbours, and preserves these proximities in the low-dimensional mapping. Optimisation is used to solve an eigen-decomposition of the knn distance construction.
- Self-organising maps (SOM): First clusters the observations into $k \times k$ groups. Uses the mean of each group laid out in a constrained 2D grid to create a 2D projection.

MONASH University



Lecturer: *Professor Di Cook*

Department of Econometrics and Business Statistics

ETC3250.Clayton-x@monash.edu

Week 4b