# COMP20008 Elements of Data Processing

## Semester 2 2018

## Lecture 8: Hierarchical Clustering and Dimension Reduction

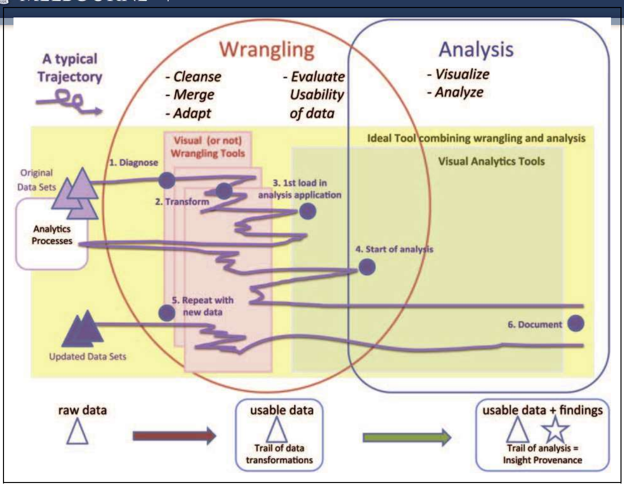


## Plan today

- Hierarchical clustering
  - Another alternative for k-means to extract clusters, visualise their relationships
- Dimension reduction
  - A technique for visualising high dimensional data (reducing many features/columns to few)



## Overview

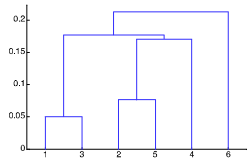


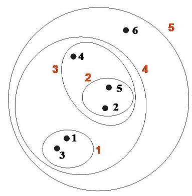

Research directions in Data Wrangling: visualisations and transformations for crediible data. S. Kandel et al, Information Visualisation 10(4), 2011.



## Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits. On LHS, y-axis is distance



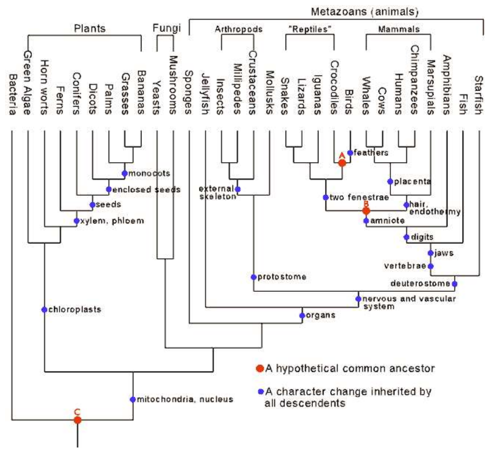
## Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by 'cutting' the dendrogram at the proper level
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)



## Tree of Life

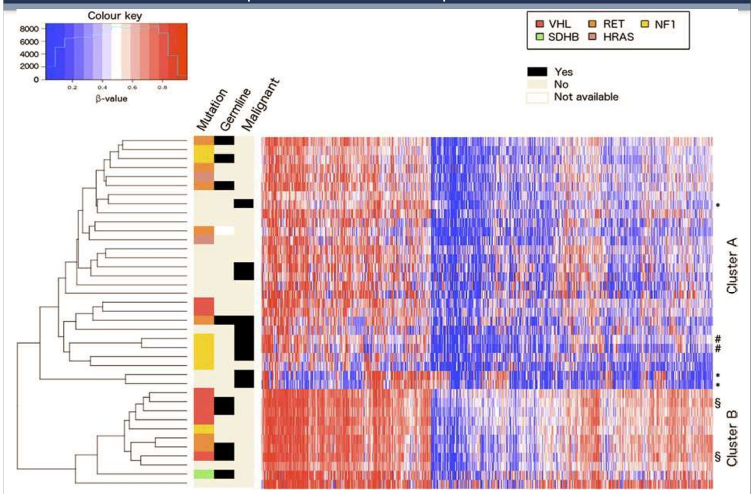
(http://www.talkorigins.org/faqs/comdesc/phylo.html)






## Rows are samples of tissue from different tumours
## Columns are biological features of a tumour

https://www.nature.com/articles/srep44943






## Hierarchical Clustering

- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

## Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. **Repeat**
  - Merge the two closest clusters
  - Update the proximity matrix

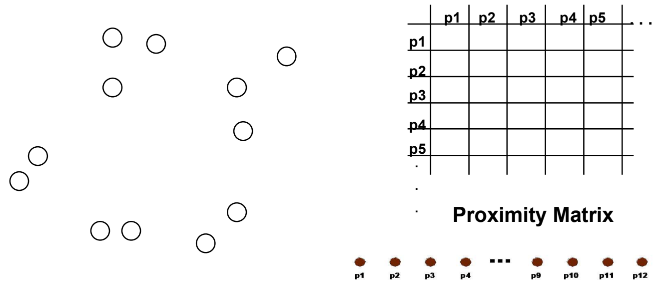
  **Until** only a single cluster remains
- Key operation is the computation of the proximity of two clusters
  – Different approaches to defining the distance between clusters distinguish the different algorithms

Let's see a step-by-step example

THE UNIVERSITY OF MELBOURNE

## Starting Situation

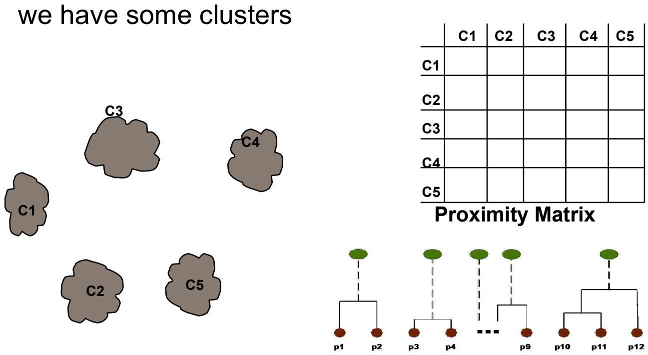
- Start with clusters of individual points and a proximity matrix




THE UNIVERSITY OF MELBOURNE

## Intermediate Situation

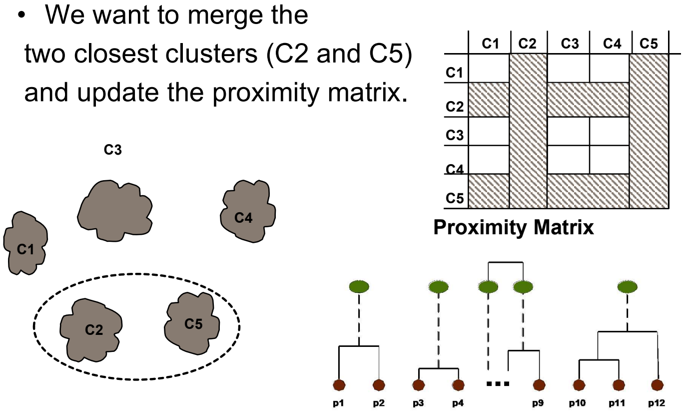
- After some merging steps, we have some clusters




THE UNIVERSITY OF MELBOURNE

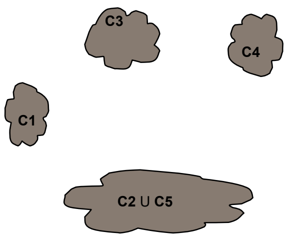
## Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



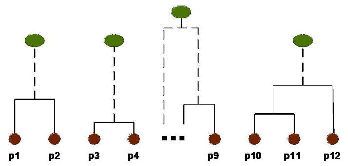
## After Merging

- The question is “How do we update the proximity matrix?”



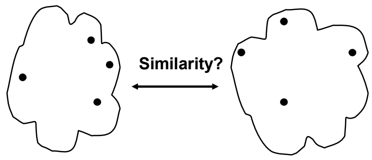

| | C1 | C2 U C5 | C3 | C4 |
|---|---|---|---|---|
| C1 | | ? | | |
| C2 U C5 | ? | ? | ? | ? |
| C3 | | ? | | |
| C4 | | ? | | |

**Proximity Matrix**






## How to Define Inter-Cluster Similarity



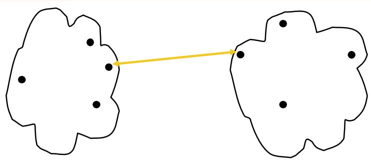

| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Proximity Matrix**

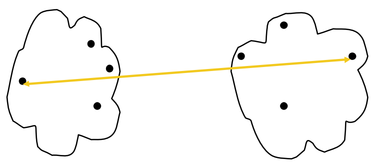
- We define the similarity to be the minimum distance between the clusters . This is also known as single linkage.
  - Other choices also possible (e.g. max or average, but we won’t cover the average linkage method)



## How to Define Inter-Cluster Similarity



- MIN (Single Linkage)



- MAX (Complete Linkage)

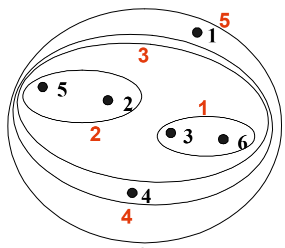
| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Proximity Matrix**

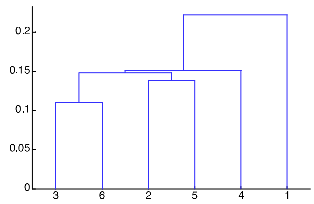


## Hierarchical Clustering: MIN or Single Linkage

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.




**Nested Clusters**



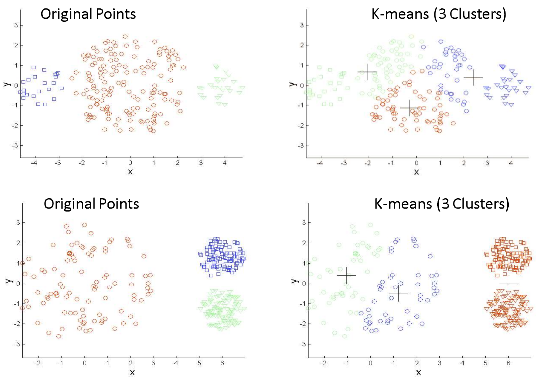

**Dendrogram**

THE UNIVERSITY OF MELBOURNE

## Clustering Reflections: K-means (1)

### Limitations of K-means

Differing sizes and denisty

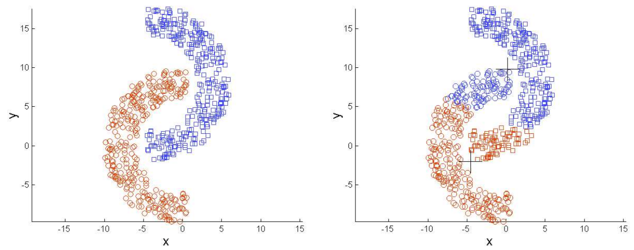



THE UNIVERSITY OF MELBOURNE

## Clustering Reflections: K-means (2)

### Limitations of K-means

Non-globular shapes



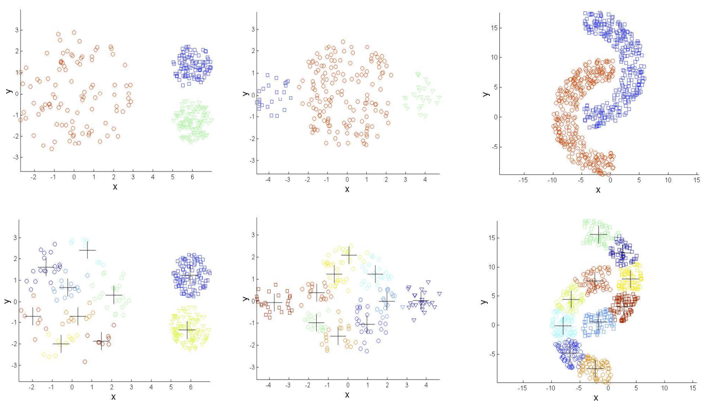
Original Points

K-means (2 Clusters)

THE UNIVERSITY OF MELBOURNE

## Clustering Reflections: K-means (3)

- One solution is to use many clusters.
- Find parts of clusters, but need to put together.

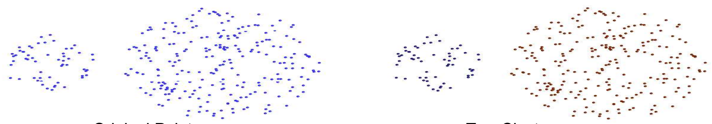


THE UNIVERSITY OF MELBOURNE

## Clustering Reflections: Agglomerative (1)

Strength of MIN: Single linkage

- Can handle non-elliptical shapes

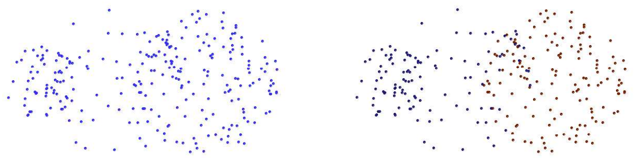


Original Points

Two Clusters

Limitations of MIN: Single linkage

- Sensitive to noise and outliers

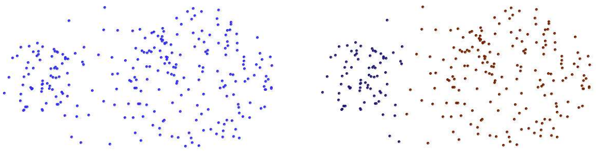


Original Points

Two Clusters

## Clustering Reflections: Agglomerative (1)

Strength of MAX: Complete linkage
- Less susceptible to noise and outliers



Limitations of Max: Complete linkage
- Tends to break large clusters

Original Points Two Clusters



## Clustering Reflections: Agglomerative (3)

Hierarchical Clustering: Problems and Limitations

- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized



## Dimension reduction

- Motivation and intuition
- Principal components analysis



## Motivation: High dimensional data

- **The curse of dimensionality**: “Data analysis techniques which work well at lower dimensions (fewer features), often perform poorly as the dimensionality of the analysed data increases (lots of features”
- As dimensionality increases, data becomes increasingly sparse and all the distances between pairs of points begin to look the same. Impacts any algorithm that is based on distances between objects.

THE UNIVERSITY OF MELBOURNE

## Dimensionality Reduction

- Purpose:
  - Avoid curse of dimensionality
  - Reduce amount of time and memory required by data processing algorithms
  - Allow data to be more easily visualized
  - May help to eliminate irrelevant features or reduce noise

THE UNIVERSITY OF MELBOURNE

## Dimensionality Reduction

- Input: A dataset with N features and K objects
- Output: A transformed dataset with n<<N features and K objects
  - n is often set to 2 or 3, so that the transformed dataset can be easily visualised
- E.g if n=2

| Object id | Feature1 | Feature2 | …. | FeatureN |
|---|---|---|---|---|
| 1 | … | … | … | … |
| … | …. | … | … | … |
| K | …. | …. | … | … |

↓ Transformation

| Object id | NewFeatureA | NewFeatureB |
|---|---|---|
| 1 | … | … |
| … | … | … |
| K | … | … |

THE UNIVERSITY OF MELBOURNE

## Transforming from N dimensions to n<<N dimensions

- The transformation must preserve the characteristics of the data
  - In particular, preserve distances between pairs of points
- If a pair of objects is close before the transformation, they should still be close after the transformation
- If a pair of objects is far apart before the transformation, they should still be far apart after the transformation
- The set of nearest neighbors of an object before the transformation should ideally be the same after the transformation

THE UNIVERSITY OF MELBOURNE

## Question

- Suppose we are given a dataset with the following N features, describing individuals in this class. Which two features would you select to represent people, in such a way that "distances" between pairs of people are likely to be preserved in the reduced dataset?

- Input: N=7 features
  1. Weighted average mark (WAM)
  2. Age (years)
  3. Height (cm)
  4. Weight (kg)
  5. Number of pets owned
  6. Number of subjects passed so far
  7. Amount of sleep last night (0=little, 1=medium, 2=a lot)
  - Output: Select 2 of the above features

## Dimension Reduction

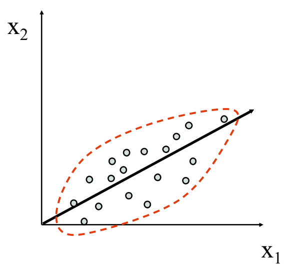
- Basic method: To reduce dimensionality, can just select a subset of the original features.
  - Scatter plots for Iris dataset shown earlier – 2D visualisations of a 4D dataset. 2 features were selected from the original 4.
- In general, when transforming a dataset from N to n<<N features
  - *The output n features do not need to be a subset of the input N features*. Rather, they can be new features whose values are constructed using some function applied to the input N features

## Principal Components Analysis (PCA)

- Find a new set of features that better captures the variability of the data
  - First dimension chosen to capture as much of the variability as possible.
  - The second dimension is orthogonal to the first, and subject to that constraint, captures as much of the remaining variability as possible,
  - The third dimension is orthogonal to the first and second, and subject to that constraint, captures as much of the remaining variability as possible.
- We will not be covering the mathematical details
  - Many tutorials available on the Web if you are interested. Nice application of linear algebra.

## Dimensionality Reduction: PCA

- Goal is to find a projection that captures the largest amount of variation in data. Below – the 1-D direction capturing most of the variation in the data. Use this to transform from 2D to 1D.

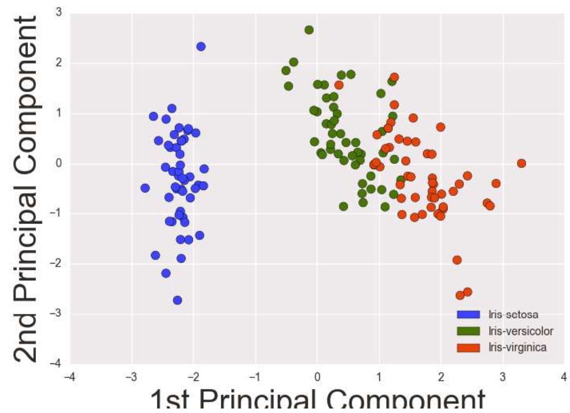



## PCA Example

- A good visualisation for PCA
  - http://setosa.io/ev/principal-component-analysis/

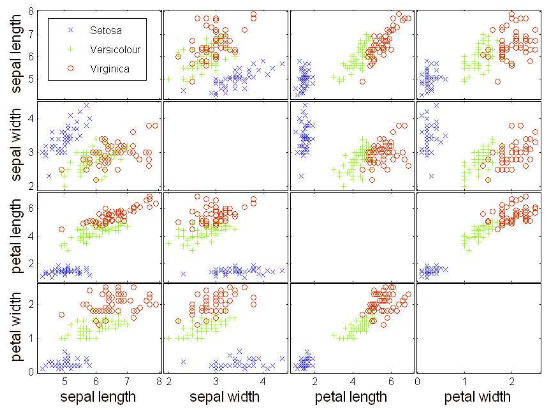
THE UNIVERSITY OF MELBOURNE

## Iris dataset: Reduced from 4 to 2 dimensions using PCA




THE UNIVERSITY OF MELBOURNE

## Contrast this against an earlier slide




Scatter plots for iris dataset

THE UNIVERSITY OF MELBOURNE

## AFL Football Dataset: From http://afltables.com/afl/stats/

[Stats Main][AFL Main]

[2013 Stats][2015 Stats]

### 2014 Player Stats

[2014 Stats Summary]

[Adelaide][Brisbane Lions][Carlton][Collingwood][Essendon][Fremantle][Geelong][Gold Coast][Greater Western Sydney][Hawthorn]
[Melbourne][North Melbourne][Port Adelaide][Richmond][St Kilda][Sydney][West Coast][Western Bulldogs]
[All Teams]

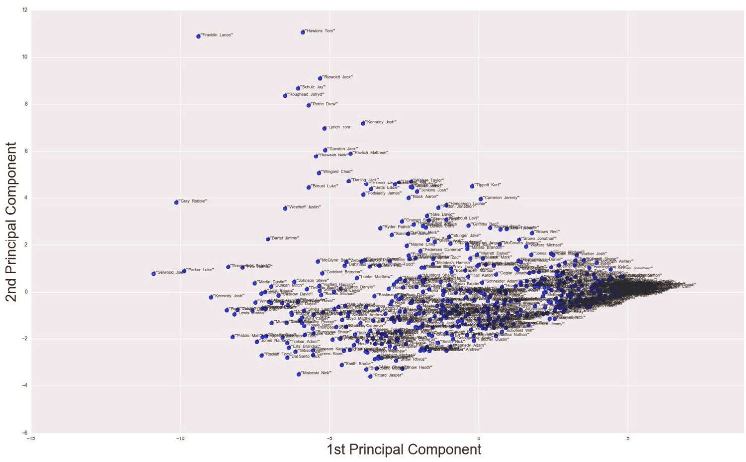
Abbreviations key

Adelaide [Game by Game]

| # | Player | GM | KI | MK | HB | DI | DA | GL | BH | HO | TK | RB | IF | CL | CG | FF | FA | BR | CP | UP | CM | MI | 1% | BO | GA | %P | SU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 32 | Dangerfield, Patrick | 22 | 276 | 74 | 272 | 548 | 24.91 | 17 | 22 | 28 | 78 | 33 | 104 | 136 | 66 | 34 | 19 | 21 | 341 | 210 | 25 | 16 | 35 | 18 | 10 | 83.7 | |
| 9 | Sloane, Rory | 22 | 269 | 105 | 252 | 521 | 23.68 | 13 | 9 | 19 | 147 | 45 | 99 | 92 | 50 | 26 | 15 | 10 | 275 | 256 | 9 | 7 | 64 | 5 | 21 | 87.2 | |
| 5 | Thompson, Scott | 19 | 257 | 69 | 262 | 519 | 27.32 | 3 | 7 | 2 | 86 | 28 | 77 | 118 | 61 | 19 | 22 | 14 | 224 | 280 | 3 | 5 | 21 | 1 | 7 | 81.7 | 0/2 |
| 33 | Smith, Brodie | 22 | 287 | 108 | 209 | 496 | 22.55 | 11 | 8 | | 35 | 109 | 76 | 18 | 45 | 9 | 6 | 4 | 142 | 319 | 7 | 2 | 56 | 46 | 7 | 87.0 | |
| 10 | Jaensch, Matthew | 22 | 297 | 126 | 166 | 463 | 21.05 | 7 | 5 | | 54 | 89 | 54 | 7 | 34 | 19 | 10 | | 106 | 325 | 16 | 1 | 57 | 34 | 3 | 81.3 | |
| 26 | Douglas, Richard | 19 | 266 | 52 | 147 | 413 | 21.74 | 11 | 8 | 4 | 91 | 21 | 96 | 91 | 38 | 22 | 17 | | 182 | 228 | 2 | 6 | 36 | 13 | 11 | 86.4 | |
| 11 | Wright, Matthew | 20 | 224 | 89 | 150 | 374 | 18.70 | 14 | 8 | | 68 | 22 | 47 | 39 | 27 | 30 | 6 | | 141 | 227 | 4 | 12 | 26 | 6 | 17 | 80.0 | 1/2 |
| 24 | Jacobs, Sam | 22 | 193 | 90 | 165 | 358 | 16.27 | 7 | 3 | 763 | 46 | 20 | 40 | 69 | 33 | 11 | 15 | 6 | 150 | 189 | 19 | 4 | 63 | 1 | 10 | 87.9 | 0/1 |
| 14 | Mackay, David | 19 | 168 | 58 | 174 | 342 | 18.00 | 11 | 7 | | 77 | 30 | 62 | 32 | 31 | 22 | 13 | | 127 | 224 | 5 | 3 | 34 | 37 | 8 | 81.1 | 0/2 |
| 18 | Betts, Eddie | 22 | 167 | 53 | 123 | 290 | 13.18 | 51 | 22 | | 74 | 8 | 37 | 30 | 39 | 19 | 16 | 4 | 149 | 136 | 3 | 29 | 21 | 8 | 29 | 87.7 | |
| 1 | Podsiadly, James | 21 | 189 | 119 | 101 | 290 | 13.81 | 26 | 14 | 2 | 37 | 17 | 52 | 2 | 63 | 14 | 25 | 4 | 132 | 165 | 41 | 35 | 60 | 1 | 16 | 90.1 | |
| 16 | Brown, Luke | 22 | 138 | 55 | 148 | 286 | 13.00 | 1 | 1 | | 54 | 37 | 16 | 8 | 18 | 13 | 5 | | 81 | 205 | 1 | 1 | 42 | 1 | 4 | 84.5 | |
| 2 | Crouch, Brad | 11 | 125 | 26 | 147 | 272 | 24.73 | 5 | 6 | 1 | 61 | 22 | 40 | 56 | 30 | 8 | 6 | | 114 | 156 | 1 | 2 | 17 | 9 | 6 | 83.7 | 0/1 |
| 36 | Martin, Brodie | 17 | 155 | 65 | 109 | 264 | 15.53 | 8 | 15 | | 45 | 30 | 38 | 23 | 40 | 13 | 11 | | 97 | 174 | 7 | 12 | 34 | 11 | 4 | 69.2 | 2/1 |
| 12 | Talia, Daniel | 22 | 167 | 105 | 93 | 260 | 11.82 | | 1 | | 24 | 45 | 25 | | 29 | 11 | 12 | | 79 | 183 | 13 | | 149 | 1 | 2 | 90.0 | 0/1 |
| 29 | Laird, Rory | 16 | 126 | 65 | 129 | 255 | 15.94 | 2 | 2 | | 37 | 21 | 34 | 15 | 31 | 8 | 8 | | 81 | 177 | 1 | 1 | 25 | 2 | 2 | 75.4 | 2/0 |
| 4 | Jenkins, Josh | 20 | 170 | 86 | 64 | 234 | 11.70 | 40 | 26 | 55 | 27 | 13 | 46 | 11 | 36 | 12 | 8 | 3 | 97 | 140 | 21 | 32 | 48 | 10 | 7 | 90.6 | |
| 13 | Walker, Taylor | 15 | 138 | 84 | 82 | 220 | 14.67 | 34 | 22 | | 24 | | 50 | | 47 | 10 | 21 | 5 | 102 | 120 | 23 | 31 | 20 | | 17 | 90.3 | |
| 3 | Reilly, Brent | 10 | 130 | 65 | 63 | 193 | 19.30 | | | | 19 | 32 | 17 | 8 | 30 | 3 | 13 | | 46 | 139 | 7 | | 19 | 24 | 1 | 81.0 | |
| 17 | Kerridge, Sam | 14 | 72 | 33 | 84 | 156 | 11.14 | 10 | 1 | | 52 | 10 | 23 | 26 | 25 | 3 | 14 | | 54 | 97 | 2 | 9 | 9 | 4 | 5 | 83.7 | 0/1 |

THE UNIVERSITY OF MELBOURNE

## AFL Football Dataset: PCA in 2D



## Principal Components Analysis Code

- Code
  - PCA in sklearn.decomposition
  - Will practice in workshop

## Acknowledgements


- Material partly adapted from
  - “Data Mining Concepts and Techniques”, Han et al, 2nd edition 2006.
  - “Introduction to Data Mining”, Tan et al 2005.