Clusters and Similarity

Week 04-BEM2031

Term2: 2023/24

Today:

- Understand the spatial interpretation of data
- Interpret the results of a principal component analysis (PCA)
- Experiment with clustering and interpret results

Today:

- Distances in data
- Dimensions and dimension reduction:
  - Multi-dimensional scaling (MDS)
  - Principal Component Analysis (PCA)
- Clustering
  - Hierarchical Clustering
  - k-Means Clustering

Supervised vs Unsupervised methods

Machine Learning

- Supervised learning
- Unsupervised learning
- Semi-supervised learning

Supervised methods have a target, an objective.

“Can we find groups of customers who have particularly high likelihoods of cancelling their service soon after their contracts expire?”

Unsupervised methods have no specific target.

“Do our customers naturally fall into different groups?”

Exploratory Data Analysis

Exploratory data analysis

(EDA) largely uses unsupervised methods to examine the structure and patterns within the data.

The objectives of EDA (according to John Tukey) are to:

- Suggest hypotheses about the causes of observed phenomena
- Assess assumptions on which statistical inference will be based
- Support the selection of appropriate statistical tools and techniques
- Provide a basis for further data collection through surveys or experiments

https://en.wikipedia.org/wiki/Exploratory_data_analysis

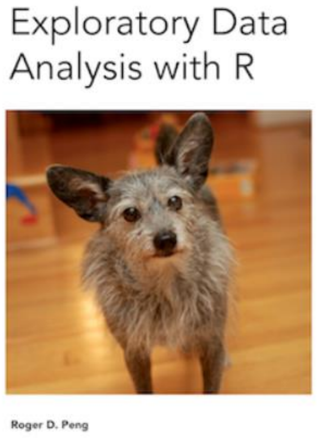
University of Exeter

Exploratory Data Analysis with R

Roger D. Peng

2020-05-01

Welcome



[Exploratory Data Analysis with R (bookdown.org)]

Similarity and distance

A distance function or metric $d(x, y)$ that tells us how far apart two data points are

- Euclidean Distance

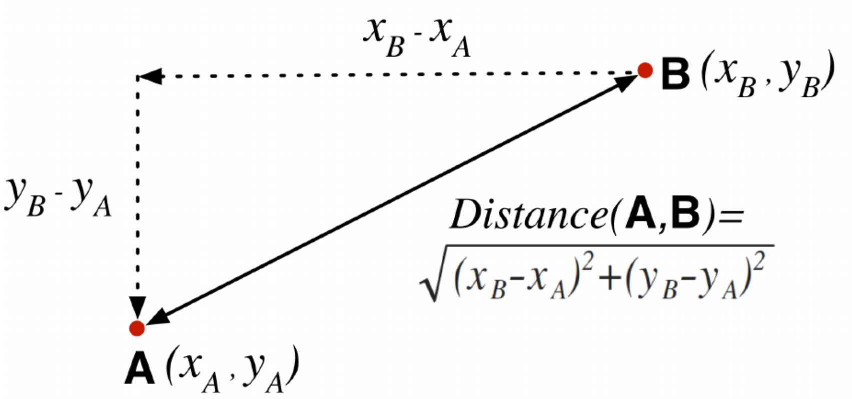
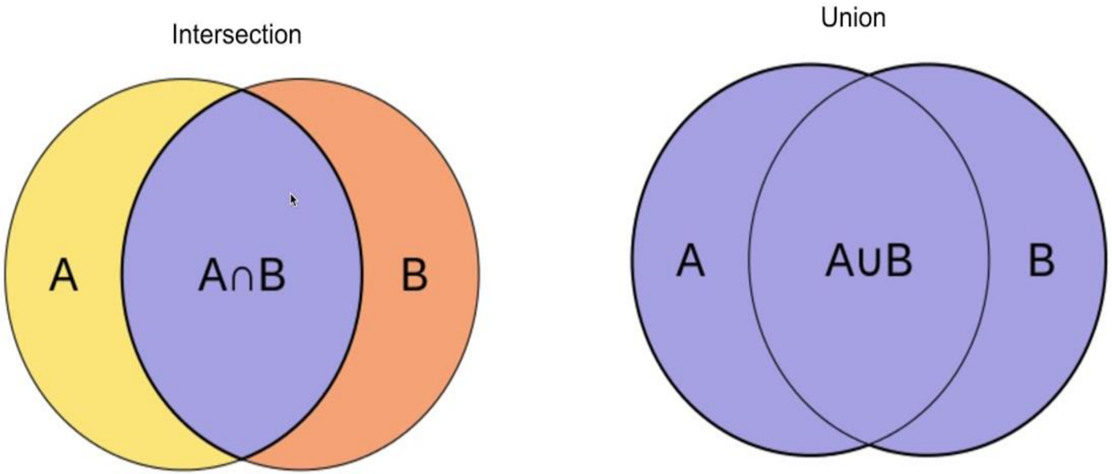


Table 6-1. Nearest neighbor example: Will David respond or not?

Customer	Age	Income (1000s)	Cards	Response (target)	Distance from David
David	*37*	*50*	*2*	*?*	0
John	35	35	3	Yes	$\sqrt{(35-37)^2 + (35-50)^2 + (3-2)^2} = 15.16$
Rachael	22	50	2	No	$\sqrt{(22-37)^2 + (50-50)^2 + (2-2)^2} = 15$
Ruth	63	200	1	No	$\sqrt{(63-37)^2 + (200-50)^2 + (1-2)^2} = 152.23$
Jefferson	59	170	1	No	$\sqrt{(59-37)^2 + (170-50)^2 + (1-2)^2} = 122$
Norah	25	40	4	Yes	$\sqrt{(25-37)^2 + (40-50)^2 + (4-2)^2} = 15.74$

A distance function or metric $d(x, y)$ that tells us how far apart two data points are

- Jaccard Similarity / Distance
- Jaccard Distance = 1 – Jaccard similarity



$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

e.g.:

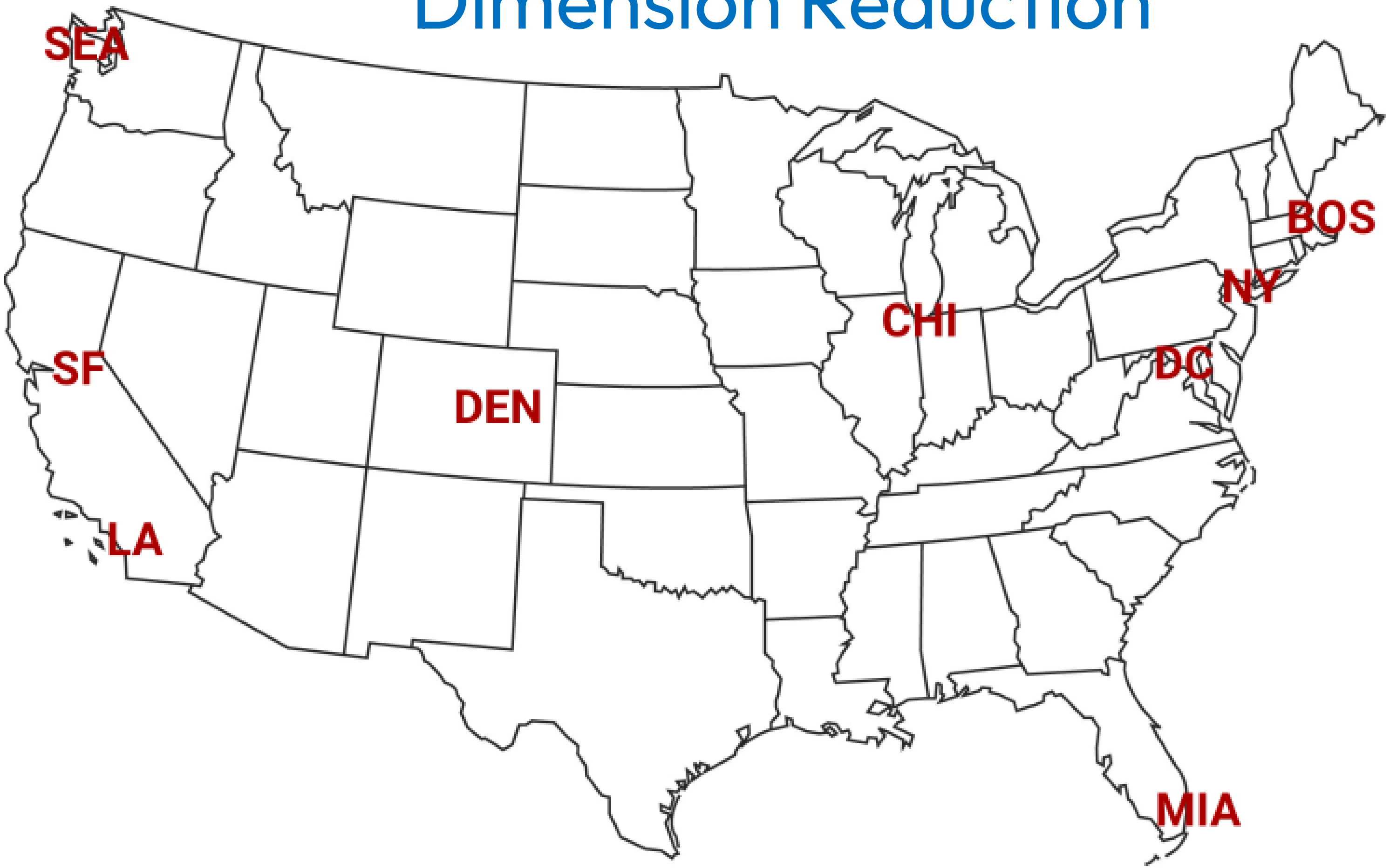
Recipe A: {salt, oil, mushrooms, bell peppers, cheese}

Recipe B: {pasta, oil, mushrooms}

J(Recipe A, Recipe B) = 2/6 = 1/3

https://youtu.be/Ah_4xqvS1WU

Dimension Reduction



Multi-dimensional Scaling (MDS)

```r
library(tidyverse)

cities <- read_csv('city_distance.csv')
view(cities)
```

We don't know the actual locations of the cities – we only know the distance between them.

Each exists in high-dimensional space (9D), not just an x and a y.

	city	BOS	NY	DC	MIA	CHI	SEA	SF	LA	DEN
1	BOS	0	206	429	1504	963	2976	3095	2979	1949
2	NY	206	0	233	1308	802	2815	2934	2786	1771
3	DC	429	233	0	1075	671	2684	2799	2631	1616
4	MIA	1504	1308	1075	0	1329	3273	3053	2687	2037
5	CHI	963	802	671	1329	0	2013	2142	2054	996
6	SEA	2976	2815	2684	3273	2013	0	808	1131	1307
7	SF	3095	2934	2799	3053	2142	808	0	379	1235
8	LA	2979	2786	2631	2687	2054	1131	379	0	1059
9	DEN	1949	1771	1616	2037	996	1307	1235	1059	0

Multi-dimensional Scaling (MDS)

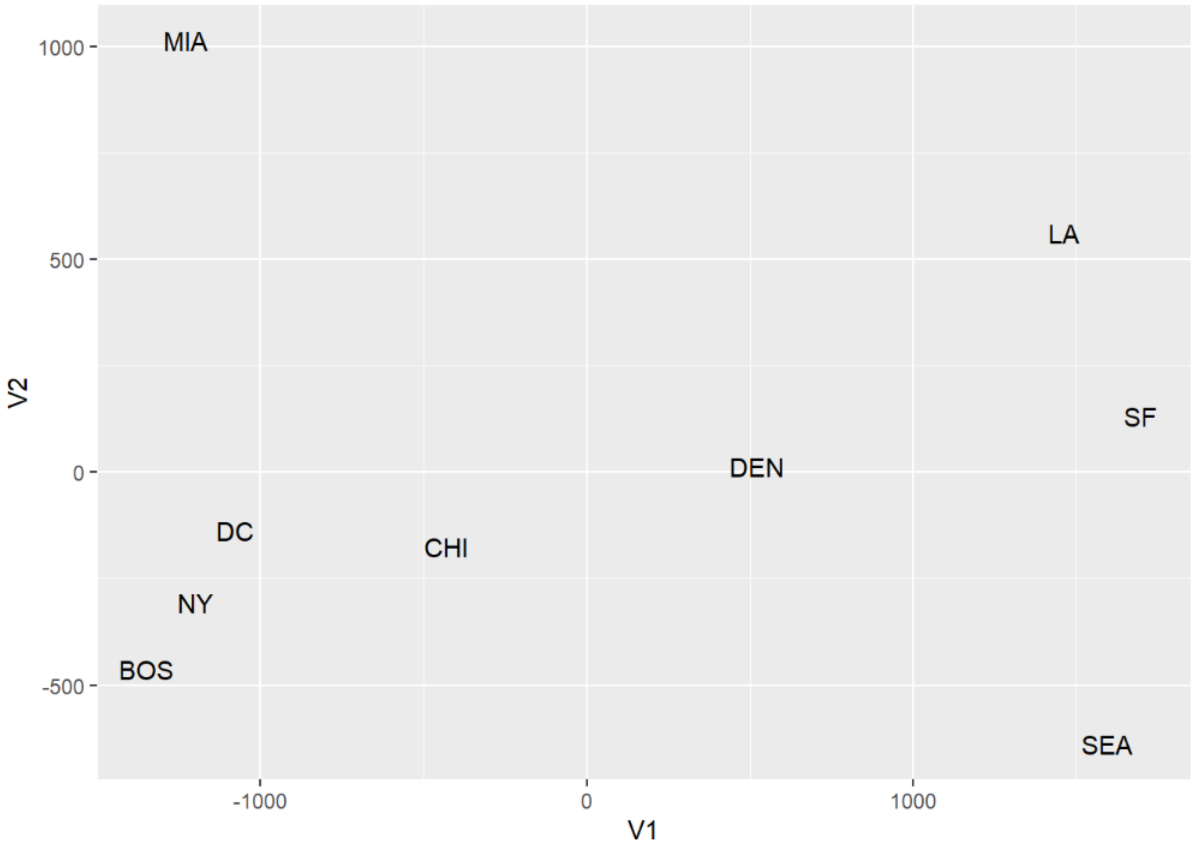
```
md_cities <- select(cities, -city) %>%
          cmdscale() %>%
          as.data.frame()

md_cities$city_name <- cities$city
view(md_cities)
```

	V1	V2	city_name
1	-1348.6683	-462.40060	BOS
2	-1198.8741	-306.54690	NY
3	-1076.9855	-136.43204	DC
4	-1226.9390	1013.62838	MIA
5	-428.4548	-174.60316	CHI
6	1596.1594	-639.30777	SEA
7	1697.2283	131.68586	SF
8	1464.0470	560.58046	LA
9	522.4871	13.39576	DEN

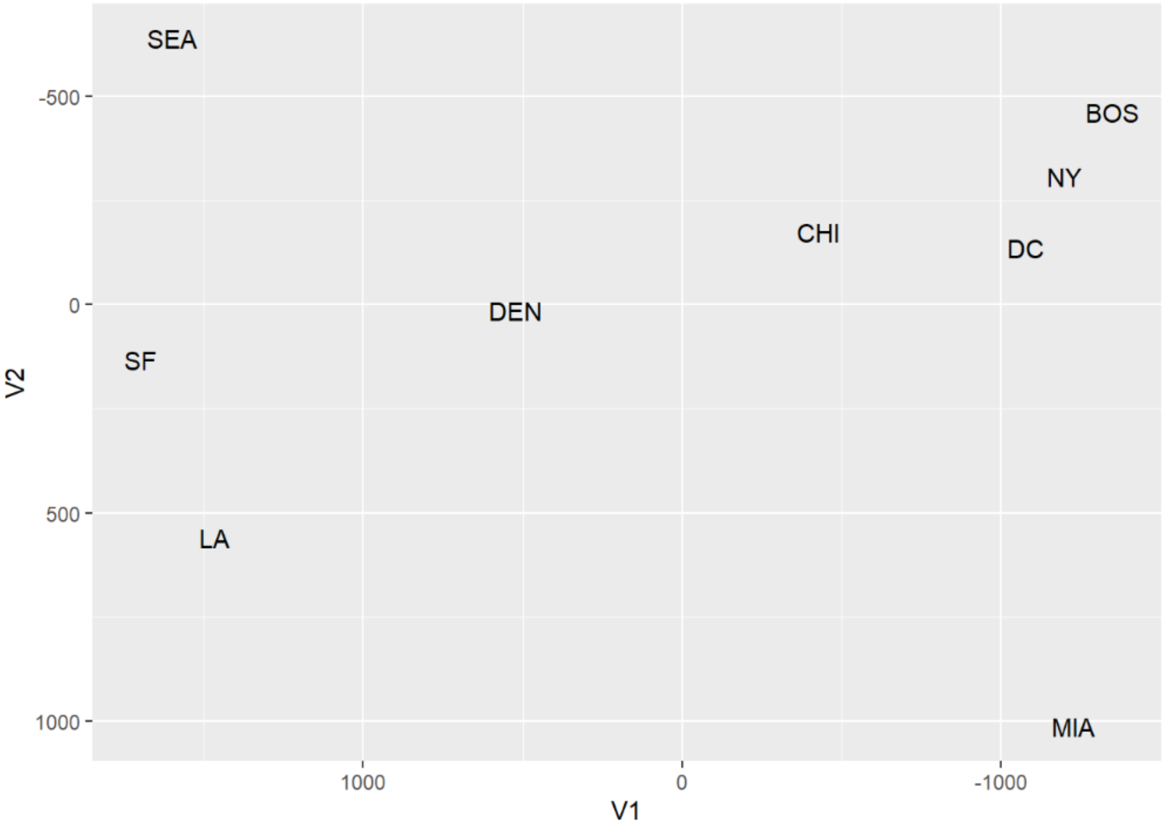
MDS creates a configuration of points in a lower-dimensional space, such that the distances between the points reflect the dissimilarities between the objects as closely as possible.

Multi-dimensional Scaling (MDS)

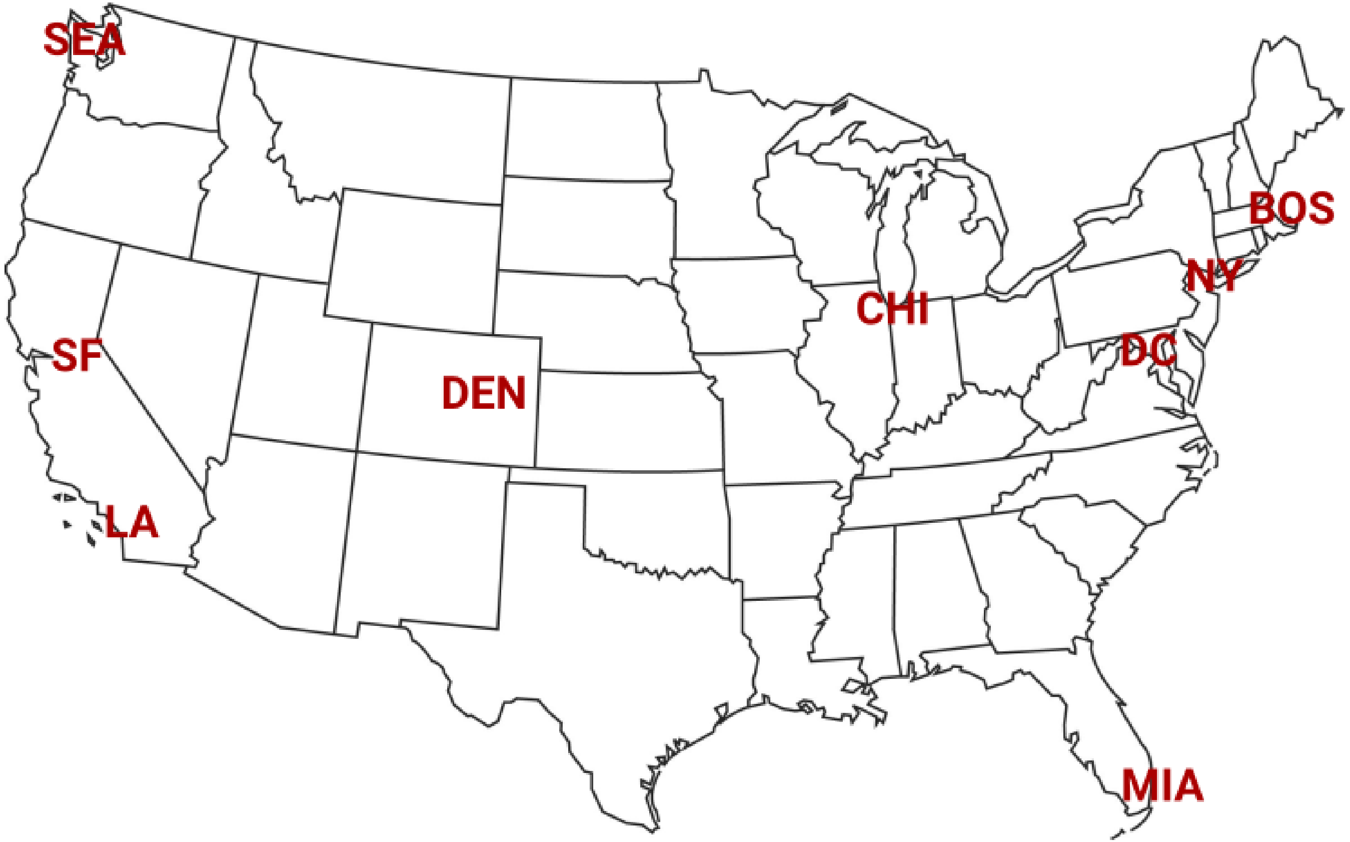
```
ggplot(md_cities, aes(x = V1, y = V2)) +
    geom_text(aes(label = city_name))
```



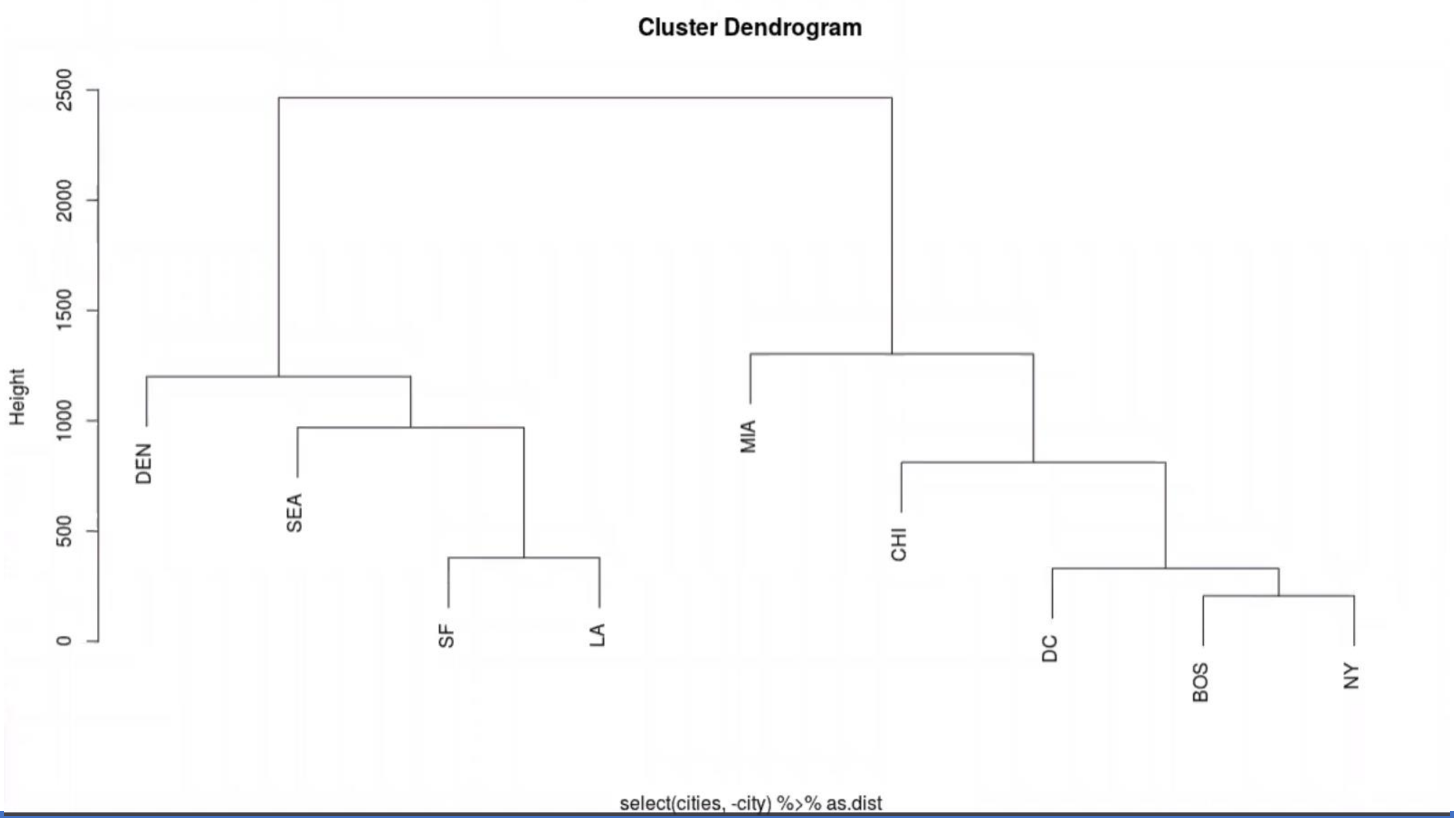
```
ggplot(md_cities, aes(x = V1, y = V2)) +
        geom_text(aes(label = city_name)) +
        scale_x_reverse() +
        scale_y_reverse()
```

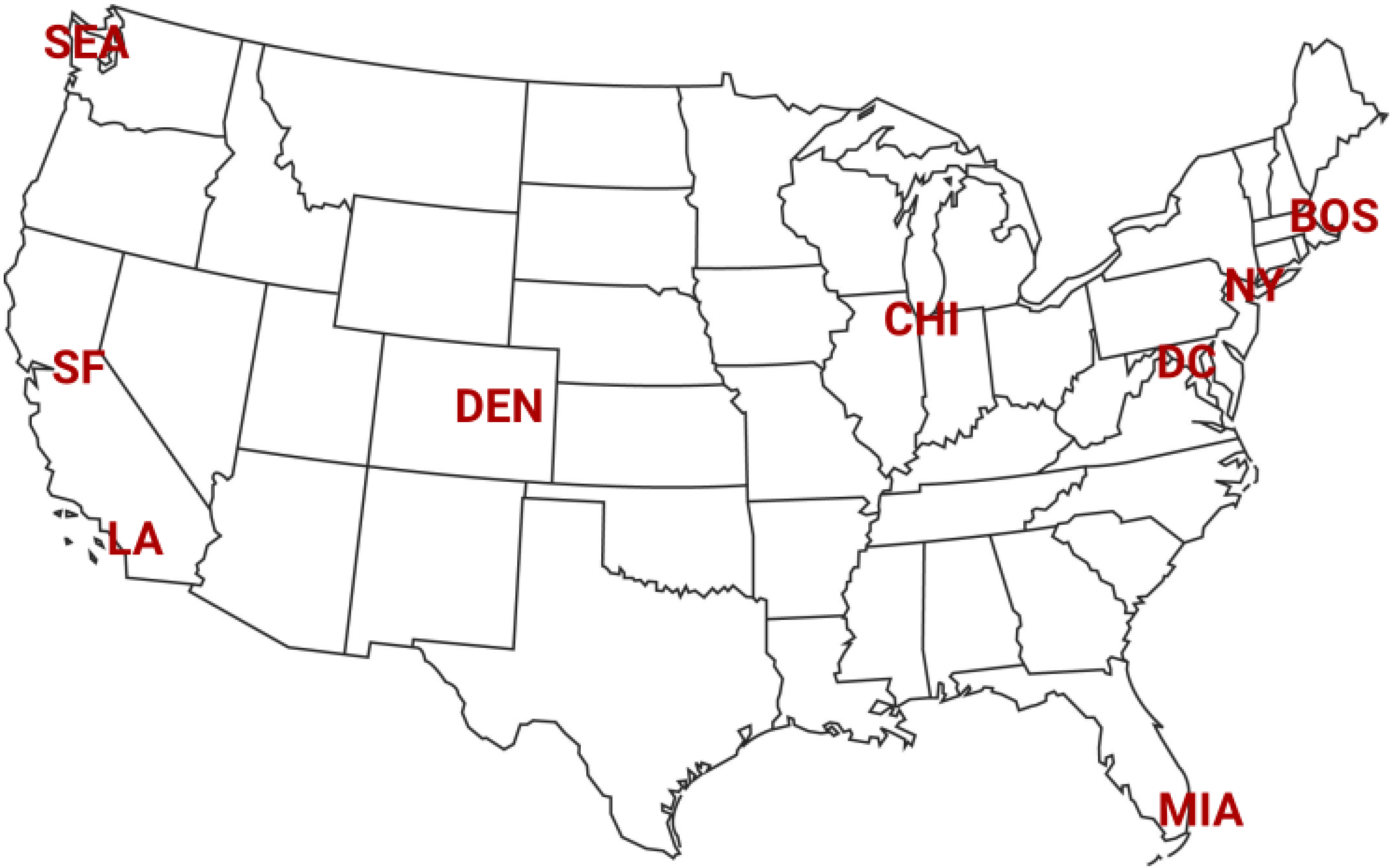


Multi-dimensional Scaling (MDS)

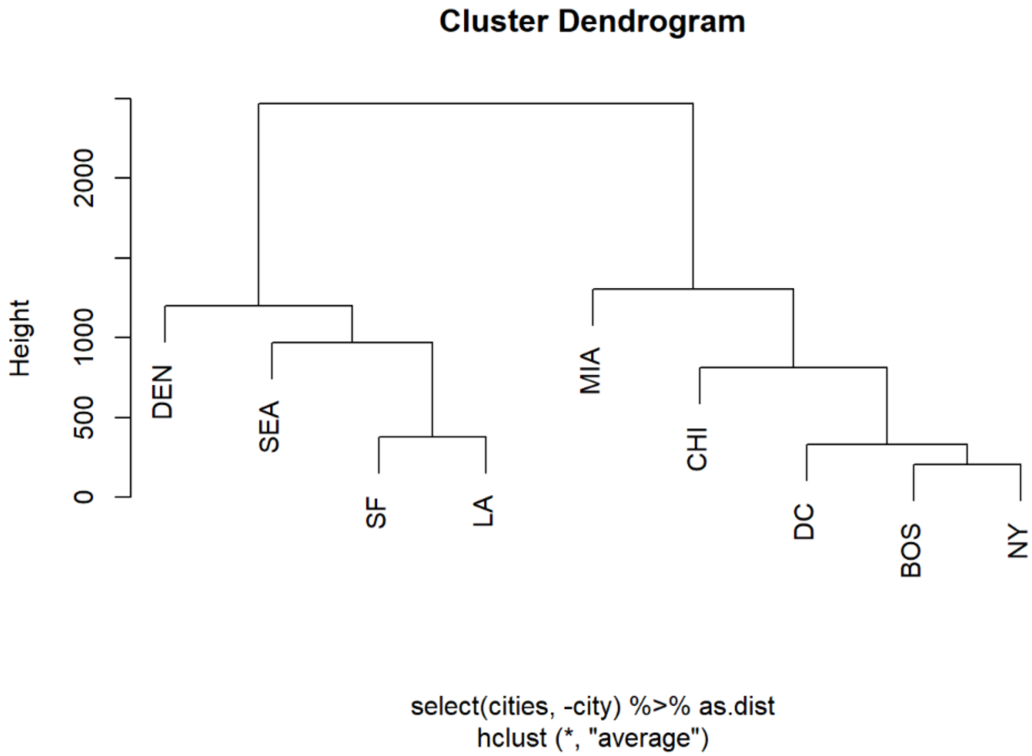


Hierarchical Clustering




SEA
SF
LA
DEN
CHI
NY
BOS
DC
MIA

```r
cities_hc <- hclust(select(cities, -city) %>% as.dist,
method = 'ave')
plot(cities_hc)
```

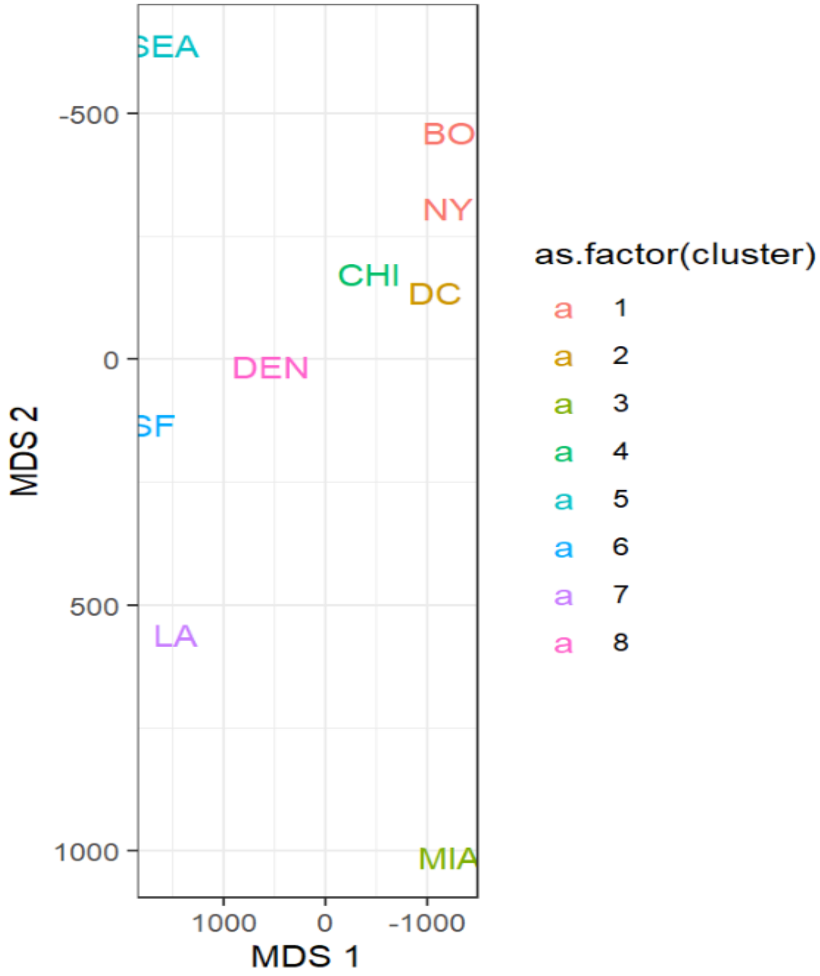


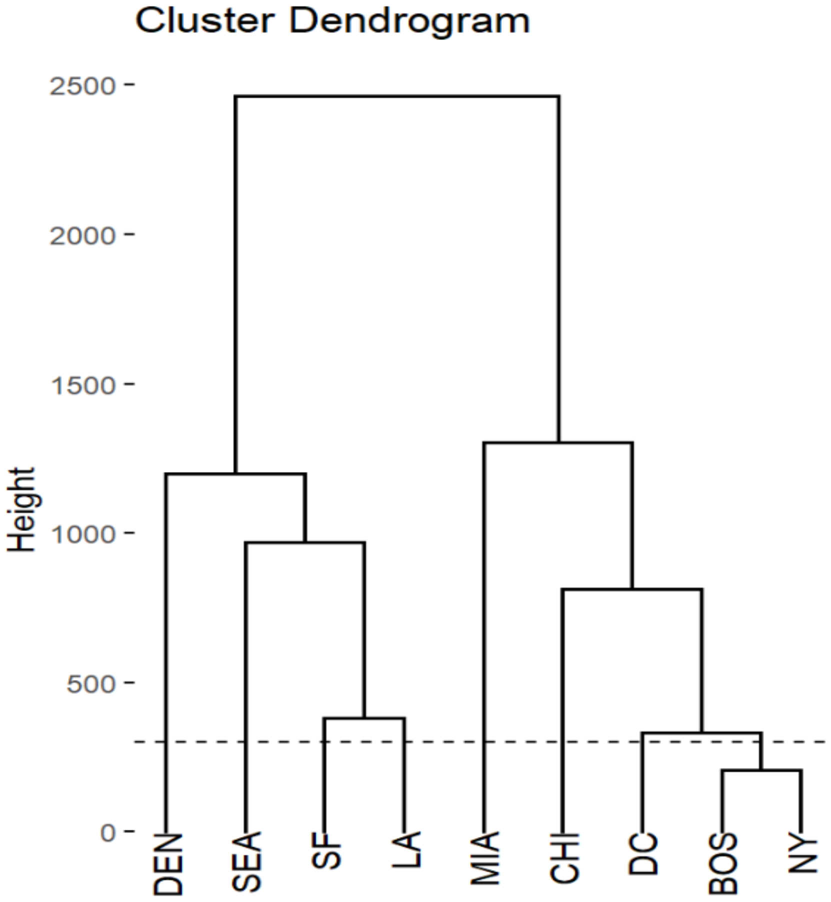
Multi-dimensional Scaling (MDS)

```
library(tidyverse)

cities <- read_csv('city_distance.csv')
view(cities)
```

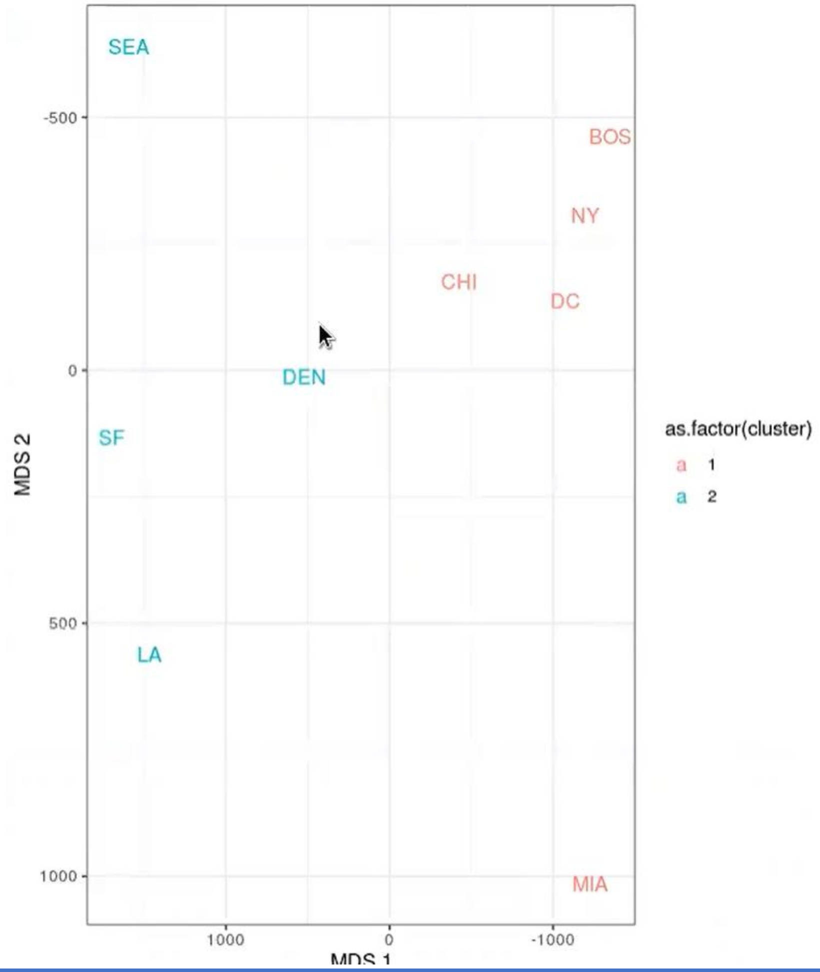
	city	BOS	NY	DC	MIA	CHI	SEA	SF	LA	DEN
1	BOS	0	206	429	1504	963	2976	3095	2979	1949
2	NY	206	0	233	1308	802	2815	2934	2786	1771
3	DC	429	233	0	1075	671	2684	2799	2631	1616
4	MIA	1504	1308	1075	0	1329	3273	3053	2687	2037
5	CHI	963	802	671	1329	0	2013	2142	2054	996
6	SEA	2976	2815	2684	3273	2013	0	808	1131	1307
7	SF	3095	2934	2799	3053	2142	808	0	379	1235
8	LA	2979	2786	2631	2687	2054	1131	379	0	1059
9	DEN	1949	1771	1616	2037	996	1307	1235	1059	0

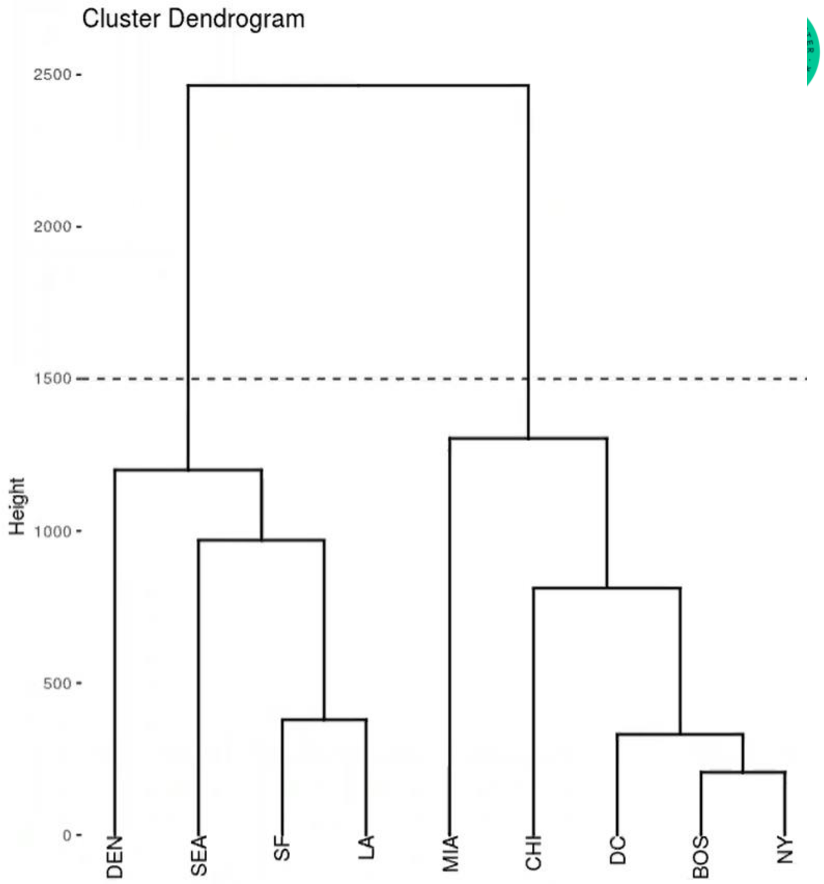
```r
library(gridExtra)

miles_apart <- 300
md_cities$cluster <- cutree(cities_hc, h = miles_apart)
p1 <- ggplot(md_cities, aes(x = V1, y = V2, color = as.factor(cluster)))+
  scale_x_reverse('MDS 1') +
  scale_y_reverse('MDS 2') +
  geom_text(aes(label = city_name)) +
  theme_bw()
p2 <- fviz_dend(cities_hc) +
  geom_hline(yintercept = miles_apart, linetype = 2)

grid.arrange(p1, p2, nrow=1)
```





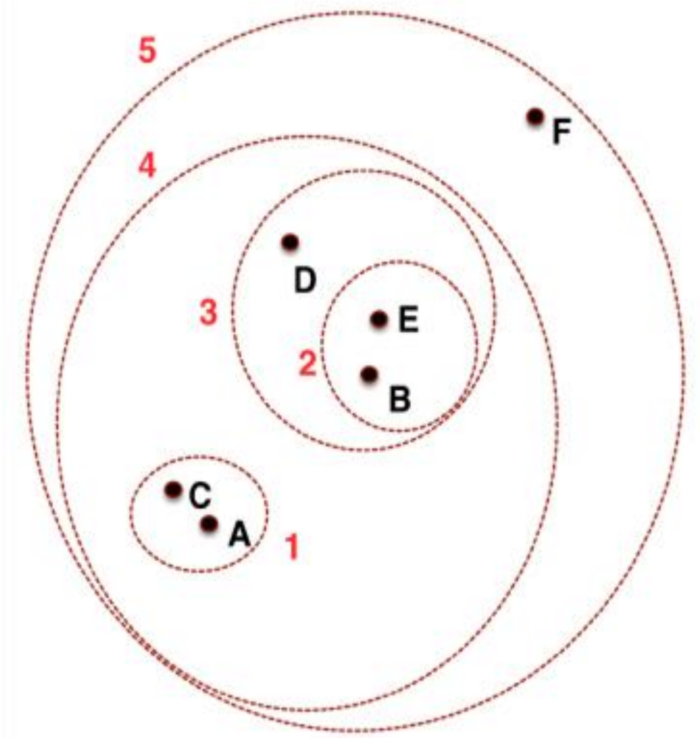
iversity
Exeter

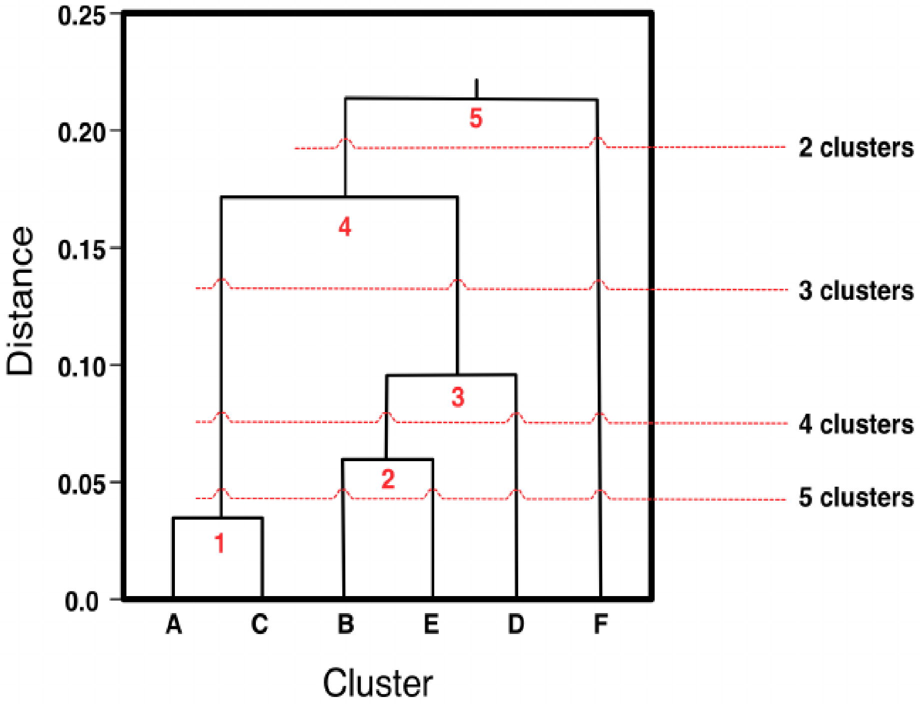


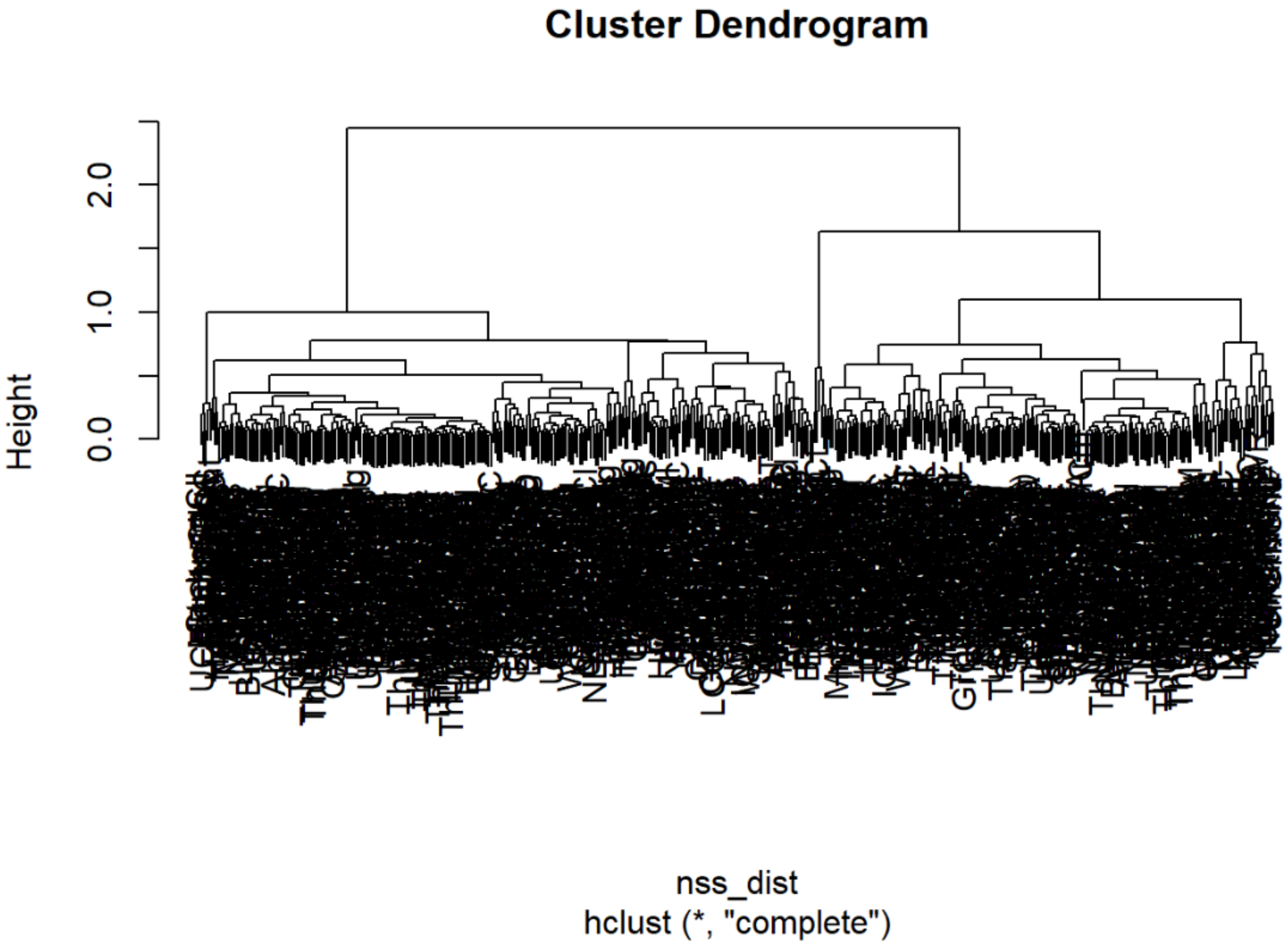


University of Exeter

Hierarchical Clustering






Cluster Dendrogram
Height
2.0
1.0
0.0
nss_dist
hclust (*, "complete")

K-means Clustering

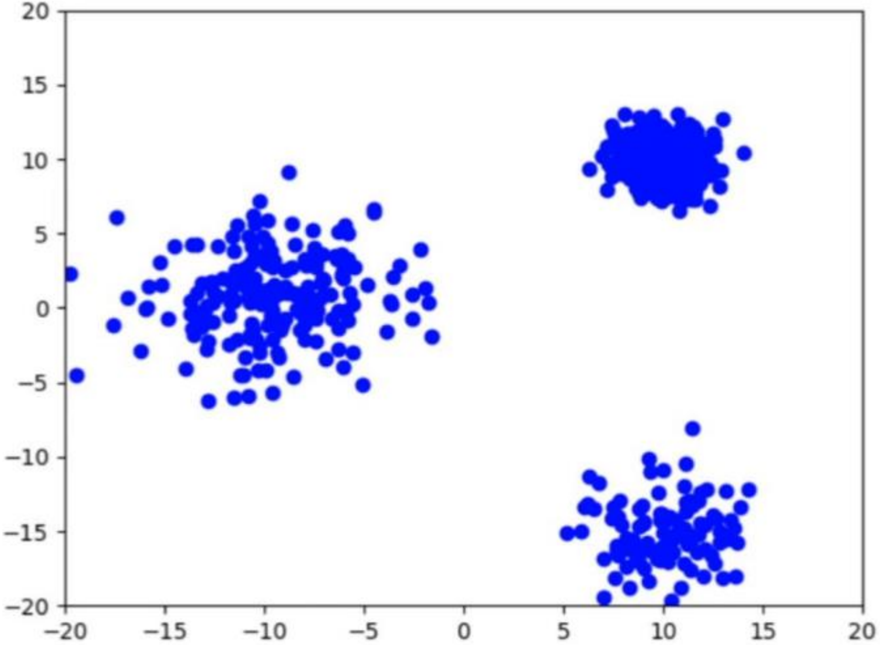
Starting with N data points $\{x_{1,} x_{2,} \ldots, x_N\}$

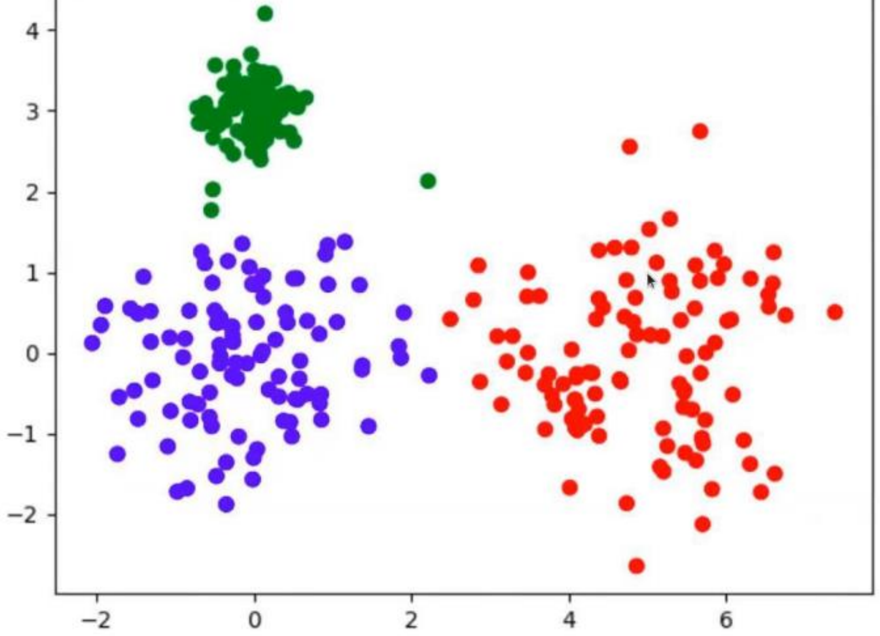
Choose k, i.e. the number of clusters

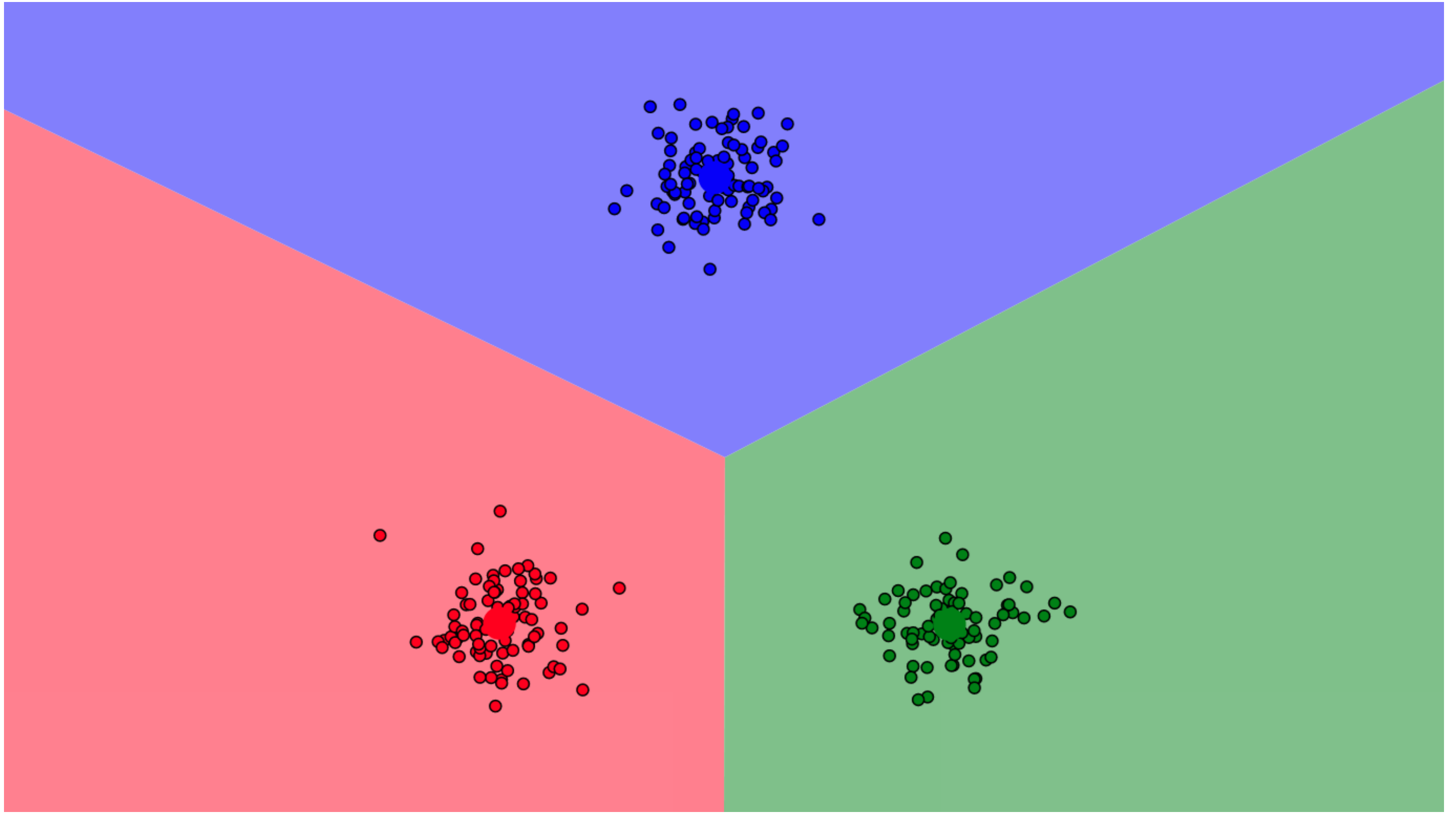
Choose k cluster centres $\{c_{1,} c_{2,} \ldots, c_N\}$

[StatQuest: K-means clustering - YouTube]

Clustering

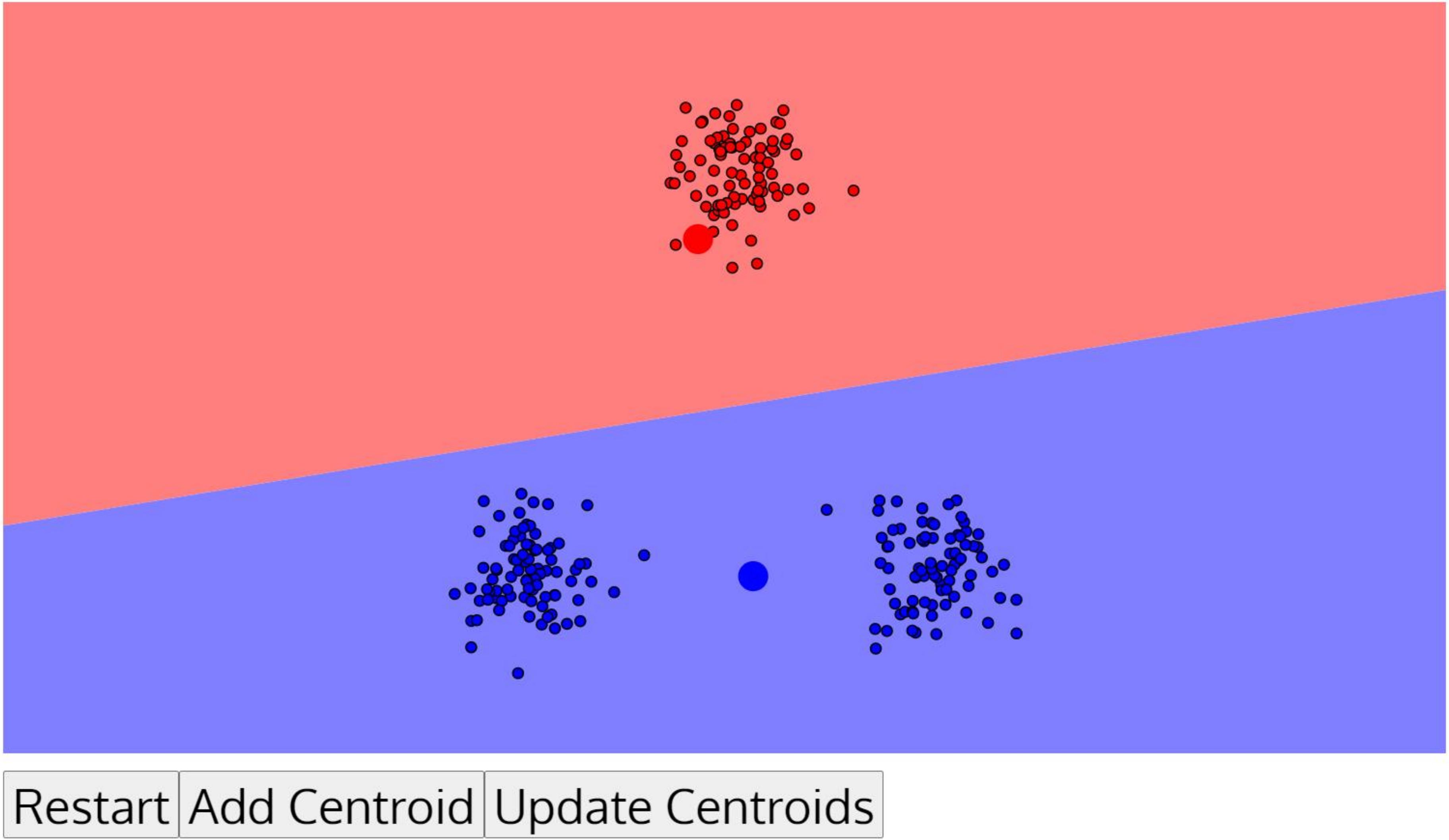


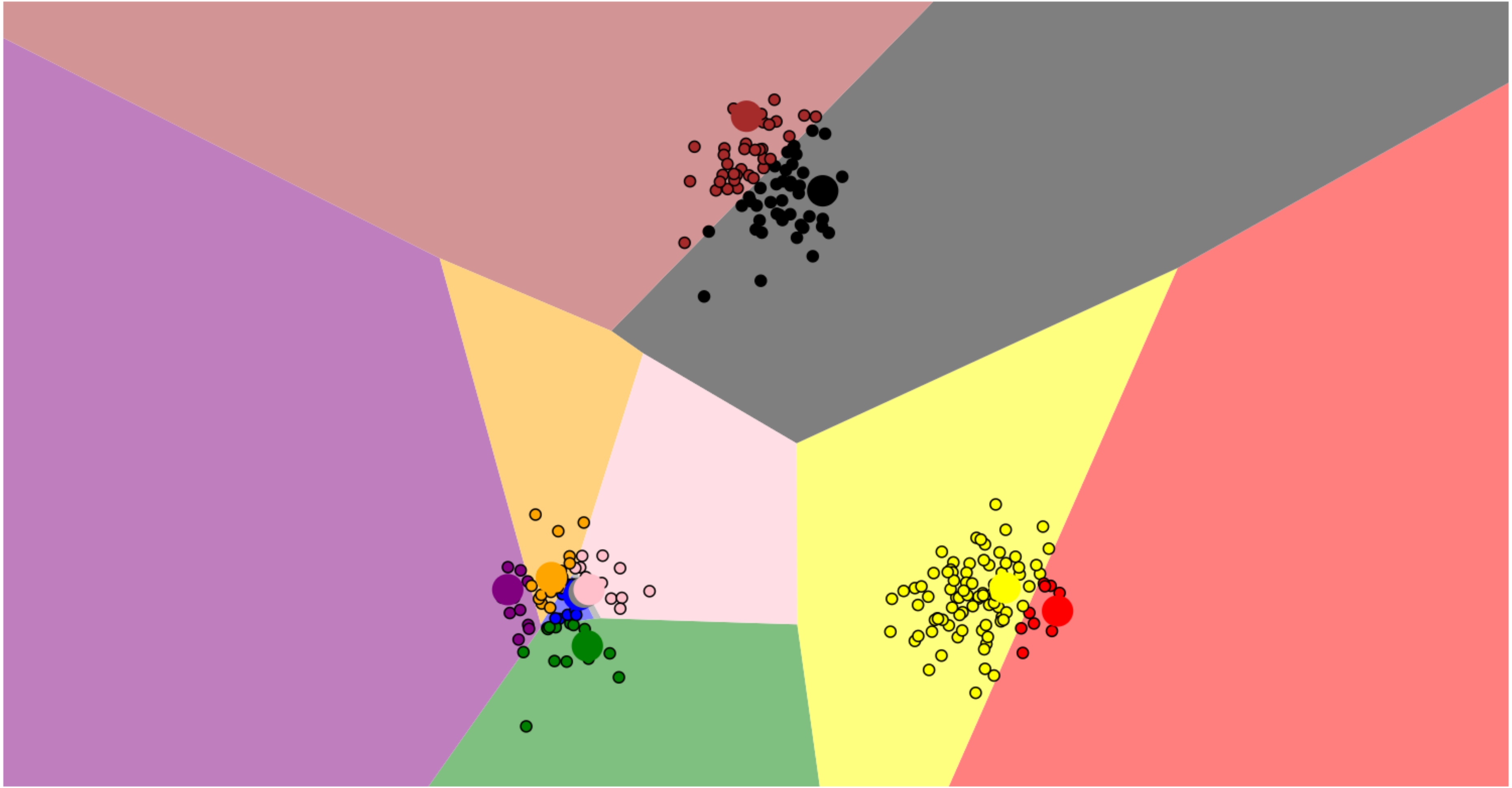




[Visualizing K-Means Clustering (naftaliharris.com)]

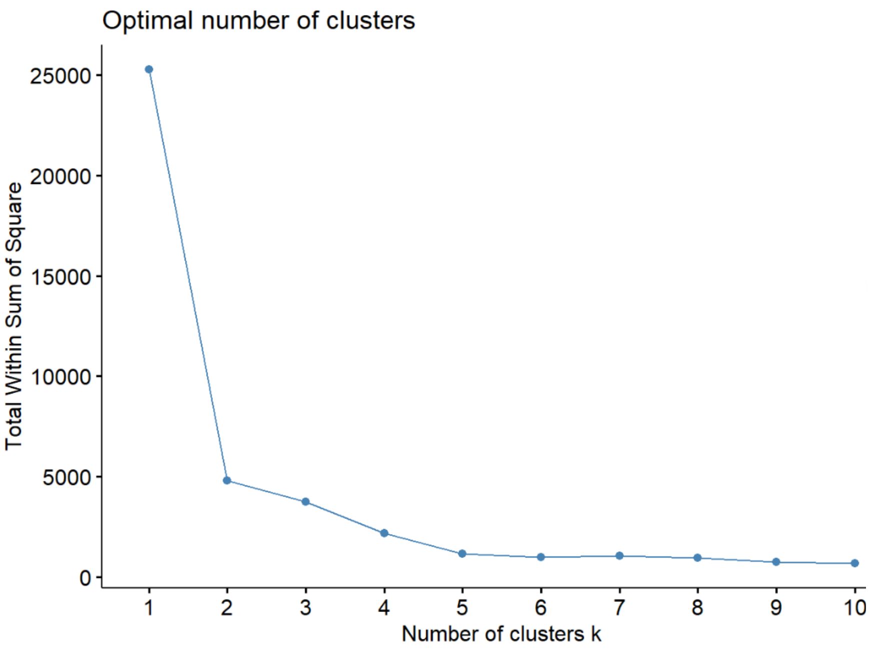
[d3.js ~ Voronoi Diagram (strongriley.github.io)]

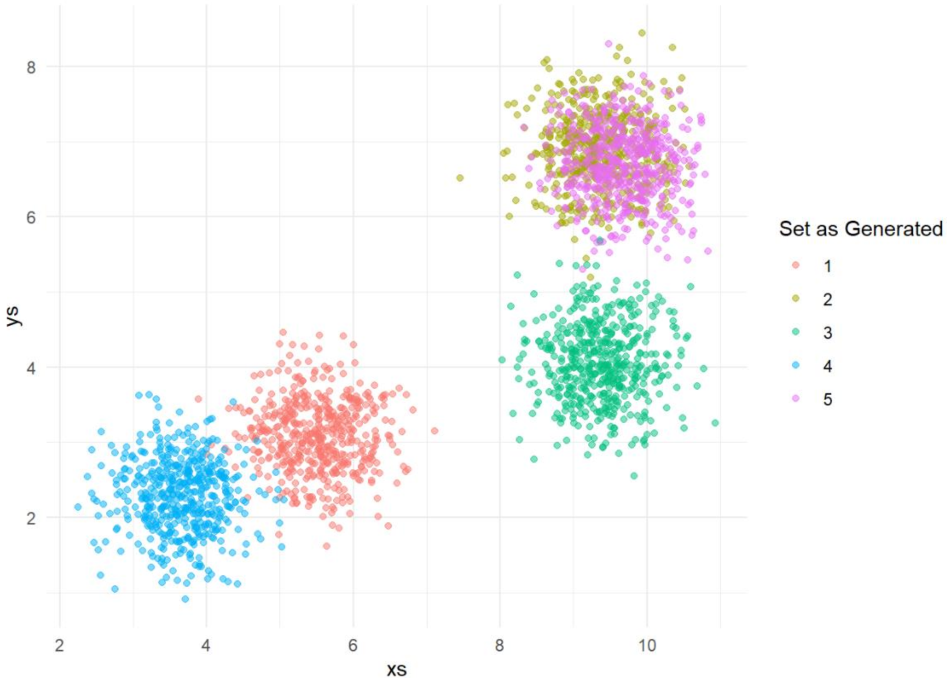

Restart
Add Centroid
Update Centroids



Restart Update Centroids

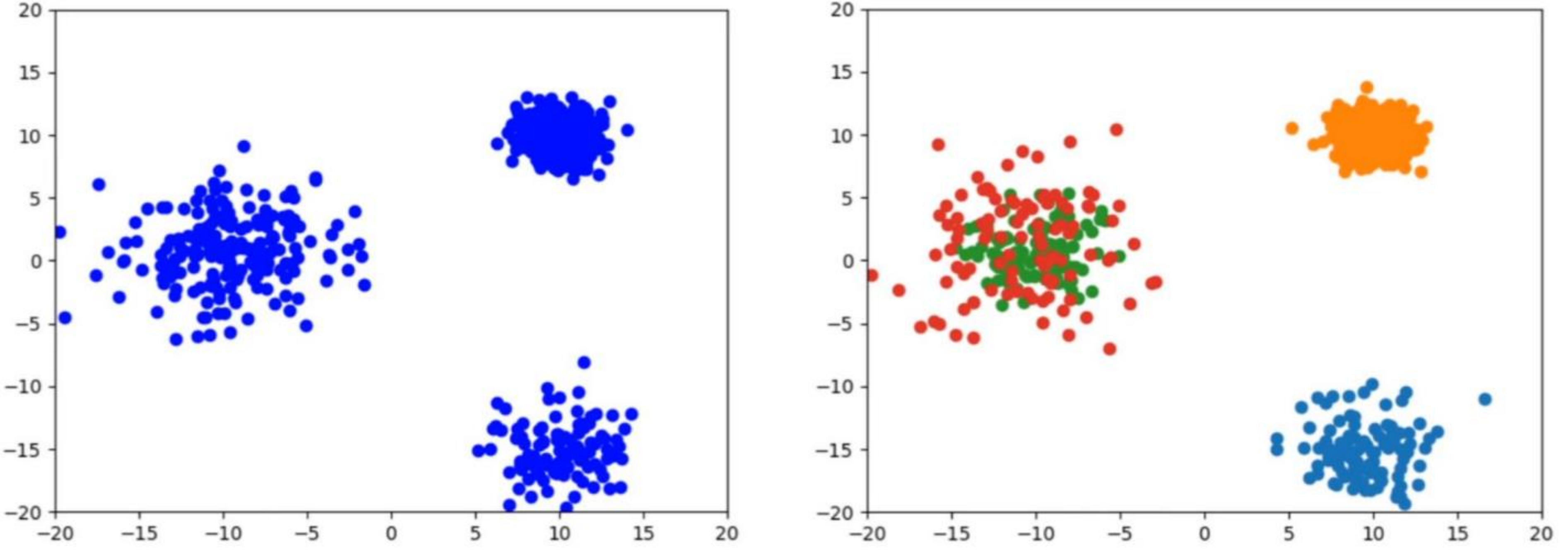
```
fviz_nbclust(d,kmeans, method = 'wss')
```



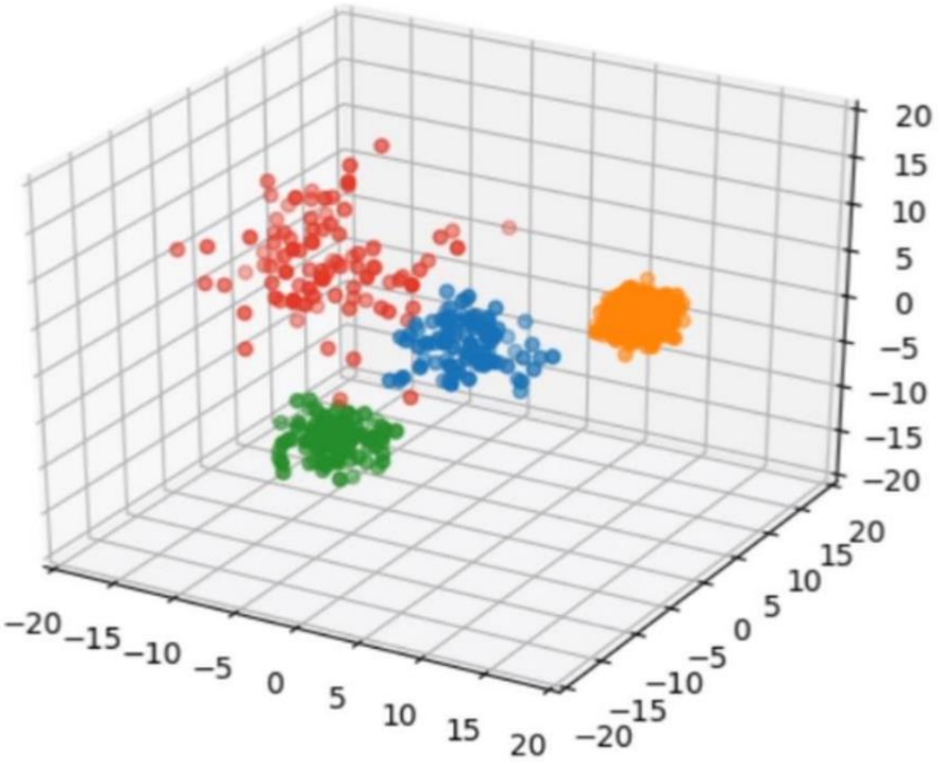


```r
set.seed(111222333)
k <- 5
N <- 500
xs <- runif(k, 0, 10)
ys <- runif(k, 0, 10)
d <- data.frame(xs = lapply(xs, function(x) rnorm(N, mean = x, sd = 0.5)) %>%
        unlist, ys = lapply(ys, function(x) rnorm(N, mean = x, sd = 0.5)) %>%
        unlist)

ggplot(d, aes(x = xs, y = ys, color = factor(rep(1:k, each = N)))) +
geom_point(alpha = 0.5) + theme_minimal() + scale_color_discrete('Set as
Generated')
```

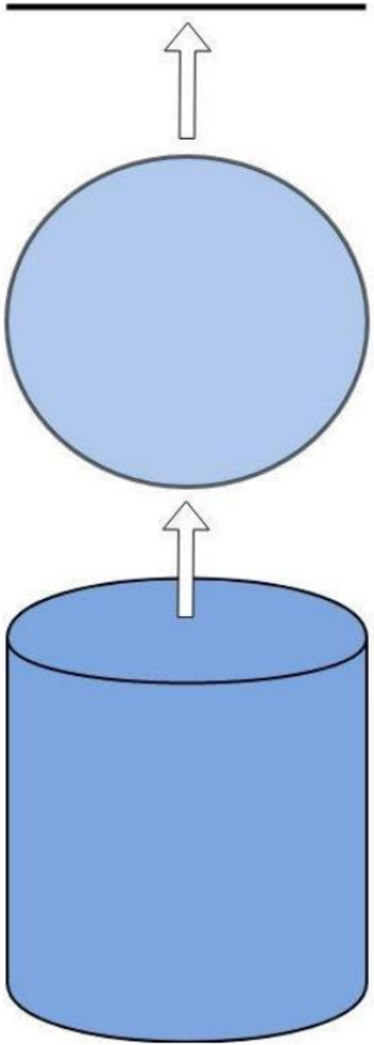
Clustering



Clustering



Dimensionality Reduction

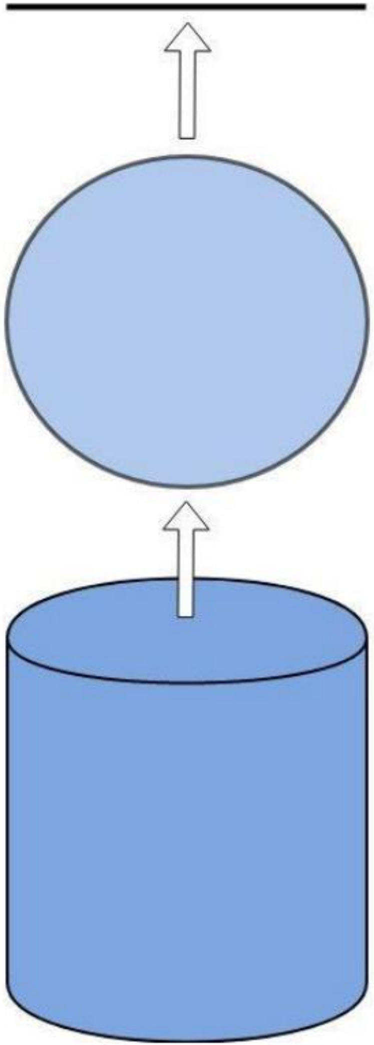


- Data can contain high level features
- Amazon books:
  - 1000s of customers
  - 1000s of books
  - Customer is a vector (0,1,0,0,0,...,0)
- What are some characteristics shared by many books/customers?

High-level features: language, genre, author etc..

Dimensionality Reduction

- Computational costs (time, memory, storage etc
- Degradation of model performance
- Feature redundancy (correlations)
- Data visualisation



Dimensionality Reduction

- **Feature Selection** – selecting a subset of relevant features (ignore redundant, irrelevant features)

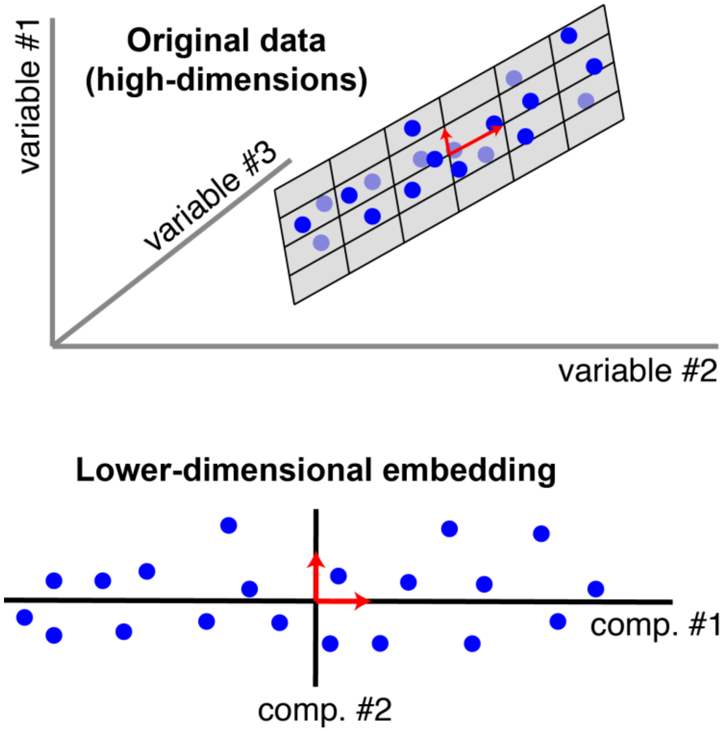
  - Original features are retained

- **Feature extraction** – finding a smaller set of features, in lower dimensional space, by extracting/deriving information from the original features in space.

  - Data is transformed by mapping it in the new lower dimensional feature space.

  - For example, multi-dimensional scaling (MDS), principal component analysis (PCA)

Principal Component Analysis (PCA)

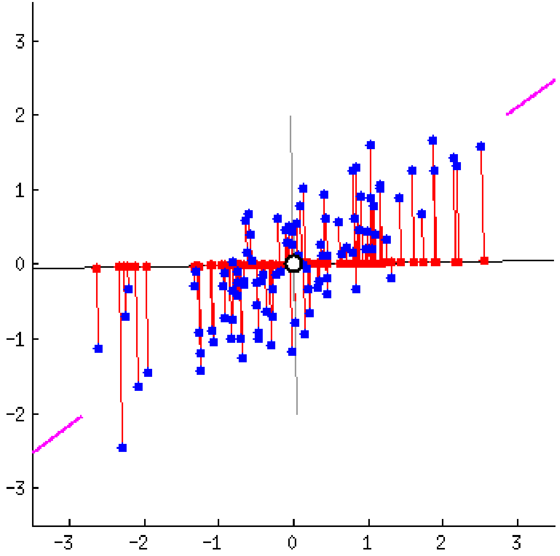


- A linear dimensionality reduction technique
- Set of new features called **principal components** are extracted from an existing set
- New features are expressed as **weighted linear combinations** of the original data

Everything you did and didn't know about PCA

http://setosa.io/ev/principal-component-analysis
Further explanation with pretty visualisations

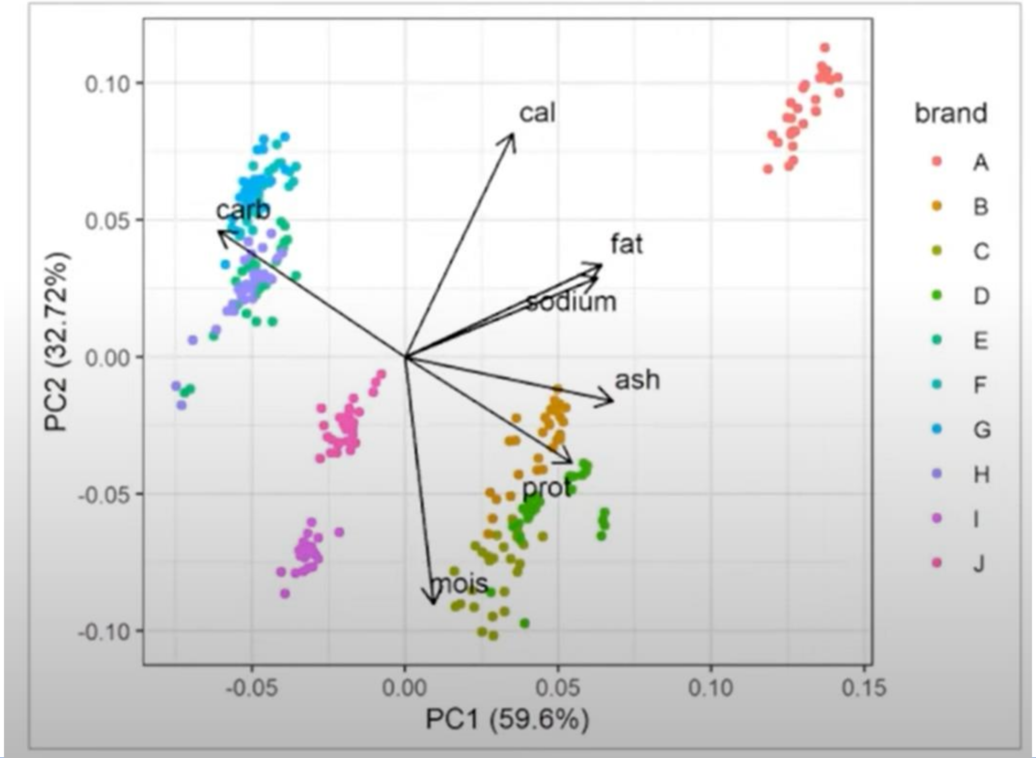
Principal Component Analysis (PCA)



As there are as many principal components as there are variables in the data, principal components are constructed in such a manner that the first principal component accounts for the **largest possible variance** in the data set.

Principal Component Analysis (PCA)

Biplot

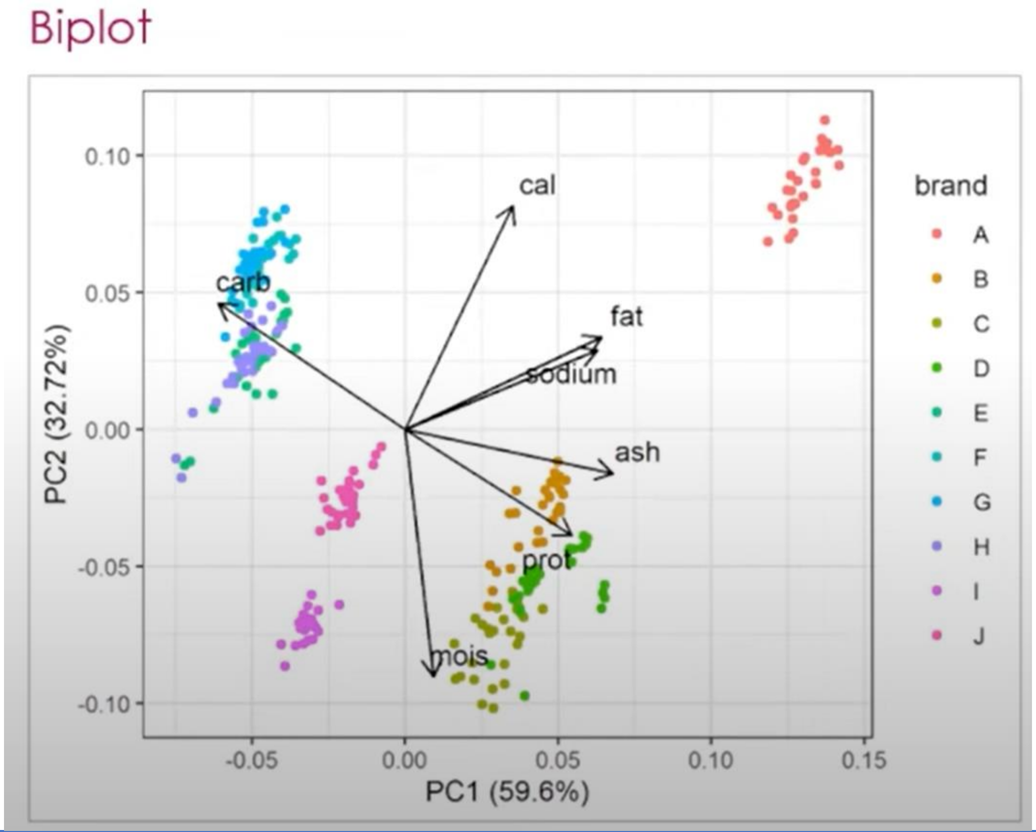


9 dimensional data (attributes of pizzas) are reduced to 2 principal components (x and y).

Each pizza data point is plotted. Colour is added by pizza brand which shows some similarity within brands.

The vectors are calculated for each original feature. We can make some interpretations of the analysis.

Principal Component Analysis (PCA)



PCA is used to identify the directions (principal components) that maximise the variance in the data. It projects the data onto new axes which are linear combinations of the original variables, while preserving as much variation as possible.

vechttps://www.youtube.com/watch?v=DngS4LNNzc8&t=0s

Next Week: Predictive Modelling

- Read Data Science for Business, chapters 3 and 4
- Watch StatQuest: Decision Trees
- Watch StatQuest: Random Forests Part 1
- Watch StatQuest: Random Forests Part 2
- Play A Visual Introduction to Machine Learning
- Play Random Forest Playground
- Play Linear Regression (try clicking and dragging on points)

Any questions?

?