SMI606: Week 10
Cluster Analysis

Dr. Calum Webb
Sheffield Methods Institute, the University of Sheffield.
c.j.webb@sheffield.ac.uk

Sign In

Learning Objectives

What will I learn? | How does this week fit into my course?

By the end of this week you will:

- Be able to identify what kinds of social science research questions cluster analysis may be useful for answering.
- Be able to explain, in basic terms, how two types of cluster analysis methods work: **k-means** clustering and **Hierarchical Cluster Analysis** clustering.
- Know which forms of cluster analysis are appropriate for the specific data and underlying clustering phenomenon you are interested in.
- Be able to run and inspect the output of **k-means** and **HCA** analyses of data in **R**
- Be able to describe the characteristics of clusters derived from **k-means** and **HCA** analyses.

Learning Objectives

What will I learn? | How does this week fit into my course?

- Cluster analysis is an increasingly important tool — both in social research itself and in understanding how data and machine learning algorithms are used in society. This week will teach you how it operates.
- Cluster analysis can be used in conjunction with regression analysis to explore interesting research questions, or to frame them in interesting ways.
- Cluster analysis can be used for the refinement of theory, particularly in the study of social ordering and social grouping. It can also be useful (along with Factor Analysis) for dealing with multicollinearity (e.g. by changing a large number of highly correlated variables into mutually distinct clustered groups).

Week 10: Cluster Analysis — Part I

What is cluster analysis and why learn it?

What is cluster analysis?

What if we have data but we are not interested in the relationship between variables?

What if, instead, our theories are more about whether our observations fall into predictable groups (or "clusters")?

Alternatively, what if we think a collection of variables are really capturing a smaller number of underlying processes (or "factors")?

What is cluster analysis?

What if we have data but we are not interested in the relationship between variables?

What if, instead, our theories are more about whether our observations fall into predictable groups (or "clusters")?

Alternatively, what if we think a collection of variables are really capturing a smaller number of underlying processes (or "factors")?

For example...

What if we want to test a theory that most people fall into two 'types' of working patterns (e.g. "sprinters", "slow-and-steadys")?

Or that learners fall into four distinct groups (e.g. "visual-learners", "auditory-learners", "kinesthetic learners", "mixed learners")?

Or that the concept of 'introversion' could be measured by questions about the places a person prefers spending their free time, what their idea of a dream holiday is, and the extent to which they find social events draining or energizing?

What is cluster analysis?

What if we have data but we are not interested in the relationship between variables?

What if, instead, our theories are more about whether our observations fall into predictable groups (or "clusters")?

Alternatively, what if we think a collection of variables are really capturing a smaller number of underlying processes (or "factors")?

Cluster Analysis

Cluster analysis refers to a collection of methods designed to **identify underlying group membership** or **hierarchical structures** in data, based on **similarities between observations**.

Factor Analysis

Factor analysis refers to a collection of methods designed to **capture and approximate an underlying construct that cannot be measured directly**, but can be approximated based on **similarities between multiple variables**.

What is cluster analysis?

What if we have data but we are not interested in the relationship between variables?

What if, instead, our theories are more about whether our observations fall into predictable groups (or "clusters")?

Alternatively, what if we think a collection of variables are really capturing a smaller number of underlying processes (or "factors")?

Cluster Analysis

"Learners fall into four distinct groups (e.g. "visual-learners", "auditory-learners", "kinesthetic learners", "mixed learners")."

Variables might be: Learner's rating of a visual-focused teaching class (0-100), Learner's rating of a auditory-focused teaching class (0-100), Learner's rating of a kinesthetic-focused teaching class (0-100), Learner's rating of a mixed-style teaching class (0-100).

Hypothesis: Learners **cluster** into three groups characterised by higher ratings of a single style, or a fourth group characterised by higher ratings of the mixed-style.

What is cluster analysis?

What if we have data but we are not interested in the relationship between variables?

What if, instead, our theories are more about whether our observations fall into predictable groups (or "clusters")?

Alternatively, what if we think a collection of variables are really capturing a smaller number of underlying processes (or "factors")?

Factor Analysis

"The concept of 'introversion' could be measured by questions about the places a person prefers spending their free time, what their idea of a dream holiday is, and the extent to which they find social events draining or energizing"

Variables might be: How much do you agree with the following statements: I prefer to spend my free time at home, rather than out and about (Strong Disagree - Strong Agree); My dream holiday would involve lots of activities, new experiences, and time socialising (Strong Disagree - Strong Agree); I find social events (birthdays, holidays, etc) energizing (Strong Disagree - Strong Agree)

Hypothesis: Reponses to the above variables correspond strongly with an **underlying factor** that can be called "introversion".

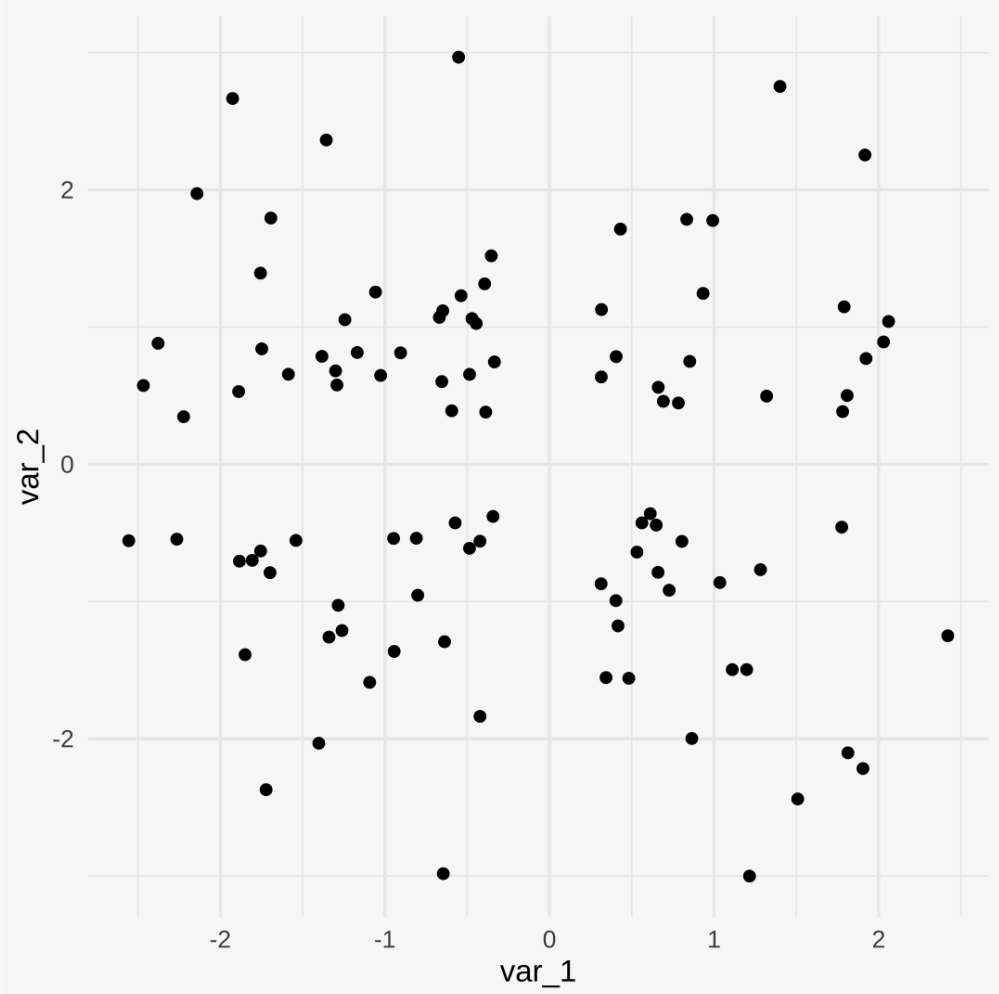
For this module, we will only be able to cover two methods of cluster analysis; however, you may find factor analysis an interesting subject (especially if you are interested in statistical measurement, e.g. psychometrics)

Week 10: Cluster Analysis — Part II

k-means: What is it and how does it work?

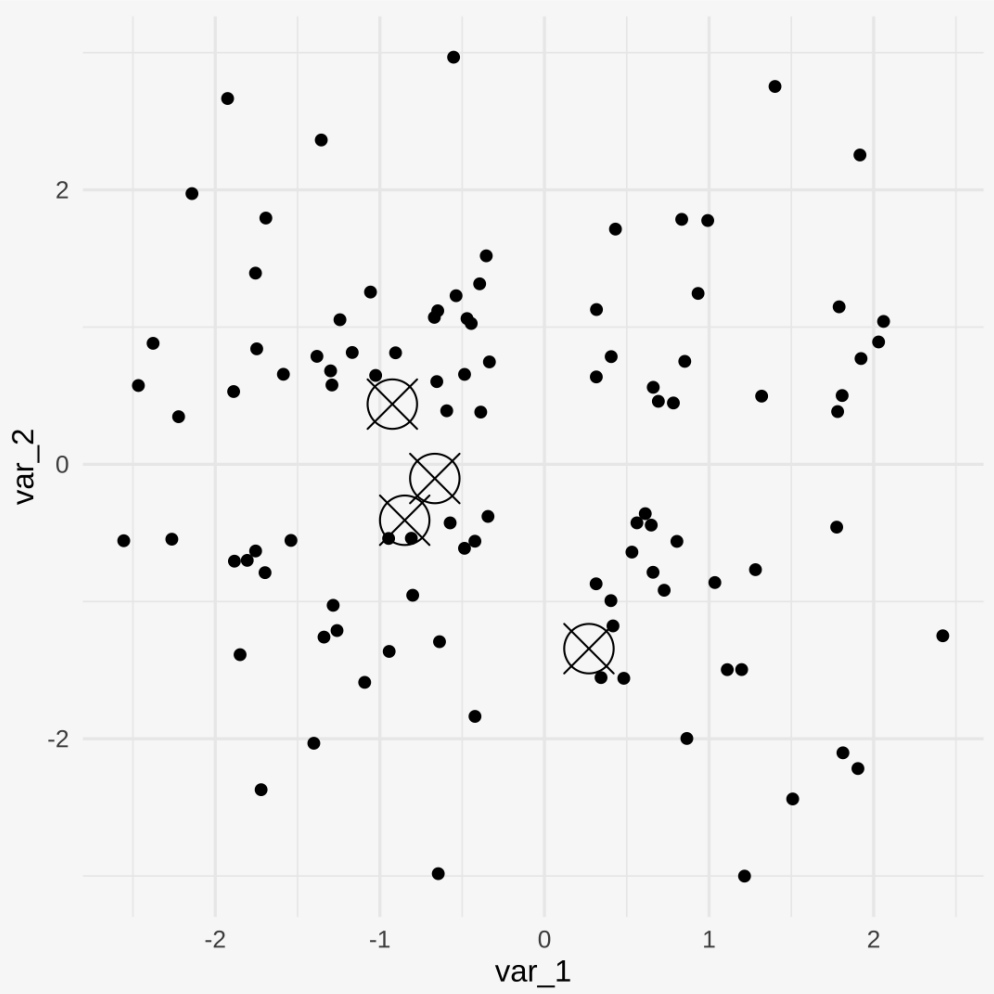
The k-means clustering algorithm: how does it work?

- k-means is a clustering algorithm that **aims to group together observations within k number of groups**. The researcher selects the number of groups and needs to decide how many should be used in the end.



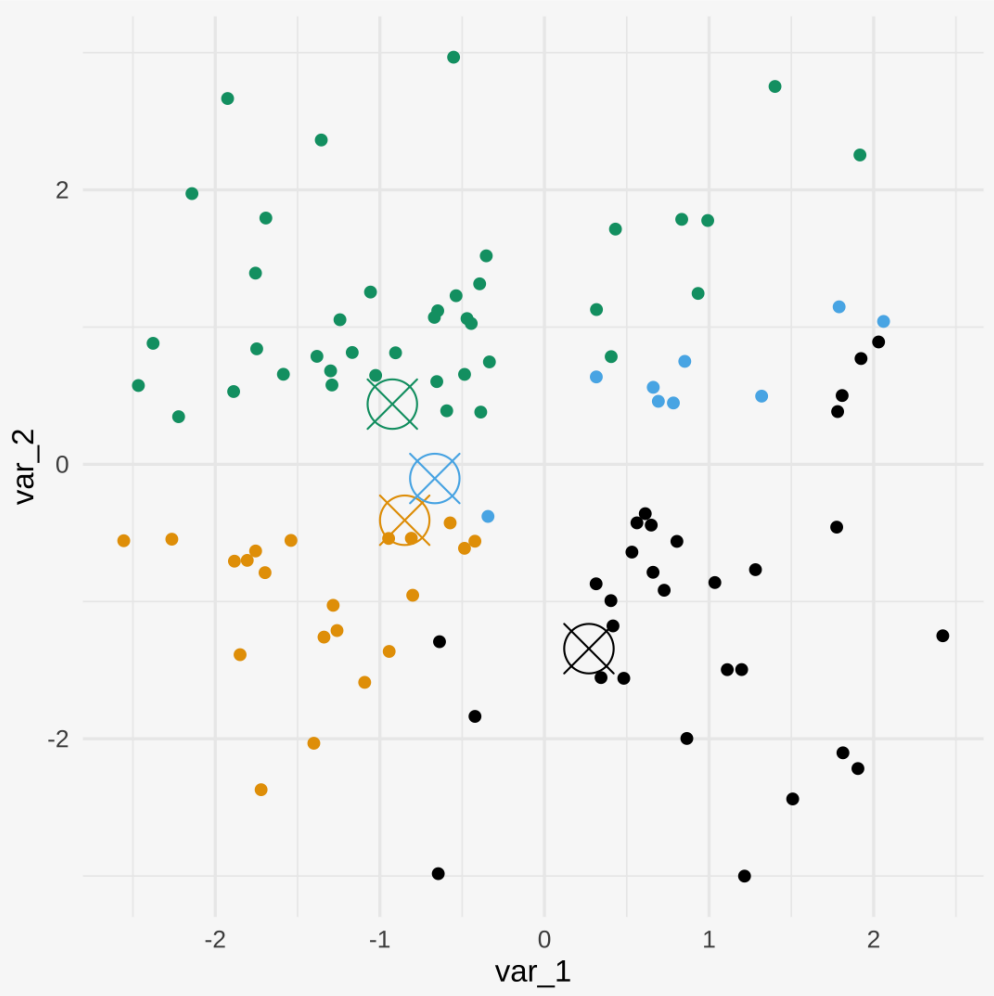
The k-means clustering algorithm: how does it work?

- k-means starts by generating k random '**centroids**'. It places these in random places in the data (indicated by circles with Xs in them).



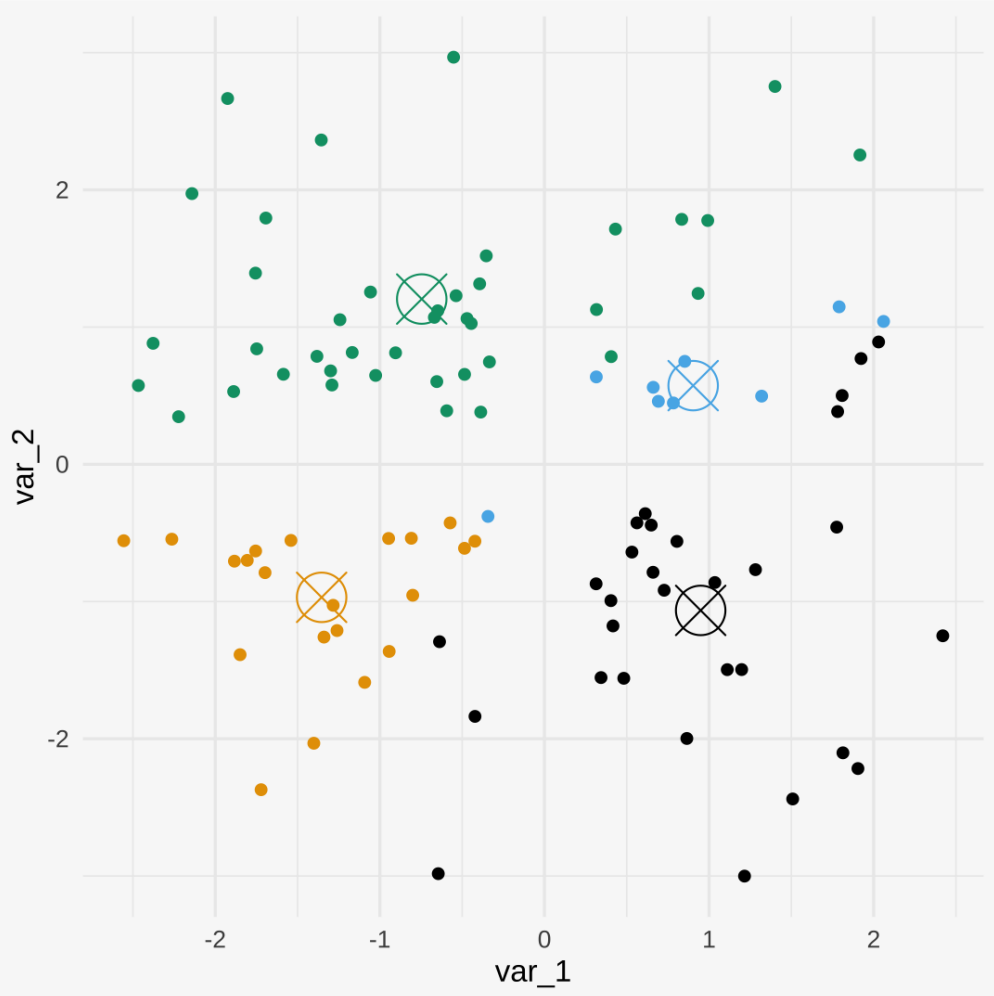
The k-means clustering algorithm: how does it work?

- It then calculates the distance between all of the points and the k centroids. It then assigns each observation to its closest centroid.



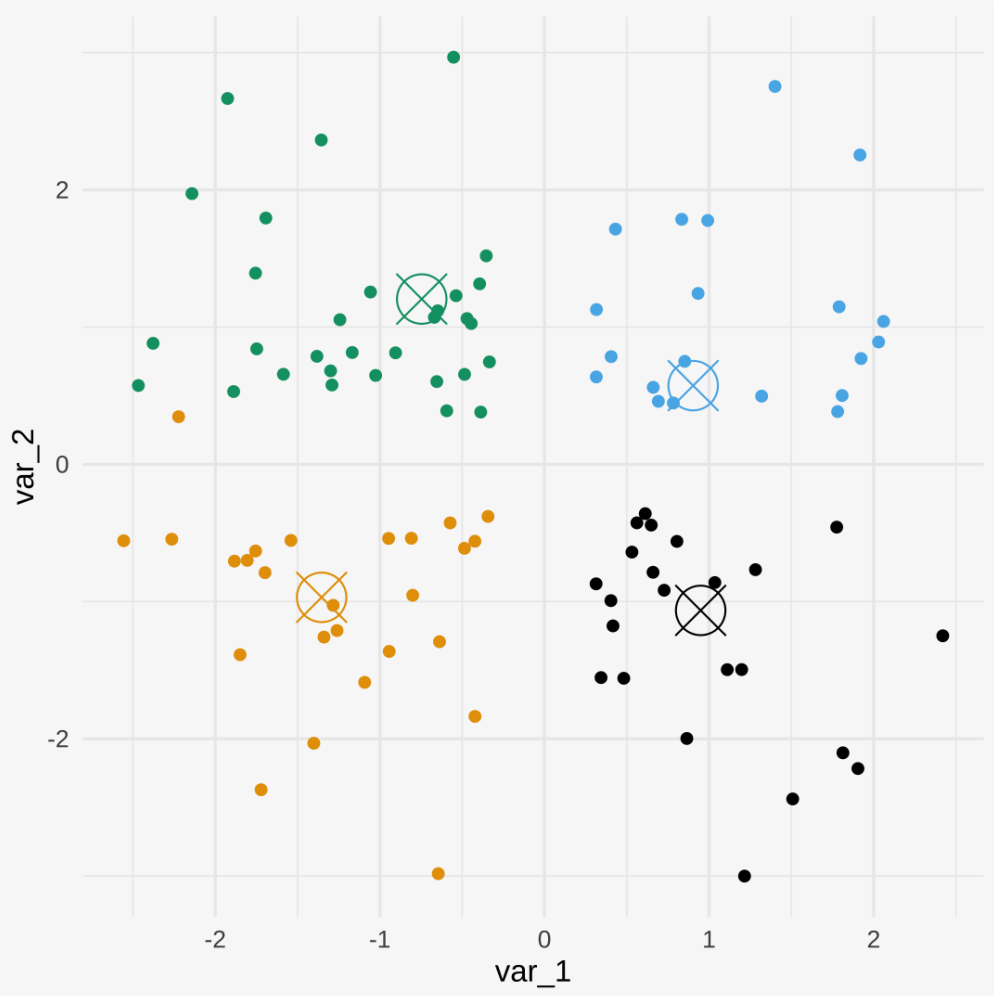
The k-means clustering algorithm: how does it work?

- The algorithm then calculates the mean values of all variables for all of the points assigned to each centroid. It then assigns a new position to the centroid at the mean position of all of the points.



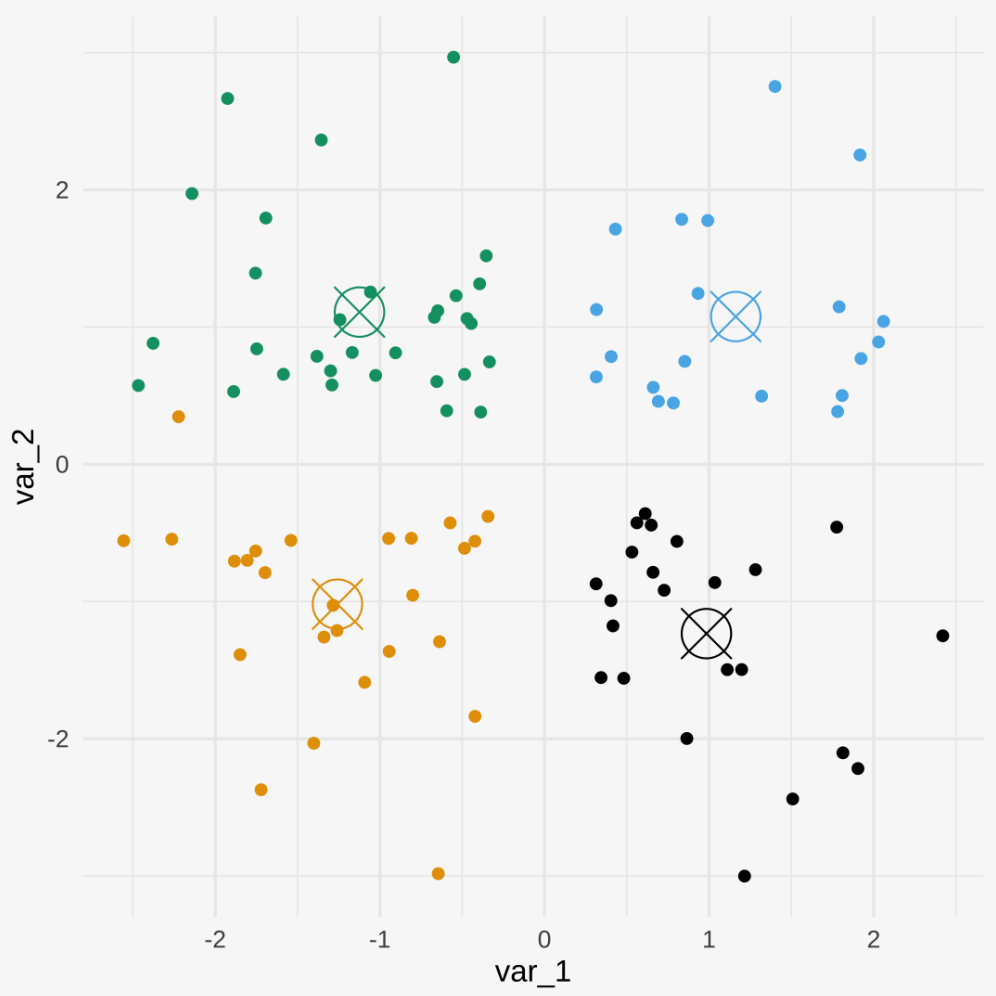
The k-means clustering algorithm: how does it work?

- The algorithm reassigns all of the points to their new closest centroid (if the closest centroid has changed).



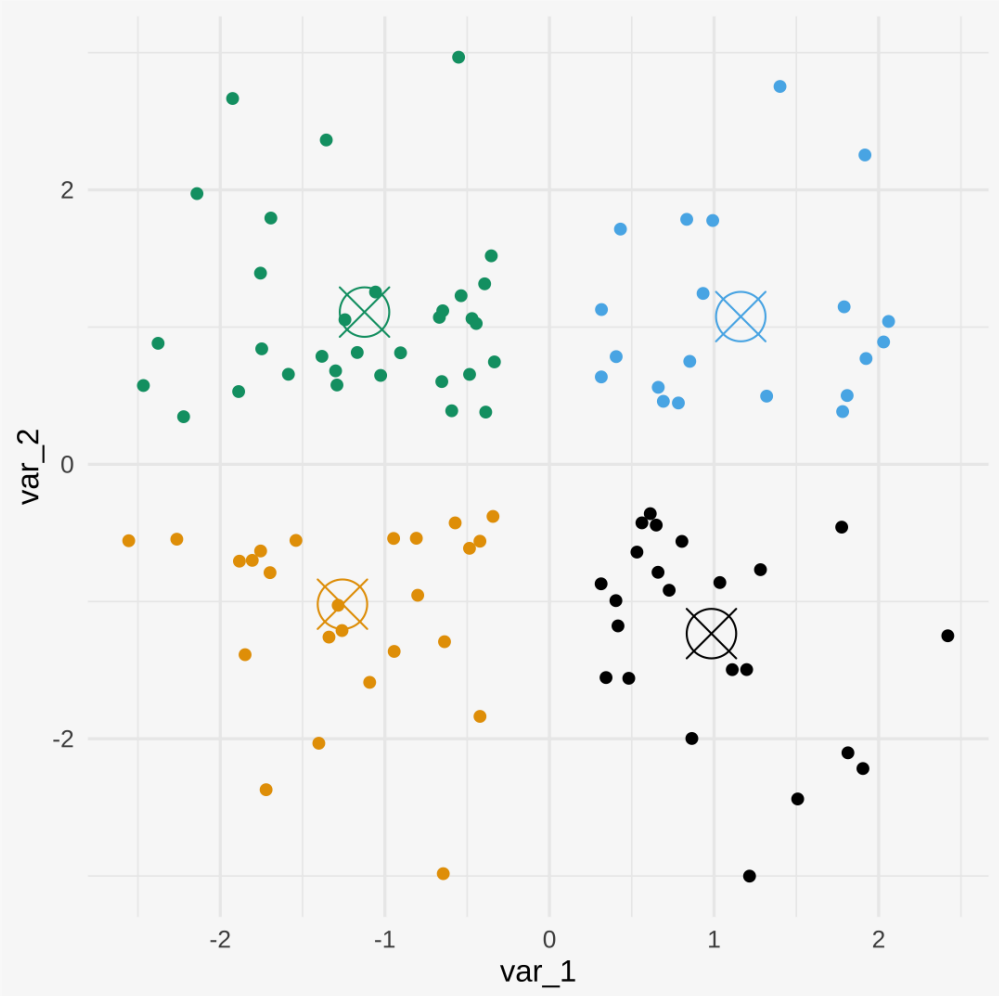
The k-means clustering algorithm: how does it work?

- The k-means algorithm then repeats the process: it calculates the mean values of all the points assigned to the centroids and moves their position according to this new value.



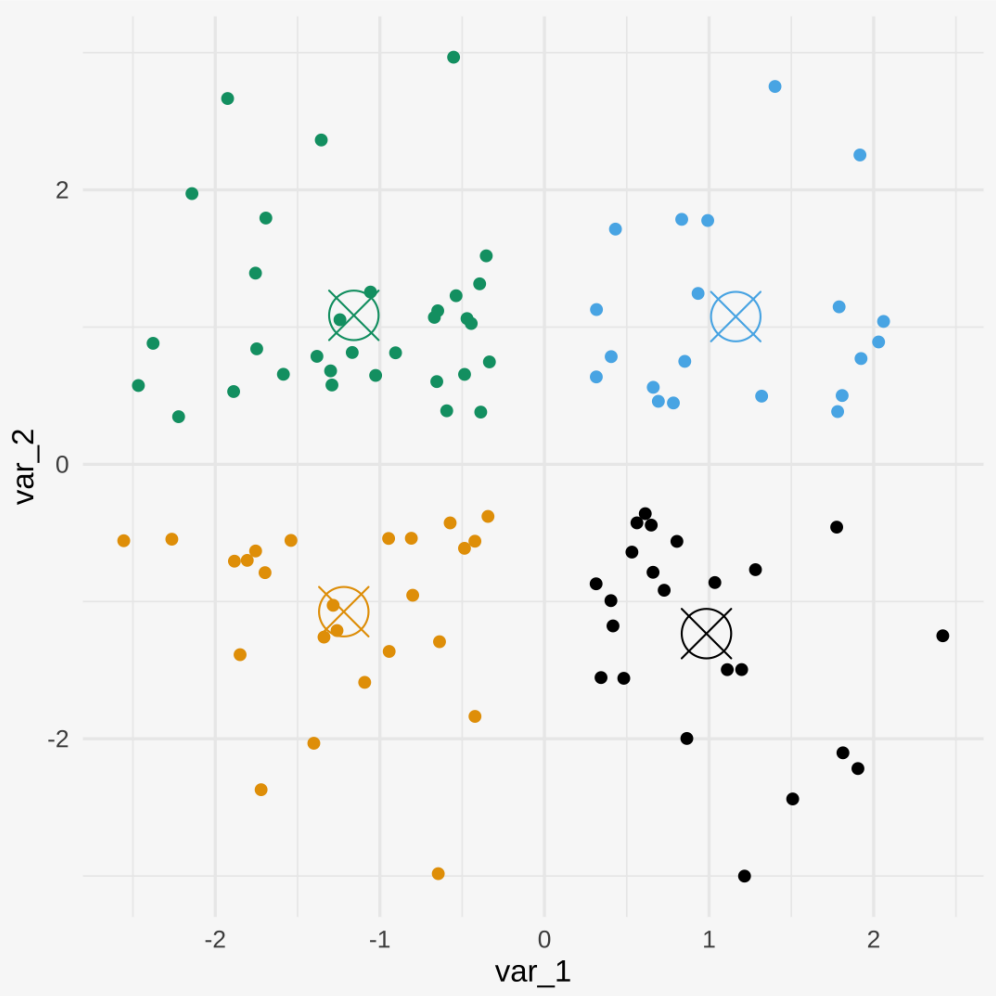
The k-means clustering algorithm: how does it work?

- It then reassigns the points again...



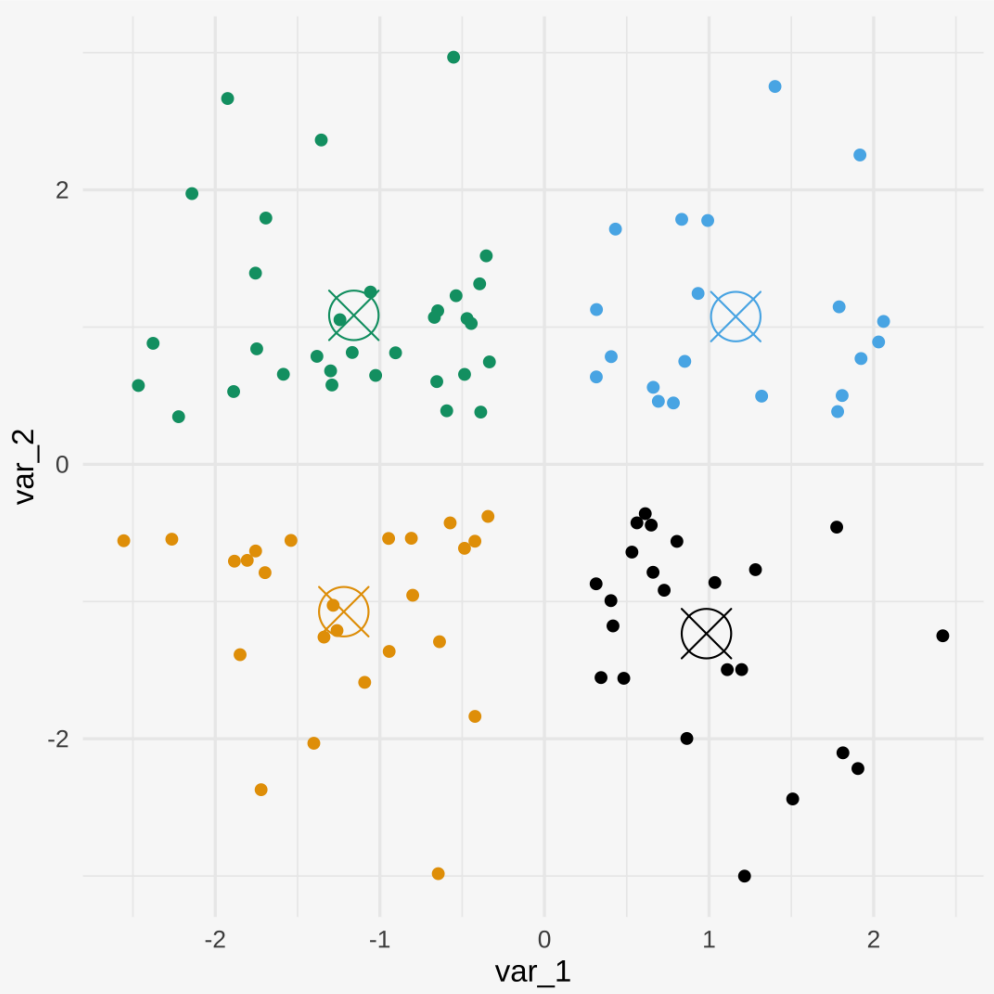
The k-means clustering algorithm: how does it work?

- And repeats the process of moving the centroids to the mean of their assigned points.



The k-means clustering algorithm: how does it work?

- It does this until the points are no longer being reassigned between centroids (**convergence**), or, more preceisely, until no movement of the centroids makes the sum of the distance between them and their assigned points any smaller.



Week 10: Cluster Analysis — Part III

k-means: What kinds of people visit museums?

k-means cluster analysis in R

Do patrons of museums fall into clearly defined clusters?

This (simulated) data is about 300 visitors to a museum. The data includes variables about number of adults the person visited with; the number of children they visited with; the donation they gave (in ££.pp); their age; the amount they spent at the gift shop; and the amount they spent at the cafe.

```
head(museum_data, 5)
```

```
## # A tibble: 5 × 7
##   museum              nadults nkids donation   age giftshop_spend cafe_spend
##   <chr>                 <dbl> <dbl>    <dbl> <dbl>          <dbl>      <dbl>
## 1 Western Park Museum       2     0       35    71           6.21       9.98
## 2 Western Park Museum       3     0       45    63           5.99      12.7
## 3 Western Park Museum       3     0       40    66           6.54      13.6
## 4 Western Park Museum       2     1       35    68           5.29      10.6
## 5 Western Park Museum       1     0       30    67           5.51      14.9
```

```
stargazer::stargazer(as.data.frame(museum_data), header = FALSE, type = "html")
```

Statistic	N	Mean	St. Dev.	Min	Max
nadults	300	2.617	1.203	1	5
nkids	300	0.953	1.409	0	6
donation	300	17.833	17.285	0.600	65.000
age	300	41.187	21.452	18	83
giftshop_spend	300	13.290	14.256	0.710	44.260
cafe_spend	300	6.313	5.015	1.400	21.060

k-means cluster analysis in R

- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- Use k-means to cluster our data.
- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- Describe the characteristics of each cluster.

We are going to use the `factoextra` and `cluster` packages to help us run our cluster analysis.

```
# install.packages("factoextra")
# install.packages("cluster")
library(factoextra)
library(cluster)
```

k-means cluster analysis in R

- **Prepare our data.**

k-means will only work with data that includes **only** numeric variables (and all of these variables should be continuous). Therefore, we need to start by creating a subset of our data of **just** the numeric variables we want to cluster on.

- Figure out the appropriate number of clusters for our data.
- Use k-means to cluster our data.
- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- Describe the characteristics of each cluster.

```
museum_data_prepped <- museum_data %>%
  select(nadults, nkids, donation,
         age, giftshop_spend, cafe_spend)

museum_data_prepped
```

```
## # A tibble: 300 × 6
##    nadults nkids donation   age giftshop_spend cafe_spend
##      <dbl> <dbl>    <dbl> <dbl>          <dbl>       <dbl>
##  1       2     0       35    71           6.21        9.98
##  2       3     0       45    63           5.99       12.7
##  3       3     0       40    66           6.54       13.6
##  4       2     1       35    68           5.29       10.6
##  5       1     0       30    67           5.51       14.9
##  6       3     0       40    69           4.46        7.36
##  7       1     0       40    65           4.33       12.6
##  8       2     0       15    72           5.68       14.5
##  9       5     0       35    70           5.19       13.5
## 10       4     0       45    67           3.61       12.2
## # ℹ 290 more rows
```

k-means cluster analysis in R

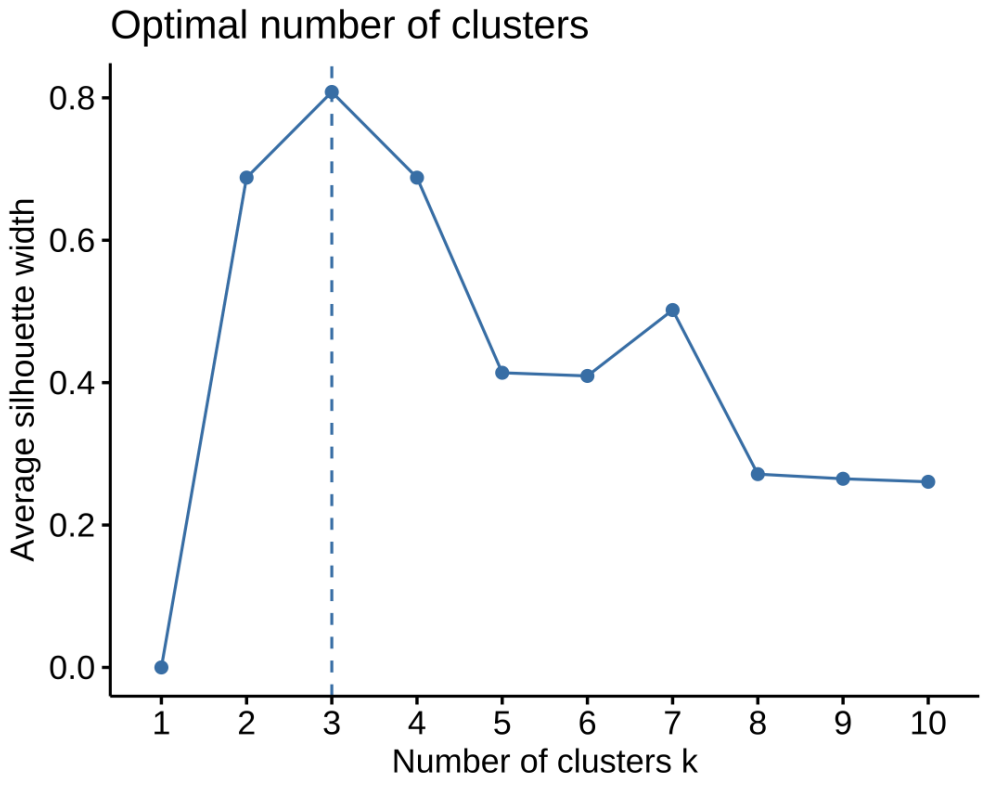
- Prepare our data.
- **Figure out the appropriate number of clusters for our data.**

Sometimes, we have a pre-defined hypothesis of how many clusters there are in our data, but often we do not. We can use a silhouette or gap statistic plot, which tells us **which number of clusters best maximises the clustering of our data**.

The highest silhouette value reflects the optimal number of clusters under this definition.

- Use k-means to cluster our data.
- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- Describe the characteristics of each cluster.

```
factoextra::fviz_nbclust(museum_data_prepped,
                         method = "silhouette",
                         FUNcluster = kmeans)
```



k-means cluster analysis in R

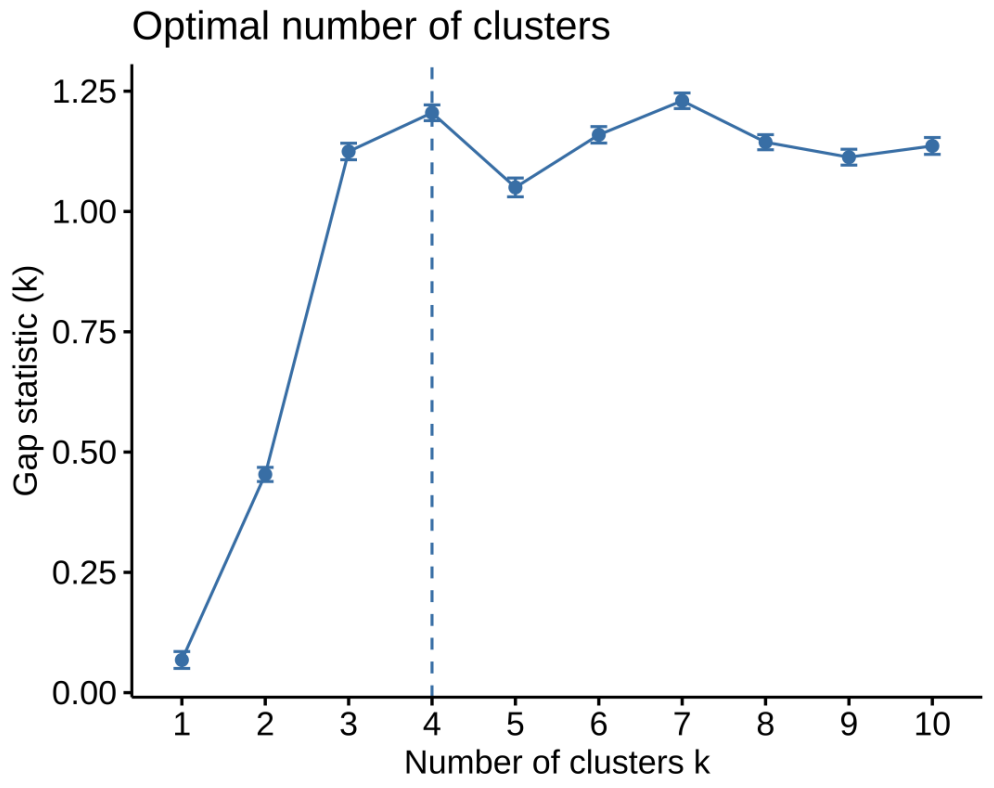
- Prepare our data.
- **Figure out the appropriate number of clusters for our data.**

Sometimes, we have a pre-defined hypothesis of how many clusters there are in our data, but often we do not. We can use a silhouette or gap statistic plot, which tells us **which number of clusters best maximises the clustering of our data**.

The highest gap statistic **above a margin of error** tells us the best solution under this condition.

- Use k-means to cluster our data.
- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- Describe the characteristics of each cluster.

```
factoextra::fviz_nbclust(museum_data_prepped,
                         method = "gap",
                         FUNcluster = kmeans, verbose = FALSE)
```



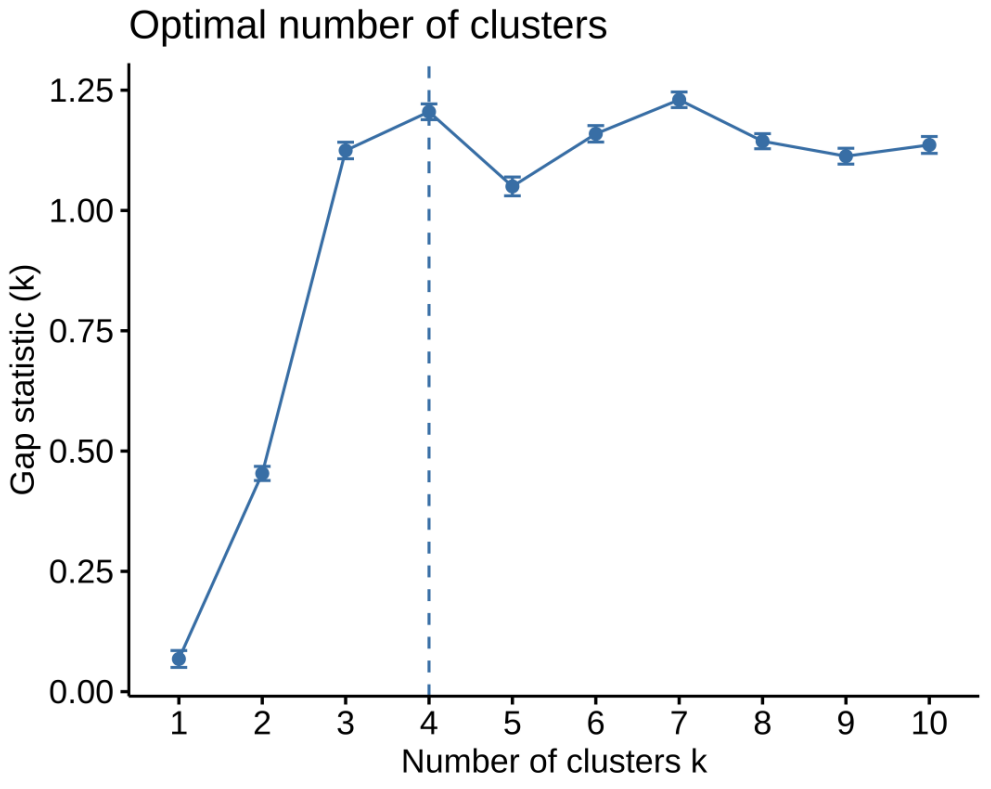
k-means cluster analysis in R

- Prepare our data.
- **Figure out the appropriate number of clusters for our data.**

Our silhouette and gap statistics seem to be indicating that either a **3** or **4** cluster solution would be a good fit to our data.

- Use k-means to cluster our data.
- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- Describe the characteristics of each cluster.

```
factoextra::fviz_nbclust(museum_data_prepped,
                         method = "gap",
                         FUNcluster = kmeans, verbose = FALSE)
```



k-means cluster analysis in R

- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- **Use k-means to cluster our data.**

Now we know an appropriate number of clusters, we can create both 3-cluster and 4-cluster solutions for our data using the `kmeans()` function. `centers = 3` is for 3 clusters, `centers = 4` for 4 clusters, etc.

```
set.seed(2021)
museum_3k <- kmeans(museum_data_prepped,
                    centers = 3)
set.seed(2021)
museum_4k <- kmeans(museum_data_prepped,
                    centers = 4)
```

- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- Describe the characteristics of each cluster.

k-means cluster analysis in R

- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- **Use k-means to cluster our data.**

Now we know an appropriate number of clusters, we can create both 3-cluster and 4-cluster solutions for our data using the `kmeans()` function. `centers = 3` is for 3 clusters, `centers = 4` for 4 clusters, etc.

If we inspect the k-means objects, we get the following information.

- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- Describe the characteristics of each cluster.

```
museum_3k
```

```
## K-means clustering with 3 clusters of sizes 100, 100, 100
##
## Cluster means:
##   nadults nkids donation   age giftshop_spend cafe_spend
## 1    1.88  2.61   10.320 33.56        33.0292     2.9782
## 2    3.38  0.05    3.028 19.97         1.9467     3.0658
## 3    2.59  0.20   40.150 70.03         4.8955    12.8940
##
## Clustering vector:
##   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [75] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [223] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [260] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [297] 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1]  3608.9634   370.0866 14963.0429
##  (between_SS / total_SS =  93.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

k-means cluster analysis in R

- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- **Use k-means to cluster our data.**

1. We see descriptive statistics for each of our clusters (we will revisit these by adding them to our data)
2. We see the cluster assigned to each observation
3. We see the proportion of variance that can be explained by cluster membership (between_SS/total_SS) (93.6%)

- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- Describe the characteristics of each cluster.

```
museum_3k
```

```
## K-means clustering with 3 clusters of sizes 100, 100, 100
##
## Cluster means:
##   nadults nkids donation   age giftshop_spend cafe_spend
## 1    1.88  2.61   10.320 33.56        33.0292     2.9782
## 2    3.38  0.05    3.028 19.97         1.9467     3.0658
## 3    2.59  0.20   40.150 70.03         4.8955    12.8940
##
## Clustering vector:
##   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [75] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [223] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [260] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [297] 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1]  3608.9634   370.0866 14963.0429
##  (between_SS / total_SS =  93.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

k-means cluster analysis in R

- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- **Use k-means to cluster our data.**

We can call specific parts of the k-means object using the `$` operator.

- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- Describe the characteristics of each cluster.

```
museum_3k$cluster
```

```
##   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [75] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [223] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [260] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [297] 2 2 2 2
```

```
museum_3k$centers
```

```
##   nadults nkids donation   age giftshop_spend cafe_spend
## 1    1.88  2.61   10.320 33.56        33.0292     2.9782
## 2    3.38  0.05    3.028 19.97         1.9467     3.0658
## 3    2.59  0.20   40.150 70.03         4.8955    12.8940
```

k-means cluster analysis in R

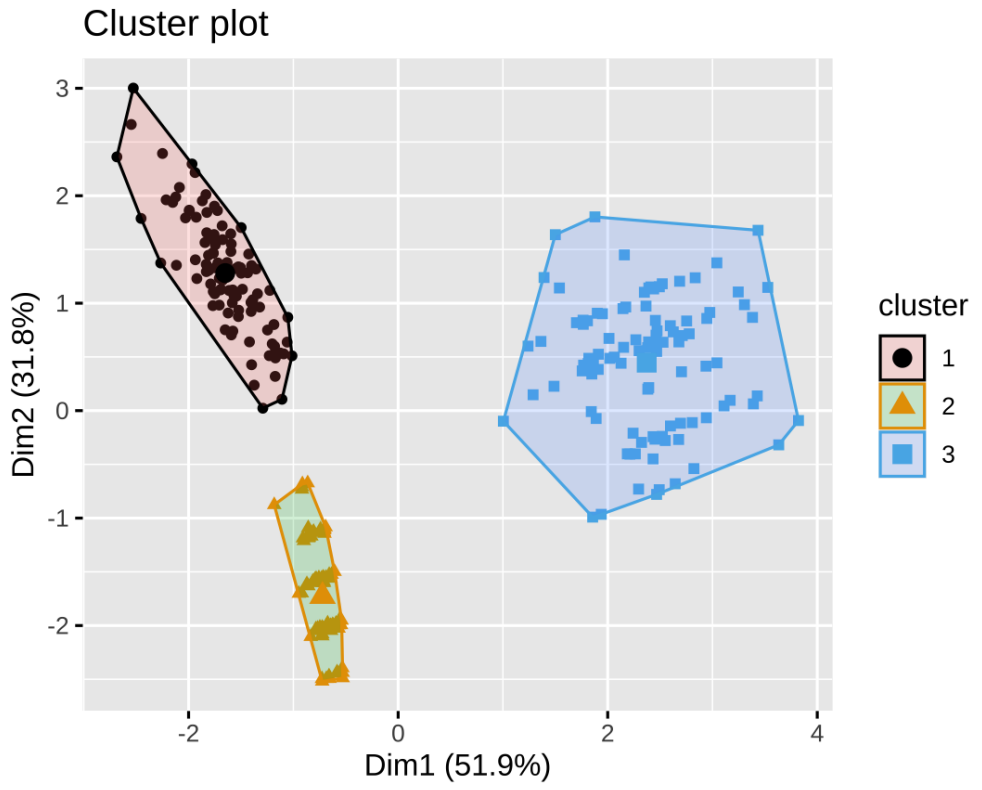
- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- Use k-means to cluster our data.
- **Visualise the clusters (and their uniqueness).**

When we have two or more possible solutions to the number of clusters it can be helpful to visualise them with a **biplot**.

A biplot uses something called Principal Components Analysis to simplify multiple dimensions of data (multiple variables) down to two dimensions. We can then plot the observations and their cluster membership using `factoextra::fviz_cluster`.

- Add the cluster membership to our data.
- Describe the characteristics of each cluster.

```
fviz_cluster(museum_3k, data = museum_data_prepped,
             geom = "point") +
  ggthemes::scale_color_colorblind()
```



k-means cluster analysis in `R`

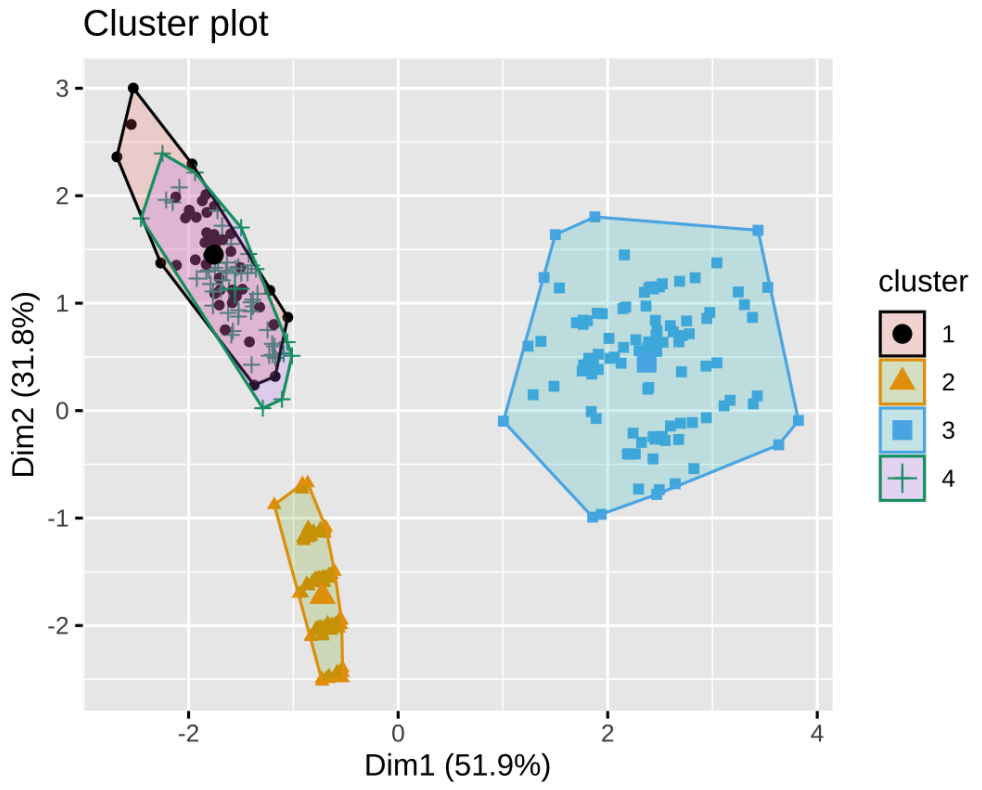
- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- Use k-means to cluster our data.
- **Visualise the clusters (and their uniqueness).**

This creates a ggplot type plot that we can customise, e.g. with colourblind friendly scales from the `ggthemes` package.

Does the four cluster solution look much better than the three cluster solution?

- Add the cluster membership to our data.
- Describe the characteristics of each cluster.

```
fviz_cluster(museum_4k, data = museum_data_prepped,
             geom = "point") +
  ggthemes::scale_color_colorblind()
```



k-means cluster analysis in R

- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- Use k-means to cluster our data.
- Visualise the clusters (and their uniqueness).
- **Add the cluster membership to our data.**

We can add the cluster membership to our data easily using the object from kmeans and the `dplyr` package from `tidyverse`.

- Describe the characteristics of each cluster.

```
museum_data <- museum_data %>%
  mutate(
    cluster = museum_3k$cluster
  )

museum_data
```

```
## # A tibble: 300 × 8
##    museum          nadults nkids donation   age giftshop_spend cafe_spend cluster
##    <chr>             <dbl> <dbl>    <dbl> <dbl>          <dbl>      <dbl>   <int>
##  1 Western Park …        2     0       35    71           6.21       9.98       3
##  2 Western Park …        3     0       45    63           5.99      12.7        3
##  3 Western Park …        3     0       40    66           6.54      13.6        3
##  4 Western Park …        2     1       35    68           5.29      10.6        3
##  5 Western Park …        1     0       30    67           5.51      14.9        3
##  6 Western Park …        3     0       40    69           4.46       7.36       3
##  7 Western Park …        1     0       40    65           4.33      12.6        3
##  8 Western Park …        2     0       15    72           5.68      14.5        3
##  9 Western Park …        5     0       35    70           5.19      13.5        3
## 10 Western Park …        4     0       45    67           3.61      12.2        3
## # ℹ 290 more rows
```

k-means cluster analysis in R

- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- Use k-means to cluster our data.
- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- **Describe the characteristics of each cluster.**

We can then carry out any bivariate statistics, tests, or other analyses with both variables that were in the cluster process and those that were not to better understand out clusters.

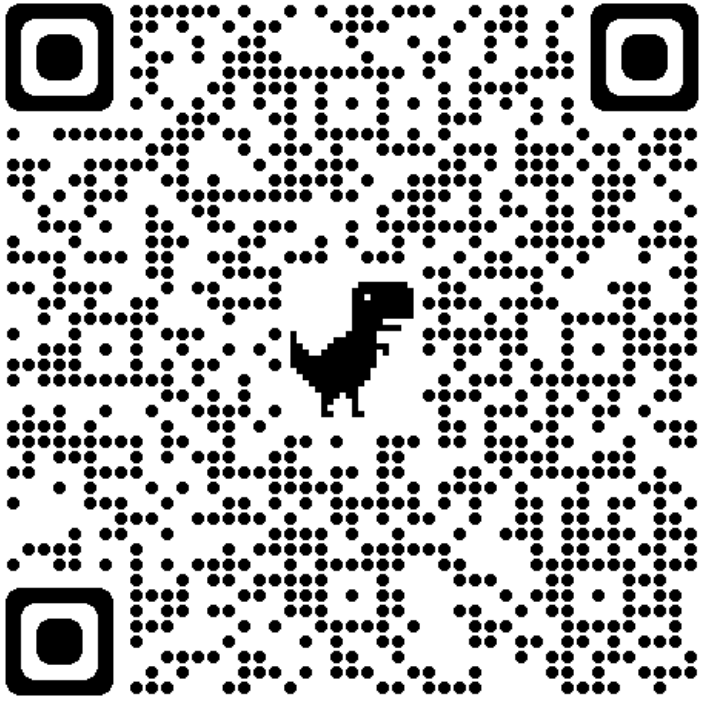
```
museum_data %>%
  group_by(cluster) %>%
  summarise(
    m_nadults = mean(nadults),
    m_nkids = mean(nkids),
    m_donation = mean(donation),
    m_age = mean(age),
    m_giftshop = mean(giftshop_spend),
    m_cafe = mean(cafe_spend)
  )
```

```
## # A tibble: 3 × 7
##   cluster m_nadults m_nkids m_donation m_age m_giftshop m_cafe
##     <int>     <dbl>   <dbl>      <dbl> <dbl>      <dbl>  <dbl>
## 1       1      1.88    2.61       10.3   33.6       33.0   2.98
## 2       2      3.38    0.05        3.03  20.0        1.95   3.07
## 3       3      2.59    0.2        40.2   70.0        4.90  12.9
```

k-means cluster analysis in R

- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- Use k-means to cluster our data.
- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- **Describe the characteristics of each cluster.**

Join the Jamboard (Week 10: Jamboard 1) using this link and try interpreting the clusters produced by k-means, based on their mean values.



k-means cluster analysis in R

- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- Use k-means to cluster our data.
- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- **Describe the characteristics of each cluster.**

e.g. 1 = Around 2 adults, large number of children, average donation, age 30s, high gift shop spend, low cafe spend... **Families with kids bringing packed lunches and buying mementos**?

```
museum_data %>%
  group_by(cluster) %>%
  summarise(
    m_nadults = mean(nadults),
    m_nkids = mean(nkids),
    m_donation = mean(donation),
    m_age = mean(age),
    m_giftshop = mean(giftshop_spend),
    m_cafe = mean(cafe_spend)
  )
```

```
## # A tibble: 3 × 7
##   cluster m_nadults m_nkids m_donation m_age m_giftshop m_cafe
##     <int>     <dbl>   <dbl>      <dbl> <dbl>      <dbl>  <dbl>
## 1       1      1.88    2.61       10.3  33.6       33.0   2.98
## 2       2      3.38    0.05        3.03 20.0        1.95  3.07
## 3       3      2.59    0.2        40.2  70.0        4.90 12.9
```

k-means cluster analysis in R

- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- Use k-means to cluster our data.
- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- **Describe the characteristics of each cluster.**

e.g. 2 = Larger number of adults, very few if ever kids, low donations, overall quite young (around 20), low giftshop spend, low cafe spend... **Groups of students killing time, on a field trip, or stopping for a coffee**?

```
museum_data %>%
  group_by(cluster) %>%
  summarise(
    m_nadults = mean(nadults),
    m_nkids = mean(nkids),
    m_donation = mean(donation),
    m_age = mean(age),
    m_giftshop = mean(giftshop_spend),
    m_cafe = mean(cafe_spend)
  )
```

```
## # A tibble: 3 × 7
##   cluster m_nadults m_nkids m_donation m_age m_giftshop m_cafe
##     <int>     <dbl>   <dbl>      <dbl> <dbl>      <dbl>  <dbl>
## 1       1      1.88    2.61       10.3  33.6       33.0   2.98
## 2       2      3.38    0.05        3.03 20.0        1.95  3.07
## 3       3      2.59    0.2        40.2  70.0        4.90 12.9
```

k-means cluster analysis in R

- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- Use k-means to cluster our data.
- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- **Describe the characteristics of each cluster.**

e.g. 3 = Often 2, sometimes 3 adults; sometimes children, but not often; average very high donation; very high average age (around 70 years); modest giftshop spend, and high cafe spend... **retired patrons of the museum who like to have a full lunch at the cafe**?

```
museum_data %>%
  group_by(cluster) %>%
  summarise(
    m_nadults = mean(nadults),
    m_nkids = mean(nkids),
    m_donation = mean(donation),
    m_age = mean(age),
    m_giftshop = mean(giftshop_spend),
    m_cafe = mean(cafe_spend)
  )
```

```
## # A tibble: 3 × 7
##   cluster m_nadults m_nkids m_donation m_age m_giftshop m_cafe
##     <int>     <dbl>   <dbl>      <dbl> <dbl>      <dbl>  <dbl>
## 1       1      1.88    2.61       10.3  33.6       33.0   2.98
## 2       2      3.38    0.05        3.03 20.0        1.95  3.07
## 3       3      2.59    0.2        40.2  70.0        4.90 12.9
```

k-means cluster analysis in R

- Prepare our data.
- Figure out the appropriate number of clusters for our data.
- Use k-means to cluster our data.
- Visualise the clusters (and their uniqueness).
- Add the cluster membership to our data.
- **Describe the characteristics of each cluster.**

It's quite common to give either straightforward or amusing names to the clusters, based on their characteristics. You will often see this in Facebook/Spotify/etc. website metadata as a way to target ads!

```
museum_data %>%
  group_by(cluster) %>%
  summarise(
    m_nadults = mean(nadults),
    m_nkids = mean(nkids),
    m_donation = mean(donation),
    m_age = mean(age),
    m_giftshop = mean(giftshop_spend),
    m_cafe = mean(cafe_spend)
  ) %>%
  mutate(
    cluster = case_when(cluster == 1 ~ "Young Families",
                        cluster == 2 ~ "Student Groups",
                        cluster == 3 ~ "High-rolling Oldies")
  )
```

```
## # A tibble: 3 × 7
##   cluster             m_nadults m_nkids m_donation m_age m_giftshop m_cafe
##   <chr>                   <dbl>   <dbl>      <dbl> <dbl>      <dbl>  <dbl>
## 1 Young Families           1.88    2.61       10.3  33.6       33.0   2.98
## 2 Student Groups           3.38    0.05        3.03 20.0        1.95  3.07
## 3 High-rolling Oldies      2.59    0.2        40.2  70.0        4.90 12.9
```

Week 10: Cluster Analysis — Part IV

k-means: Assumptions.

k-means assumptions

k-means does not quite have the same kinds of 'assumptions' as statistical models like regression, t-tests, etc., but it does make some assumptions about the data and it does have some requirements.

Data must be continuous

It is not recommended to use k-means with binary or other forms of data, though pseudo-continuous variables (e.g. likert scales) may be fine.

k-means assumptions

k-means does not quite have the same kinds of 'assumptions' as statistical models like regression, t-tests, etc., but it does make some assumptions about the data and it does have some requirements.

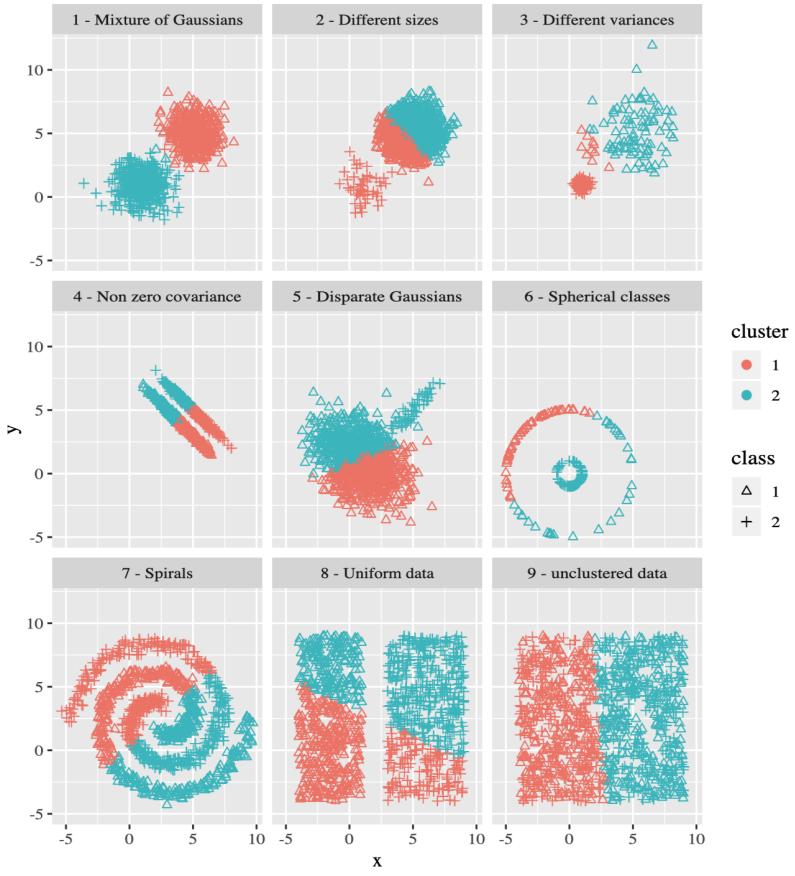
Data must be continuous

It is not recommended to use k-means with binary or other forms of data, though pseudo-continuous variables (e.g. likert scales) may be fine.

Clusters must be (approximately) spherical, of similar size, and with similar variance

k-means also assumes that the best fitting clusters to your data are approximately spherical (e.g. not oblong, or any other weird shape). This can be checked using your biplot (to see where misclassifications may be occuring). There are some methods designed to handle non-spherical clustering.

k-means assumptions



Clusters must be (approximately) spherical, of similar size, and with similar variance

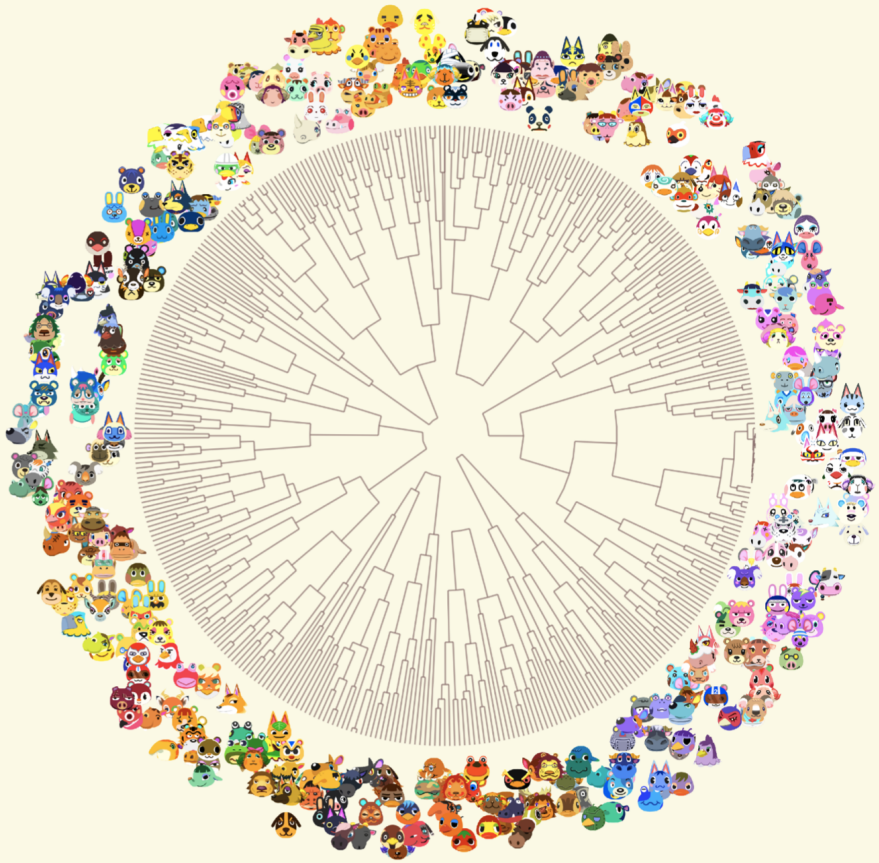
k-means also assumes that the best fitting clusters to your data are approximately spherical (e.g. not oblong, or any other weird shape). This can be checked using your biplot (to see where misclassifications may be occuring). There are some methods designed to handle non-spherical clustering.

Source: Morbieu, 2018

Visualising the Villagers

datavis meets Animal Crossing

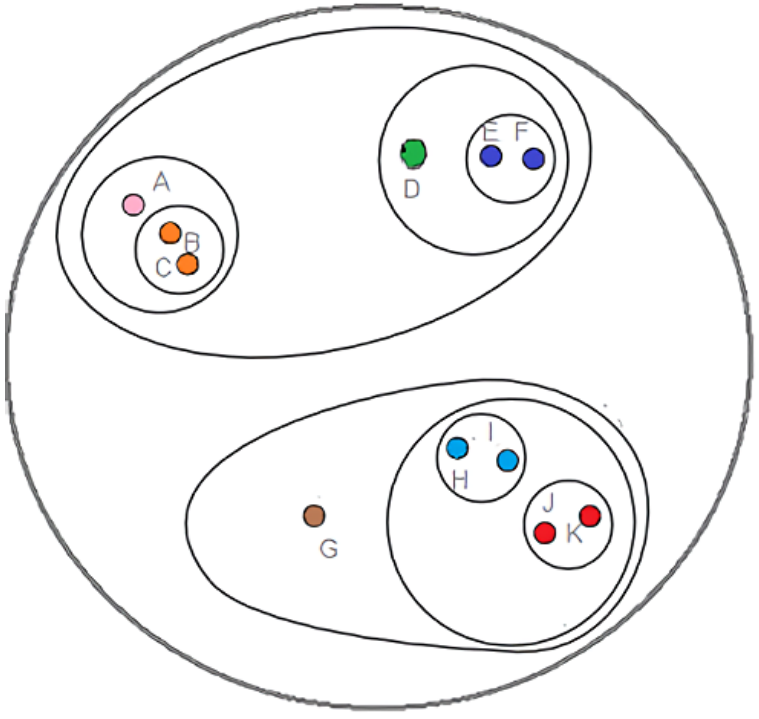
Sources: Animal Crossing Icons (Nookipedia), species, and personality data (animalcrossing.fandom), Villager Popularity Rankings compiled by Pandoria, Mairen, & bloobelle from the Bell Tree Forums (28th March 2020)

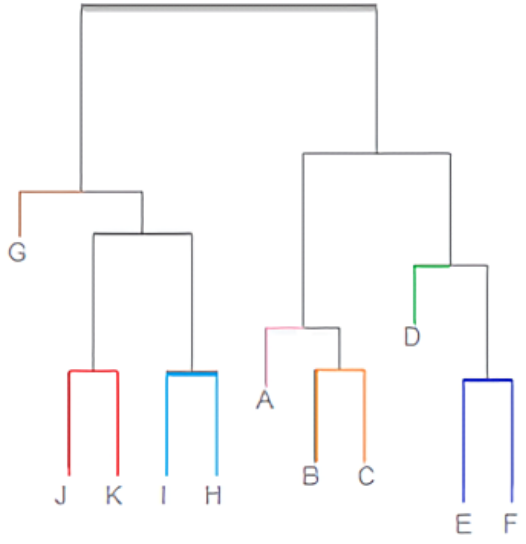


Week 10: Cluster Analysis — Part V

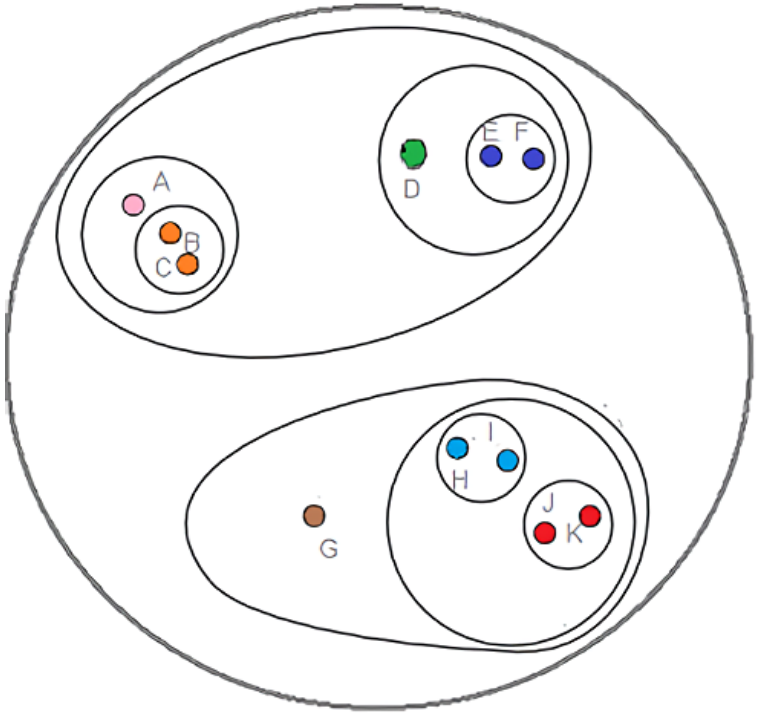
Hierarchical Cluster Analysis: How does it work?

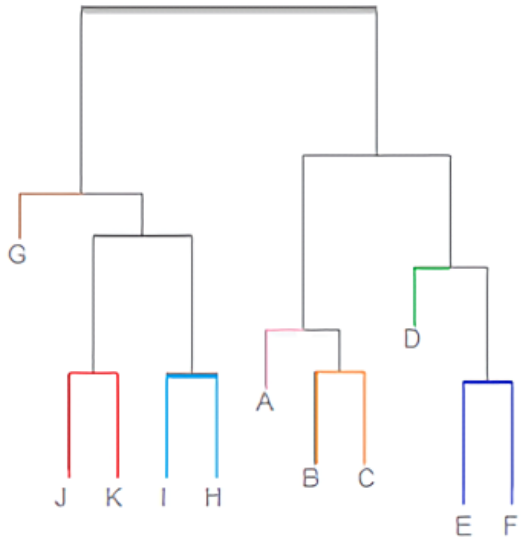
Hierarchical Cluster Analysis





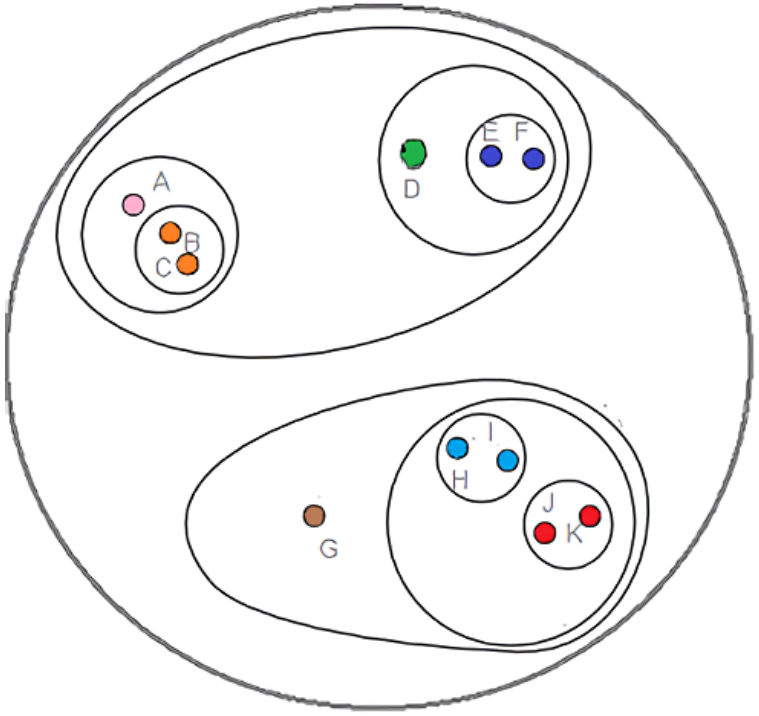
Hierarchical Cluster Analysis

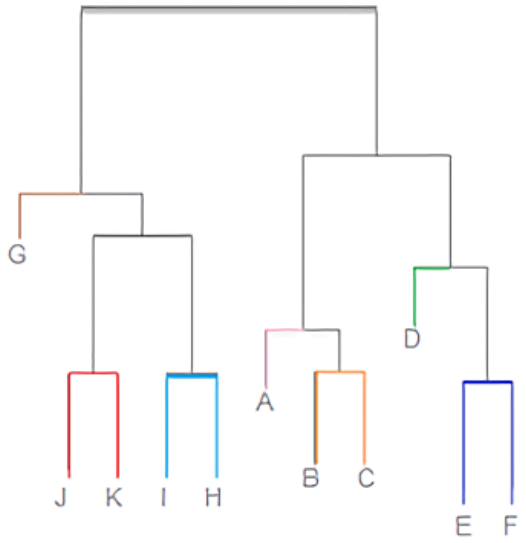




- **Calculate an appropriate distance (dissimilarity) measure between all points** in a matrix.

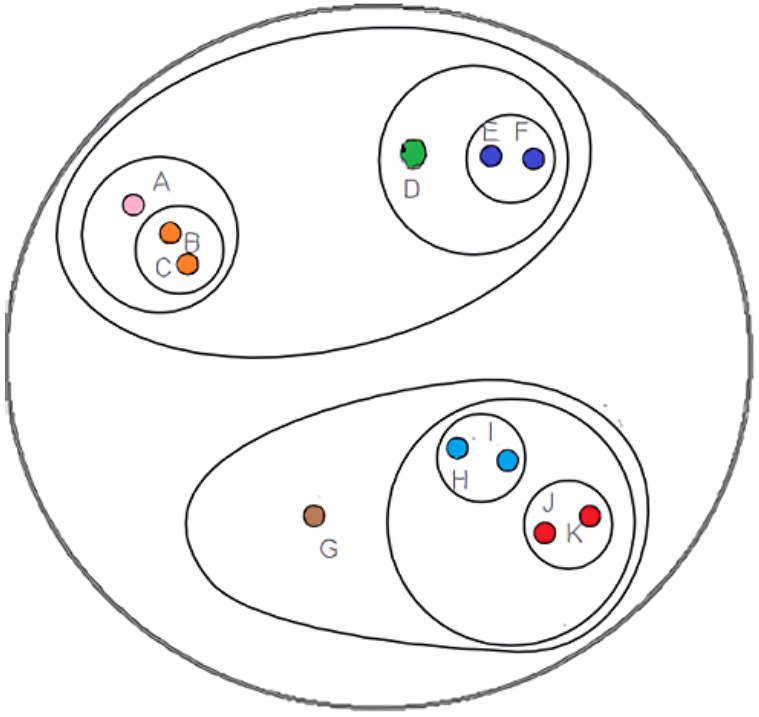
Hierarchical Cluster Analysis

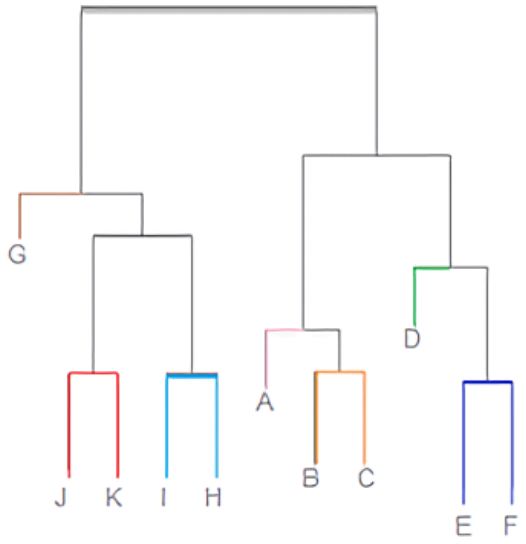




- **Calculate an appropriate distance (dissimilarity) measure between all points** in a matrix.
- Use one of several algorithms to **agglomoratively** or **divisively** link all points, depending on the theoretical type of clusters.

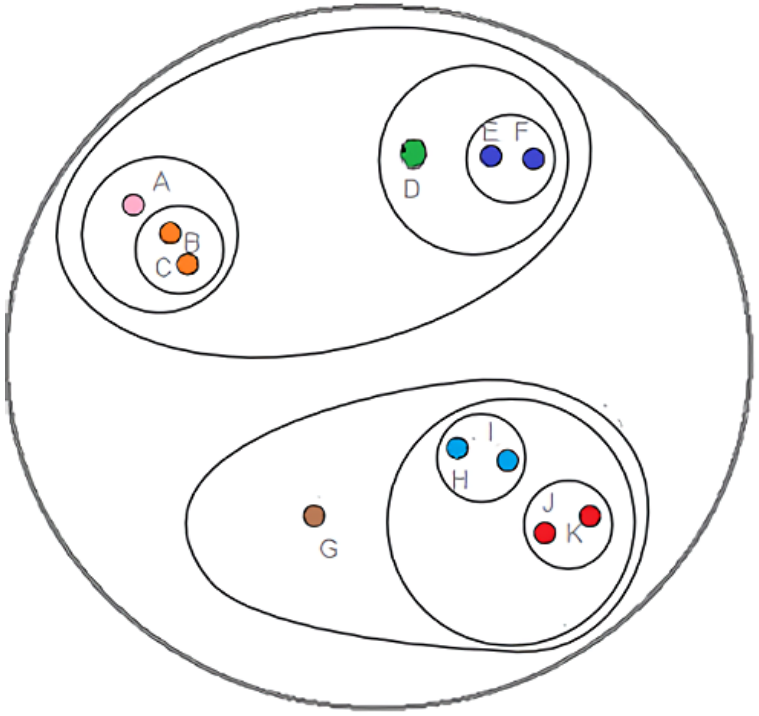
Hierarchical Cluster Analysis

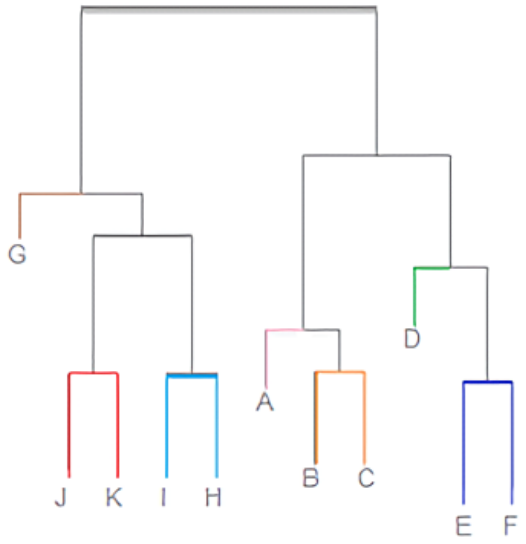




- **Calculate an appropriate distance (dissimilarity) measure between all points** in a matrix.
- Use one of several algorithms to **agglomoratively** or **divisively** link all points, depending on the theoretical type of clusters.
- Visualise result in a **dendrogram** and decide on (one or more) solutions.

Hierarchical Cluster Analysis





- **Calculate an appropriate distance (dissimilarity) measure between all points** in a matrix.
- Use one of several algorithms to **agglomoratively** or **divisively** link all points, depending on the theoretical type of clusters.
- Visualise result in a **dendrogram** and decide on (one or more) solutions.
- **"Cut" tree** to assign observations to clusters, and then describe clusters with bivariate statistics.

Hierarchical Cluster Analysis

Pros and Cons compared to k-means

Pros

- Highly flexible — range of methods means that non-spherical, unusual, types of clustering (e.g. single point of power causing a 'chain' effect; identifying a common "type"; 'spheres of influence' that diffuse gradually; 'tight', highly defined clusters; or general unity). Often better defines strangely shaped clusters.

Cons

- High degree of subjectivity in choice of hierarchical clustering algorithm, as well as choice of distance matrix and final number of clusters. No straightforward statistics for deciding on optimal number of clusters.
- Can be harder to work with, visualise nicely, etc, without more programming knowledge.

Week 10: Cluster Analysis — Part VI

Hierarchical Cluster Analysis: What groups do Youtube viewers fall into?

HCA: Example

Can we find a general typology of Youtube viewers?

We collected a random sample of videos watched on 200 Youtube users and then recorded which types of videos they had watched in the random sample of 10 that was collected for each viewer. If the person watched a video of the genre listed, a 1 was recorded. If they did not watch a video of that genre, a 0 was recorded.

```
youtube_data
```

```
## # A tibble: 200 × 12
##    animation comedy documentaries music news  reviews sport tutorials trailers
##    <fct>     <fct>  <fct>         <fct> <fct> <fct>   <fct> <fct>     <fct>
##  1 0         1      0             0     0     0       0     0         0
##  2 1         0      0             0     0     1       0     0         1
##  3 1         0      0             1     0     0       0     1         1
##  4 1         0      0             1     0     1       0     0         0
##  5 1         0      0             1     0     0       0     1         1
##  6 1         0      0             1     0     0       0     0         1
##  7 1         0      0             1     0     1       0     1         1
##  8 1         1      0             1     0     0       0     1         0
##  9 1         0      1             1     0     1       0     0         0
## 10 0         1      0             1     0     0       0     0         1
## # ℹ 190 more rows
## # ℹ 3 more variables: videoessays <fct>, videogames <fct>, vlogs <fct>
```

HCA: Example

Step 1: Select the most appropriate dissimilarity/distance calculation.

- **Euclidean** distance: Distance calculated as a straight line ("as the bird flies"). Most appropriate form of distance when all variables are continuous.
- **Manhattan** distance: Distance calculated as steps of travel (like navigating a city via blocks). Most appropriate form when we have a large number of continuous variables and/or continuous variables with very different scales, variance, or mixtures of ratio, discrete, and interval.
- **Gower** distance: Uses a range of distance measures depending on variable type. Most appropriate when you have all categorical binary, ordinal, or mixtures of categorical, binary, and continuous data.

HCA: Example

Step 1: Select the most appropriate dissimilarity/distance calculation.

- **Euclidean** distance: Distance calculated as a straight line ("as the bird flies"). Most appropriate form of distance when all variables are continuous.
- **Manhattan** distance: Distance calculated as steps of travel (like navigating a city via blocks). Most appropriate form when we have a large number of continuous variables and/or continuous variables with very different scales, variance, or mixtures of ratio, discrete, and interval.
- **Gower** distance: Uses a range of distance measures depending on variable type. Most appropriate when you have all categorical binary, ordinal, or mixtures of categorical, binary, and continuous data.

We can calculate a distance/dissimilarity matrix (distance between all points) using the `daisy()` function from the `cluster` package.

Binary continuous variables **must** be coded as factors for `daisy` to calculate Gower's distance correctly.

```
library(cluster)

# Calculate a distance matrix and save it as youtube_d
youtube_d <- daisy(youtube_data, metric = "gower")
```

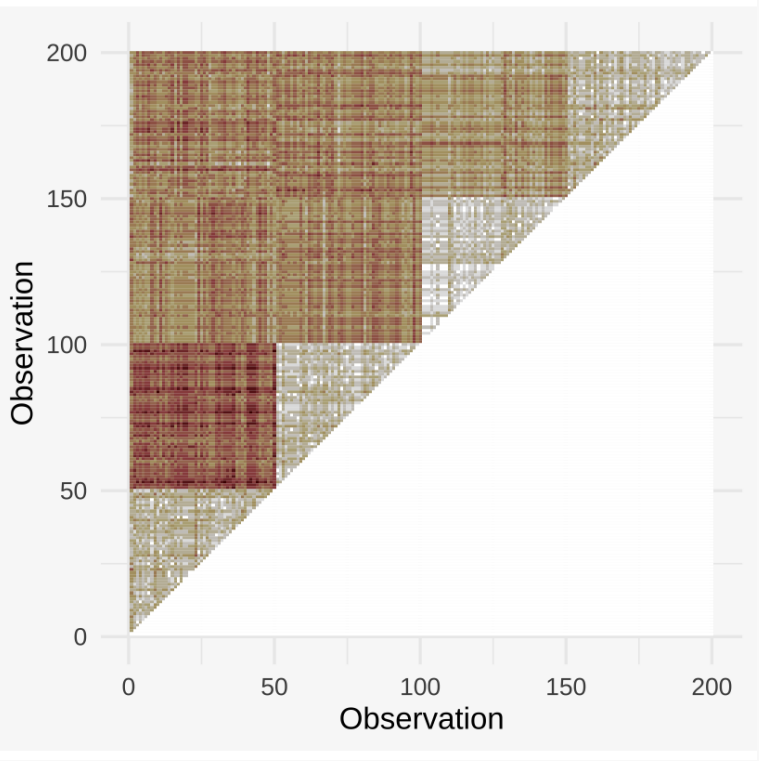
HCA: Example

Step 1: Select the most appropriate dissimilarity/distance calculation.

- **Euclidean** distance: Distance calculated as a straight line ("as the bird flies"). Most appropriate form of distance when all variables are continuous.
- **Manhattan** distance: Distance calculated as steps of travel (like navigating a city via blocks). Most appropriate form when we have a large number of continuous variables and/or continuous variables with very different scales, variance, or mixtures of ratio, discrete, and interval.
- **Gower** distance: Uses a range of distance measures depending on variable type. Most appropriate when you have all categorical binary, ordinal, or mixtures of categorical, binary, and continuous data.

Visualisation of Dissimilarity Matrix

You don't need to do this, I'm just illustrating what `daisy` is calculating! Darker colours = more dissimilar observations.



HCA: Example

Step 2: Decide on an appropriate clustering method

- **Ward**: Identifies underlying defined 'types' (e.g. species). Should only be used with Euclidean distance.
- **single**: Identifies underlying hierarchical 'chains' — strictly most similar to least.
- **complete**: Identifies underlying 'circles' (as in: runs in the same circles). Discriminates between dissimilar overall clusters as well as individuals.
- **centroid**: Identifies clusters based on the there being a clear center with dispersion in the membership. Comparable to k-means. Should only be used with Euclidean distance.
- **median**: Same as centroid but less sensitive to outliers. Should only be used with Euclidean distance.
- **average**: Like complete but tends towards identifying clusters of approximately equal size, defined very generically (i.e. not a chain of command, or a clear center). Based on average dissimilarity of each pair/group from all other pairs/groups as clusters are formed.

General rule of thumb: **average** and **complete** linkage is a good choice for any underlying theory and data.

HCA: Example

Step 2: Decide on an appropriate clustering method

- **Ward**: Identifies underlying defined 'types' (e.g. species). Should only be used with Euclidean distance.
- **single**: Identifies underlying hierarchical 'chains' — strictly most similar to least.
- **complete**: Identifies underlying 'circles' (as in: runs in the same circles). Discriminates between dissimilar overall clusters as well as individuals.
- **centroid**: Identifies clusters based on the there being a clear center with dispersion in the membership. Comparable to k-means. Should only be used with Euclidean distance.
- **median**: Same as centroid but less sensitive to outliers. Should only be used with Euclidean distance.
- **average**: Like complete but tends towards identifying clusters of approximately equal size, defined very generically (i.e. not a chain of command, or a clear center). Based on average dissimilarity of each pair/group from all other pairs/groups as clusters are formed.

General rule of thumb: **average** and **complete** linkage is a good choice for any underlying theory and data.

- Not going to use any that require Euclidean distance

HCA: Example

Step 2: Decide on an appropriate clustering method

- **Ward**: Identifies underlying defined 'types' (e.g. species). Should only be used with Euclidean distance.
- **single**: Identifies underlying hierarchical 'chains' — strictly most similar to least.
- **complete**: Identifies underlying 'circles' (as in: runs in the same circles). Discriminates between dissimilar overall clusters as well as individuals.
- **centroid**: Identifies clusters based on the there being a clear center with dispersion in the membership. Comparable to k-means. Should only be used with Euclidean distance.
- **median**: Same as centroid but less sensitive to outliers. Should only be used with Euclidean distance.
- **average**: Like complete but tends towards identifying clusters of approximately equal size, defined very generically (i.e. not a chain of command, or a clear center). Based on average dissimilarity of each pair/group from all other pairs/groups as clusters are formed.

General rule of thumb: **average** and **complete** linkage is a good choice for any underlying theory and data.

- Not going to use any that require Euclidean distance
- Not going to use single as it doesn't fit underlying theory (no highly weighted individual youtube viewer)

HCA: Example

Step 3: Cluster data using `hclust`

- Cluster data based on **complete linkage**

```
set.seed(2021)
hca_comp <- hclust(d = youtube_d,
                   method = "complete")
```

- Cluster data based on **average linkage**

```
set.seed(2021)
hca_avg <- hclust(d = youtube_d,
                  method = "average")
```

HCA: Example

Step 3: Cluster data using `hclust`

- Cluster data based on **complete linkage**

```
set.seed(2021)
hca_comp <- hclust(d = youtube_d,
                   method = "complete")
```

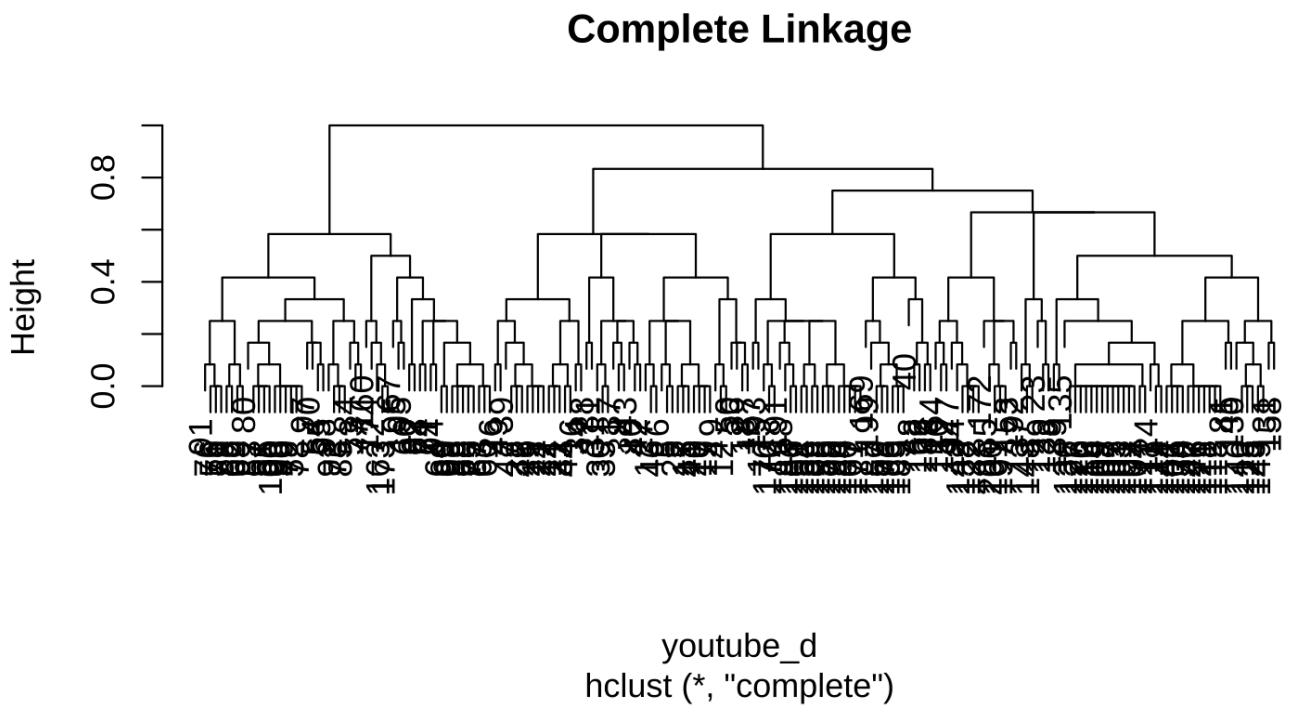
- Cluster data based on **average linkage**

```
set.seed(2021)
hca_avg <- hclust(d = youtube_d,
                  method = "average")
```

Complete: Maybe 4, maybe 6 clusters?

Step 4: Visualise HCAs using dendrograms

```
plot(hca_comp, main = "Complete Linkage")
```



HCA: Example

Step 3: Cluster data using `hclust`

- Cluster data based on **complete linkage**

```
set.seed(2021)
hca_comp <- hclust(d = youtube_d,
                   method = "complete")
```

- Cluster data based on **average linkage**

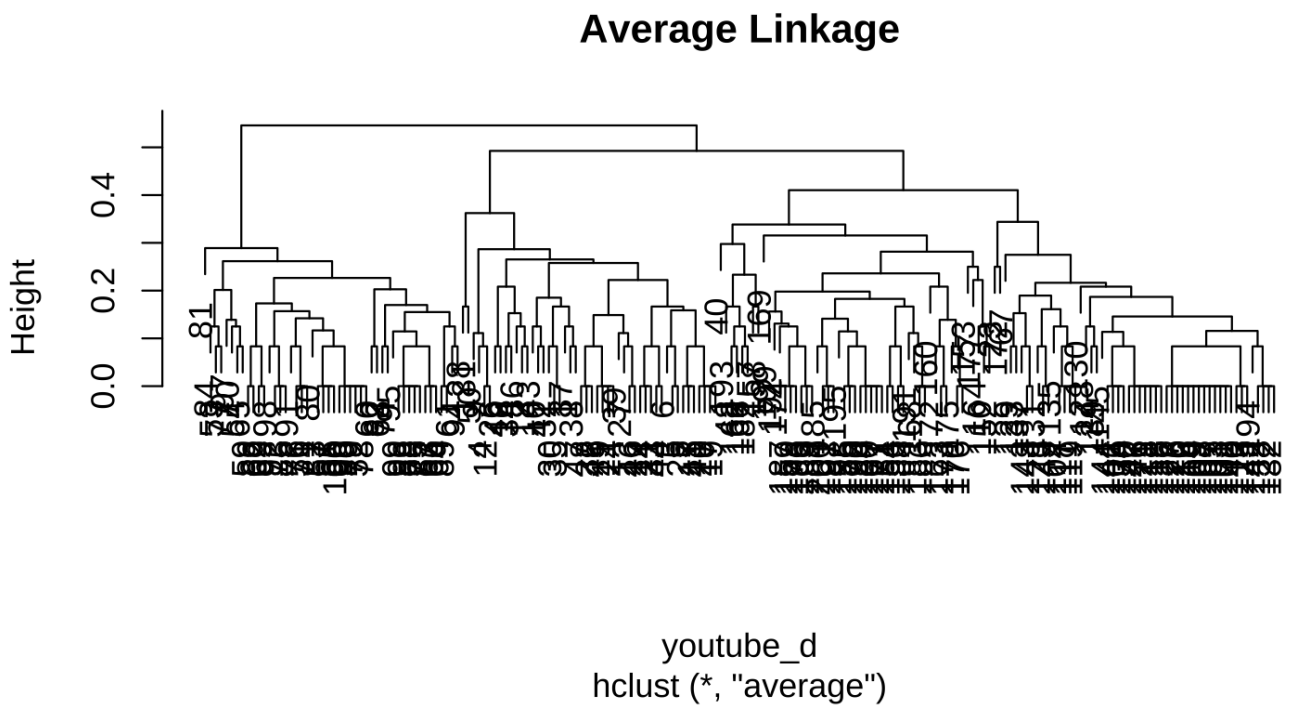
```
set.seed(2021)
hca_avg <- hclust(d = youtube_d,
                  method = "average")
```

Complete: Maybe 4, maybe 6 clusters?

Average: Probably 4 clusters

Step 4: Visualise HCAs using dendrograms

```
plot(hca_avg, main = "Average Linkage")
```

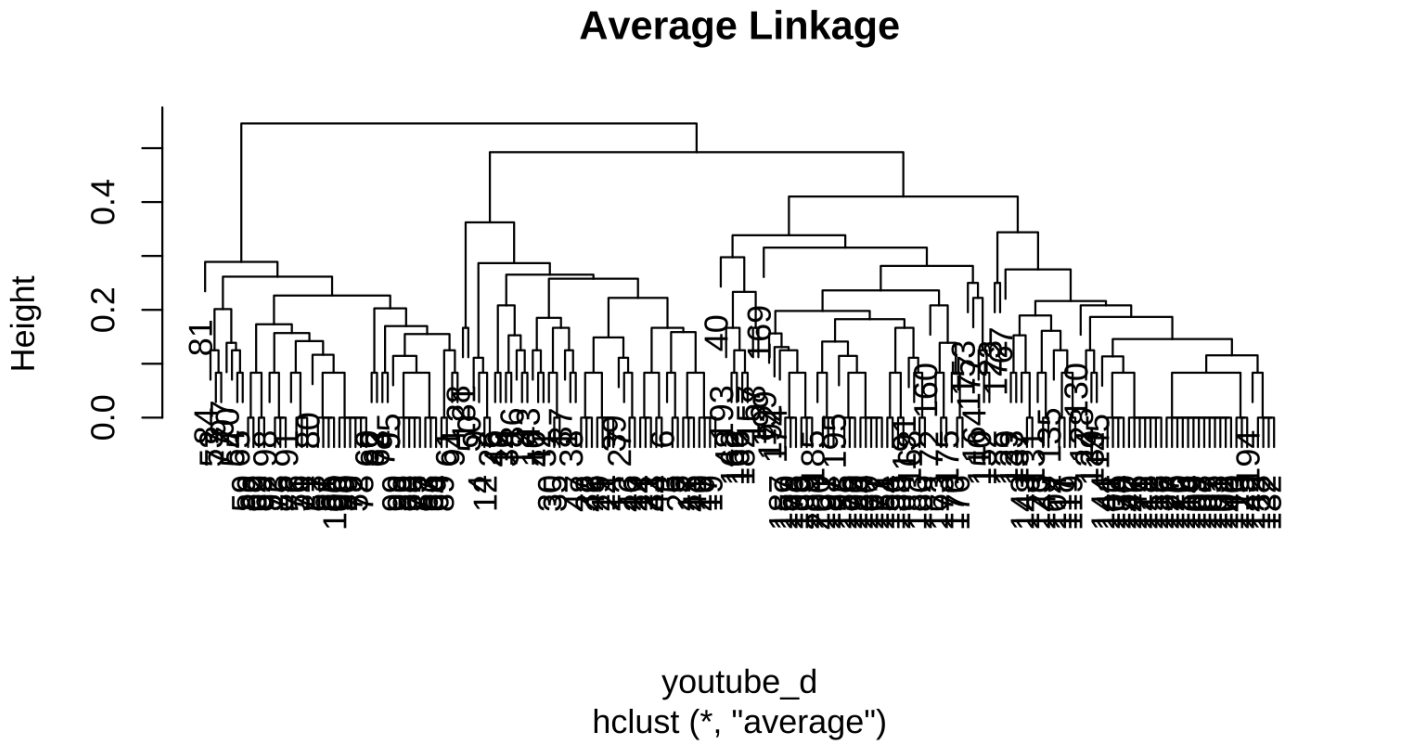


HCA: Example

Step 5: "Cut" tree to decide on cluster membership

- The `cutree` (NB: only one "t") can be used to cut based on either height (h =) or resultant number of clusters.

```
plot(hca_avg, main = "Average Linkage")
```

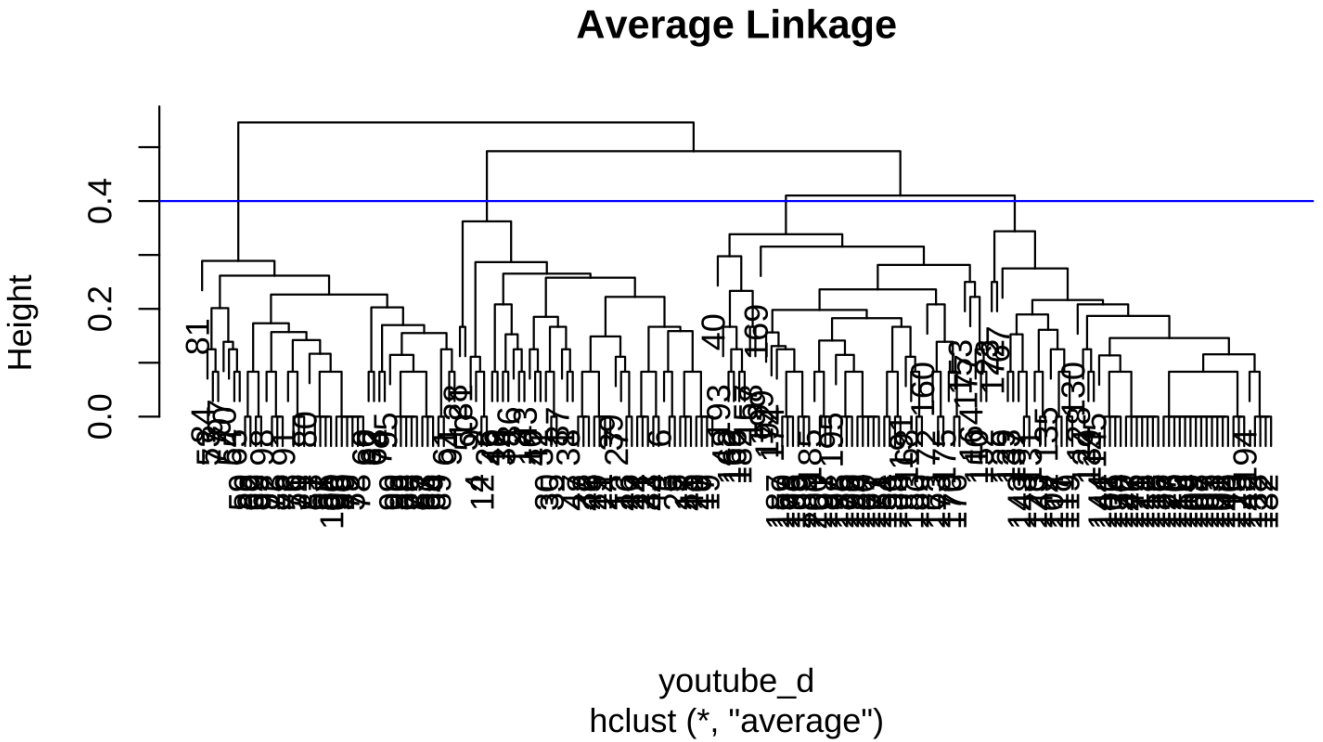


HCA: Example

Step 5: "Cut" tree to decide on cluster membership

- The `cutree` (NB: only one "t") can be used to cut based on either height (h =) or resultant number of clusters.
- e.g. `cutree(hca_avg, h = 0.4)` would cut the tree here.

```
plot(hca_avg, main = "Average Linkage")
abline(a = 0.4, b = 0, col = "blue")
```



HCA: Example

Step 5: "Cut" tree to decide on cluster membership

- The `cutree` (NB: only one "t") can be used to cut based on either height (h =) or resultant number of clusters.
- e.g. `cutree(hca_avg, h = 0.4)` would cut the tree here.
- Usually easiest to just specify the number of groups and have `R` calculate the equivalent height, e.g. `cutree(hca_avg, k = 4)`

```
hca_avg_k4  <- cutree(hca_avg, k = 4)
hca_comp_k4 <- cutree(hca_comp, k = 4)
hca_comp_k6 <- cutree(hca_comp, k = 6)

# Example of what you get

hca_avg_k4
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 3 1 1 1 1 1 1 1 3 1 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 3 4 4
##  [75] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 2 2 2 2 2 2 3 2
## [112] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [149] 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 1 3 3 3 3
## [186] 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3
```

HCA: Example

Step 6: Add cluster membership

We can add the cluster membership to our data using the `mutate` function.

```
youtube_data_results <- youtube_data %>%
  mutate(
    hca_avg_k4 = hca_avg_k4,
    hca_comp_k4 = hca_comp_k4,
    hca_comp_k6 = hca_comp_k6
  )

youtube_data_results
```

```
## # A tibble: 200 × 15
##    animation comedy documentaries music news  reviews sport tutorials trailers
##    <fct>     <fct>  <fct>         <fct> <fct> <fct>   <fct> <fct>     <fct>
##  1 0         1      0             0     0     0       0     0         0
##  2 1         0      0             0     0     1       0     0         1
##  3 1         0      0             1     0     0       0     1         1
##  4 1         0      0             1     0     1       0     0         0
##  5 1         0      0             1     0     0       0     1         1
##  6 1         0      0             1     0     0       0     0         1
##  7 1         0      0             1     0     1       0     1         1
##  8 1         1      0             1     0     0       0     1         0
##  9 1         0      1             1     0     1       0     0         0
## 10 0         1      0             1     0     0       0     0         1
## # ℹ 190 more rows
## # ℹ 6 more variables: videoessays <fct>, videogames <fct>, vlogs <fct>,
## #   hca_avg_k4 <int>, hca_comp_k4 <int>, hca_comp_k6 <int>
```

HCA: Example

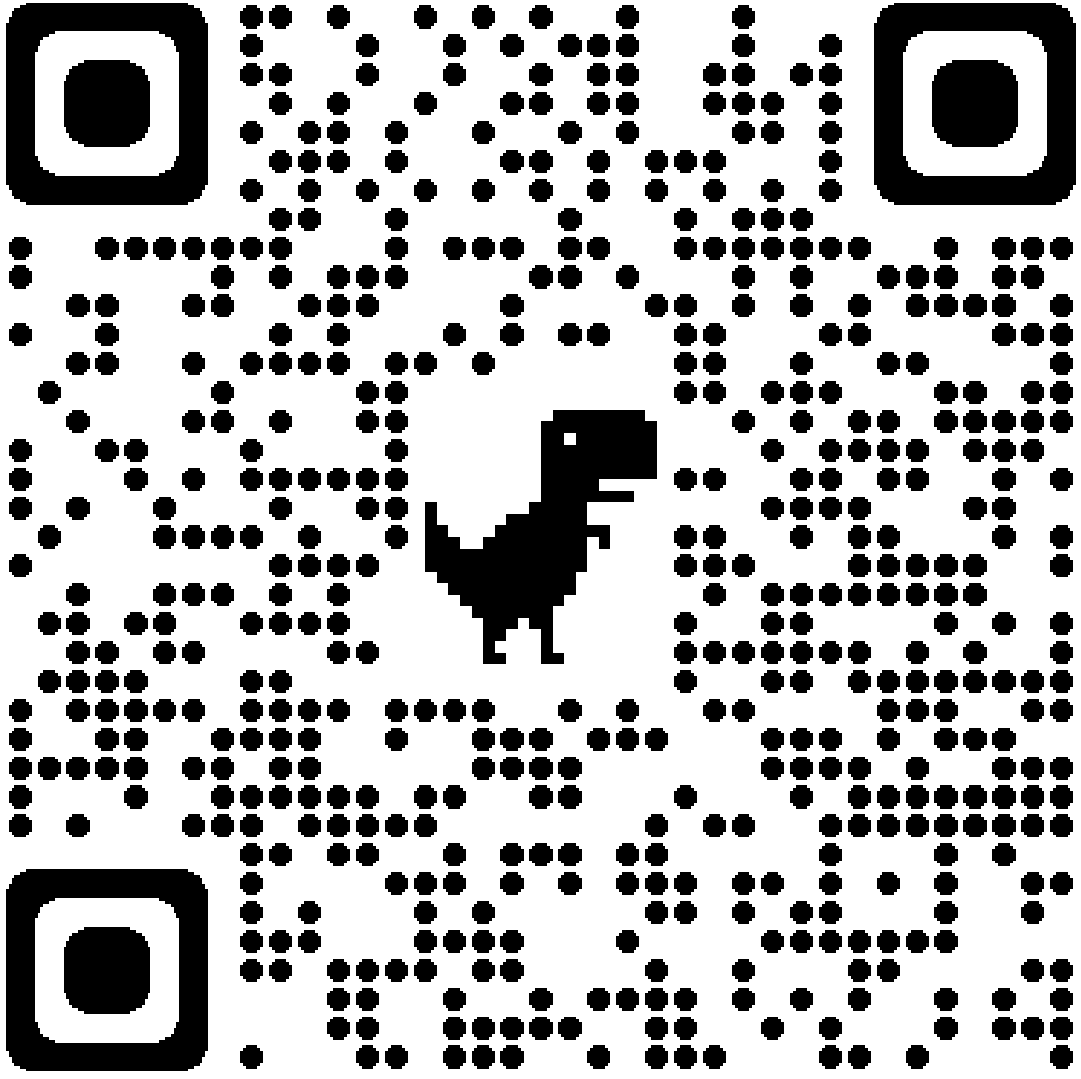
Step 7: Explore how the clusters differ with bivariate statistics

```
youtube_data_results %>%
  # change my factor variables to numeric for
  # calculating proportions
  mutate_at(vars(animation:vlogs), ~as.numeric(.)-1) %>%
  group_by(hca_avg_k4) %>%
  # Means for all youtube genres
  summarise_at(vars(animation:vlogs), ~mean(., na.rm = TRUE))
```

```
## # A tibble: 4 × 13
##   hca_avg_k4 animation comedy documentaries music  news reviews sport tutorials trailers videoessays videogames  vlogs
##        <int>     <dbl>  <dbl>         <dbl> <dbl> <dbl>   <dbl> <dbl>     <dbl>    <dbl>       <dbl>      <dbl>  <dbl>
## 1          1    0.854  0.667        0.0208 0.688 0       0.354 0         0.583    0.542       0          0.958  0.917
## 2          2    0.0943 0.0377       0.0189 0.868 0.358   0.774 0.151     0.283    0.981       0          0.0377 0
## 3          3    0.118  0.392        0.0980 0.216 0.157   0.196 0.922     0.0392   0.137       0          0.392  0.0980
## 4          4    0.0625 0.229        0.917  0.125 0.938   0.25  0.146     0.938    0.0208      0.625      0.0208 0.0833
```

Class Activity: Join the Jamboard and try to give appropriate labels to each 'group' of Youtube Viewers

Link



HCA: Example

Step 7: Explore how the clusters differ with bivariate statistics

```
youtube_data_results %>%
  # change my factor variables to numeric for
  # calculating proportions
  mutate_at(vars(animation:vlogs), ~as.numeric(.)-1) %>%
  group_by(hca_avg_k4) %>%
  # Means for all youtube genres
  summarise_at(vars(animation:vlogs), ~mean(., na.rm = TRUE))
```

```
## # A tibble: 4 × 13
##   hca_avg_k4 animation comedy documentaries music  news reviews sport tutorials trailers videoessays videogames  vlogs
##        <int>     <dbl>  <dbl>         <dbl> <dbl> <dbl>   <dbl> <dbl>     <dbl>    <dbl>       <dbl>      <dbl>  <dbl>
## 1          1    0.854  0.667        0.0208 0.688 0       0.354 0         0.583    0.542       0          0.958  0.917
## 2          2    0.0943 0.0377       0.0189 0.868 0.358   0.774 0.151     0.283    0.981       0          0.0377 0
## 3          3    0.118  0.392        0.0980 0.216 0.157   0.196 0.922     0.0392   0.137       0          0.392  0.0980
## 4          4    0.0625 0.229        0.917  0.125 0.938   0.25  0.146     0.938    0.0208      0.625      0.0208 0.0833
```

- **Average Linkage 4 Group**
  - group 1: High (>60%) animation, comedy, music, videogames, vlogs — "Young People Youtube"?
  - group 2: High (>60%) music, reviews, trailers — "Traditional Media Adjacent"?
  - group 3: High (>60%) sport, and nothing else — "Sport fans"
  - group 4: High (>60%) documentaries, news, tutorials, video essays — "Info-tainment seekers"?

HCA: Example

Step 7: Explore how the clusters differ with bivariate statistics

```
youtube_data_results %>%
  # change my factor variables to numeric for
  # calculating proportions
  mutate_at(vars(animation:vlogs), ~as.numeric(.)-1) %>%
  group_by(hca_comp_k4) %>%
  # Means for all youtube genres
  summarise_at(vars(animation:vlogs), ~mean(., na.rm = TRUE))
```

```
## # A tibble: 4 × 13
##   hca_comp_k4 animation comedy documentaries  music  news reviews sport tutorials trailers videoessays videogames  vlogs
##         <int>     <dbl>  <dbl>         <dbl>  <dbl> <dbl>   <dbl> <dbl>     <dbl>    <dbl>       <dbl>      <dbl>  <dbl>
## 1           1    0.854  0.667         0.0208 0.688  0       0.354 0         0.583    0.542       0          0.958  0.917
## 2           2    0.125  0.0781        0.0312 0.75   0.297   0.734 0.297     0.234    0.891       0          0.0469 0.0156
## 3           3    0.0882 0.382         0.0588 0.294  0.118   0.118 0.912     0.0294   0           0          0.559  0.118
## 4           4    0.0556 0.278         0.852  0.0926 0.907   0.222 0.222     0.852    0.0556      0.556      0.0185 0.0741
```

- **Complete Linkage 4 Group**
  - group 1: High (>60%) animation, comedy, music, videogames, vlogs — "Young People Youtube"?
  - group 2: High (>60%) music, reviews, trailers — "Traditional Media Adjacent"?
  - group 3: High (>60%) sport, and nothing else (maybe + videogames?) — "Sport fans"
  - group 4: High (>60%) documentaries, news, tutorials — "Info-tainment seekers"?

HCA: Example

Step 7: Explore how the clusters differ with bivariate statistics

```
youtube_data_results %>%
  # change my factor variables to numeric for
  # calculating proportions
  mutate_at(vars(animation:vlogs), ~as.numeric(.)-1) %>%
  group_by(hca_comp_k6) %>%
  # Means for all youtube genres
  summarise_at(vars(animation:vlogs), ~mean(., na.rm = TRUE))
```

```
## # A tibble: 6 × 13
##   hca_comp_k6 animation comedy documentaries  music  news reviews  sport tutorials trailers videoessays videogames
##         <int>     <dbl>  <dbl>         <dbl>  <dbl> <dbl>   <dbl>  <dbl>     <dbl>    <dbl>       <dbl>      <dbl>
## 1           1    0.854  0.667         0.0208 0.688  0       0.354 0         0.583    0.542        0         0.958
## 2           2    0.5    0             0      1      0.667   1     0.333     1        1            0         0.167
## 3           3    0.0882 0.382         0.0588 0.294  0.118   0.118 0.912     0.0294   0            0         0.559
## 4           4    0.0556 0.278         0.852  0.0926 0.907   0.222 0.222     0.852    0.0556       0.556     0.0185
## 5           5    0.0238 0.0476        0.0476 0.881  0.357   0.690 0.0238    0.190    0.976        0         0.0476
## 6           6    0.25   0.188         0      0.312  0       0.75  1         0.0625   0.625        0         0
## # ℹ 1 more variable: vlogs <dbl>
```

- **Complete Linkage 6 Group**
  - group 1: High (>60%) animation, comedy, music, videogames, vlogs — "Young People Youtube"?
  - group 2: High (>60%) music, news, reviews, tutorials — "Traditional Media Adjacent + Guides"?
  - group 3: High (>60%) sport, and nothing else (maybe + videogames?) — "Sport fans"
  - group 4: High (>60%) documentaries, news, tutorials — "Info-tainment seekers"?
  - group 5: High (>60%) music, reviews, trailers — "Traditional Media Adjacent"?
  - group 6: High (>60%) reviews and sport only — "Traditional Media Adjacent + Sport"?

With Cluster Analysis you can use the derived clusters for anything: e.g. interesting visualisations or further regression models on other variables.

Week 10: Cluster Analysis — Summary & Practical

Summary & Practical.

Summary

- Cluster Analysis can be used in quantitative social research when we are **interested in identifying an underlying grouping — or typology — of observations**.
- There are multiple forms of cluster analysis that can be used for defining clusters under different conditions: differently shaped clusters, different types of hierarchy in their definition, and different types of data. We have learned how two of these **k-means** and **Hierarchical Cluster Analysis (HCA)** can be estimated using **R**.
- Generally speaking, in Cluster analysis, we need to:
    - **Choose the most appropriate method** (k-means or HCA) and measurements (distance metric & clustering method for HCA)
    - Use data visualisation and Silhouette/Gap statistics (in k-means) to **decide on a number of underlying clusters**
    - **Describe our clusters** using bivariate statistics (we only used summary tables here, but we could also use data visualisations and other tools to show variation, ranges, different strength of correlations, etc.)

Practical

Do US states and English Community Safety Partnerships fall into clearly defined clusters of types and scales of criminal offences committed?

- Download and extract the data and .Rmd files from the `week-10-exercise.zip`. Open the exercise Rmarkdown file and follow the instructions to complete the practical.