Week 10 Exercise: Cluster Analysis

Calum Webb

27/11/2021

In this practical activity we will be using cluster analysis to try and explore whether there are underlying clusters of crime incidence in US states and English Community Safety Partnerships, using data from the CORGIS Dataset Project (https://corgis-edu.github.io/corgis/csv/state_crime/) and the Office for National Statistics (https://www.ons.gov.uk/peoplepopulationandcommunity/crimeandjustice/datasets/recordedcrimedatabycommunitysafetypartnershiparea). The researchers in this activity are interested in finding out if states/community partnerships simply cluster into 'high' or 'low' crime, or whether there are clusters of specific types of crime (assault, murder, sexual crime, theft, etc.)

First, you will be asked to follow along and interpret the output from some code analysing clusters of crime in US states. Then, you will be asked to use this code as a template to explore clusters of crime in community safety partnerships in England.

Part I: Clusters of Crime Types in US States

Start by loading (or installing and then loading) the relevant libraries used for cluster analysis.

```
# Don't forget to install any packages you don't have installed.
#install.packages("tidyverse")
#install.packages("cluster")
#install.packages("factoextra")

library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

First, the researchers read in the data and save it in an object called `usa_data`

```
usa_data <- read_csv("state_crime_rates.csv")
```

```
## Rows: 51 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## chr (1): state
## dbl (6): property_burglary, property_larceny, property_motor, violent_assaul...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
usa_data
```

```
## # A tibble: 51 x 7
##    state       property_burglary property_larceny property_motor violent_assault
##    <chr>                   <dbl>            <dbl>          <dbl>           <dbl>
##  1 Alabama                  532.            1886.           256.            381
##  2 Alaska                   487.            2066            358.            596
##  3 Arizona                  394.            1797.           249.            312.
##  4 Arkansas                 600.            2013.           246.            448.
##  5 California               386.            1586.           359.            267.
##  6 Colorado                 348.            1858.           384             246.
##  7 Connecticut              181.            1079.           167.            105
##  8 Delaware                 305.            1783.           165.            305.
##  9 Florida                  295.            1669.           182.            258.
## 10 Georgia                  372.            1780.           224.            232
## # i 41 more rows
## # i 2 more variables: violent_murder <dbl>, violent_robbery <dbl>
```

Next, the researchers decide that because their variables of interest are all continuous they will start by using k-means to try and identify relevant clusters. They start by removing any non-numeric variables from their dataset, keeping only the numeric ones.

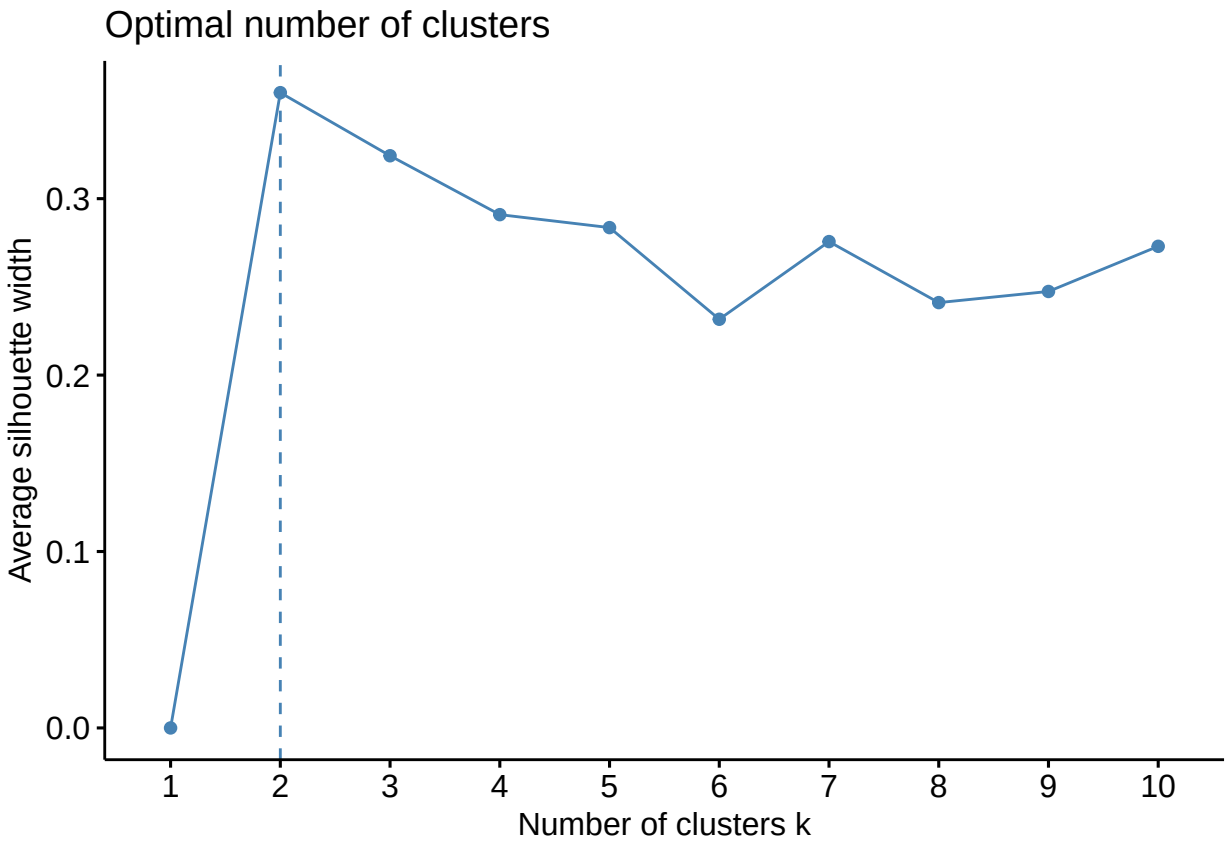
```
usa_data_prepped <- usa_data %>%
  select(-state)
```

It's generally also recommended to standardise all of the continuous variables that you are using in a cluster analysis too, as what might look like big gaps between data points might just be linked to the scale that they're on (e.g. the difference between measuring something as being 1000 milimeters away compared to 1 meter). This can be achieved using the `scale()` function. Don't worry, we'll be adding our clusters back onto our original data so we'll still be able to make sense of the differences between groups.

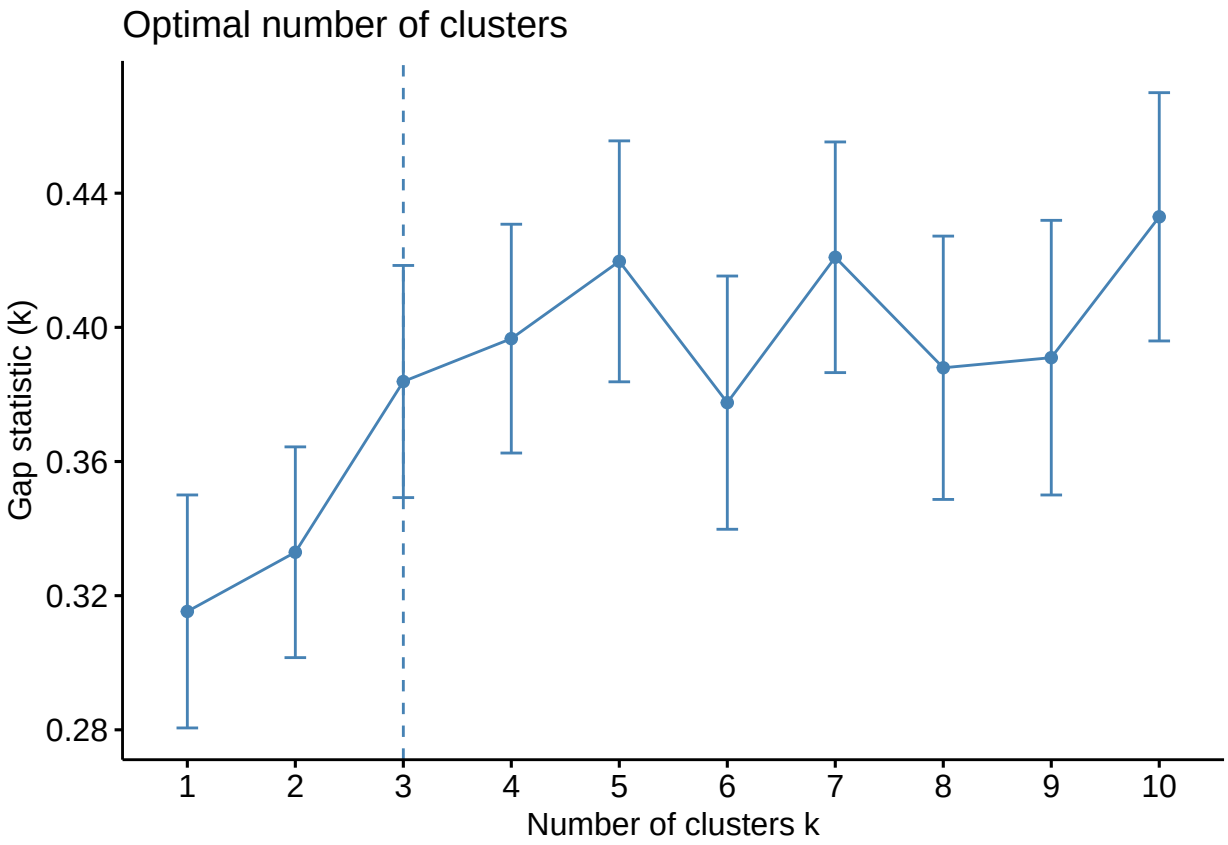
```
usa_data_prepped <- usa_data_prepped %>%
  mutate_all(scale)
```

They start by using the `factoextra` package, and the `fviz_nbclust` function to try and determine how many clusters they should try to identify.

```
# Silhouette Plot
fviz_nbclust(x = usa_data_prepped,
             FUNcluster = kmeans,
             method = "silhouette")
```




```
# Gap statistic plot
fviz_nbclust(x = usa_data_prepped,
             FUNcluster = kmeans,
             method = "gap")
```



- Interpret the above plots: what is the optimal number of clusters according to the silhouette statistic and what is the optimal number of clusters according to the gap statistic?

The optimal number of clusters according to the silhouette statistic is 2, whereas the optimal number of clusters according to the gap statistic is 3.

The researchers decide that they will create a 2-cluster solution (as suggested by the silhouette plot), as well as a 3-cluster solution suggested by the gap statistic. They use the `kmeans` to first estimate the clusters, they then visualise the clusters using the `fviz_cluster` function.

```
# run kmeans analysis
set.seed(2021)
usa_k2 <- kmeans(usa_data_prepped, centers = 2)
usa_k2
```

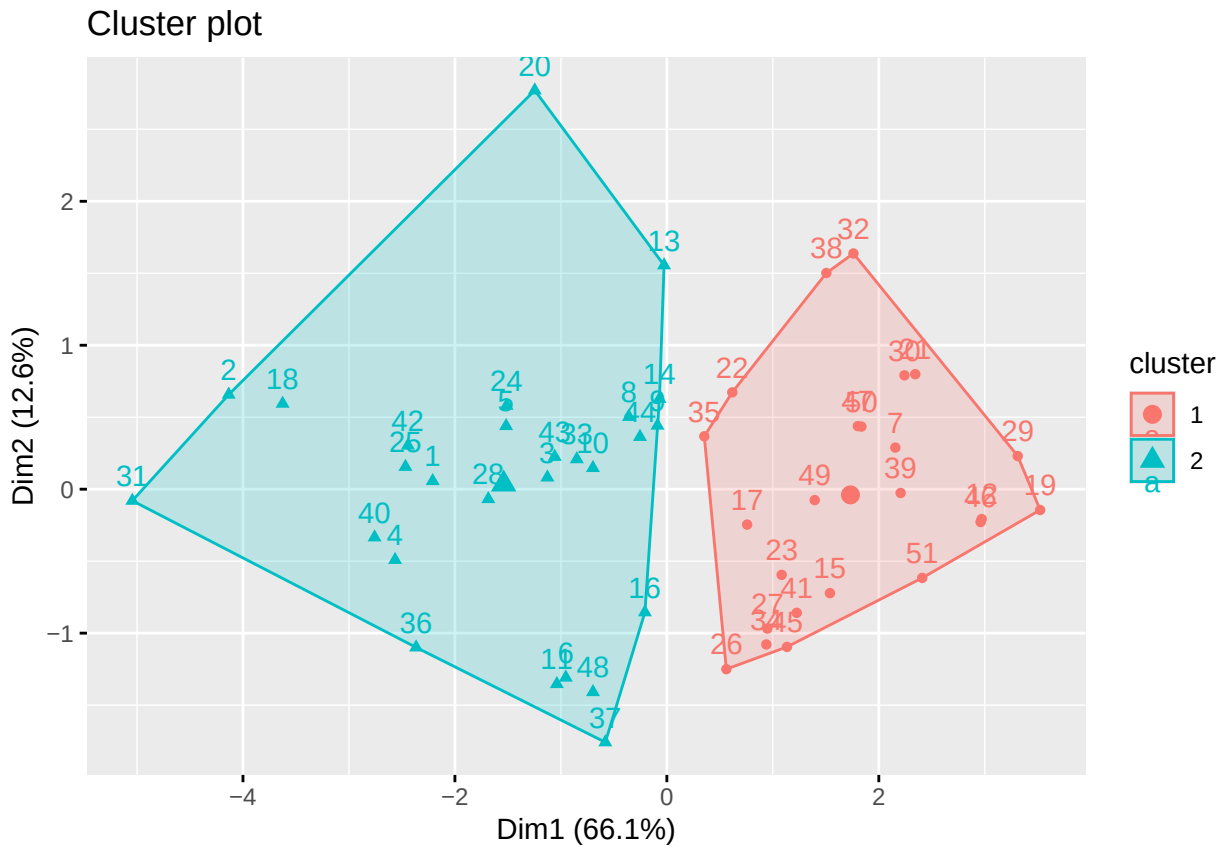
```
## K-means clustering with 2 clusters of sizes 24, 27
##
## Cluster means:
##   property_burglary property_larceny property_motor violent_assault
## 1        -0.7314813       -0.8133296     -0.7111823      -0.5996541
## 2         0.6502056        0.7229597      0.6321621       0.5330258
##   violent_murder violent_robbery
## 1     -0.6848029      -0.7101602
## 2      0.6087137       0.6312535
##
## Clustering vector:
```

```
##  [1] 2 2 2 2 2 2 1 2 2 2 2 1 2 2 1 2 1 2 1 2 1 1 1 2 2 1 1 2 1 1 2 1 2 1 1 2 2 1
## [39] 1 2 1 2 2 2 1 1 1 2 1 1 1
##
## Within cluster sum of squares by cluster:
## [1]  49.51358 112.88988
##  (between_SS / total_SS =  45.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

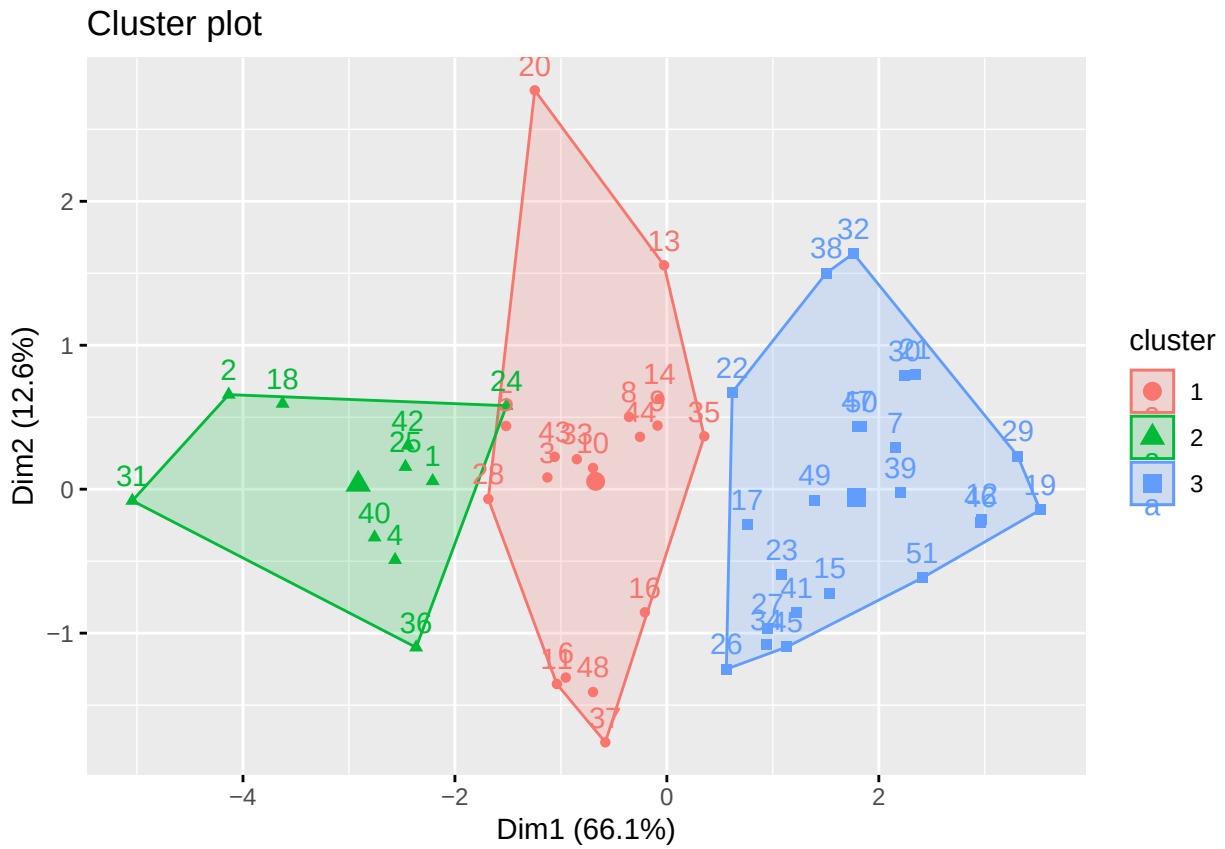
```
# run kmeans analysis
set.seed(2021)
usa_k3 <- kmeans(usa_data_prepped, centers = 3)
usa_k3
```

```
## K-means clustering with 3 clusters of sizes 18, 10, 23
##
## Cluster means:
##   property_burglary property_larceny property_motor violent_assault
## 1         0.1498748        0.4328118      0.4350552     -0.02077013
## 2         1.5058837        1.1684405      0.8681425      1.40374373
## 3        -0.7720254       -0.8467399     -0.7179312     -0.59406847
##   violent_murder violent_robbery
## 1     0.04296655       0.6827466
## 2     1.55919719       0.5151203
## 3    -0.71153782      -0.7582887
##
## Clustering vector:
##  [1] 2 2 1 2 1 1 3 1 1 1 1 3 1 1 3 1 3 2 3 1 3 3 3 2 2 3 3 1 3 3 2 3 1 3 1 2 1 3
## [39] 3 2 3 2 1 1 3 3 3 1 3 3 3
##
## Within cluster sum of squares by cluster:
## [1] 41.92760 30.60742 46.27448
##  (between_SS / total_SS =  60.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
# visualise clusters - 2 cluster
fviz_cluster(usa_k2,
             data = usa_data_prepped)
```




```
# visualise clusters - 3 cluster
fviz_cluster(usa_k3,
             data = usa_data_prepped)
```



- Approximately what proportion of variance could be explained by the cluster membership solution chosen in the 2-cluster and 3-cluster solutions?

The proportion of variance explained in the 2 cluster solution was around 45.9%, whereas the proportion explained in the 3 cluster solution was 60.4%.

- Do the clusters look well defined? Are there any states that may have been misclassified by the algorithm?

Both 2-clusters and 3 cluster solutions appear to have minimal overlap. In the three cluster solution, there is some ambiguity as to whether state 24 (Mississippi) and state 5 (California) belong in Cluster 1 or Cluster 2.

The researchers then decide to add the cluster membership to the original data and then explore how the clusters differ in mean values of each crime rate.

```
# Add cluster results to data and save in `usa_data_results`
usa_data_results <- usa_data %>%
  mutate(
    cluster_k2 = usa_k2$cluster,
    cluster_k3 = usa_k3$cluster
  )

# Summarise all numeric variables with their mean
# Tip: Uncomment the %>% view() section of the code to view all output
usa_data_results %>%
  group_by(cluster_k2) %>%
```

```
  summarise_all(~mean(., na.rm = TRUE)) # %>% view()
```

```
## Warning: There were 2 warnings in `summarise()`.
## The first warning was:
## i In argument: `state = (structure(function (..., .x = ..1, .y = ..2, . = ..1)
##   ...`.
## i In group 1: `cluster_k2 = 1`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```
## # A tibble: 2 x 9
##   cluster_k2 state property_burglary property_larceny property_motor
##        <int> <dbl>             <dbl>            <dbl>          <dbl>
## 1          1    NA              244.            1241.           145.
## 2          2    NA              439.            1799.           274.
## # i 4 more variables: violent_assault <dbl>, violent_murder <dbl>,
## #   violent_robbery <dbl>, cluster_k3 <dbl>
```

```
usa_data_results %>%
  group_by(cluster_k3) %>%
  summarise_all(~mean(., na.rm = TRUE)) # %>% view()
```

```
## Warning: There were 3 warnings in `summarise()`.
## The first warning was:
## i In argument: `state = (structure(function (..., .x = ..1, .y = ..2, . = ..1)
##   ...`.
## i In group 1: `cluster_k3 = 1`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## i Run `dplyr::last_dplyr_warnings()` to see the 2 remaining warnings.
```

```
## # A tibble: 3 x 9
##   cluster_k3 state property_burglary property_larceny property_motor
##        <int> <dbl>             <dbl>            <dbl>          <dbl>
## 1          1    NA              368.            1694.           255.
## 2          2    NA              559.            1961.           296.
## 3          3    NA              238.            1229.           144.
## # i 4 more variables: violent_assault <dbl>, violent_murder <dbl>,
## #   violent_robbery <dbl>, cluster_k2 <dbl>
```

- Describe and label the two kinds of clusters found in the 2-cluster solution.

The two cluster solution appears to be broken into high-crime and low-crime clusters. Cluster 1 has higher average incidence of all forms of recorded crime whereas cluster 2 has lower average incidence of all forms of crime.

- Describe and label the three kinds of clusters found in the 3-cluster solution.

The three cluster solution appears to have clustered the states into low, middling, and high incidence of crime, with cluster 3 being the lowest and cluster 2 being the highest. The exception is that cluster 1 seems to have the highest incidences of violent robberies, which may be more of a unique feature of this cluster.

Some plots have been created to help visualise the differences between clusters more easily. This is achieved using the `pivot_longer` function to first put all of the variables on their own rows. Notice how this is done

using the standardised scores so that they are easier to compare. Look at how the structure of the dataset changes:

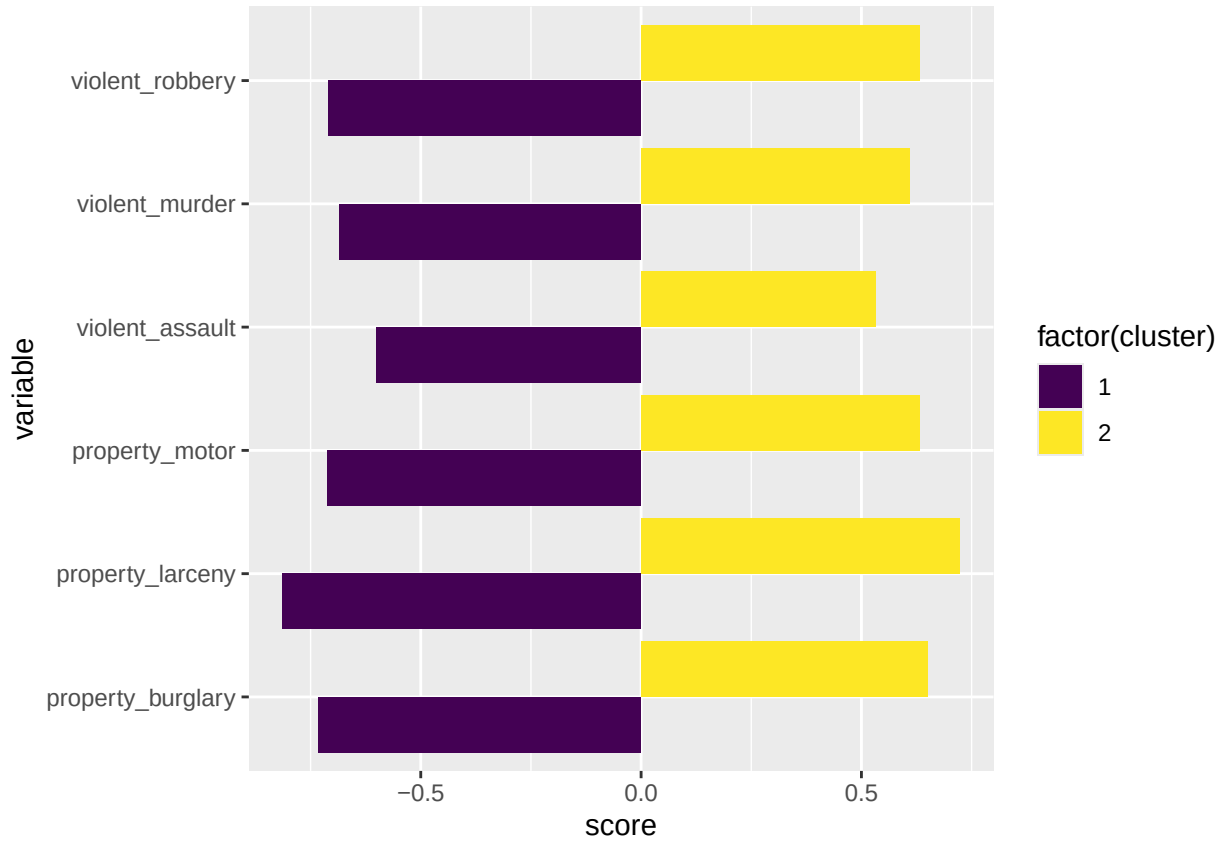
```
# Create a data frame from our results. Save our cluster in
# the variable "cluster"
k2_results <- as_tibble(usa_k2$centers) %>%
  rowid_to_column(var = "cluster")
k2_results
```

```
## # A tibble: 2 x 7
##   cluster property_burglary property_larceny property_motor violent_assault
##     <int>             <dbl>            <dbl>          <dbl>           <dbl>
## 1       1            -0.731           -0.813         -0.711          -0.600
## 2       2             0.650            0.723          0.632           0.533
## # i 2 more variables: violent_murder <dbl>, violent_robbery <dbl>
```

```
k2_results_long <- k2_results %>%
  # make all of our variable, but not our cluster ID, long format.
  # Make our variable column name "variable" and our value name "score"
  pivot_longer(-cluster, names_to = "variable", values_to = "score")

k2_results_long
```

```
## # A tibble: 12 x 3
##    cluster variable          score
##      <int> <chr>             <dbl>
##  1       1 property_burglary -0.731
##  2       1 property_larceny  -0.813
##  3       1 property_motor    -0.711
##  4       1 violent_assault   -0.600
##  5       1 violent_murder    -0.685
##  6       1 violent_robbery   -0.710
##  7       2 property_burglary  0.650
##  8       2 property_larceny   0.723
##  9       2 property_motor     0.632
## 10       2 violent_assault    0.533
## 11       2 violent_murder     0.609
## 12       2 violent_robbery    0.631
```

```
k2_results_long %>%
  ggplot() +
  geom_col(aes(x = score, y = variable, fill = factor(cluster)),
           position = "dodge") +
  scale_fill_viridis_d() # make the scale more colourblind friendly
```



Of course, with only two clusters this isn't much more helpful than just looking at the table. But the visualisations can be more useful if we want to use multiple clusters.

Try copy and pasting the code above to the R chunk below and editing it so that it visualises the three cluster solution instead:

```
# Create a data frame from our results. Save our cluster in
# the variable "cluster"
k3_results <- as_tibble(usa_k3$centers) %>%
  rowid_to_column(var = "cluster")
k3_results
```

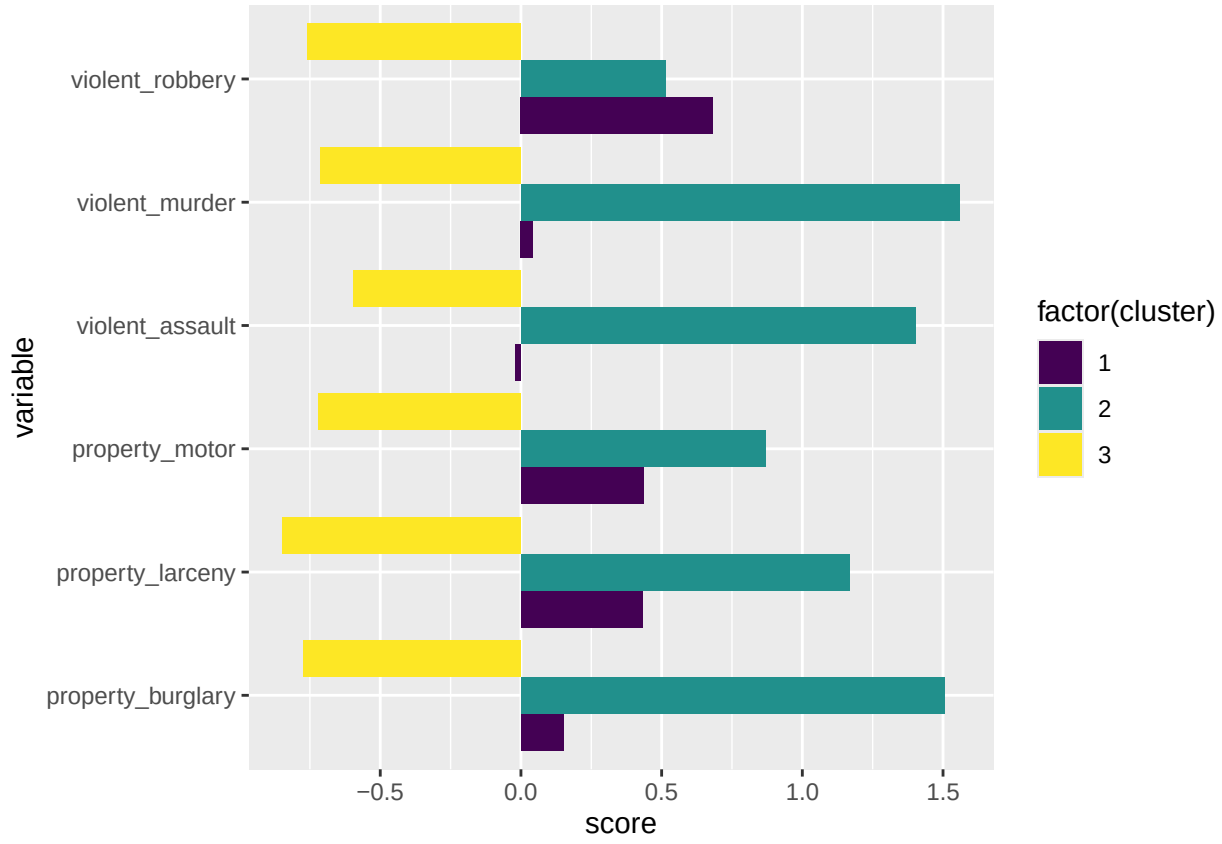
```
## # A tibble: 3 x 7
##   cluster property_burglary property_larceny property_motor violent_assault
##     <int>             <dbl>            <dbl>          <dbl>           <dbl>
## 1       1             0.150            0.433          0.435         -0.0208
## 2       2             1.51             1.17           0.868          1.40
## 3       3            -0.772           -0.847         -0.718         -0.594
## # i 2 more variables: violent_murder <dbl>, violent_robbery <dbl>
```

```
k3_results_long <- k3_results %>%
  # make all of our variable, but not our cluster ID, long format.
  # Make our variable column name "variable" and our value name "score"
  pivot_longer(-cluster, names_to = "variable", values_to = "score")

k3_results_long
```

```
## # A tibble: 18 x 3
```

```
##    cluster variable            score
##      <int> <chr>               <dbl>
##  1       1 property_burglary  0.150
##  2       1 property_larceny   0.433
##  3       1 property_motor     0.435
##  4       1 violent_assault   -0.0208
##  5       1 violent_murder     0.0430
##  6       1 violent_robbery    0.683
##  7       2 property_burglary  1.51
##  8       2 property_larceny   1.17
##  9       2 property_motor     0.868
## 10       2 violent_assault    1.40
## 11       2 violent_murder     1.56
## 12       2 violent_robbery    0.515
## 13       3 property_burglary -0.772
## 14       3 property_larceny  -0.847
## 15       3 property_motor    -0.718
## 16       3 violent_assault   -0.594
## 17       3 violent_murder    -0.712
## 18       3 violent_robbery   -0.758
```

```r
k3_results_long %>%
  ggplot() +
  geom_col(aes(x = score, y = variable, fill = factor(cluster)),
           position = "dodge") +
  scale_fill_viridis_d() # make the scale more colourblind friendly
```



The researchers then decide to check whether they find similar results when using hierarchical cluster analysis.

They decide that since all of their data is continuous, they will create a dissimilarity matrix based on Euclidean distance.

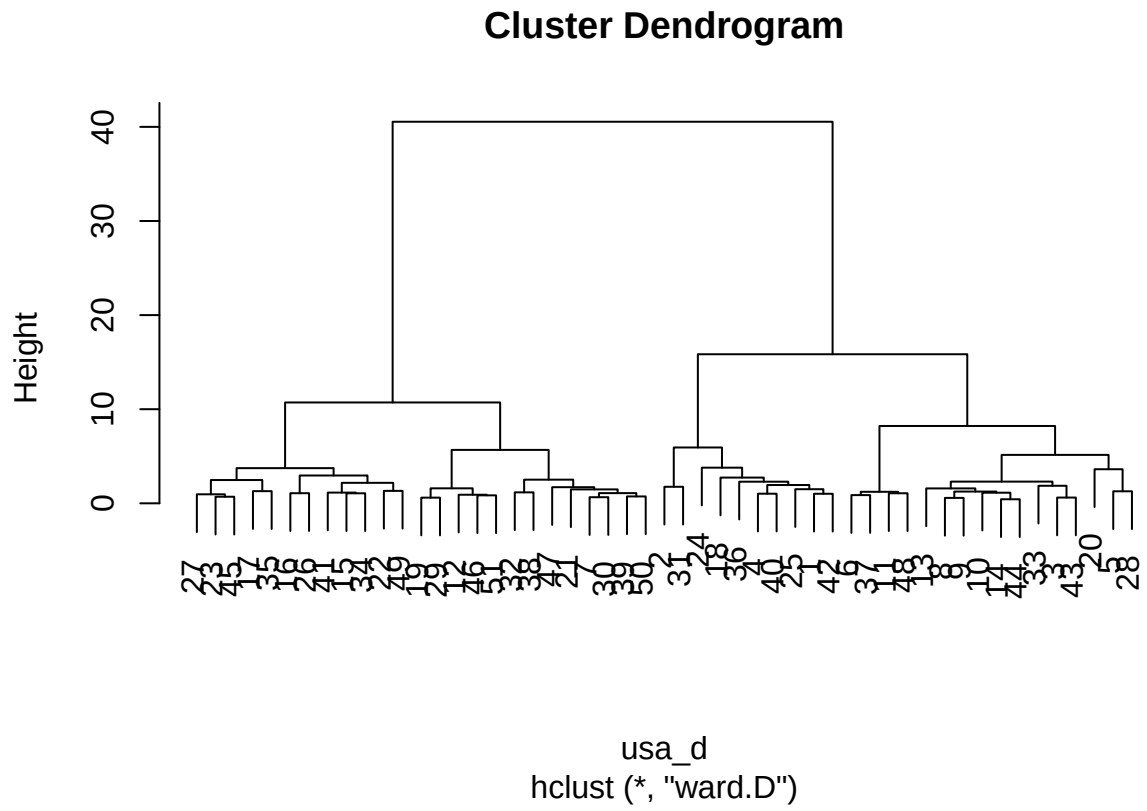
```
usa_d <- daisy(as.matrix(usa_data_prepped), metric = "euclidean")
```

They decide to use two different methods for hierarchical cluster analysis: the Ward's linkage method and complete linkage method.

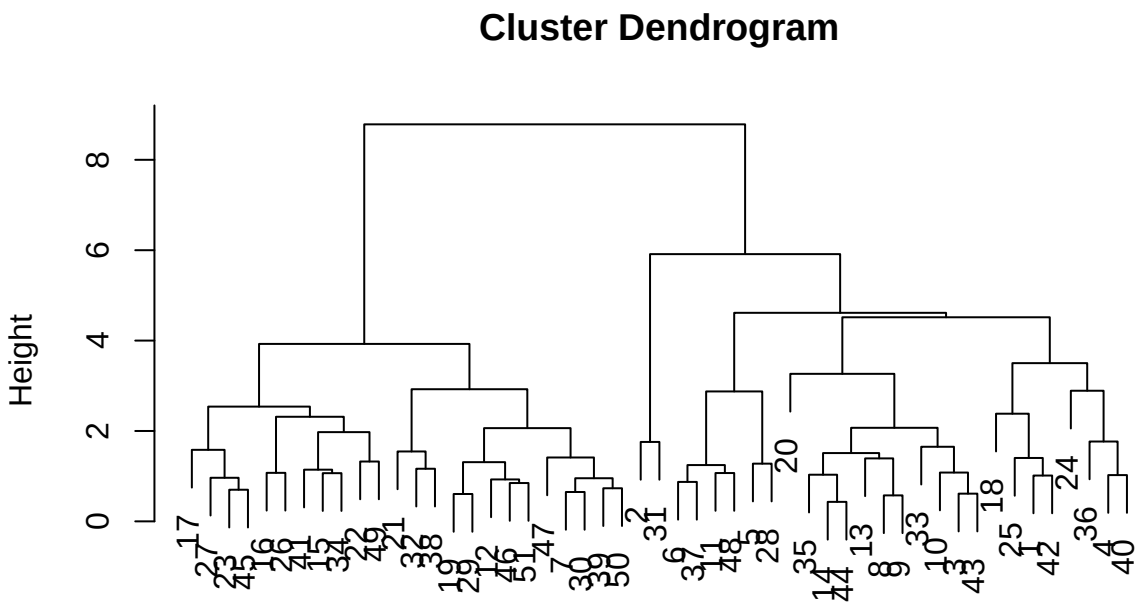
```
set.seed(2021)
usa_ward     <- hclust(d = usa_d, method = "ward.D")
set.seed(2021)
usa_complete <- hclust(d = usa_d, method = "complete")
```

Then then visualise a dendrogram of their results.

```
plot(usa_ward)
```




```
plot(usa_complete)
```



- Based on the two dendrograms, how many different clusters might be reasonable to extract from the data and why?

HCA using Ward linkage seems to show two very large clusters, but there could be good arguments for a three cluster solution. Similarly, with complete linkage there is a good argument for a two cluster solution, a three cluster solution with two "outlier" states, or a more complex 6 cluster solution.

The researchers decide to test a 3-cluster (Ward and complete) solution to their data clustering. They use the `cutree` function to achieve this.

```
usa_ward_k3 <- cutree(usa_ward, k = 3)
usa_comp_k3 <- cutree(usa_complete, k = 3)
```

They add the cluster results to their data and generate some descriptive statistics for each solution.

```
usa_data_hca_results <- usa_data %>%
  mutate(
    ward_k3 = usa_ward_k3,
    comp_k3 = usa_comp_k3
  )

# Results for 3 cluster ward
usa_data_hca_results %>%
  group_by(ward_k3) %>%
  summarise_all(~mean(., na.rm = TRUE)) # %>% view()
```

```
## Warning: There were 3 warnings in `summarise()`.
```

```
## The first warning was:
## i In argument: `state = (structure(function (..., .x = ..1, .y = ..2, . = ..1)
##   ...`.
## i In group 1: `ward_k3 = 1`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## i Run `dplyr::last_dplyr_warnings()` to see the 2 remaining warnings.
```

```
## # A tibble: 3 x 9
##   ward_k3 state property_burglary property_larceny property_motor
##     <int> <dbl>             <dbl>            <dbl>          <dbl>
## 1       1    NA              559.             1961.           296.
## 2       2    NA              369.             1703.           261.
## 3       3    NA              248              1260.           149.
## # i 4 more variables: violent_assault <dbl>, violent_murder <dbl>,
## #   violent_robbery <dbl>, comp_k3 <dbl>
```

```
# Results for 3 cluster complete
usa_data_hca_results %>%
  group_by(comp_k3) %>%
  summarise_all(~mean(., na.rm = TRUE)) # %>% view()
```

```
## Warning: There were 3 warnings in `summarise()`.
## The first warning was:
## i In argument: `state = (structure(function (..., .x = ..1, .y = ..2, . = ..1)
##   ...`.
## i In group 1: `comp_k3 = 1`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## i Run `dplyr::last_dplyr_warnings()` to see the 2 remaining warnings.
```

```
## # A tibble: 3 x 9
##   comp_k3 state property_burglary property_larceny property_motor
##     <int> <dbl>             <dbl>            <dbl>          <dbl>
## 1       1    NA              428.             1773.           261.
## 2       2    NA              592.             2027.           392.
## 3       3    NA              243.             1250.           149.
## # i 4 more variables: violent_assault <dbl>, violent_murder <dbl>,
## #   violent_robbery <dbl>, ward_k3 <dbl>
```

- Write a description and labels for the 3-cluster solution found through Ward's linkage

The Ward linkage HCA appears to show three clusters of states that could be labelled high crime (1), medium crime (2), and low crime (3), with the exception of violent robberies.

- Write a description and labels for the 3-cluster solution found through complete linkage

Complete data linkage appears to have found clusters very similar to Ward linkage and k-means, but the very high cluster has been limited to two states and therefore stands out more. Cluster two has the highest rate of all crimes, and followed by cluster 1 and then by cluster 3, which has the lowest rates.

- How would you summarise the research? Did the researchers find evidence that state-level crime fell into distinct categories of crimes committed, or did clusters largely reflect rates of all crimes?

Cluster analysis did not seem to suggest that, at the state level, communities face very different clusters characterised by types of crime. We might have expected that some states face more difficulties with property crimes while others face more problems with violent crimes, but the evidence from the cluster analysis did not seem to support this. Rather, the clustering of crime was associated with the prevalence of all forms of crime rather than specific forms.

In the Ward Linkage, Cluster 1 contained states with the highest levels of crime. These tended to be states with high levels of poverty, including Alabama, Mississippi, and Arkansas. States in the cluster with the lowest crime rates tended to be concentrated in the North East and included, for example, New York, New Jersey, New Hampshire, and Vermont, but there were also some unexpected states such as West Virginia.

Part II: Clusters of Crime Types in English Community & Safety Partnerships

Now we'll explore whether we find similar or different results for crime rates in English Community and Safety Partnerships.

- Start by loading the `"england_crime_rates.csv"` data into `R` using the `read_csv` function. Save the result to an object called `english_crime`.

```
english_crime <- read_csv("england_crime_rates.csv")
```

```
## Rows: 300 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): community_safety_partnership_code, community_safety_partnership_name
## dbl (8): violence_against_the_person, sexual_offences, robbery, theft_offenc...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
english_crime
```

```
## # A tibble: 300 x 10
##    community_safety_partnership_~1 community_safety_par~2 violence_against_the~3
##    <chr>                           <chr>                                   <dbl>
##  1 E22000001                       Bath and North East S~                   18.3
##  2 E22000002                       Bristol, City of                         34.5
##  3 E22000003                       North Somerset                           25.6
##  4 E22000369                       Somerset                                 25.1
##  5 E22000006                       South Gloucestershire                    19.5
##  6 E22000009                       Bedford                                  29.1
##  7 E22000353                       Central Bedfordshire                     17.6
##  8 E22000010                       Luton                                    32.8
##  9 E22000013                       Cambridge                                30.4
## 10 E22000014                       East Cambridgeshire                      19.0
## # i 290 more rows
## # i abbreviated names: 1: community_safety_partnership_code,
## #   2: community_safety_partnership_name, 3: violence_against_the_person
## # i 7 more variables: sexual_offences <dbl>, robbery <dbl>,
## #   theft_offences <dbl>, vehicle_offences <dbl>,
## #   criminal_damage_and_arson <dbl>, drug_offences <dbl>,
## #   public_order_offences <dbl>
```

- Check the kinds of variables in the data and decide whether you could use k-means, Hierarchical Cluster Analysis, or both methods for exploring clusters of crime.

Both k-means and Hierarchical Cluster Analysis could be used for analysing this data as all variables of interest are continuous.

- Create a version of the data that contains only the numeric type variables and store it in an object called `english_crime_prepped` so that it can be used for k-means and HCA. Then, standardise this

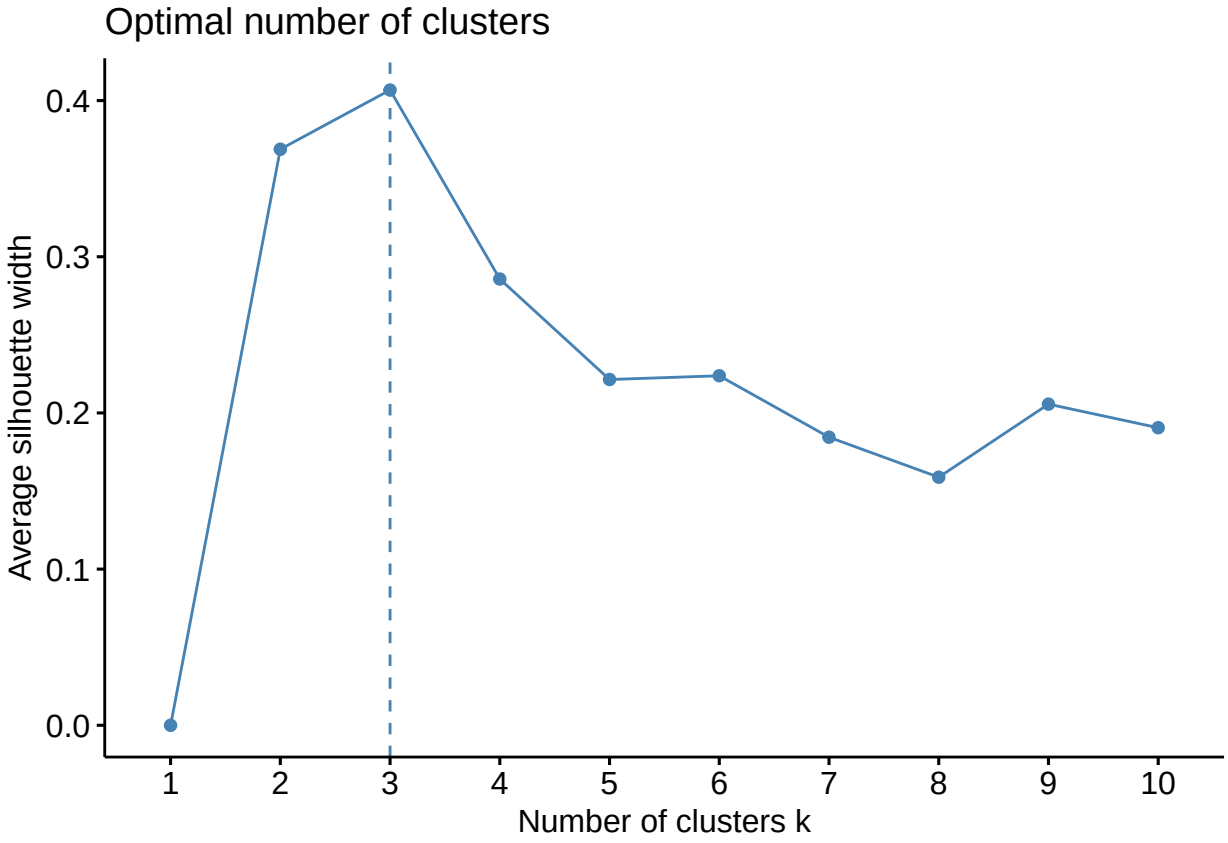
dataset using the `scale` function.

```
english_crime_prepped <- english_crime %>%
  select_if(is.numeric)

english_crime_prepped <- scale(english_crime_prepped)
```

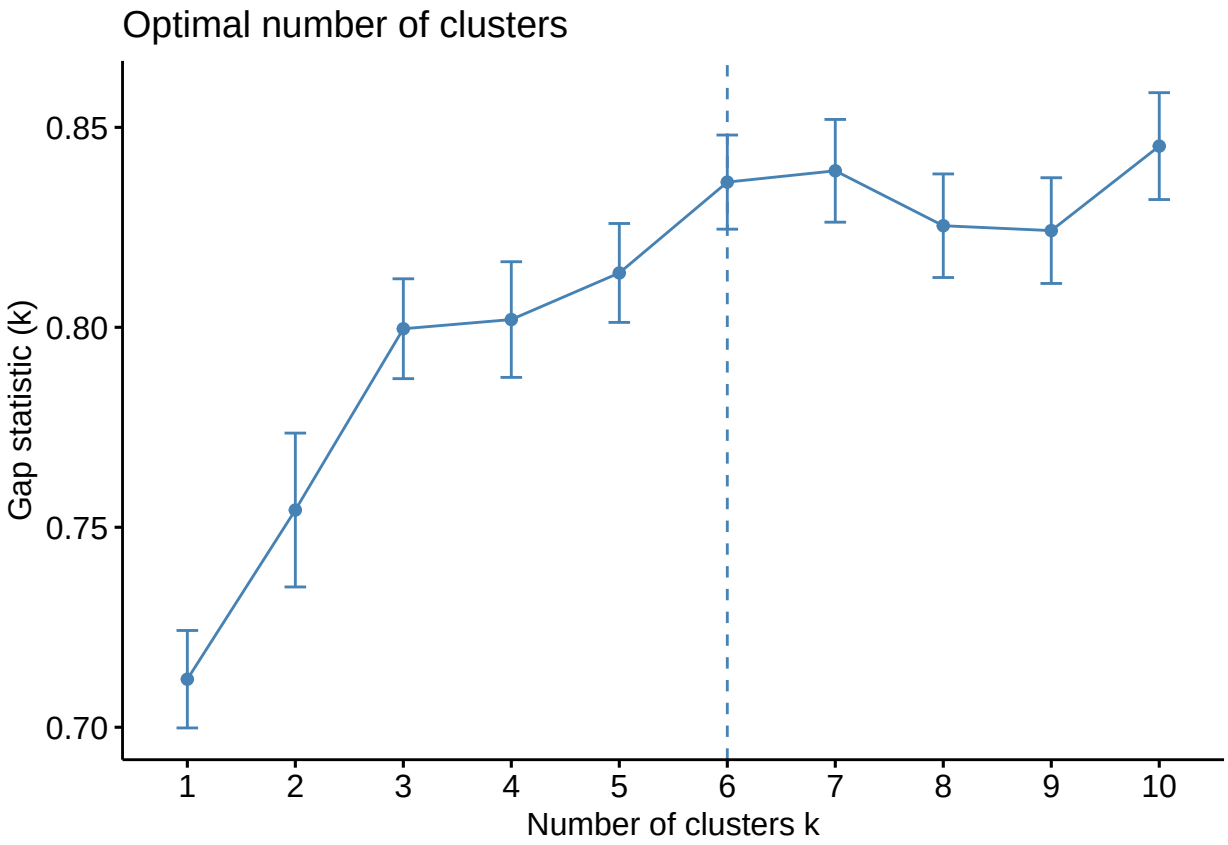
Let's start with k-means analysis.

- Use the `fviz_nbclust` function from the `factoextra` package to identify the optimal cluster solution under both the silhouette method and the gap statistic method.

```
fviz_nbclust(english_crime_prepped, FUNcluster = kmeans, method = "silhouette")
```




```
fviz_nbclust(english_crime_prepped, FUNcluster = kmeans, method = "gap")
```



- How many clusters do the silhouette and gap statistic methods recommend respectively?

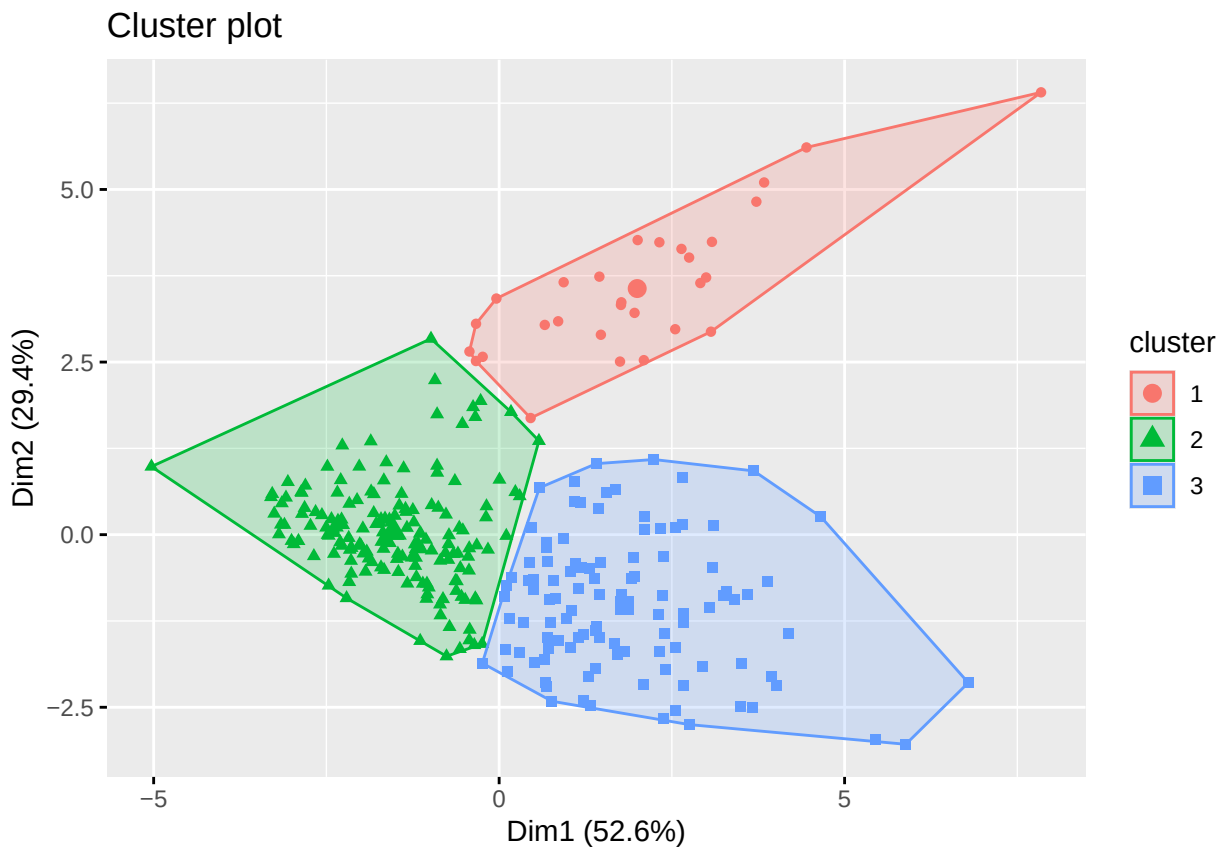
The silhouette method recommends a three cluster solution and the gap statistic recommends a six cluster solution.

- Create a 3-cluster and a 6-cluster solution for the English crime data using the `kmeans` function. Remember to save the results to an object for later use.

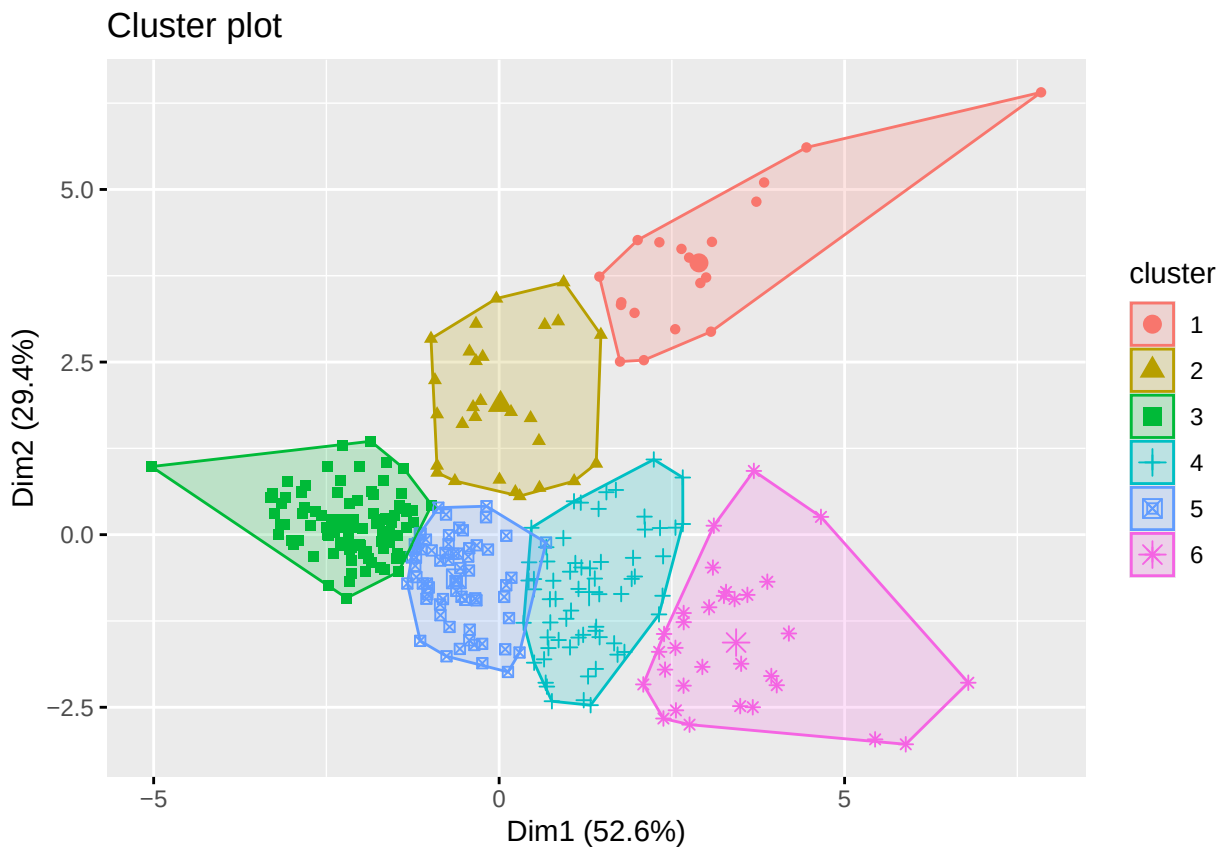
```
set.seed(2021)
english_km3 <- kmeans(english_crime_prepped, centers = 3)
set.seed(2021)
english_km6 <- kmeans(english_crime_prepped, centers = 6)
```

- Visualise the 3 cluster and 6 cluster solutions using the `fviz_clusters` function.

```
fviz_cluster(english_km3, data = english_crime_prepped, geom = "point")
```




```
fviz_cluster(english_km6, data = english_crime_prepped, geom = "point")
```



- By calling the k-means objects created earlier, report the proportion of variance that can be explained by the 3-cluster solution and the 6-cluster solution.

```
english_km3
```

```
## K-means clustering with 3 clusters of sizes 29, 165, 106
##
## Cluster means:
##   violence_against_the_person sexual_offences    robbery theft_offences
## 1                  -0.3516035      -0.2074953  2.3189822      1.8912353
## 2                  -0.6374748      -0.5954636 -0.4874661     -0.5555854
## 3                   1.0884891       0.9836685  0.1243530      0.3474129
##   vehicle_offences criminal_damage_and_arson drug_offences
## 1       2.13532745                -0.7489274     1.5517727
## 2      -0.41051062                -0.5144797    -0.5016671
## 3       0.05480902                 1.0057362     0.3563553
##   public_order_offences
## 1            -0.3301983
## 2            -0.5681002
## 3             0.9746442
##
## Clustering vector:
##   [1] 2 3 2 2 2 3 2 3 3 2 2 2 3 2 2 3 3 3 3 3 3 3 2 3 3 2 2 2 2 2 3 3 2 2 2 2 2
##  [38] 2 2 2 2 2 3 2 3 3 2 3 3 2 2 3 2 3 2 2 2 3 3 2 3 2 2 3 3 3 2 2 2 2 3 2 2 3
##  [75] 3 2 3 3 2 2 2 3 3 2 2 2 3 3 2 2 2 2 2 2 2 3 2 1 2 2 3 3 3 3 3 3 3 3 3 3 3
## [112] 2 3 3 2 2 3 3 3 2 2 3 2 2 2 2 2 3 2 2 2 2 2 2 3 2 2 2 2 2 3 2 2 2 2 3 3 3
## [149] 3 3 3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 2 1 1 1 1 2 2
```

```
## [186] 3 2 2 3 2 3 3 2 2 2 3 2 2 3 2 2 3 3 3 3 3 2 3 3 2 2 3 2 3 2 2 3 2 2 2 2
## [223] 3 3 3 3 2 2 2 2 2 2 2 3 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 2
## [260] 2 2 2 2 2 2 2 2 3 3 3 3 2 2 2 2 2 2 2 3 2 2 2 2 2 2 3 3 3 2 3 2 3 3 3 3 3
## [297] 3 3 2 2
##
## Within cluster sum of squares by cluster:
## [1] 172.6303 354.3655 532.7698
##  (between_SS / total_SS =  55.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
english_km6
```

```
## K-means clustering with 6 clusters of sizes 19, 28, 94, 63, 65, 31
##
## Cluster means:
##   violence_against_the_person sexual_offences     robbery theft_offences
## 1                  -0.1841280      0.08242351  3.00235361      2.3548287
## 2                  -0.4544185     -0.68003466  0.46109421      0.6843945
## 3                  -0.9341749     -0.86430489 -0.58195201     -0.7495941
## 4                   0.8686339      0.75608197  0.02612084      0.2390736
## 5                  -0.1023543     -0.05324522 -0.46462294     -0.5217654
## 6                   1.8052801      1.75959220  0.42912938      0.8196828
##   vehicle_offences criminal_damage_and_arson drug_offences
## 1       2.26505348               -0.62360758     2.0200199
## 2       1.38329255               -0.68436267     0.2258272
## 3      -0.55951765               -0.74112244    -0.7431282
## 4       0.02667415                0.79652538     0.2036994
## 5      -0.58367759               -0.05972581    -0.1621858
## 6       0.22854916                1.75410602     0.7374041
##   public_order_offences
## 1           -0.12443704
## 2           -0.45807951
## 3           -0.89174117
## 4            0.60401120
## 5            0.06521674
## 6            1.82975502
##
## Clustering vector:
##   [1] 3 6 5 5 3 2 3 2 4 3 5 3 4 3 5 5 6 4 6 6 4 4 5 4 4 5 3 3 5 5 4 6 3 5 3 3 3
##  [38] 3 3 5 3 3 4 3 4 4 3 4 4 5 5 5 5 4 5 2 5 4 4 2 6 3 3 4 4 4 3 5 3 3 4 3 3 4
##  [75] 5 3 6 5 3 3 3 5 5 5 3 5 6 6 5 3 2 3 3 2 3 3 4 3 2 2 3 6 6 4 4 4 4 4 4 4 6
## [112] 3 4 6 5 5 4 6 6 3 3 4 5 5 5 3 5 6 3 3 3 5 3 5 6 5 5 5 3 5 5 6 3 3 3 3 4 6
## [149] 5 4 5 1 2 2 1 2 1 1 1 1 1 1 1 1 2 2 2 2 1 1 2 1 1 2 1 2 2 1 2 1 1 2 1 5 3
## [186] 6 5 3 6 3 4 4 5 5 5 4 3 3 4 3 3 4 6 4 4 6 4 5 4 6 5 2 4 3 4 3 5 4 5 5 5 3
## [223] 4 6 4 4 3 3 3 3 3 3 3 4 3 3 5 6 3 5 3 3 5 3 3 5 2 3 3 3 5 5 5 4 3 4 4 4 3
## [260] 3 3 3 3 5 3 3 5 4 4 4 4 3 3 3 3 2 3 2 4 5 3 3 5 3 5 5 6 2 2 4 2 4 4 6 6 4
## [297] 6 6 5 3
##
## Within cluster sum of squares by cluster:
## [1] 101.07162  66.26089 103.01447 192.73525  98.02709 163.42444
##  (between_SS / total_SS =  69.7 %)
```

```
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

- Do you find any particular solution preferable?

The variance explained in the three cluster solution was 55.7%, compared to 69.7% in the six cluster solution. The added complexity of three additional clusters may not be worth the tradeoff in variance explained.

Now, we need to describe the clusters.

- Get summary statistics for all of the clusters from the 3-cluster and 6- cluster solution that can be used to describe them. You can either: call the cluster centres table directly from the results, add the cluster membership to the original data and then use group_by and summarise, or modify the code for the visualisation we used above and re-purpose it for this visualisation (or even better, have a go at all three!)

```
# get the proportion in each cluster
proportions(table(english_km3$cluster))
```

```
##
##          1          2          3
## 0.09666667 0.55000000 0.35333333
```

```
proportions(table(english_km6$cluster))
```

```
##
##          1          2          3          4          5          6
## 0.06333333 0.09333333 0.31333333 0.21000000 0.21666667 0.10333333
```

```
english_crime_km_results <- english_crime %>%
  mutate(
    cluster_km3 = english_km3$cluster,
    cluster_km6 = english_km6$cluster,
  )

english_crime_km_results %>%
  group_by(cluster_km3) %>%
  summarise_if(is.numeric, mean) # %>% view()
```

```
## # A tibble: 3 x 10
##   cluster_km3 violence_against_the_person sexual_offences robbery theft_offences
##         <int>                       <dbl>           <dbl>   <dbl>          <dbl>
## 1           1                        25.8            2.40   2.82            37.6
## 2           2                        22.8            2.09   0.337           15.0
## 3           3                        40.8            3.35   0.878           23.3
## # i 5 more variables: vehicle_offences <dbl>, criminal_damage_and_arson <dbl>,
## #   drug_offences <dbl>, public_order_offences <dbl>, cluster_km6 <dbl>
```

```
english_crime_km_results %>%
  group_by(cluster_km6) %>%
  summarise_if(is.numeric, mean) # %>% view()
```

```
## # A tibble: 6 x 10
##   cluster_km6 violence_against_the_person sexual_offences robbery theft_offences
##         <int>                       <dbl>           <dbl>   <dbl>          <dbl>
```

```
## 1           1                        27.5            2.63   3.42            41.9
## 2           2                        24.7            2.02   1.18            26.5
## 3           3                        19.7            1.88   0.253           13.2
## 4           4                        38.5            3.17   0.791           22.3
## 5           5                        28.4            2.52   0.357           15.3
## 6           6                        48.3            3.97   1.15            27.7
## # i 5 more variables: vehicle_offences <dbl>, criminal_damage_and_arson <dbl>,
## #   drug_offences <dbl>, public_order_offences <dbl>, cluster_km3 <dbl>
```

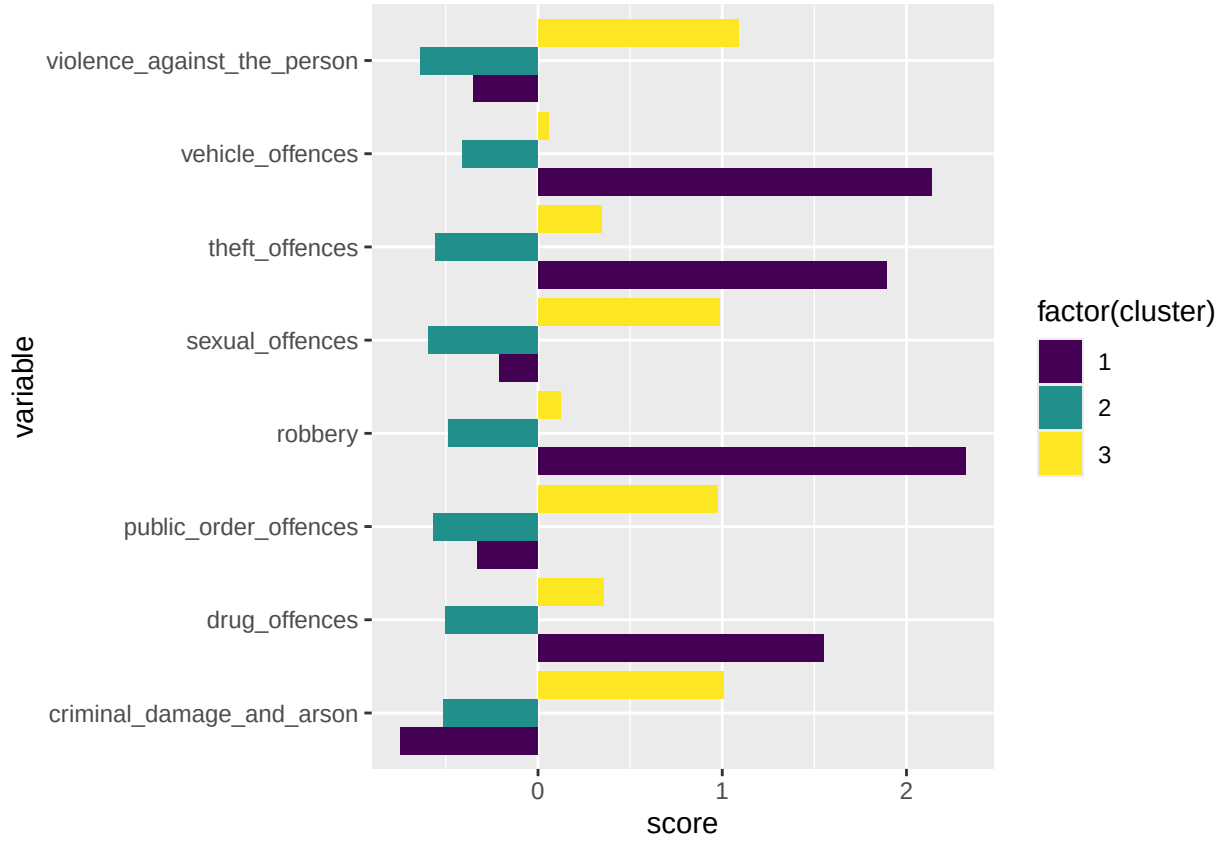
```
# Create a data frame from our results. Save our cluster in
# the variable "cluster"
k3_results <- as_tibble(english_km3$centers) %>%
  rowid_to_column(var = "cluster")
k3_results
```

```
## # A tibble: 3 x 9
##   cluster violence_against_the_person sexual_offences robbery theft_offences
##     <int>                       <dbl>           <dbl>   <dbl>          <dbl>
## 1       1                      -0.352          -0.207   2.32           1.89
## 2       2                      -0.637          -0.595  -0.487         -0.556
## 3       3                       1.09            0.984   0.124          0.347
## # i 4 more variables: vehicle_offences <dbl>, criminal_damage_and_arson <dbl>,
## #   drug_offences <dbl>, public_order_offences <dbl>
```

```
k3_results_long <- k3_results %>%
  # make all of our variable, but not our cluster ID, long format.
  # Make our variable column name "variable" and our value name "score"
  pivot_longer(-cluster, names_to = "variable", values_to = "score")

k3_results_long
```

```
## # A tibble: 24 x 3
##    cluster variable                      score
##      <int> <chr>                         <dbl>
##  1       1 violence_against_the_person -0.352
##  2       1 sexual_offences             -0.207
##  3       1 robbery                      2.32
##  4       1 theft_offences               1.89
##  5       1 vehicle_offences             2.14
##  6       1 criminal_damage_and_arson   -0.749
##  7       1 drug_offences                1.55
##  8       1 public_order_offences       -0.330
##  9       2 violence_against_the_person -0.637
## 10       2 sexual_offences             -0.595
## # i 14 more rows
```

```
k3_results_long %>%
  ggplot() +
  geom_col(aes(x = score, y = variable, fill = factor(cluster)),
           position = "dodge") +
  scale_fill_viridis_d() # make the scale more colourblind friendly
```




```
# Create a data frame from our results. Save our cluster in
# the variable "cluster"
k6_results <- as_tibble(english_km6$centers) %>%
  rowid_to_column(var = "cluster")
k6_results
```

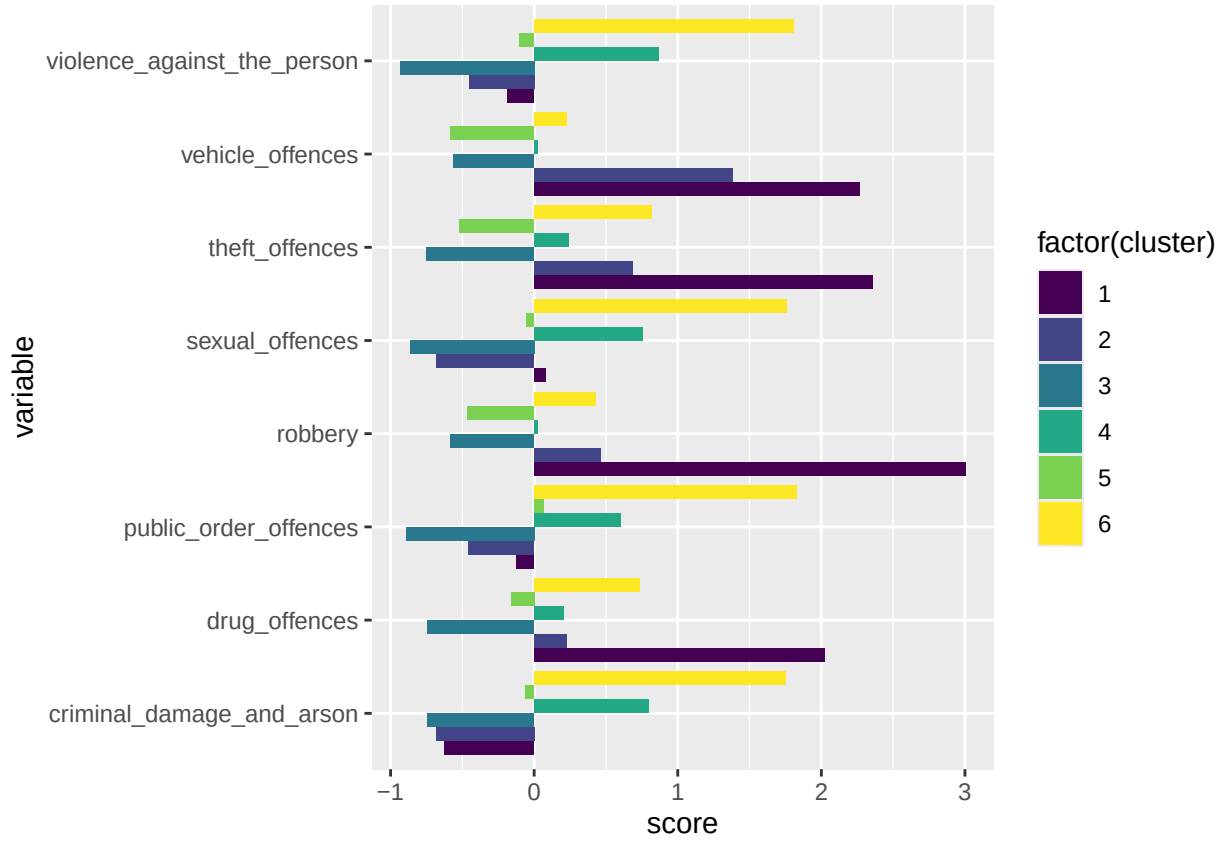
```
## # A tibble: 6 x 9
##   cluster violence_against_the_person sexual_offences robbery theft_offences
##     <int>                       <dbl>           <dbl>   <dbl>          <dbl>
## 1       1                      -0.184          0.0824  3.00           2.35
## 2       2                      -0.454         -0.680   0.461          0.684
## 3       3                      -0.934         -0.864  -0.582         -0.750
## 4       4                       0.869          0.756   0.0261         0.239
## 5       5                      -0.102         -0.0532 -0.465         -0.522
## 6       6                       1.81           1.76    0.429          0.820
## # i 4 more variables: vehicle_offences <dbl>, criminal_damage_and_arson <dbl>,
## #   drug_offences <dbl>, public_order_offences <dbl>
```

```
k6_results_long <- k6_results %>%
  # make all of our variable, but not our cluster ID, long format.
  # Make our variable column name "variable" and our value name "score"
  pivot_longer(-cluster, names_to = "variable", values_to = "score")

k6_results_long
```

```
## # A tibble: 48 x 3
##    cluster variable                    score
```

```
##       <int> <chr>                          <dbl>
##  1        1 violence_against_the_person -0.184
##  2        1 sexual_offences              0.0824
##  3        1 robbery                      3.00
##  4        1 theft_offences               2.35
##  5        1 vehicle_offences             2.27
##  6        1 criminal_damage_and_arson   -0.624
##  7        1 drug_offences                2.02
##  8        1 public_order_offences       -0.124
##  9        2 violence_against_the_person -0.454
## 10        2 sexual_offences             -0.680
## # i 38 more rows
```

```
k6_results_long %>%
  ggplot() +
  geom_col(aes(x = score, y = variable, fill = factor(cluster)),
           position = "dodge") +
  scale_fill_viridis_d() # make the scale more colourblind friendly
```



- Describe the clusters found in the 3-cluster and 6-cluster solution.

3 Cluster Solution

The three cluster solution can broadly be described as consisting of a low crime cluster (Cluster 2, green) and two high crime clusters (Cluster 1 and 3, purple and yellow), which differ somewhat in the nature of the crimes committed. For example, Cluster 3 has the highest rate of violence against people, sexual offences, criminal damage and arson, and public order offences than any other cluster, whereas Cluster 1 has far more vehicle, theft, robbery, and drug offences. There is therefore some evidence that high crime areas are split

somewhat between violent crimes and property crimes, unlike in the USA.

Only around 9.6% of areas fall within the first cluster of high property crime areas, while 35.3% of areas fall within the high violent crime areas of cluster 3. Most areas fall within the low crime cluster (55%).

6 Cluster Solution

Cluster 1 appears to be characterised by relatively high property, vehicle, and drug crime (high robbery, high theft, high vehicle offences), but with relatively typical levels of other forms of crime and low levels of criminal damage and arson. Cluster 2 is somewhat similar as it has similar elevated levels of property-related offences, but at a much lower level than cluster 1. Both of these clusters have relatively low rates of violent crime and public order offences, and cluster 2 has a particularly low rate of sexual offences. Together, clusters 1 and 2 make up 6.3% and 9.3% of the areas in the data.

Cluster 3 represents areas with low rates of all crimes. These areas make up around 31.3% of all of the areas in the data.

Cluster 4 represents areas with a mixture of violent crime and property crime at a fairly moderately high level. Cluster 4 has average or above average rates of all crimes and represents around 21% of the areas in the data.

Cluster 5 represents areas with atypically low levels of property crime — vehicle offences, theft offences, and robberies — but with fairly average levels of other forms of offences. Around 21.6% of areas were allocated to this cluster.

Cluster 6 represents high rates of crime across almost all areas, with a mixture of violent and property crime. It represents around 10.3% of the areas in the data.

- Is there evidence in either of these cluster solutions of areas being clustered into different types of criminal offences, or do clusters only reflect low or high crime as in the United States?

Yes, as opposed to the US data the three cluster solution appears to show the English community safety partnerships can be classified by prevalence of certain types of crime as well as overall crime.

Let's also see if we get similar results from a hierarchical cluster analysis. Before we can do that, we need to pick an appropriate dissimilarity/distance measure and linkage method. Your answers may start to differ from mine here — that's totally fine and to me expected! A lot of cluster analysis is subjective.

- What might be an appropriate distance measure for this data and why? (Hint: see slide 55)

Manhattan distance may be an appropriate measure of dissimilarity due to the large number of variables included in the data (8).

- What might be an appropriate linkage method for this data and why? (Hint: see slide 58)

Ward, Centroid, and Median linkage should be avoided because the distance matrix will be based on Manhattan and now Euclidean distance. Further, single linkage may not be appropriate as we do not assume a 'chain of command' type hierarchy in the data. As such, complete or average linkage may be most appropriate.

- Create a distance matrix for the English data using the dissimilarity measure of your choice.

```
english_crime_d <- daisy(english_crime_prepped, metric = "manhattan")
```

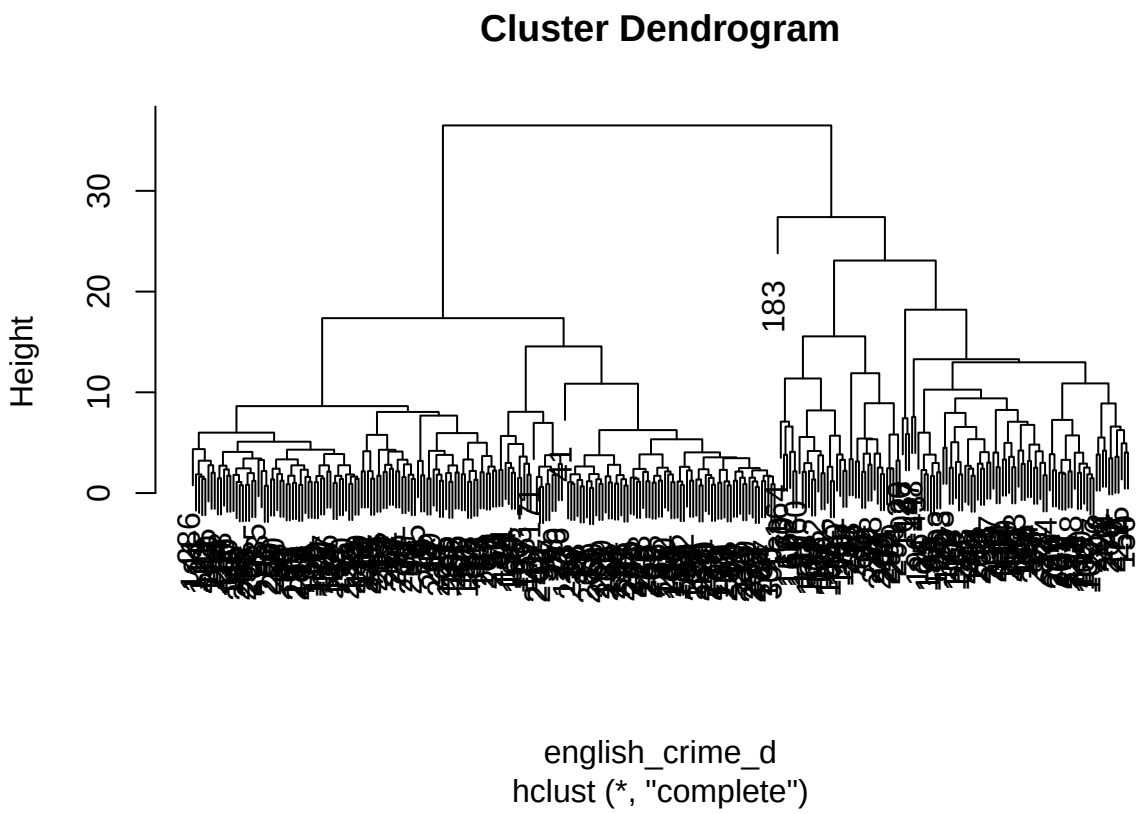
Now we can run the Hierarchical Cluster Analysis algorithm (or algorithms, if trying multiple) that we decided on about.

- Use the `hclust` function to cluster the data according to the linkage method that you chose. Don't forget to save the result to a new object.

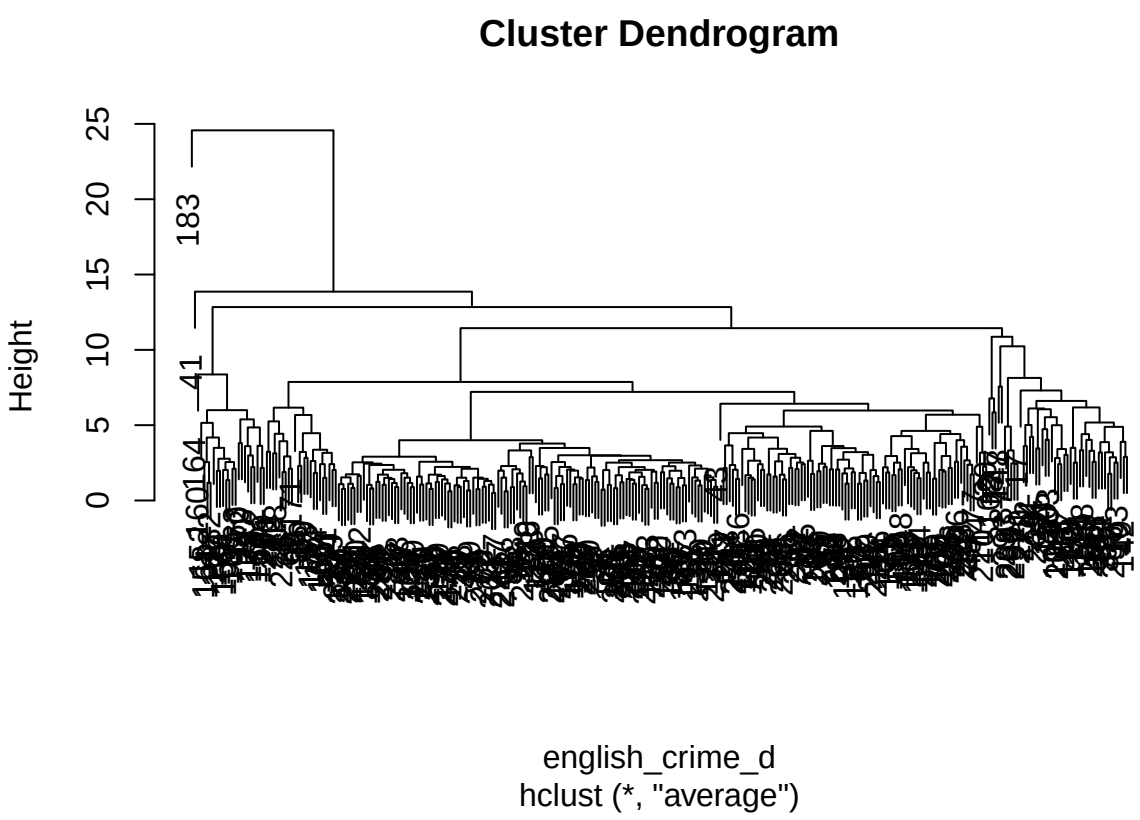
```r
set.seed(2021)
english_crime_comp <- hclust(english_crime_d, method = "complete")

set.seed(2021)
english_crime_avg <- hclust(english_crime_d, method = "average")
```

- Plot the results of your HCA with a dendrogram using the `plot` function

```r
plot(english_crime_comp) # Maybe 4, with one group having only one area (183)
```




```r
plot(english_crime_avg) # Average linkage doesn't seem to lead to any well-defined groups
```



- Come up with a sensible number of clusters from the above plots that you think the data could be clustered into. Write how many clusters you think there may be in the data based on each dendrogram (if more than one).

Four seems to be an appropriate number of clusters for both dendrograms, though there could be some equally valid larger numbers of clusters. The complete linkage dendrogram appears to have a singleton clade (Community 183), which may indicate outliers or potentially bad fitting clusters. Average linkage doesn't seem to lead to any well-defined groups.

Now we can cut our dendrogram into the number of clusters we believe we identified to explore how we might describe them.

- Use the `cutree` function to cut your dendrogram(s) into the chosen number of clusters.

```
english_hca_comp_results <- cutree(english_crime_comp, k = 4)
```

- Add your cluster solution(s) to the original data using the mutate function and the stored results above.

```
english_crime_hca_results <- english_crime %>%
  mutate(
    hca_comp = english_hca_comp_results,
  )
```

- Now produce some bivariate statistics showing how the crime rates differ by cluster. You can also produce a plot if you think it would be helpful.

```
hca_summary <- english_crime_hca_results %>%
  select(-1, -2) %>%
```

```
  group_by(hca_comp) %>%
  summarise_all(~mean(., na.rm = TRUE))# %>% view()

hca_summary
```

```
## # A tibble: 4 x 9
##   hca_comp violence_against_the_person sexual_offences robbery theft_offences
##      <int>                       <dbl>           <dbl>   <dbl>          <dbl>
## 1        1                        24.0            2.18   0.382           15.4
## 2        2                        43.0            3.54   0.872           24.4
## 3        3                        30.5            2.56   2.28            33.4
## 4        4                        32.0            3.63   6.21            79.6
## # i 4 more variables: vehicle_offences <dbl>, criminal_damage_and_arson <dbl>,
## #   drug_offences <dbl>, public_order_offences <dbl>
```

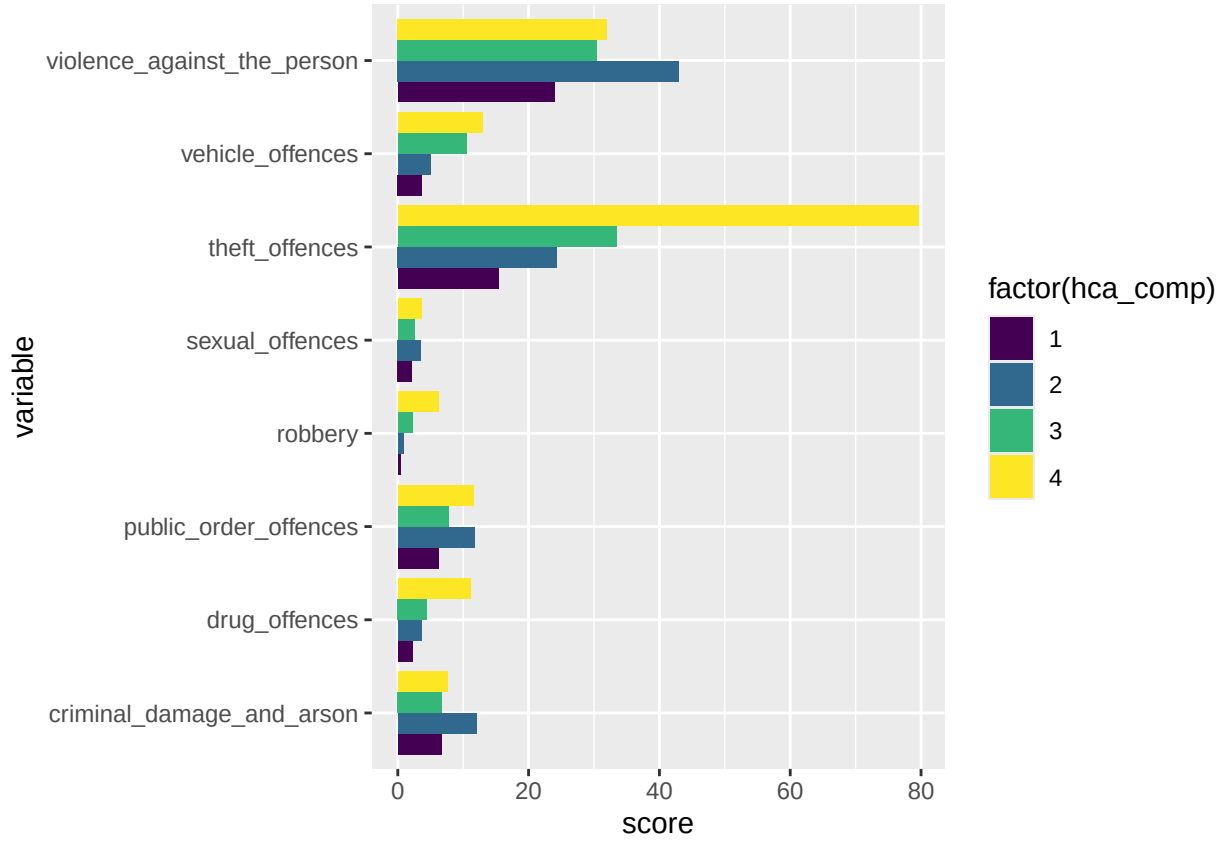
```
prop.table(table(english_crime_hca_results$hca_comp))
```

```
##
##           1           2           3           4
## 0.623333333 0.243333333 0.130000000 0.003333333
```

```
# Create a plot of the HCA results
hca_summary_long <- hca_summary %>%
  # make all of our variables, but not our cluster ID, long format.
  # Make our variable column name "variable" and our value name "score"
  pivot_longer(-hca_comp, names_to = "variable", values_to = "score")

hca_summary_long
```

```
## # A tibble: 32 x 3
##    hca_comp variable                     score
##       <int> <chr>                        <dbl>
##  1        1 violence_against_the_person 24.0
##  2        1 sexual_offences              2.18
##  3        1 robbery                      0.382
##  4        1 theft_offences              15.4
##  5        1 vehicle_offences             3.68
##  6        1 criminal_damage_and_arson    6.70
##  7        1 drug_offences                2.26
##  8        1 public_order_offences        6.23
##  9        2 violence_against_the_person 43.0
## 10        2 sexual_offences              3.54
## # i 22 more rows
```

```
hca_summary_long %>%
  ggplot() +
  geom_col(aes(x = score, y = variable, fill = factor(hca_comp)),
           position = "dodge") +
  scale_fill_viridis_d() # make the scale more colourblind friendly
```



- Describe the clusters found above (including from multiple linkage methods, if relevant). If possible, label the clusters found.

Complete Linkage Cluster (4)

Cluster 1: The lowest rates of all crimes. (62.3% of all areas)

Cluster 2: The highest rates of violence against the person, criminal damage and arson, and public order offences. (24.3% of all areas)

Cluster 3: Higher rates of property crime than cluster 2, and lower rates of personal/violent crime, crminal damage, and arson. Somewhat elevated rates of drug offences. (13% of all areas)

Cluster 4: This singleton cluster is made up of an area with extremely high levels of theft and drug offences, indicating it is an outlier. This makes sense as inspecting the data shows the area is Westminster.

- Do these clusters differ from the clusters found using k-means? Which do you prefer as a typology of crime in English community and safety partnerships and why?

The HCA analysis was very similar to the k-means analysis and both have substantively the same interpretations, but it did pick out one particular area which was an outlier, which could be helpful.

Week 10 Challenge

- Practice using some of the skills we learned in Week 3 (bivariate data visualisation and statistics) to further illustrate the differences and similarities between your favoured cluster analysis of the English crime data. This might make it easier to see the characteristics of clusters than using the means of all variables; it might also show you some interesting differences in terms of variation.