# k-means clustering



## Example:

**Process the dataset**

- Verify results of clustering by plotting them with ggplot2

```
>ggplot(data = df2, aes(x = Sepal.Length, y = Sepal.Width)) + geom_point(aes(col =
        as.factor(clusterResult$cluster)))# Plot to see how Sepal.Length and Sepal.Width data points
have been distributed in clusters
>ggplot(data = df2, aes(x = Sepal.Length, y = Sepal.Width)) + geom_point(aes(col = df1))# Plot to see
how Sepal.Length and Sepal.Width data points have been distributed originally as per "class" attribute
in dataset
> ggplot(data = df2, aes(x = Petal.Length, y = Petal.Width)) + geom_point(aes(col =
        as.factor(clusterResult$cluster)))# Plot to see how Petal.Length and Petal.Width data points have
been distributed in clusters
> ggplot(data = df2, aes(x = Petal.Length, y = Petal.Width)) + geom_point(aes(col = df1))# Plot to see
how Petal.Length and Petal.Width data points have been distributed originally as per "class" attribute
in dataset
```

- **as.factor()** is necessary for the cluster because the variable is otherwise interpreted as a continuous one

# k-means clustering



## How to determine the optimal number of clusters :

- **Choosing an appropriate *k***

**Elbow method**

- Minimize the total intra-cluster variation
- Known as total within-cluster variation or total within-cluster sum of square

- ***Algorithm:***
  - Compute *k-means* clustering for different values
  - For each k, calculate the total **within-cluster sum of square** (**wss**)
  - Plot the curve of *wss* according to the number of clusters k.
  - The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

**wss measures the compactness of the clustering.**

# k-means clustering



## Example:

**Choosing an appropriate *k* with Elbow method**

```
# Create an empty vector
# Write a for-loop for different k-values  from 1 to 20
# Apply k-means clustering algorithm for each k-value
# Calculate the sum of wss-values for each k-value with withinss-variable
# Save them in your vector  and plot them.
```

# k-means clustering



**Example:**

**Choosing an appropriate *k* with Elbow method**

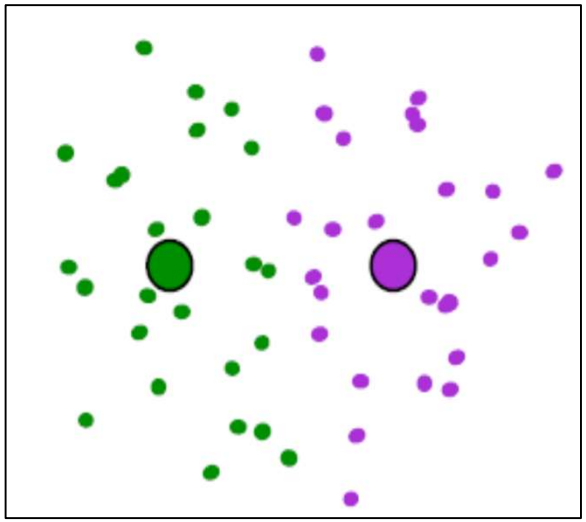
```
# create an empty vector
> wss=c()
# Write a  for-loop for different k-values  from 1 to 20
# Apply k-means clustering algorithm for each k-value
# Calculate the sum of wss-values for each k-value with withinss-variable & save them in your vector
> for(k in 1:20){
      clusters=kmeans(df2, centers=k)
      wss[k]=sum(clusters$withinss) # sum of within-cluster sum of squares
   }
# Plot the wss values with plot function and ggplot2 library
>plot(1:length(wss), wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")
 >plotValues=data.frame(k=1:length(wss), values=wss)
 >p = ggplot(plotValues, aes(x=k, y=values))
 >p = p + geom_point() + geom_line()
 >p
```

# k-means clustering

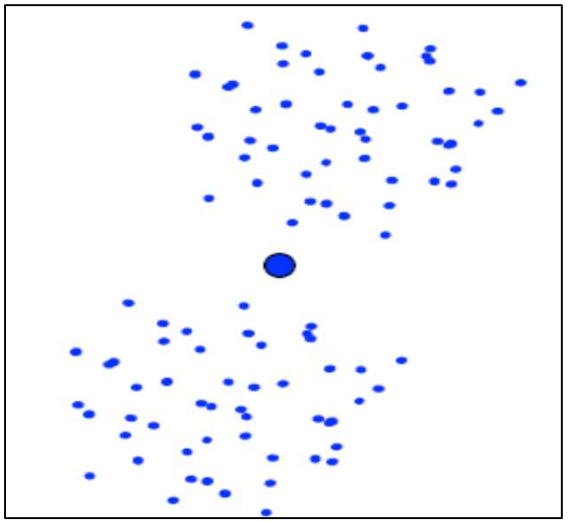


## Limitiations:

- **K-means getting stuck**
  - **A local optimum**: Possible solution is to run k-means multiple times with different initial clusters



**Would it be better to have one cluster here**



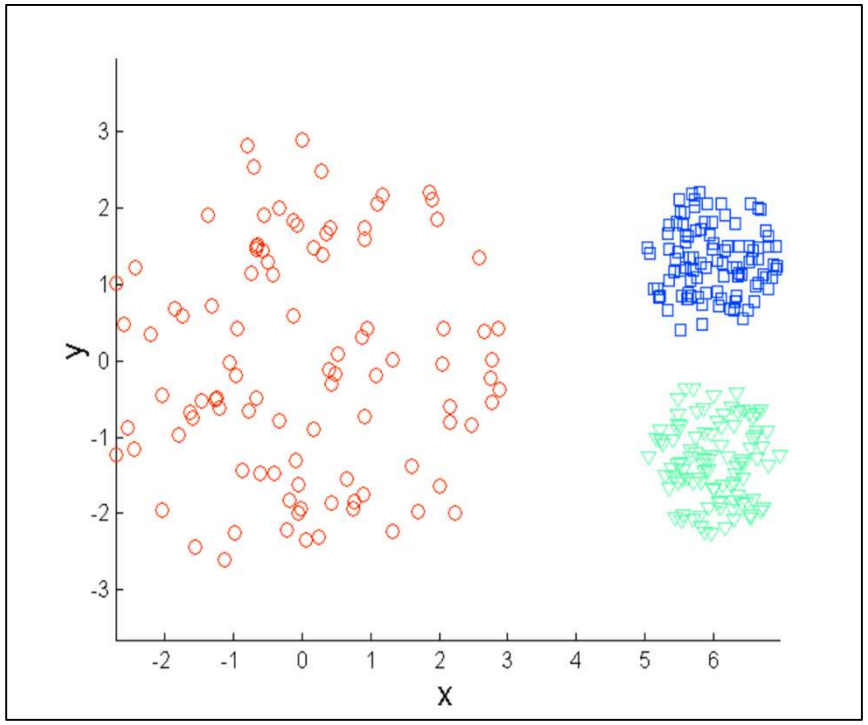
**… to have two cluster here**

# k-means clustering

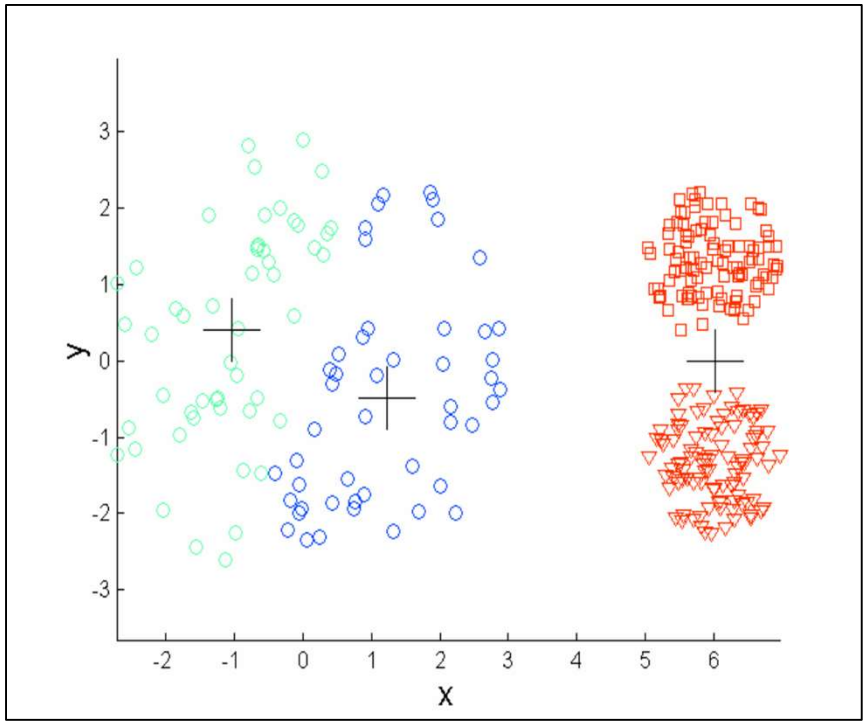


## Limitiations:

- **K-means getting stuck**
  - **Different density**




Original points



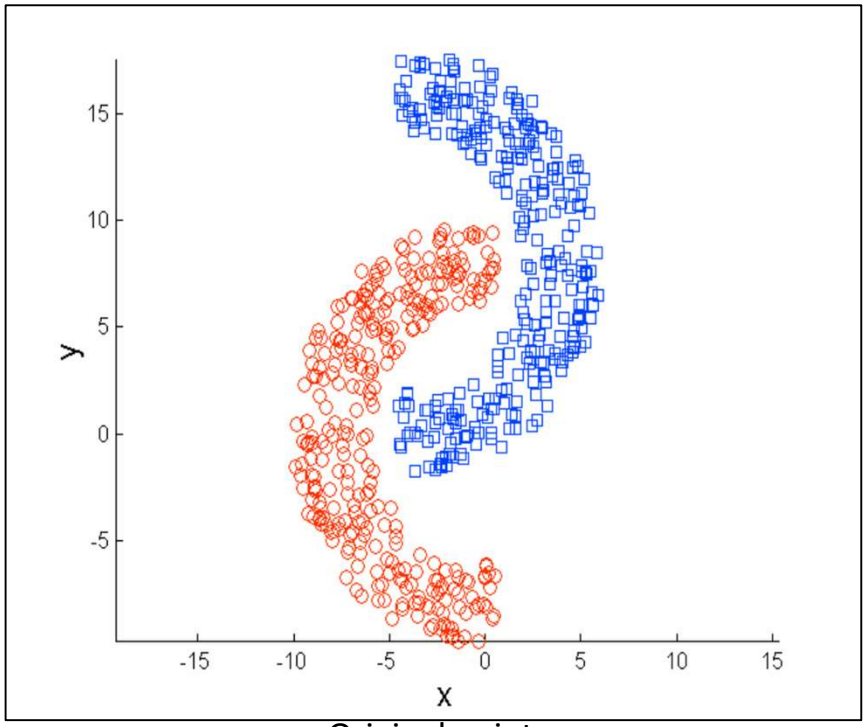

… k-means (3 Clusters)

# k-means clustering

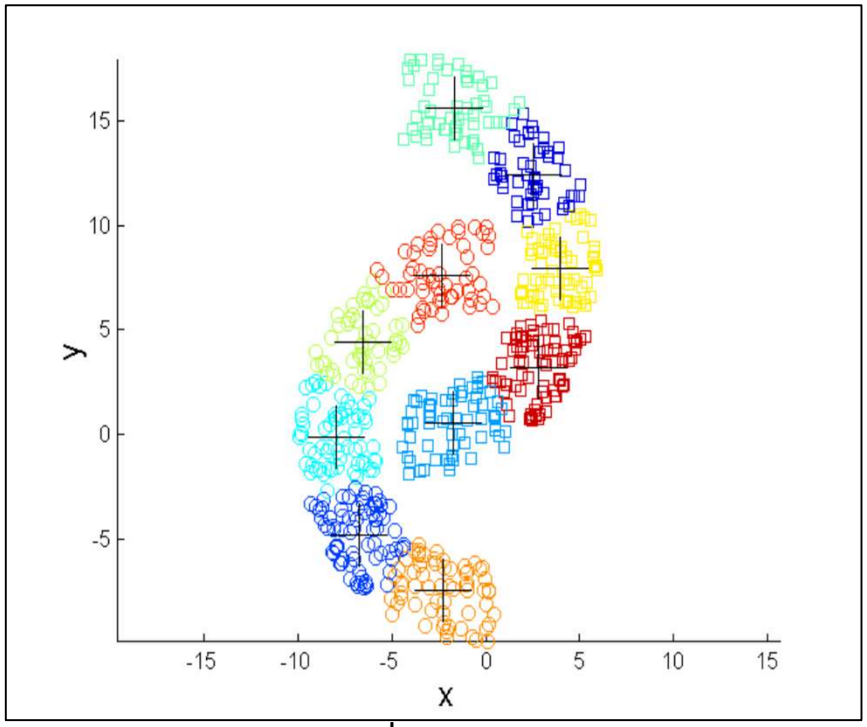


## Limitiations:

- **K-means getting stuck**
  - **Non-globular shapes**




Original points



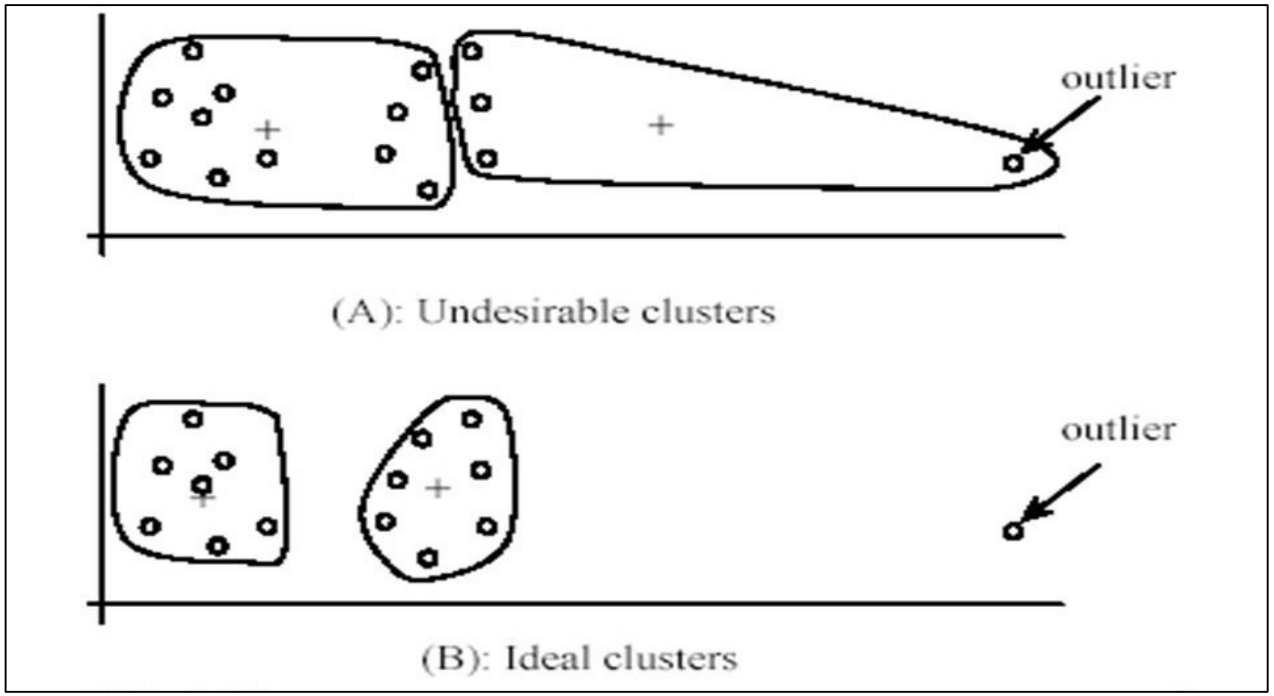

… k-means

# k-means clustering



## Limitiations:

- **K-means getting stuck**
  - **Effect of outliers**




… k-means is sensitive to Outliers

lww.quora.com: k-means weeakness

# Hierarchical Clustering Algorithm

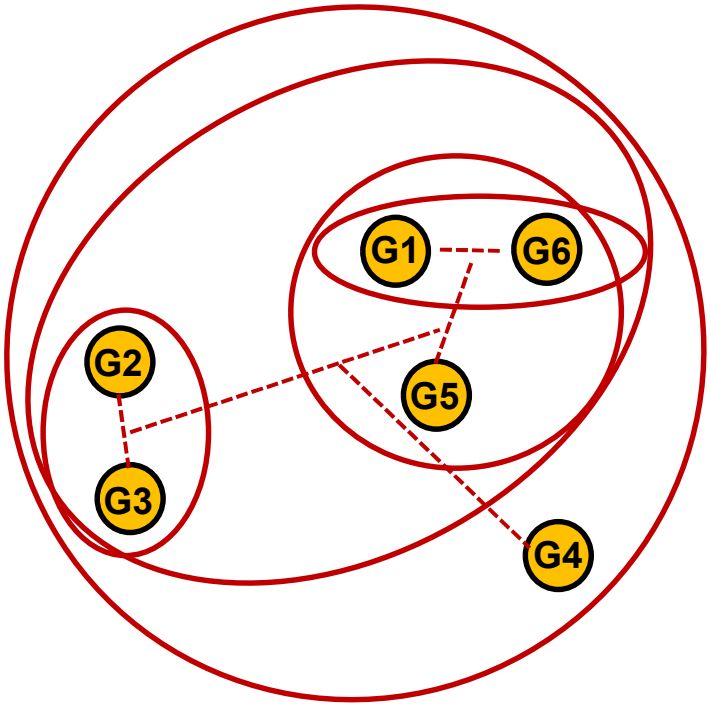


- **Hierarchical clustering**
  - an alternative approach which builds a **hierarchy** from the bottom-up based on a **distance matrix**
- The specification of the number of clusters is not required
- The algorithm works as follows:
  - Put each data point in its own cluster.

# Hierarchical Clustering Algorithm



- **Hierarchical clustering**
  - an alternative approach which builds a **hierarchy** from the bottom-up based on a **distance matrix**
- The specification of the number of clusters is not required
- The algorithm works as follows:
  - Put each data point in its own cluster.
  - Identify the closest two clusters and combine them into one cluster.
  - Repeat the above step till all the data points are in a single cluster.

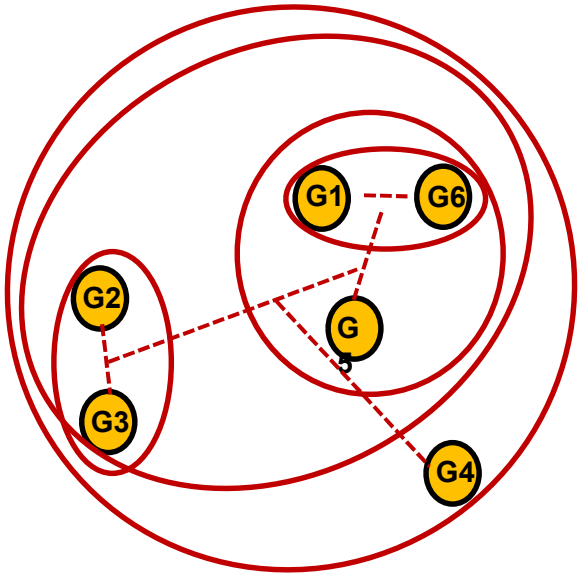



https://www.datascienceplus.com

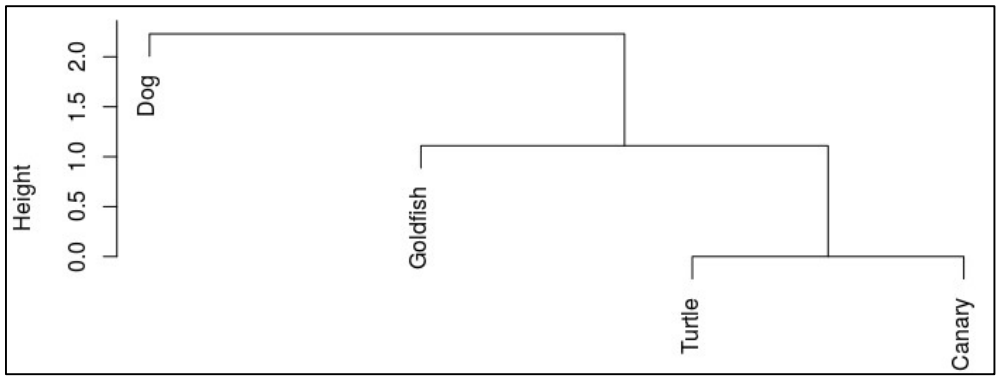
# Hierarchical Clustering Algorithm



- **Hierarchical clustering**
- Once this is done, it is usually represented by a **dendrogram** like structure




| | Dog | Turtle | Canary | Goldfish |
|---|---|---|---|---|
| Dog | 0 | 1.73 | 1.73 | 2.0 |
| Turtle | 1.73 | 0 | 0 | 1.0 |
| Canary | 1.73 | 0 | 0 | 1.0 |
| Goldfish | 2.0 | 1.0 | 1.0 | 0 |




https://www.datascienceplus.com

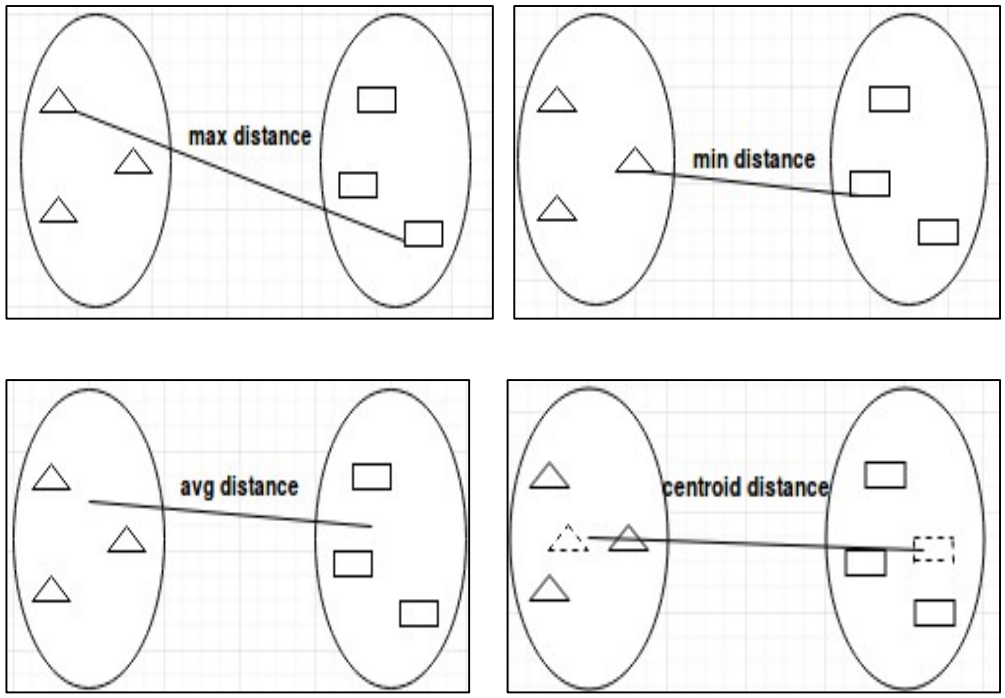
# Hierarchical Clustering Algorithm

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

## Identification of how close two clusters are:

- **Linkage methods**
  - measure the distance between clusters in order to decide the rules for clustering

1. **Complete-linkage:** Find the maximum possible distance between points belonging to two different clusters
2. **Single linkage clustering:** Find the minimum possible distance between points belonging to two different clusters.
3. **Mean linkage clustering (UPGMA):** Find all possible pairwise distances for points belonging to two different clusters and then calculate the average.
4. **Centroid linkage clustering:** Find the centroid of each cluster and calculate the distance between centroids of two clusters.




https://www.datacamp.com

# Hierarchical Clustering Algorithm

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

**Example:** Hierarchical clustering based on Complete-Linkage (use maximum distance as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster

| | A | B | C | D | E |
|---|---|---|---|---|---|
| **A** | **0** | **12** | **6** | **4** | **8** |
| **B** | | **0** | **10** | **10** | **18** |
| **C** | | | **0** | **6** | **2** |
| **D** | | | | **0** | **14** |
| **E** | | | | | **0** |

A B C D E

# Hierarchical Clustering Algorithm

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

**Example:** Hierarchical clustering based on Complete-Linkage (use maximum distance as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 0 | 12 | 6 | 4 | 8 |
| B | | 0 | 10 | 10 | 18 |
| C | | | 0 | 6 | 2 |
| D | | | | 0 | 14 |
| E | | | | | 0 |

A B C D E

# Hierarchical Clustering Algorithm

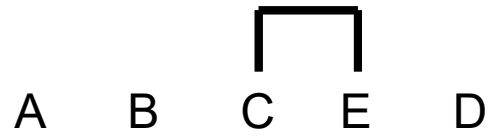


**Example:** Hierarchical clustering based on Complete-Linkage (use maximum distance as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them

| | A | B | C | D | E |
|---|---|---|---|---|---|
| **A** | **0** | **12** | **6** | **4** | **8** |
| **B** | | **0** | **10** | **10** | **18** |
| **C** | | | **0** | **6** | **2** |
| **D** | | | | **0** | **14** |
| **E** | | | | | **0** |



# Hierarchical Clustering Algorithm

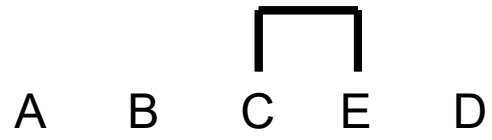


**Example:** Hierarchical clustering based on Complete-Linkage (use maximum distance as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. Update the distance matrix with distances between the new cluster and the other ones

| | A | B | C | D | E |
|---|---|---|---|---|---|
| **A** | **0** | **12** | **6** | **4** | **8** |
| **B** | | **0** | **10** | **10** | **18** |
| **C** | | | **0** | **6** | **2** |
| **D** | | | | **0** | **14** |
| **E** | | | | | **0** |



# Hierarchical Clustering Algorithm

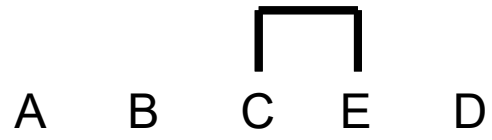


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. Update the distance matrix with distances between the new cluster and the other ones
   - **Example**: Original distances : **A-C = 6** **A-E = 8**
     New distance is the maximum of old distances : **A-(C,E) = 8**

| | A | B | C | D | E | C,E |
|---|---|---|---|---|---|---|
| **A** | **0** | **12** | **6** | **4** | **8** | **8** |
| **B** | | **0** | **10** | **10** | **18** | |
| **C** | | | **0** | **6** | **2** | |
| **D** | | | | **0** | **14** | |
| **E** | | | | | **0** | |
| **C,E** | | | | | | **0** |



# Hierarchical Clustering Algorithm

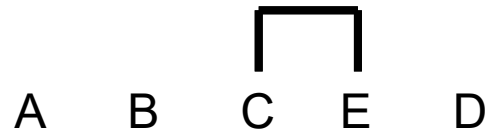


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. Update the distance matrix with distances between the new cluster and the other ones
   - **Example**: Original distances : **B-C = 10 B-E = 18**
     New distance is the maximum of old distances : **B-(C,E) = 18**

| | A | B | C | D | E | C,E |
|---|---|---|---|---|---|---|
| **A** | **0** | **12** | **6** | **4** | **8** | **8** |
| **B** | | **0** | **10** | **10** | **18** | **18** |
| **C** | | | **0** | **6** | **2** | |
| **D** | | | | **0** | **14** | |
| **E** | | | | | **0** | |
| **C,E** | | | | | | **0** |



# Hierarchical Clustering Algorithm

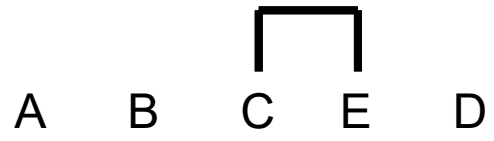


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. Update the distance matrix with distances between the new cluster and the other ones
   - **Example**: Original distances : **D-C = 6** **D-E = 14**
     New distance is the maximum of old distances : **D-(C,E) = 14**

| | A | B | C | D | E | C,E |
|---|---|---|---|---|---|---|
| **A** | **0** | **12** | **6** | **4** | **8** | **8** |
| **B** | | **0** | **10** | **10** | **18** | **18** |
| **C** | | | **0** | **6** | **2** | |
| **D** | | | | **0** | **14** | **14** |
| **E** | | | | | **0** | |
| **C,E** | | | | | | **0** |



# Hierarchical Clustering Algorithm

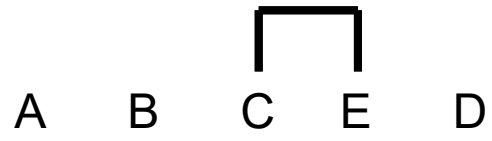
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. Update the distance matrix with distances between the new cluster and the other ones
   - After the calculation of the new distances is done the clustered elements can be removed from the table

| | A | B | C | D | E | C,E |
|---|---|---|---|---|---|---|
| A | 0 | 12 | 6 | 4 | 8 | 8 |
| B | | 0 | 10 | 10 | 18 | 18 |
| C | | | 0 | 9 | 7 | |
| D | | | | 0 | 14 | 14 |
| E | | | | | 0 | |
| C,E | | | | | | 0 |



# Hierarchical Clustering Algorithm

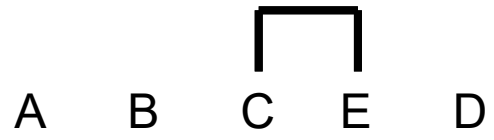


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. Update the distance matrix with distances between the new cluster and the other ones
   - After the calculation of the new distances is done the clustered elements can be removed from the table

| | A | B | D | C,E |
|---|---|---|---|---|
| **A** | **0** | **12** | **4** | **8** |
| **B** | | **0** | **10** | **18** |
| **D** | | | **0** | **14** |
| **C,E** | | | | **0** |



# Hierarchical Clustering Algorithm

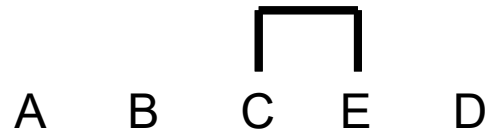


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. Update the distance matrix with distances between the new cluster and the other ones
4. If not all elements are in one cluster go back to step 2

| | A | B | D | C,E |
|---|---|---|---|---|
| **A** | **0** | **12** | **4** | **8** |
| **B** | | **0** | **10** | **18** |
| **D** | | | **0** | **14** |
| **C,E** | | | | **0** |



# Hierarchical Clustering Algorithm

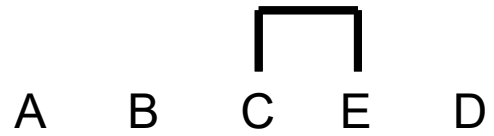
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. **Identify the two clusters with the smallest distance and combine them**
3. Update the distance matrix with distances between the new cluster and the other ones
4. If not all elements are in one cluster go back to step 2

| | A | B | D | C,E |
|---|---|---|---|---|
| **A** | **0** | **12** | **4** | **8** |
| **B** | | **0** | **10** | **18** |
| **D** | | | **0** | **14** |
| **C,E** | | | | **0** |



# Hierarchical Clustering Algorithm

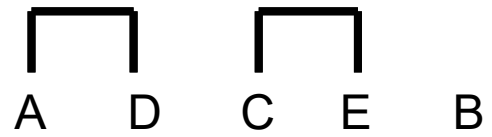


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. **Identify the two clusters with the smallest distance and combine them**
3. Update the distance matrix with distances between the new cluster and the other ones
4. If not all elements are in one cluster go back to step 2

| | A | B | D | C,E |
|---|---|---|---|---|
| **A** | **0** | **12** | **4** | **8** |
| **B** | | **0** | **10** | **18** |
| **D** | | | **0** | **14** |
| **C,E** | | | | **0** |



# Hierarchical Clustering Algorithm

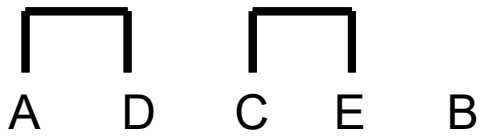


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. **Update the distance matrix with distances between the new cluster and the other ones**
4. If not all elements are in one cluster go back to step 2

| | A | B | D | C,E | A,D |
|---|---|---|---|---|---|
| **A** | **0** | **12** | **4** | **8** | |
| **B** | | **0** | **10** | **18** | |
| **D** | | | **0** | **14** | |
| **C,E** | | | | **0** | |
| **A,D** | | | | | **0** |



# Hierarchical Clustering Algorithm

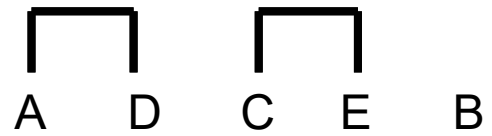


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. **Update the distance matrix with distances between the new cluster and the other ones**
   - **Example**: Original distances **: B-A = 12 B-D = 10**
     New distance is the maximum of old distances : **B-(A,D) = 12**

| | A | B | D | C,E | A,D |
|---|---|---|---|---|---|
| A | 0 | 12 | 4 | 8 | |
| B | | 0 | 10 | 18 | 12 |
| D | | | 0 | 14 | |
| C,E | | | | 0 | |
| A,D | | | | | 0 |



# Hierarchical Clustering Algorithm

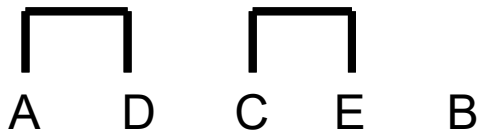


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. **Update the distance matrix with distances between the new cluster and the other ones**
   - **Example**: Original distances : **(C,E)-A = 8 , (C,E)-D = 14**
     New distance is the maximum of old distances : **(C,E)-(A,D) = 14**

| | A | B | D | C,E | A,D |
|---|---|---|---|---|---|
| A | 0 | 12 | 4 | 8 | |
| B | | 0 | 10 | 18 | 12 |
| D | | | 0 | 14 | |
| C,E | | | | 0 | 14 |
| A,D | | | | | 0 |



# Hierarchical Clustering Algorithm

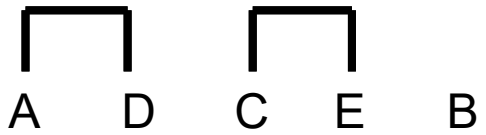


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. Update the distance matrix with distances between the new cluster and the other ones
4. **If not all elements are in one cluster go back to step 2**

| | **B** | **C,E** | **A,D** |
|---|---|---|---|
| **B** | **0** | **18** | **12** |
| **C,E** | | **0** | **14** |
| **A,D** | | | **0** |



# Hierarchical Clustering Algorithm

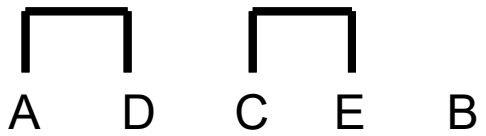


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. **Identify the two clusters with the smallest distance and combine them**
3. Update the distance matrix with distances between the new cluster and the other ones
4. If not all elements are in one cluster go back to step 2

| | B | C,E | A,D |
|---|---|---|---|
| B | 0 | 18 | 12 |
| C,E | | 0 | 14 |
| A,D | | | 0 |



# Hierarchical Clustering Algorithm

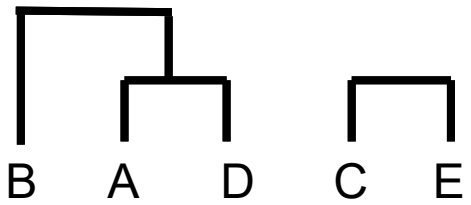


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. **Identify the two clusters with the smallest distance and combine them**
3. Update the distance matrix with distances between the new cluster and the other ones
4. If not all elements are in one cluster go back to step 2

| | B | C,E | A,D |
|---|---|---|---|
| **B** | **0** | **18** | **12** |
| **C,E** | | **0** | **14** |
| **A,D** | | | **0** |



# Hierarchical Clustering Algorithm

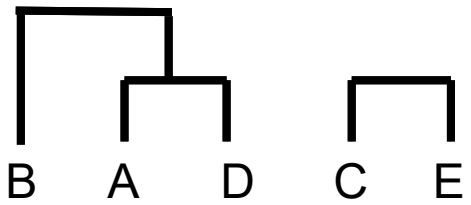


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. **Update the distance matrix with distances between the new cluster and the other ones**
4. If not all elements are in one cluster go back to step 2

| | **B** | **C,E** | **A,D** | **A,D,B** |
|---|---|---|---|---|
| **B** | **0** | **18** | **12** | |
| **C,E** | | **0** | **14** | |
| **A,D** | | | **0** | |
| **A,D,B** | | | | **0** |



# Hierarchical Clustering Algorithm

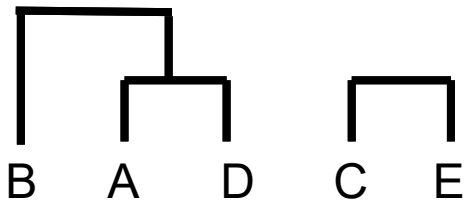


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. **Update the distance matrix with distances between the new cluster and the other ones**
   - **Example**: Original distances : **(C,E)-(A,D) = 14** **(C,E)-B = 18**
     New distance is the maximum of old distances : **(C,E)-(A,D,B) = 18**

| | B | C,E | A,D | A,D,B |
|---|---|---|---|---|
| **B** | **0** | **18** | **12** | |
| **C,E** | | **0** | **14** | **18** |
| **A,D** | | | **0** | |
| **A,D,B** | | | | **0** |



# Hierarchical Clustering Algorithm

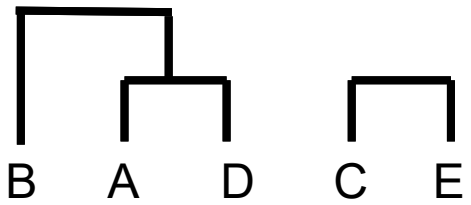


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. Update the distance matrix with distances between the new cluster and the other ones
4. **If not all elements are in one cluster go back to step 2**

| | **C,E** | **A,D,B** |
|---|---|---|
| **C,E** | **0** | **18** |
| **A,D,B** | | **0** |



# Hierarchical Clustering Algorithm

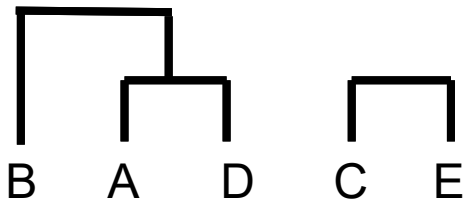


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. **Identify the two clusters with the smallest distance and combine them**
3. Update the distance matrix with distances between the new cluster and the other ones
4. If not all elements are in one cluster go back to step 2

| | **C,E** | **A,D,B** |
|---|---|---|
| **C,E** | **0** | **18** |
| **A,D,B** | | **0** |



# Hierarchical Clustering Algorithm

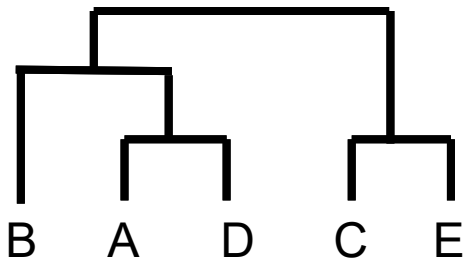


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. **Identify the two clusters with the smallest distance and combine them**
3. Update the distance matrix with distances between the new cluster and the other ones
4. If not all elements are in one cluster go back to step 2

| | C,E | A,D,B |
|---|---|---|
| C,E | 0 | 18 |
| A,D,B | | 0 |



# Hierarchical Clustering Algorithm

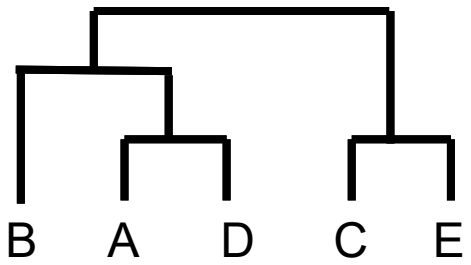


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. **Update the distance matrix with distances between the new cluster and the other ones**
   - We can skip this step since no other cluster remains

| | **C,E** | **A,D,B** |
|---|---|---|
| **C,E** | **0** | **18** |
| **A,D,B** | | **0** |



# Hierarchical Clustering Algorithm

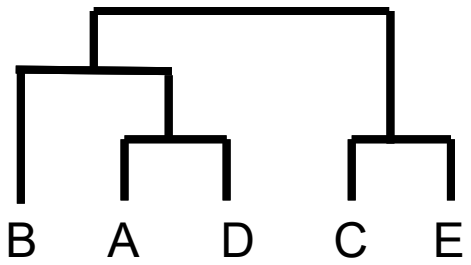


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. **Update the distance matrix with distances between the new cluster and the other ones**
   - We can skip this step since no other cluster remains

| | **A,D,B,C,E** |
|---|---|
| **A,D,B,C,E** | **0** |



# Hierarchical Clustering Algorithm

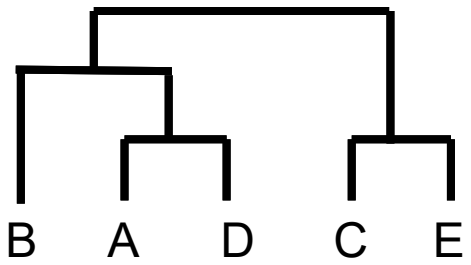


**Example:** Hierarchical clustering based on Complete-Linkage (use **maximum distance** as distance between clusters)

- Given a matrix containing the distances between 5 objects

1. First consider each object as a single cluster
2. Identify the two clusters with the smallest distance and combine them
3. Update the distance matrix with distances between the new cluster and the other ones
4. **If not all elements are in one cluster go back to step 2 ⇨ Done**

| | **A,D,B,C,E** |
|---|---|
| **A,D,B,C,E** | **0** |

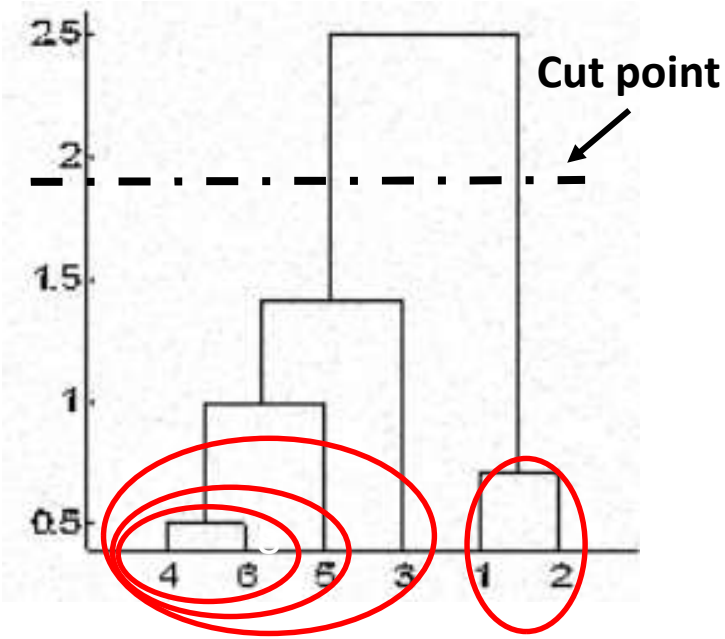


# Hierarchical Clustering Algorithm

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

## Dendrograms:

- Tree like diagrams




1. **Cluster 1:** Points **4** and **6**
2. **Cluster 2:** Points **1** and **2**
3. **Point 5** was merged in the same **cluster 1** followed by point **3** resulting in two clusters
4. **Final step:** two clusters are merged into a single cluster

### When should we stop merging the clusters?

- depends on your knowledge about the data
- you can leverage the results from the dendrogram to approximate the number of clusters
    - Select the maximum distance up and down without intersecting the merging point

https://www.datacamp.com

# Hierarchical Clustering Algorithm



## Example:

Import the data file called **"animalData.csv**" into the variable **mydata**

```
> mydata= read.table("animalData.csv", header=TRUE, sep="," , stringsAsFactors = TRUE)
# Check the dimensions of the data
# View statistical  summary of dataset
# View the complete data
```

**Process the dataset**

- Divide **mydata** in two data frames as **df1** and **df2**
  - **df1** contains the class attribute "animal"
  - **df2** contains the remaining information
- Create a distance matrix for all entries (function in R: **dist**)
- Apply the **hierarchical** clustering algorithm (function in R: **hclust**)
- Verify results of clustering by plotting them

# Hierarchical Clustering Algorithm



## Example:

**Process the dataset**

- Divide **mydata** in two data frames as **df1** and **df2**

  *> df1=mydata[, "animal"]*

  *> df2= subset(mydata, select=-c(animal))* # or mydata[, 1:85]

- Create a distance matrix for all entries (?dist for help)

  *> distMat = dist(df2)*

- Apply **hierarchical** clustering algorithm (?hclust for help)

  *> clusterResult= hclust(distMat, method = "average")*

  #apply hierarchical clustering algorithm with mean linkage clustering

- Verify results of clustering by plotting them

# Hierarchical Clustering Algorithm



## Example:

**Process the dataset**

– Verify results of clustering by plotting them with plot()

*>plot(clusterResult, labels = df1*) #Using labels we can add the class attribute “animal” to the plot

# Hierarchical Clustering Algorithm

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

## Example:

**Process the dataset**

- Verify results of clustering by plotting them with ggplot2
- This is not directly easily possible
- We need another package **ggdendro**

```
>install.packages("ggdendro")
>library(ggdendro)
>ggdendrogram(clusterResult)
#For adding labels we need an extra step at the beginning before the clustering
>rownames(df2) = df1 #We assign the animal names as row names to df2
>distMat = dist(df2)
>clusterResult = hclust(distMat, method = "average")
>ggdendrogram(clusterResult)
```

# Clustering Exercises

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

## Exercise: beansData

Import the data file called **"DryBeanData_smallData.csv**" into the variable **bean_data**

**Process the dataset**

- Check the dimensions of the dataset and what type of data it contains
- Check for missing values and replace them with the median
- Divide **bean_data** in two data frames as **df1** and **df2**
  - **df1** contains the attribute "Class"
  - **df2** contains the remaining information
- Normalize the values in **df2** between 0 and 1 using our own function
- Create a subset of 100 randomly selected observations
- Perform **k-means clustering**
- Find the optimal k by using the elbow method. Verify results of clustering by plotting them for the optimal k
- Apply **hierarchical clustering** for a subset of 100 randomly selected observations
- Visualize your results with a phylogenetic dendrogram