

# ISA 401: Business Intelligence & Data Visualization

## 26: A Short Introduction to Clustering

Fadel M. Megahed, PhD

Raymond E. Glos Professor in Business  
Farmer School of Business  
Miami University

 @FadelMegahed

 fmegahed

 fmegahed@miamioh.edu

 Automated Scheduler for Office Hours

Fall 2025

# A Recap of What we Learned Last Class

- Describe the goals & functions of data mining
- Understand the statistical limits on data mining
- Describe the data mining process
- What is “frequent itemsets” & the application of this concept
- Explain how and why “association rules” are constructed
- Use  to populate both concepts

# Kahoot: A Recap of Phase 3 of Class So Far

Let us go to Kahoot and compete for a \$10  Starbucks gift card. To evaluate your understanding of the material, please answer the questions correctly and as quickly as possible to get the most points.

# Learning Objectives for Today's Class

- Describe the different steps of the  $k$ -means algorithm
- Cluster using  $k$ -means (by hand)
- Cluster using  $k$ -means (software)
  - 
  - Tableau

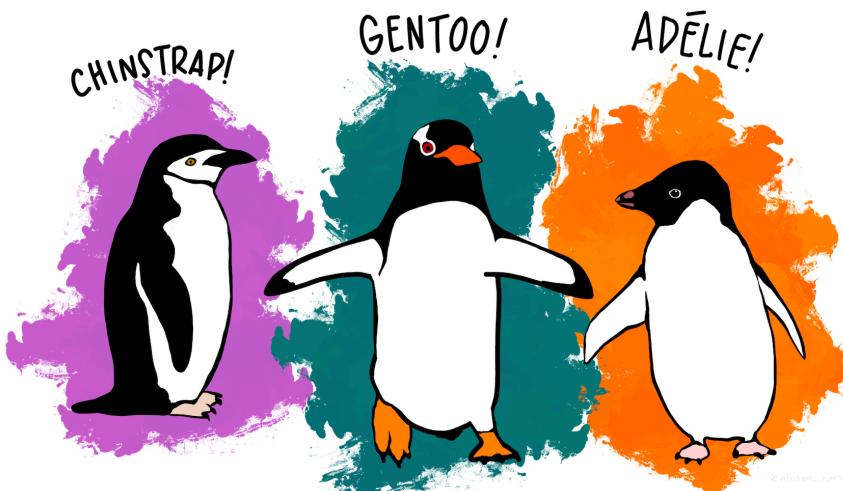
# An Overview of Clustering Techniques

# The Problem of Clustering

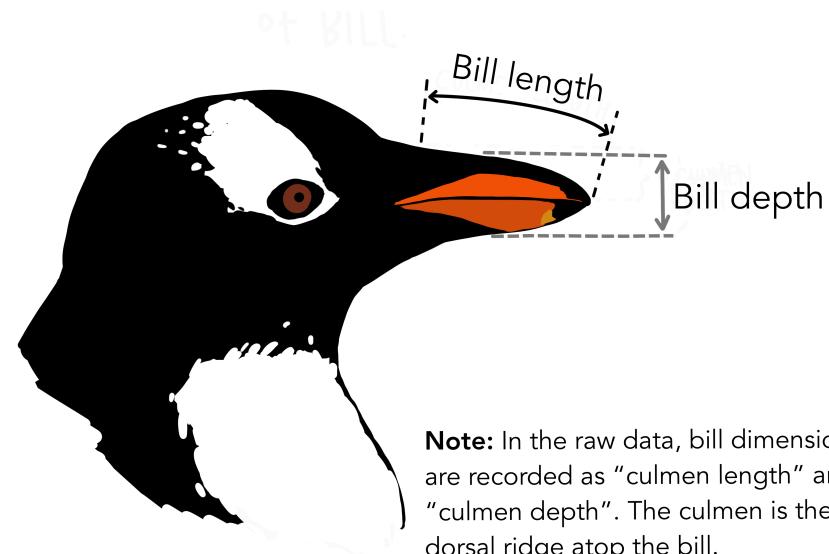
- Given a **set of (high-dimensional) observations**, with a notion of **distance** between observations, **group the observations** into **some number of clusters**, so that:
  - Members of a cluster are close/similar to each other
  - Members of different clusters are dissimilar
- **Usually:**
  - The observations are in a high-dimensional space
  - Similarity is defined using a distance measure, e.g.,
    - Euclidean, Cosine, Jaccard, edit distance, etc.

# Clustering in 2D Space

## Meet the Palmer penguins



## Anatomical description of the dataset:



**Note:** In the raw data, bill dimensions are recorded as "culmen length" and "culmen depth". The culmen is the dorsal ridge atop the bill.

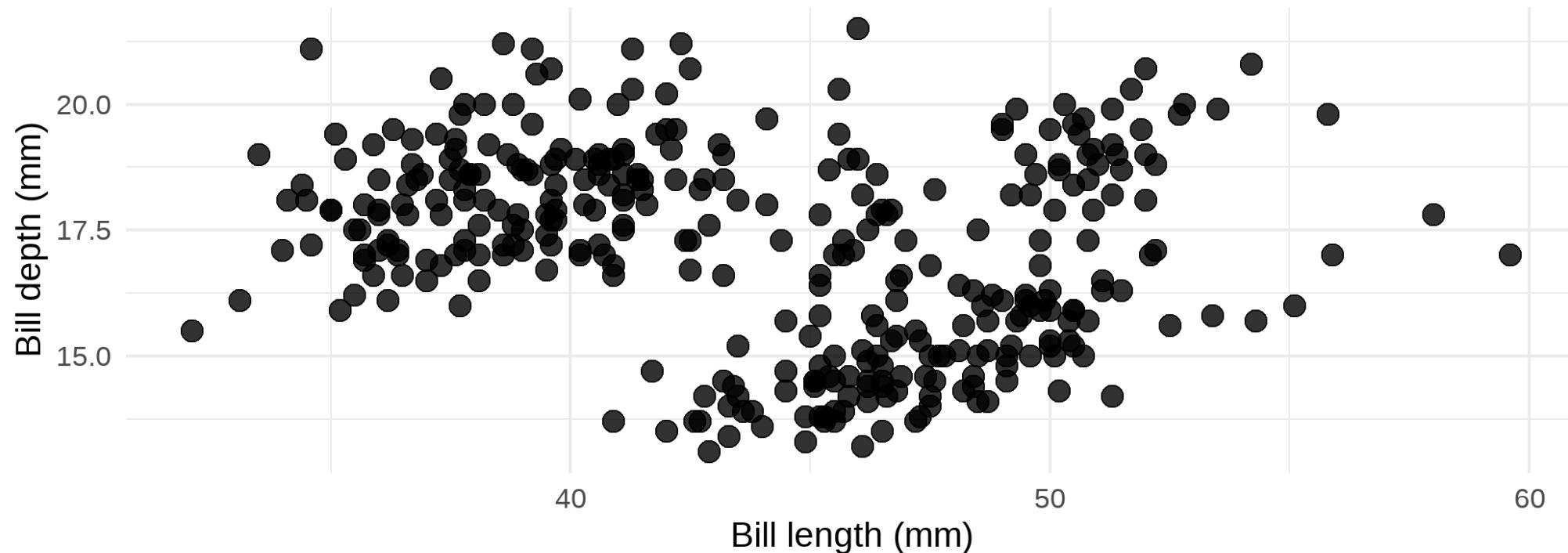
# Clustering in 2D Space: Formulation

- Given a **set of observations (each containing bill length and depth)**, with a notion of **Euclidean distance** between observations, **group the observations into 3 clusters**, so that:
  - Members of a cluster are close/similar to each other
  - Members of different clusters are dissimilar
- Note we are assuming that we did not have a "label/type" for each penguin.

# Clustering in 2D Space: Raw Data

## Penguin bill dimensions

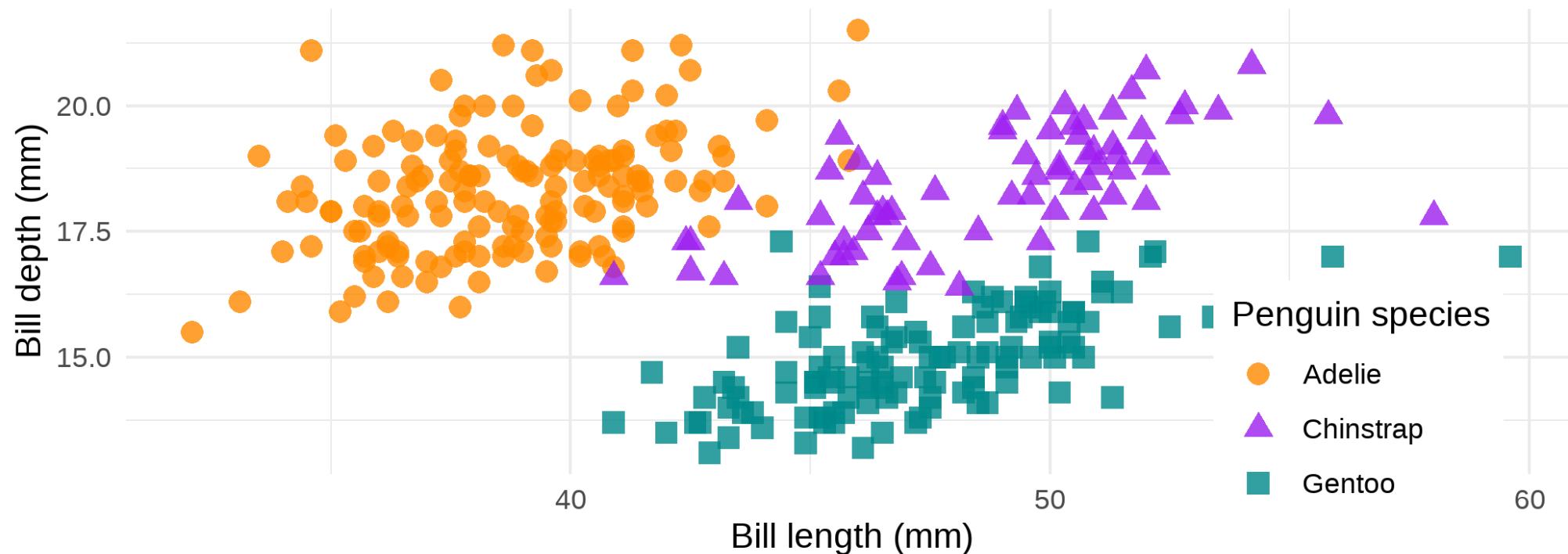
Bill length and depth for Adelie, Chinstrap and Gentoo Penguins at Palmer Station LTER



# Clustering in 2D Space: Labeled Raw Data

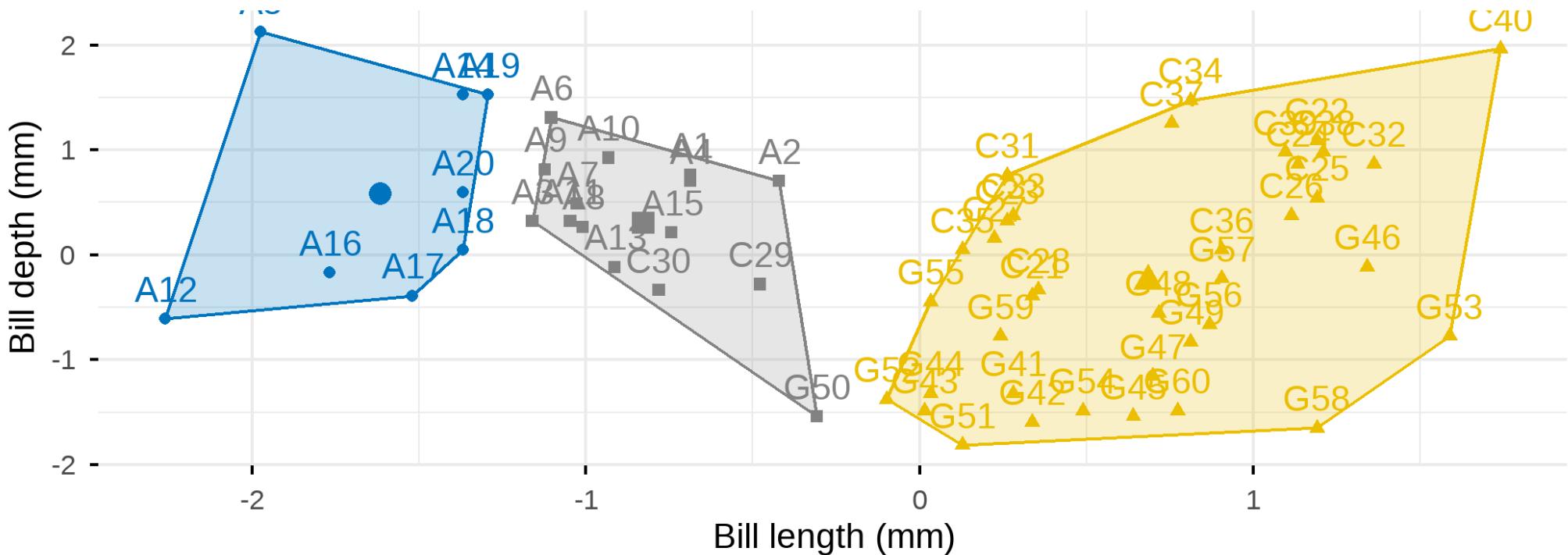
## Penguin bill dimensions

Bill length and depth for Adelie, Chinstrap and Gentoo Penguins at Palmer Station LTER



# Clustering in 2D Space: Clustering Results

Clustering of a sample of 60 penguins into three groups  
Using only bill length and depth



# Comments on the 2D Clustering Problem

Even though the 2D Space clustering problem is the easiest problem to "solve" since we can benefit by plotting the data, **clustering is hard**.

## Some important questions:

- With all the variables being numerical, we often assume **Euclidean distance**. This can be problematic when:
  - variables have significantly different scales
  - we are including information that is not pertinent to grouping
- How do you determine the number of clusters ( $k$ )?
- How to represent a cluster of many points?
- How do we determine the "nearness" of clusters?

# An Overview of Clustering Methods

| Categories               | Abb. name         | Volume          |                              |                     | Variety                   |                | Velocity complexity of Algorithm                                       | Other criterion Input Parameter |
|--------------------------|-------------------|-----------------|------------------------------|---------------------|---------------------------|----------------|--|---------------------------------|
|                          |                   | Size of Dataset | Handling High Dimensionality | Handling Noisy Data | Type of Dataset           | Clusters Shape |  |                                 |
| Partitional algorithms   | K-Means [25]      | Large           | No                           | No                  | Numerical                 | Non-convex     | $O(nkd)$   | 1                               |
|                          | K-modes [19]      | Large           | Yes                          | No                  | Categorical               | Non-convex     | $O(n)$   | 1                               |
|                          | K-medoids [33]    | Small           | Yes                          | Yes                 | Categorical               | Non-convex     | $O(n^2dt)$   | 1                               |
|                          | PAM [31]          | Small           | No                           | No                  | Numerical                 | Non-convex     | $O(k(n-k)^2)$  | 1                               |
|                          | CLARA [23]        | Large           | No                           | No                  | Numerical                 | Non-convex     | $O(k(40+k)^2+k(n-k))$  | 1                               |
|                          | CLARANS [32]      | Large           | No                           | No                  | Numerical                 | Non-convex     | $O(kn^2)$  | 2                               |
|                          | FCM [6]           | Large           | No                           | No                  | Numerical                 | Non-convex     | $O(n)$   | 1                               |
| Hierarchical algorithms  | BIRCH [40]        | Large           | No                           | No                  | Numerical                 | Non-convex     | $O(n)$   | 2                               |
|                          | CURE [14]         | Large           | Yes                          | Yes                 | Numerical                 | Arbitrary      | $O(n^2 \log n)$  | 2                               |
|                          | ROCK [15]         | Large           | No                           | No                  | Categorical and Numerical | Arbitrary      | $O(n^2+nmmma+n^2\log n)$   | 1                               |
|                          | Chameleon [22]    | Large           | Yes                          | No                  | All type of data          | Arbitrary      | $O(n^2)$   | 3                               |
|                          | ECHIDNA [26]      | Large           | No                           | No                  | Multivariate Data         | Non-convex     | $O(N * B(1 + \log_B m))$   | 2                               |
| Density-based algorithms | DBSCAN [9]        | Large           | No                           | No                  | Numerical                 | Arbitrary      | $O(n\log n)$ If a spatial index is used<br>Otherwise, it is $O(n^2)$ . | 2                               |
|                          | OPTICS [5]        | Large           | No                           | Yes                 | Numerical                 | Arbitrary      | $O(n\log n)$   | 2                               |
|                          | DBCLASD [39]      | Large           | No                           | Yes                 | Numerical                 | Arbitrary      | $O(3n^2)$  | No                              |
|                          | DENCLUE [17]      | Large           | Yes                          | Yes                 | Numerical                 | Arbitrary      | $O(\log D )$   | 2                               |
| Grid- based algorithms   | Wave-Cluster [34] | Large           | No                           | Yes                 | Special data              | Arbitrary      | $O(n)$   | 3                               |
|                          | STING [37]        | Large           | No                           | Yes                 | Special data              | Arbitrary      | $O(k)$   | 1                               |
|                          | CLIQUE [21]       | Large           | Yes                          | No                  | Numerical                 | Arbitrary      | $O(Ck + mk)$   | 2                               |
|                          | OptiGrid [18]     | Large           | Yes                          | Yes                 | Special data              | Arbitrary      | Between $O(nd)$ and $O(nd \log n)$                                     | 3                               |
| Model- based algorithms  | EM [8]            | Large           | Yes                          | No                  | Special data              | Non-convex     | $O(knp)$   | 3                               |
|                          | COBWEB [12]       | Small           | No                           | No                  | Numerical                 | Non-convex     | $O(n^2)$   | 1                               |
|                          | CLASSIT [13]      | Small           | No                           | No                  | Numerical                 | Non-convex     | $O(n^2)$   | 1                               |
|                          | SOMs [24]         | Small           | Yes                          | No                  | Multivariate Data         | Non-convex     | $O(n^2m)$  | 2                               |

# $k$ -means Algorithm

# General Idea

The  $k$ -means algorithm clusters data by trying to separate samples in  $n$  groups of equal variance, minimizing a criterion known as the **inertia** or **within-cluster sum-of-squares** (see below). This algorithm requires the **number of clusters to be specified**.

$$\sum_{i=0}^n \min_{\mu_j \in C} (\|x_i - \mu_j\|^2)$$

**Inertia is a measure of how internally coherent clusters are; however, it suffers from various drawbacks:**

- Inertia makes the assumption that clusters are convex and isotropic, which is not always the case. It responds poorly to elongated clusters, or manifolds with irregular shapes.
- Inertia is not a normalized metric: we just know that lower values are better and zero is optimal. But in very high-dimensional spaces, Euclidean distances tend to become inflated.

# The Steps of the $k$ -means Algorithm

In basic terms, the algorithm has three steps.

- Step 0 chooses the initial centroids, with the most basic method being to choose  $k$  samples from the dataset  $X$ . After initialization,  $k$ -means consists of looping between the remaining two steps.
- Step 1 assigns each sample to its nearest centroid.
- Step 2 creates new centroids by taking the mean value of all of the samples assigned to each previous centroid. The difference between the old and the new centroids are computed.

**The algorithm repeats these last two steps the centroids do not move significantly.**

# Out-Of-Class Activity: Finish by Friday

Use the  $k$ -means algorithm to cluster the following observations. Use  $k = 2$  and Euclidean distance. **Use this handout to go through the  $k$ -means algorithm implementation (by hand).**

| Observation | X1  | X2  |
|-------------|-----|-----|
| 1           | 1.0 | 1.0 |
| 2           | 1.5 | 2.0 |
| 3           | 3.0 | 4.0 |
| 4           | 5.0 | 7.0 |
| 5           | 3.5 | 5.0 |
| 6           | 4.5 | 5.0 |
| 7           | 3.5 | 4.5 |

# Practical Issues with $k$ -means Clustering

| Data  | Prep | k-means (k=3) | Optimal k | Viz Clusters |
|---|------|---------------|-----------|--------------|
| <pre>penguins_tbl = palmerpenguins::penguins # our data for today<br/>penguins_tbl # printing it out</pre>  |      |               |           |              |
| <pre>## # A tibble: 344 × 8<br/>##   species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g<br/>##   &lt;fct&gt;   &lt;fct&gt;        &lt;dbl&gt;        &lt;dbl&gt;          &lt;int&gt;        &lt;int&gt;<br/>## 1 Adelie   Torgersen      39.1       18.7           181        3750<br/>## 2 Adelie   Torgersen      39.5       17.4           186        3800<br/>## 3 Adelie   Torgersen      40.3        18            195        3250<br/>## 4 Adelie   Torgersen       NA         NA             NA          NA<br/>## 5 Adelie   Torgersen      36.7       19.3           193        3450<br/>## 6 Adelie   Torgersen      39.3       20.6           190        3650<br/>## 7 Adelie   Torgersen      38.9       17.8           181        3625<br/>## 8 Adelie   Torgersen      39.2       19.6           195        4675<br/>## 9 Adelie   Torgersen      34.1       18.1           193        3475<br/>## 10 Adelie  Torgersen       42         20.2           190        4250<br/>## # i 334 more rows<br/>## # i 2 more variables: sex &lt;fct&gt;, year &lt;int&gt;</pre> |      |               |           |              |

# Summary of Practical Issues

- Rescale numeric data prior to  $k$ -means implementation. The scaling can be:
  - a z-transformation similar to what we did in the example
  - a 0-1 scaling
  - converting count data into percentage or counts per a certain number of the population
  - etc.
- Use more than one metric to determine  $k$  when using  $k$ -means clustering
- Your cluster solution is not the end result, you will need to:
  - visualize it in appropriate way (simple representation as in the previous slide, [spatially](#), [time-based](#), etc.)
  - Attempt to explain the cluster membership using an appropriate binomial/multinomial model (e.g., see [this analysis](#))

# *k*-means in Tableau

Let us use Tableau to implement the *k*-means clustering implementation on the 60 sample observations from the penguins dataset as shown in Slide 11 of this presentation.

# Recap

# Summary of Main Points

- Describe the different steps of the  $k$ -means algorithm
- Cluster using  $k$ -means (by hand)
- Cluster using  $k$ -means (software)
  -  R
  - Tableau

