STAT 209
Clustering

August 10, 2021

Colin Reimer Dawson

Outline

Machine Learning

> *"[Machine Learning is a] field of study that gives computers the ability to learn without being explicitly programmed."*
> *— Arthur Samuel*

Learning relies on finding *patterns* and *relationships* in data

Machine Learning vs. Statistics

- Statistics is about finding patterns and relationships too. What's the difference?
- Not sure there really is one, fundamentally.
- Existence of two names is mainly historical: Statistics as a field grew from math, ML as a field grew from CS (which had previously grown from math).
- Accordingly, statisticians tend to emphasize data-generating models and inferences from data about those models, whereas (many non-statistician) ML people tend to think in terms of optimization algorithms instead

Types of Learning

- Supervised Learning: Learning to make predictions when you have many examples of “correct answers”
  - Classification: answer is a category / label
  - Regression: answer is a number
- Unsupervised Learning: Finding structure in unlabeled data
- Reinforcement Learning: Finding actions that maximize long-run reward

Some Unsupervised Learning Problems

1. Clustering: Divide observations into groups
    - Recommender systems: segment customers into "types" based on their product preferences; then recommend products based on what other customers of your "type" have bought
    - Gene function prediction: Group genes that carry out similar functions; hypothesize new properties by generalizing within a cluster.
    - Cognitive science: How should new concepts/labels be generalized?
2. Association Mining: Discover that X predicts Y
    - Recommender systems: People who like X tend to like Y
    - Medicine: Characteristic X associated with risk of Y
3. Segmentation/chunking: Divide spatial/temporal data into chunks/regions
    - Speech recognition
    - Image segmentation/understanding
    - Finding functional components in social/neural networks

Outline

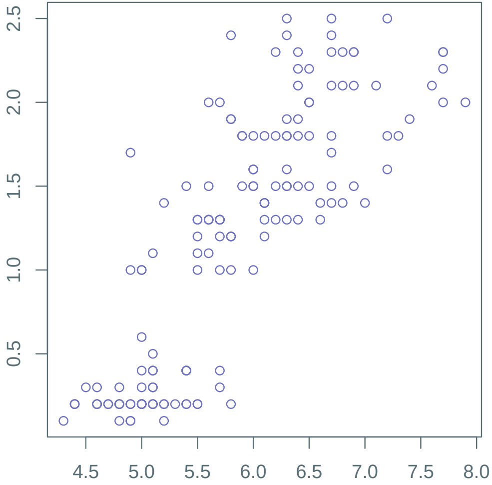
A Brief Detour into Machine Learning

Clustering

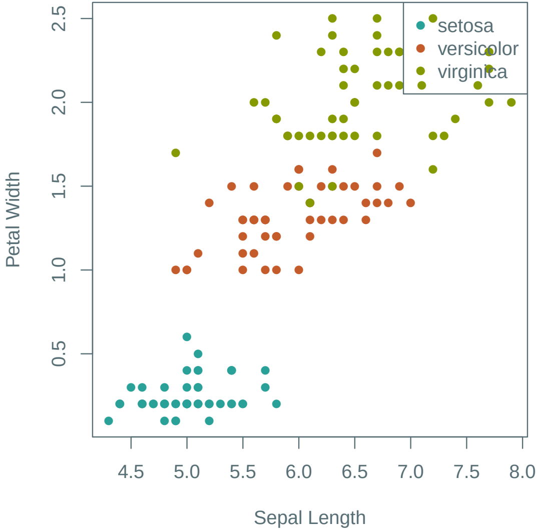
K-means

Hierarchical Clustering

Example: What are the Clusters?



The Answer: Species of Irises (Flowers)

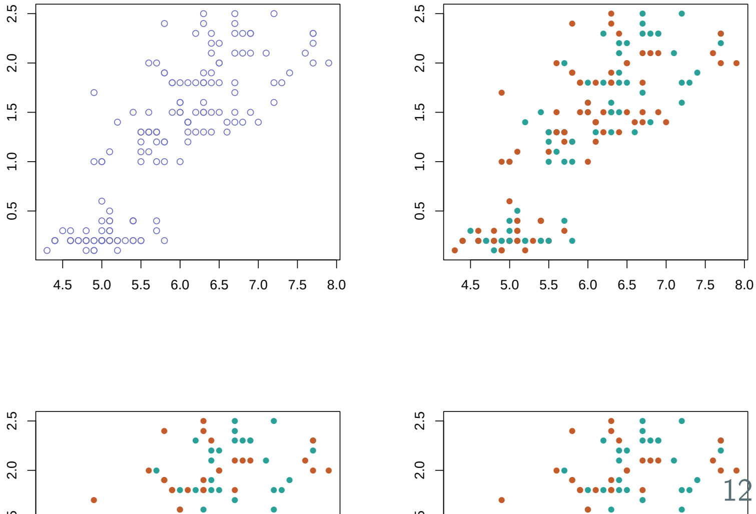


Outline

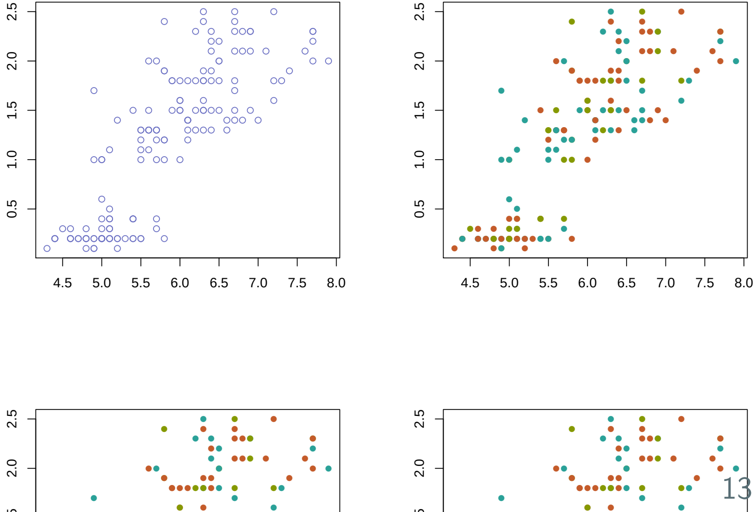
The K-means Algorithm

1. Initialize points to K clusters (randomly?)
2. While not converged:
   (a) Find centers (means) of each current cluster
   (b) Reassign points to closest center
   (c) If no change, stop; else iterate

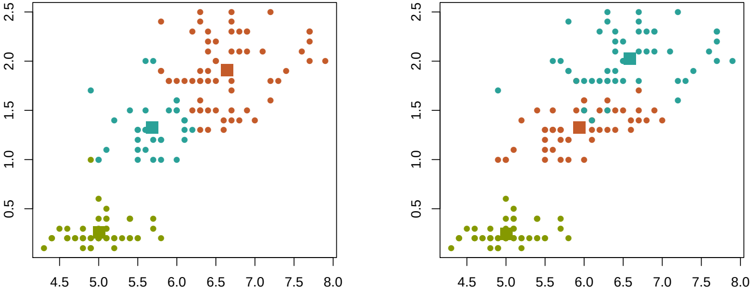
K-means on Iris Data ($K = 2$)



K-means on Iris Data ($K = 3$)

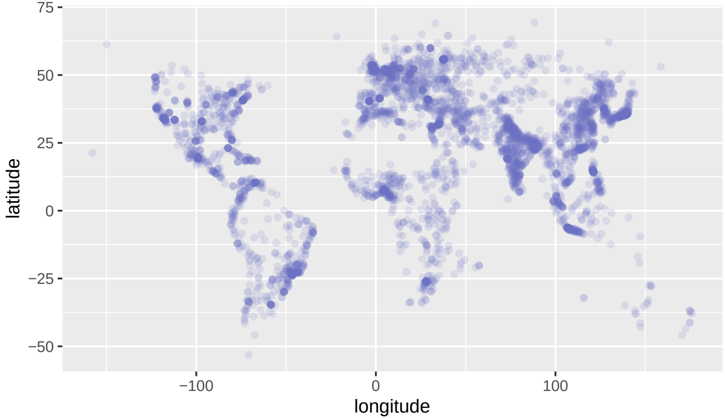


Iris Data: Ground Truth



Example: Largest Cities

```
library(tidyverse); library(mdsr); data(world_cities)
BigCities <- world_cities %>% filter(population > 100000)
BigCities %>%
    ggplot(aes(x = longitude, y = latitude)) +
    geom_point(color = solar["violet"], alpha = 0.15)
```

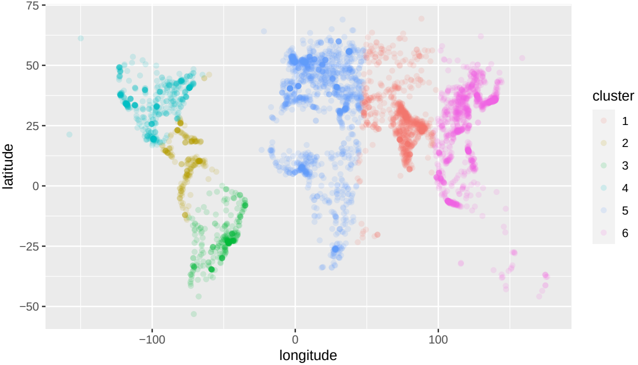


Clustering Cities via K-means

```
library(mclust); set.seed(15)
cluster_model <- BigCities %>%
    select(longitude, latitude) %>%
    kmeans(centers = 6)
cluster_model %>% pluck("centers")

     longitude  latitude
  1  74.32534  27.38729
  2 -75.96901  11.50652
  3 -51.32861 -22.23020
  4 -98.25924  34.15903
  5  18.11938  33.33487
  6 120.42226  23.54011

cluster_model %>% pluck("cluster") %>% head(n = 50)

   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 3 3 3 3 3 3
  [36] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

Plotting the Clusters

```
BigCities <- BigCities %>%
    mutate(
        cluster = cluster_model %>% pluck("cluster") %>% factor())
BigCities %>%
    ggplot(aes(x = longitude, y = latitude, col = cluster)) +
    geom_point(alpha = 0.15)
```



Note: Initialization is Random

```
set.seed(42) # only change is to the random seed
cluster_model <- BigCities %>%
    select(longitude, latitude) %>%
    kmeans(centers = 6)
cluster_model %>% pluck("centers")
```

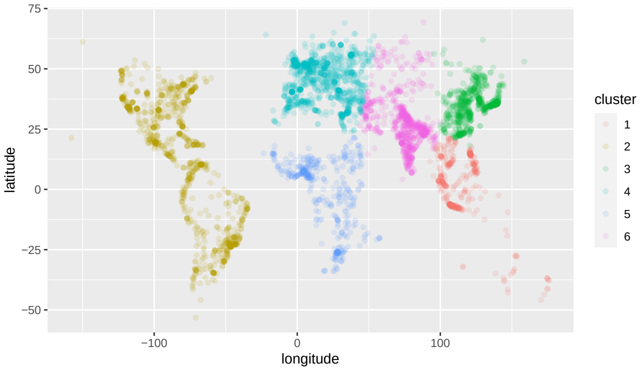
```
   longitude  latitude
 1 114.52357  2.208719
 2 -79.36394 12.351992
 3 123.41967 34.801488
 4  18.64109 45.484237
 5  18.80086 -1.171748
 6  75.02845 27.621635
```

```
cluster_model %>% pluck("cluster") %>% head(n = 50)
```

```
  [1] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 4 4 4 4 4 4 5 5 5 5 5 5 5 2 2 2 2 2 2
 [36] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

Different Initialization

```
# same plotting code as before
BigCities <- BigCities %>%
    mutate(
        cluster = cluster_model %>% pluck("cluster") %>% factor())
BigCities %>%
    ggplot(aes(x = longitude, y = latitude, col = cluster)) +
    geom_point(alpha = 0.15)
```



Multiple Starts: Minimize Distances Within Clusters

```
set.seed(42) # only change is to the random seed
cluster_model <- BigCities %>%
    select(longitude, latitude) %>%
    kmeans(centers = 6, nstart = 10) # run 10 random initializations
cluster_model %>% pluck("centers")
```

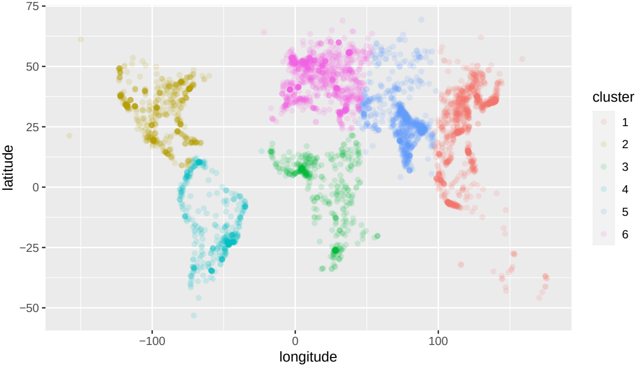
```
   longitude   latitude
1 120.44399  23.542662
2 -94.65728  31.293156
3  18.90771  -1.212412
4 -56.79367 -15.434155
5  75.12489  27.604277
6  18.64109  45.484237
```

```
cluster_model %>% pluck("cluster") %>% head(n = 50)
```

```
 [1] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 3 3 3 3 3 3 3 4 4 4 4 4 4
[36] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
```

Ten Random Initializations

```
# same plotting code as before
BigCities <- BigCities %>%
    mutate(
        cluster = cluster_model %>% pluck("cluster") %>% factor())
BigCities %>%
    ggplot(aes(x = longitude, y = latitude, col = cluster)) +
    geom_point(alpha = 0.15)
```

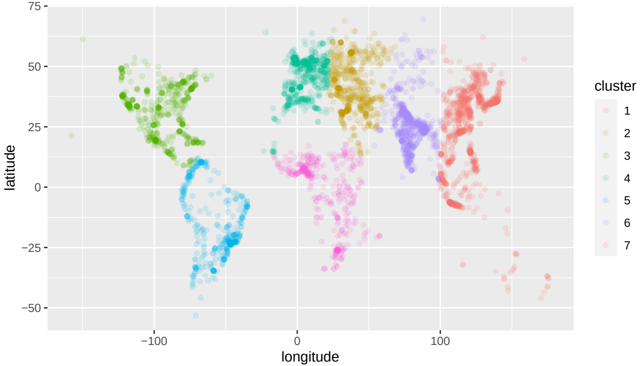


Changing the Number of Clusters

```
set.seed(15)
cluster_model <- BigCities %>%
    select(longitude, latitude) %>%
    kmeans(centers = 7, nstart = 10) # only change is to number of centers
cluster_model %>% pluck("centers")


       longitude  latitude
     1 120.874140  23.76997
     2  38.798273  42.42577
     3 -94.657284  31.29316
     4   5.514612  45.86114
     5 -56.871614 -15.50527
     6  79.645942  25.45844
     7  18.953222  -2.25623

cluster_model %>% pluck("cluster") %>% head(n = 50)


      [1] 6 2 2 2 2 2 6 6 6 6 6 6 6 6 6 6 4 4 4 2 2 2 7 7 7 7 7 7 7 5 5 5 5 5 5
     [36] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
```

Seven Clusters

```
# same plotting code as before
BigCities <- BigCities %>%
    mutate(
        cluster = cluster_model %>% extract2("cluster") %>% factor())
BigCities %>%
    ggplot(aes(x = longitude, y = latitude, col = cluster)) +
    geom_point(alpha = 0.15)
```



Technical Note: Scaling Data

- K-means finds a "local optimum" for within-cluster distance
- Distance is D-dimensional Euclidean distance

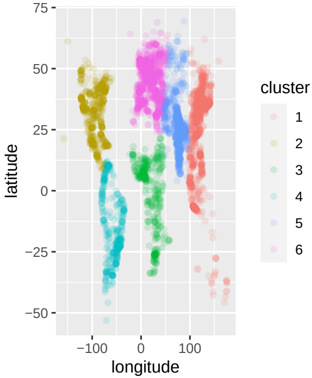
$$\mathbf{x}_i := (x_{i1}, x_{i2}, \ldots, x_{iD})$$

$$d(\mathbf{x}_i, \mathbf{x}_j) := \left[\sum_{d=1}^{D} (x_{id} - x_{jd})^2\right]^{1/2}$$

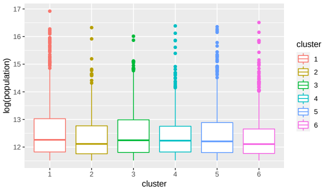
- This weights every dimension equally
- This may not be desireable (for example, cluster people using income in $ and age; age will barely register)
- Usually advised to rescale data before clustering. Popular scalings:
  - z-score = $\frac{x_{id} - \bar{x}_d}{s}$
  - Unit scaling: $\frac{x_{id} - \min_i(x_{id})}{\max_i(x_{id}) - \min_i(x_{id})}$

Using Clusters in Another Plot

```
BigCities %>%
    ggplot(aes(
        x     = longitude,
        y     = latitude,
        color = cluster)) +
    geom_point(alpha = 0.15)
```




```
BigCities %>%
    ggplot(aes(
        x     = cluster,
        y     = log(population),
        color = cluster)) +
    geom_boxplot()
```



Outline

A Brief Detour into Machine Learning

Clustering

K-means

Hierarchical Clustering

```r
cars <- mpg %>%
    rename(
        make         = manufacturer,
        model        = model,
        displacement = displ,
        cylinders    = cyl,
        city_mpg     = cty,
        hwy_mpg      = hwy) %>%
    select(make, model, displacement, cylinders, city_mpg, hwy_mpg) %>%
    distinct(model, .keep_all = TRUE) %>%
    mutate(make_model = paste(make, model)) %>%
    select(-make,-model) %>%
    column_to_rownames("make_model")
head(cars)
```

```
                             displacement cylinders city_mpg hwy_mpg
 audi a4                              1.8         4       18      29
 audi a4 quattro                      1.8         4       18      26
 audi a6 quattro                      2.8         6       15      24
 chevrolet c1500 suburban 2wd         5.3         8       14      20
 chevrolet corvette                   5.7         8       16      26
 chevrolet k1500 tahoe 4wd            5.3         8       14      19
```

Computing Pairwise Distances

```
model_diffs <-
    cars %>%
    dist()
dist_mat <-
    model_diffs %>%
    as.matrix()
```

```
dist_mat %>% extract(1:4, 1:4) %>% round(digits = 2)
```

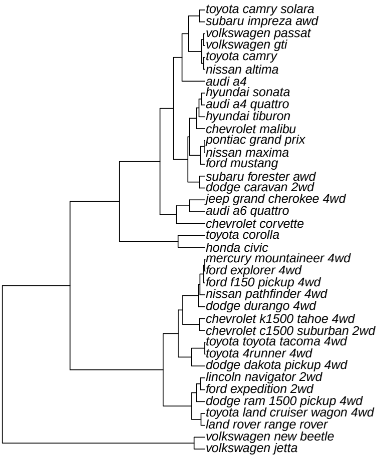
```
                           audi a4 audi a4 quattro audi a6 quattro
  audi a4                     0.00            3.00            6.24
  audi a4 quattro             3.00            0.00            4.24
  audi a6 quattro             6.24            4.24            0.00
  chevrolet c1500 suburban 2wd   11.19            8.96            5.22
                           chevrolet c1500 suburban 2wd
  audi a4                                         11.19
  audi a4 quattro                                  8.96
  audi a6 quattro                                  5.22
  chevrolet c1500 suburban 2wd                     0.00
```

Hierarchical Clusters

```
library(ape)
cluster_tree <- model_diffs %>%
    hclust() %>%
    as.phylo()
```

Clustering Tree

```
cluster_tree %>% plot(cex = 0.9, label.offset = 0.1)
```



Again With Standardized Distances

```
cars_scaled <- cars %>%
    transmute_all(list(scale))
rownames(cars_scaled) <- rownames(cars)
model_diffs <- cars_scaled %>% dist()
dist_mat <- model_diffs %>% as.matrix()
dist_mat %>% extract(1:4, 1:4) %>% round(digits = 2)
```

```
                             audi a4 audi a4 quattro audi a6 quattro
    audi a4                     0.00            0.43            1.75
    audi a4 quattro             0.43            0.00            1.62
    audi a6 quattro             1.75            1.62            0.00
    chevrolet c1500 suburban 2wd    4.11            3.99            2.50
                             chevrolet c1500 suburban 2wd
    audi a4                                          4.11
    audi a4 quattro                                  3.99
    audi a6 quattro                                  2.50
    chevrolet c1500 suburban 2wd                     0.00
```

Hierarchical Clusters

```
library(ape)
cluster_tree <- model_diffs %>%
    hclust() %>%
    as.phylo()
```

Clustering Tree

```
plot(cluster_tree, cex = 0.9, label.offset = 0.1)
```