Cluster Analysis: Discovery
Winter Institute in Data Science

Ryan T. Moore

9 January 2020

Matrices and Lists

Matrices and Lists

- `matrix`: rectangular array with a few more restrictions than `data.frame`

Matrices and Lists

- `matrix`: rectangular array with a few more restrictions than `data.frame`
- `list`: more general type of array that can be rectangular, but need not be. A `data.frame` is an example of a `list`

Matrices and Lists

- `matrix`: rectangular array with a few more restrictions than `data.frame`
- `list`: more general type of array that can be rectangular, but need not be. A `data.frame` is an example of a `list`

Matrices and Lists

- `matrix`: rectangular array with a few more restrictions than `data.frame`
- `list`: more general type of array that can be rectangular, but need not be. A `data.frame` is an example of a `list`

We can extract variable `v1` from data frame `df` with `df$v1`.

Can we extract column `c1` from matrix `m` with `m$c1`?

4. Describe list `ll` created below:

```
v1 <- 1:8
v2 <- letters[1:5]
m <- matrix(1:9, 3, 3)
ll <- list(x = v1, y = v2, z = m)
```

5. What is `ll$y`?
6. What is `ll[[1]]`?
7. What is `ll[[3]][2, 2]`?

Supervised and Unsupervised Learning

Supervised and Unsupervised Learning

Supervised: Modeling with **known** outcomes

Supervised and Unsupervised Learning

Supervised: Modeling with **known** outcomes

Unsupervised: Discovery w/ **unknown** outcomes

Supervised Learning

- Linear regression (LS)

Supervised Learning

- Linear regression (LS)
- Generalized linear regression (logistic, probit, Poisson, beta)

Supervised Learning

- Linear regression (LS)
- Generalized linear regression (logistic, probit, Poisson, beta)
- Bayesian modeling

Supervised Learning

- Linear regression (LS)
- Generalized linear regression (logistic, probit, Poisson, beta)
- Bayesian modeling
- LASSO

Supervised Learning

- Linear regression (LS)
- Generalized linear regression (logistic, probit, Poisson, beta)
- Bayesian modeling
- LASSO
- Neural networks (CNNs)

Supervised Learning

- Linear regression (LS)
- Generalized linear regression (logistic, probit, Poisson, beta)
- Bayesian modeling
- LASSO
- Neural networks (CNNs)
- Decision trees, forests, etc.

Supervised Learning

- Linear regression (LS)
- Generalized linear regression (logistic, probit, Poisson, beta)
- Bayesian modeling
- LASSO
- Neural networks (CNNs)
- Decision trees, forests, etc.

Supervised Learning

- Linear regression (LS)
- Generalized linear regression (logistic, probit, Poisson, beta)
- Bayesian modeling
- LASSO
- Neural networks (CNNs)
- Decision trees, forests, etc.

If you have $y = f(X)$, it's "supervised".

Unsupervised Learning

- Network analysis

Unsupervised Learning

- Network analysis
- Principal components analysis (PCA)

Unsupervised Learning

- Network analysis
- Principal components analysis (PCA)
- (Some neural networks)

Unsupervised Learning

- Network analysis
- Principal components analysis (PCA)
- (Some neural networks)
- Clustering algorithms (most)

Clustering is a method of "unsupervised learning" – of trying to derive structure from data when no outcome labels are given.

Clustering is a method of "unsupervised learning" – of trying to derive structure from data when no outcome labels are given.

- Networks: "community detection"

Clustering is a method of "unsupervised learning" – of trying to derive structure from data when no outcome labels are given.

- Networks: "community detection"
  - no predefined community labels

Clustering is a method of “unsupervised learning” – of trying to derive structure from data when no outcome labels are given.

- Networks: “community detection”
    - no predefined community labels
- Roll call voting: party/faction detection

Clustering is a method of "unsupervised learning" – of trying to derive structure from data when no outcome labels are given.

- Networks: "community detection"
  - no predefined community labels
- Roll call voting: party/faction detection
  - no predefined party/faction labels

Clustering is a method of "unsupervised learning" – of trying to derive structure from data when no outcome labels are given.

- Networks: "community detection"
  - no predefined community labels
- Roll call voting: party/faction detection
  - no predefined party/faction labels
- Geographic clustering: daily activities

Clustering is a method of "unsupervised learning" – of trying to derive structure from data when no outcome labels are given.

- Networks: "community detection"
  - no predefined community labels
- Roll call voting: party/faction detection
  - no predefined party/faction labels
- Geographic clustering: daily activities
  - no "home"/"work"/"leisure" labels

Clustering

- Hierarchical clustering

Clustering

- Hierarchical clustering
  - Find best groups at many levels

Clustering

- Hierarchical clustering
  - Find best groups at many levels
  - Units in one cluster at each level of hierarchy

Clustering

- Hierarchical clustering
  - Find best groups at many levels
  - Units in one cluster at each level of hierarchy
- Partitional clustering

Clustering

- Hierarchical clustering
  - Find best groups at many levels
  - Units in one cluster at each level of hierarchy
- Partitional clustering
  - Find splits in full set

Clustering

- Hierarchical clustering
  - Find best groups at many levels
  - Units in one cluster at each level of hierarchy
- Partitional clustering
  - Find splits in full set
  - Units in only one cluster

Hierarchical Clustering

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Divisive clustering

- Start at $\{a, b, c, d, e\}$

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Divisive clustering

- Start at $\{a, b, c, d, e\}$
- Find $\{a, b, d\}$ and $\{c, e\}$

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Divisive clustering

- Start at $\{a, b, c, d, e\}$
- Find $\{a, b, d\}$ and $\{c, e\}$
- $\ldots$

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Divisive clustering

- Start at $\{a, b, c, d, e\}$
- Find $\{a, b, d\}$ and $\{c, e\}$
- $\ldots$
- End at $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Divisive clustering

- Start at $\{a, b, c, d, e\}$
- Find $\{a, b, d\}$ and $\{c, e\}$
- $\ldots$
- End at $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Divisive clustering

- Start at $\{a, b, c, d, e\}$
- Find $\{a, b, d\}$ and $\{c, e\}$
- $\ldots$
- End at $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$

Divisive clustering is "top-down"

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Agglomerative clustering

- Start at $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Agglomerative clustering

- Start at $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$
- Find $\{a, b\}$ and leave $\{c\}$, $\{d\}$, $\{e\}$

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Agglomerative clustering

- Start at $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$
- Find $\{a, b\}$ and leave $\{c\}$, $\{d\}$, $\{e\}$
- ...

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Agglomerative clustering

- Start at $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$
- Find $\{a, b\}$ and leave $\{c\}$, $\{d\}$, $\{e\}$
- $\dots$
- End at $\{a, b, c, d, e\}$

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Agglomerative clustering

- Start at $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$
- Find $\{a, b\}$ and leave $\{c\}$, $\{d\}$, $\{e\}$
- ...
- End at $\{a, b, c, d, e\}$

Hierarchical clustering

Consider a set of units open to cluster discovery, $\{a, b, c, d, e\}$

Agglomerative clustering

- Start at $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$
- Find $\{a, b\}$ and leave $\{c\}$, $\{d\}$, $\{e\}$
- ...
- End at $\{a, b, c, d, e\}$

Agglomerative clustering is "bottom-up"

Hierarchical Clustering

Each unit in different clustering at every level.

Agglomerative Hierarchical Clustering

Complete linkage clustering: greedily create clusters

Agglomerative Hierarchical Clustering

Complete linkage clustering: greedily create clusters

1. Merge closest pair (into, say, $\{a, b\}$)

Agglomerative Hierarchical Clustering

Complete linkage clustering: greedily create clusters

1. Merge closest pair (into, say, $\{a, b\}$)
2. Update `dist`: delete rows/cols for a, b; add row for $\{a, b\}$ (with max)
3. Merge closest "pair"
4. Update `dist`
5. …

Agglomerative Hierarchical Complete Linkage Clusters

	a	b	c	d
a	0	1	2	3
b	1	0	6	7
c	2	6	0	8
d	3	7	8	0

Agglomerative Hierarchical Complete Linkage Clusters

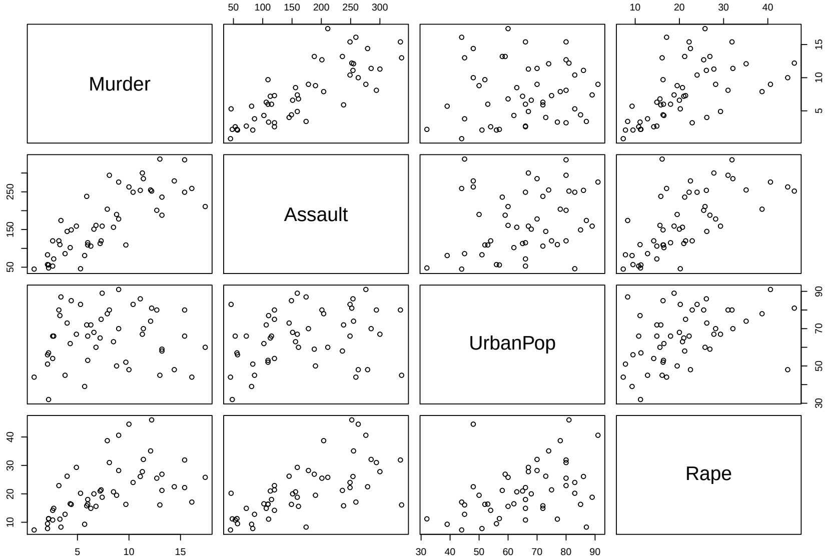
	a	b	c	d
a	0	1	2	3
b	1	0	6	7
c	2	6	0	8
d	3	7	8	0

	(a,b)	c	d
(a,b)	0	6	7
c	6	0	8
d	7	8	0

Agglomerative Hierarchical Clustering

Agglomerative Hierarchical Clustering

```r
pairs(USArrests)
```



Agglomerative Hierarchical Clustering

```r
d <- dist(USArrests, method = "euclidean")
d %>% round(0)
```

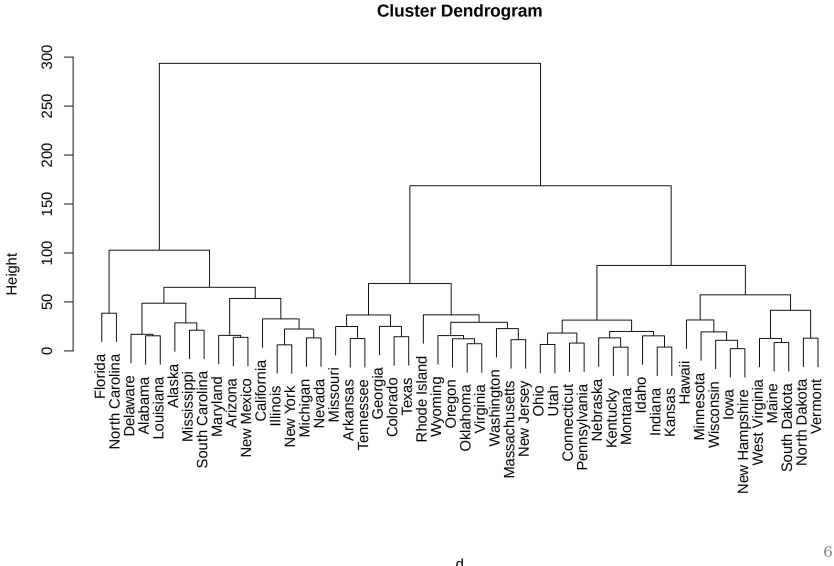
```
##                Alabama Alaska Arizona Arkansas Californ
## Alaska              37
## Arizona             63     47
## Arkansas            47     77     109
## California          56     45      23       98
## Colorado            42     66      90       37         7
## Connecticut        128    159     185       85        16
## Delaware            17     45      59       53         4
## Florida            102     80      42      149         6
## Georgia             26     57      86       26         7
## Hawaii             192    221     248      148        23
## Idaho              117    146     177       71        16
## Illinois            28     43      46       68         3
## Indiana            123    153     182       78        16
## Iowa               181    210     240      135        22
## Kansas             122    151     180       77        16
```

Agglomerative Hierarchical Clustering

```r
hag <- hclust(d, method = "complete")
plot(hag)
```

Agglomerative Hierarchical Clustering

```r
hag <- hclust(d, method = "complete")
plot(hag)
```



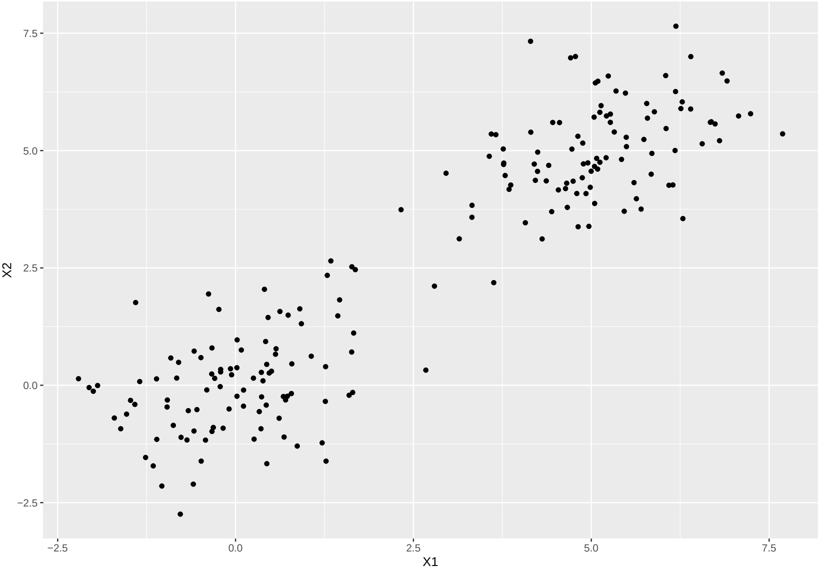
Partitional Clustering

k-means

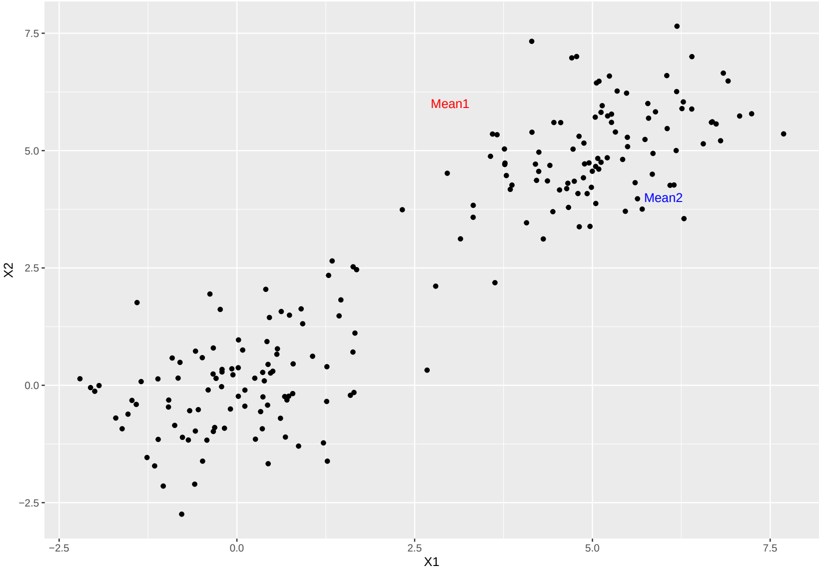
0. Standardize variables, or ensure comparable scales. A difference of 1 unit on x should be comparable to a difference of 1 unit on y.
1. Choose k, the number of clusters to identify.
2. Select the location of a center for each cluster.
3. Assign each observation to the cluster defined by the center closest to it.
4. Relocate each cluster's center to the mean of the observations currently in that cluster.
5. Repeat 3. and 4. until no observations gets assigned to a new cluster.

Suppose we have a set of points measured in a two-dimensional space, with `X1` and `X2` on comparable scales.

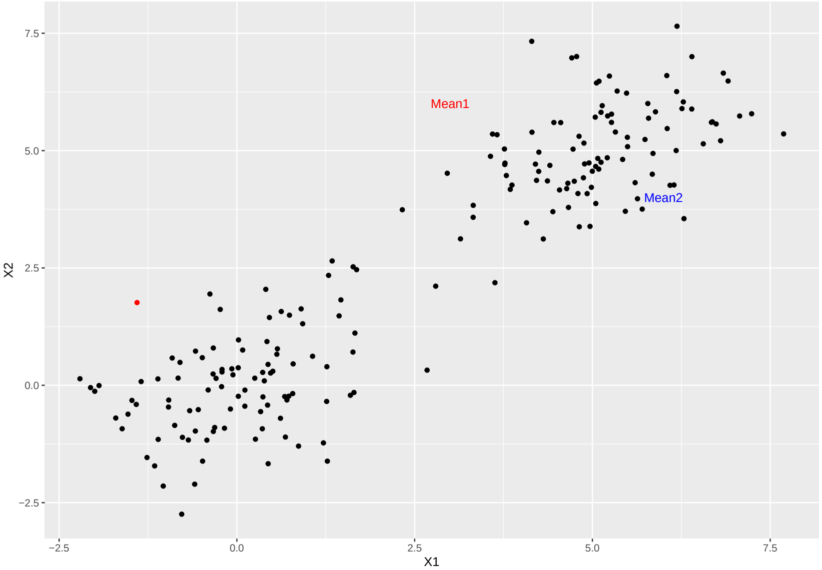
We will find $k = 2$ clusters. Ideas?



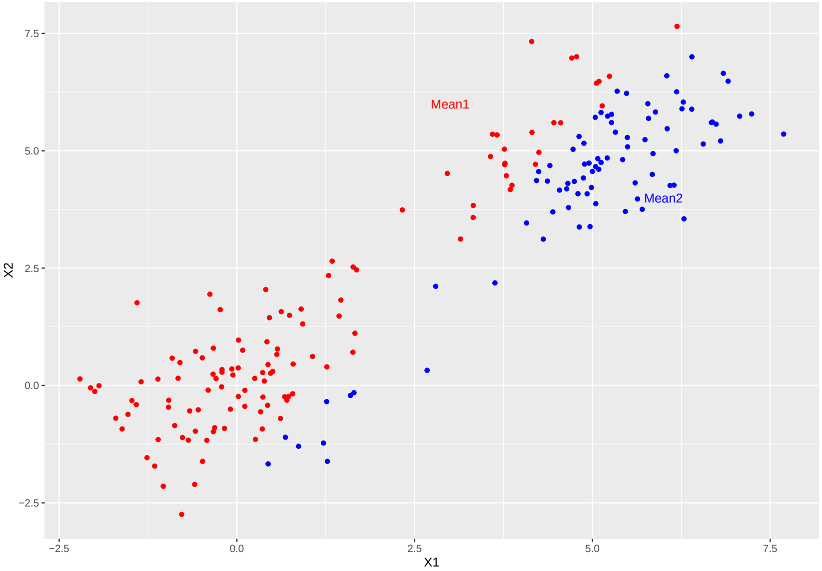
2. Let's randomly select two centroids:



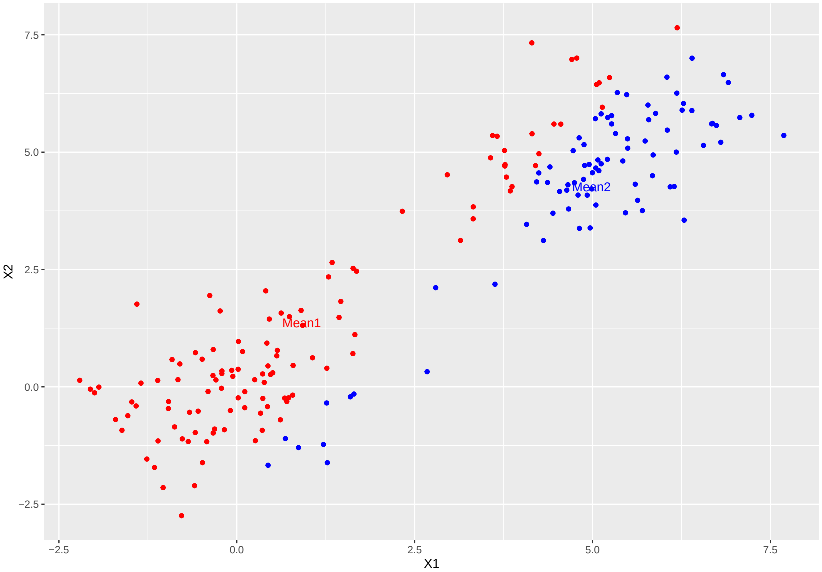
Where "should" this point go?



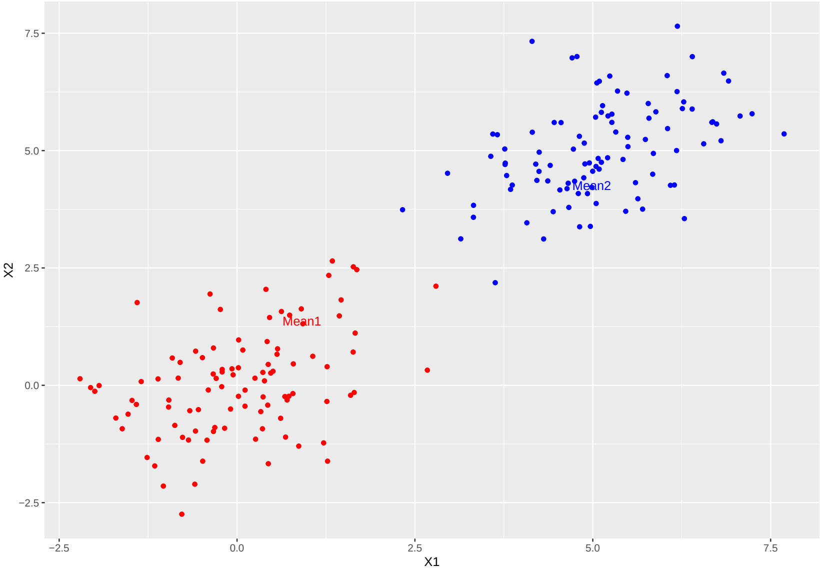
3. Find (Euclidean) distance between each point and the centroids; assign each point to closer centroid:



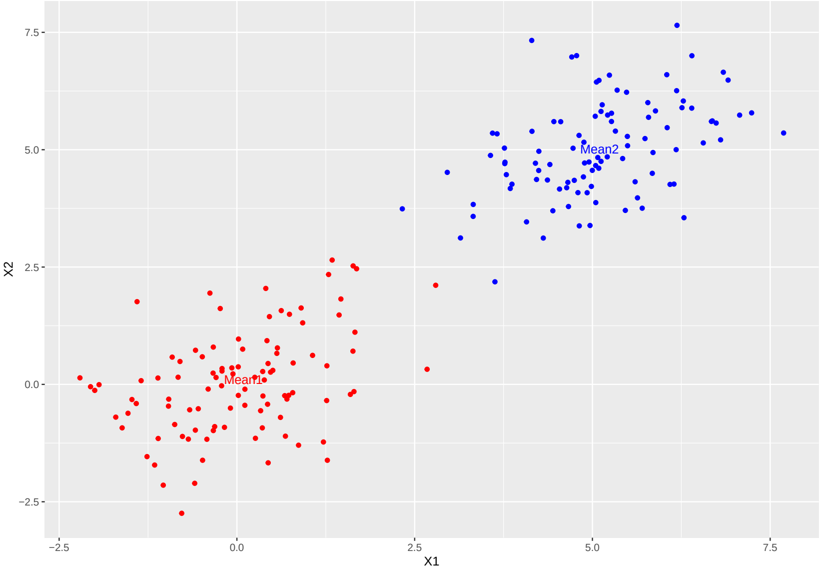
4. Relocate centroids to the mean (`X1`, `X2`) value for each cluster:



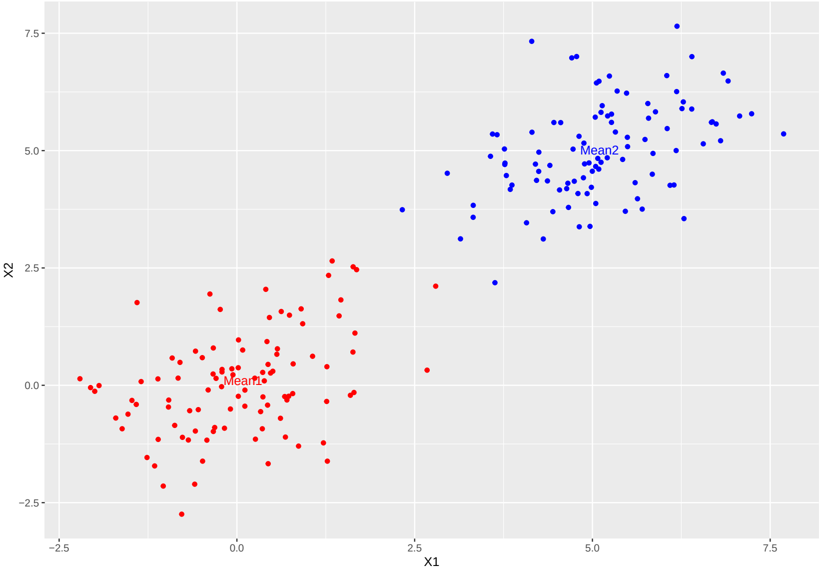
3. Reassign each observation to the closer centroid:



4. Recalculate the centroid locations ...

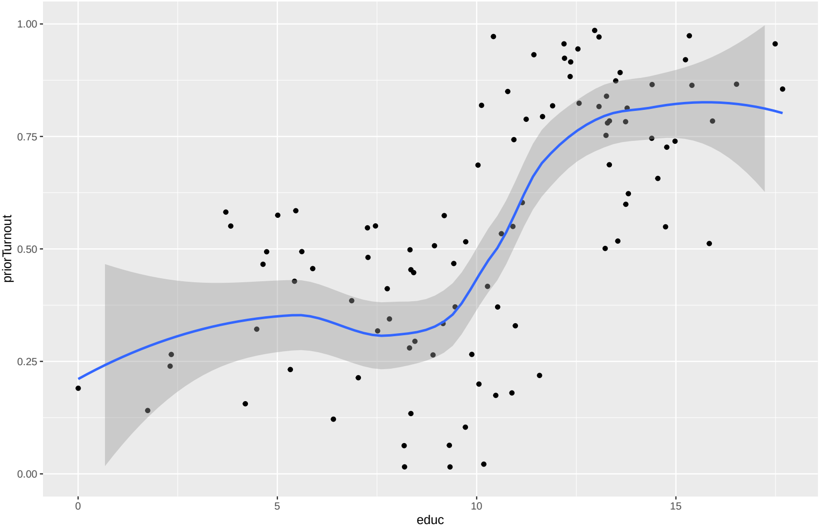


3. Reassign points to clusters . . .



Standardization: Why, before clustering?

Suppose we have prior turnout $[0, 1]$ and education (yrs):



```r
k2 <- kmeans(df2, centers = 2)
names(k2)
```

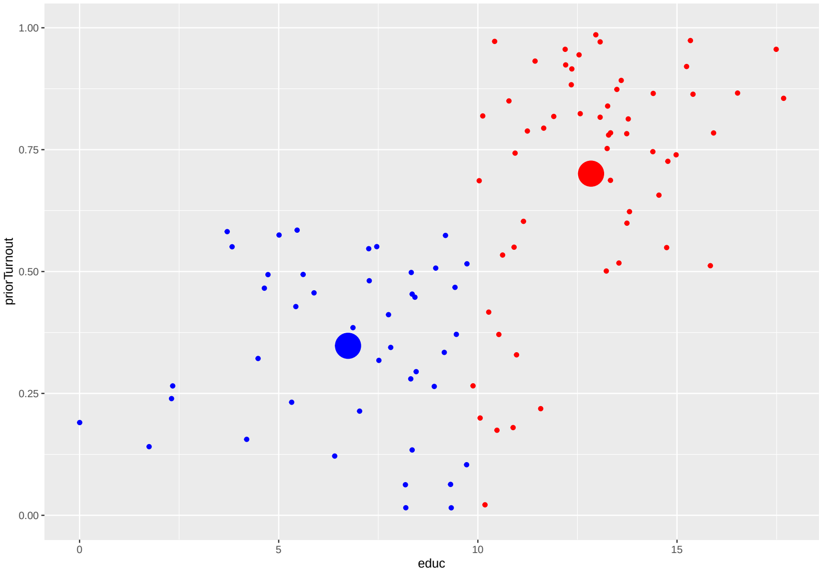
```
## [1] "cluster"      "centers"      "totss"        "withi
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
table(k2$cluster)
```

```
##
##  1  2
## 57 43
```

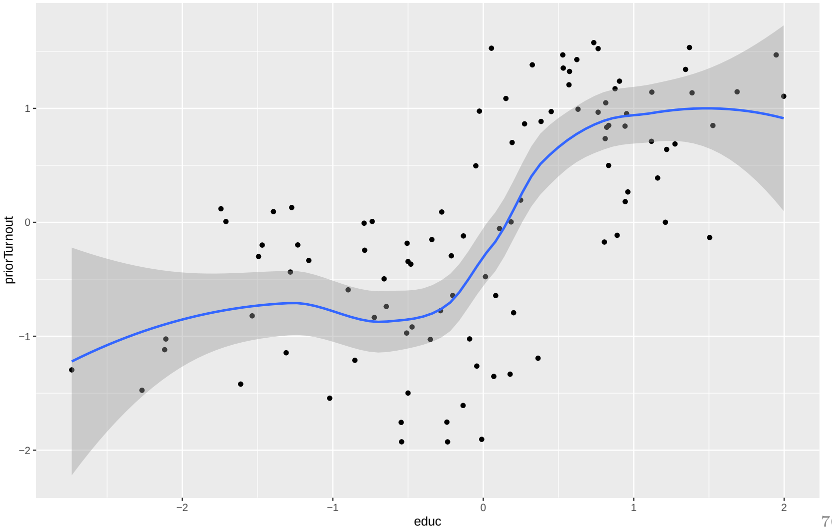
```r
k2$centers
```

```
##        educ priorTurnout
## 1 12.841805    0.7007953
## 2  6.740813    0.3477128
```

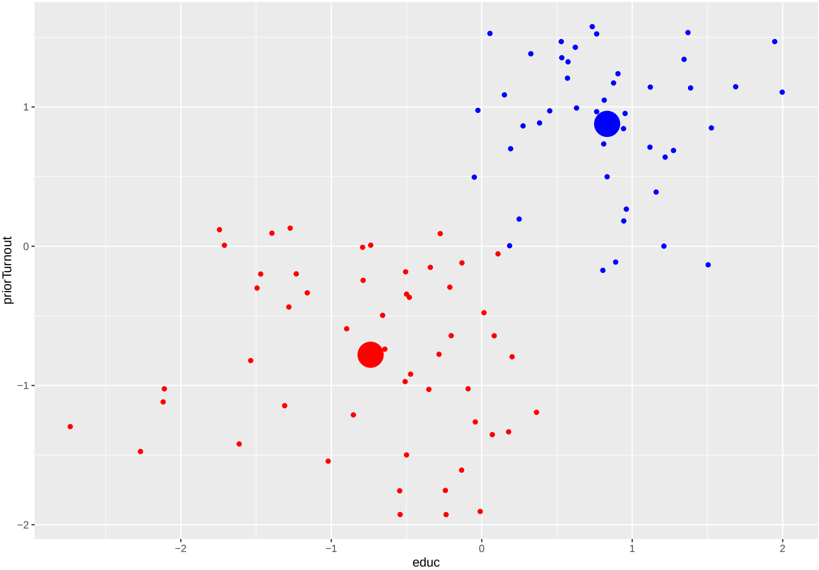

1.00
0.75
0.50
0.25
0.00
priorTurnout
0
5
10
15
educ

Standardized:

```r
df2.standard <- data.frame(scale(df2))
ggplot(df2.standard, aes(educ, priorTurnout)) +
  geom_point() + geom_smooth()
```



```
##         educ priorTurnout
## 1 -0.7388515   -0.7791249
## 2  0.8331730    0.8785876
```



Other applications: Geolocations

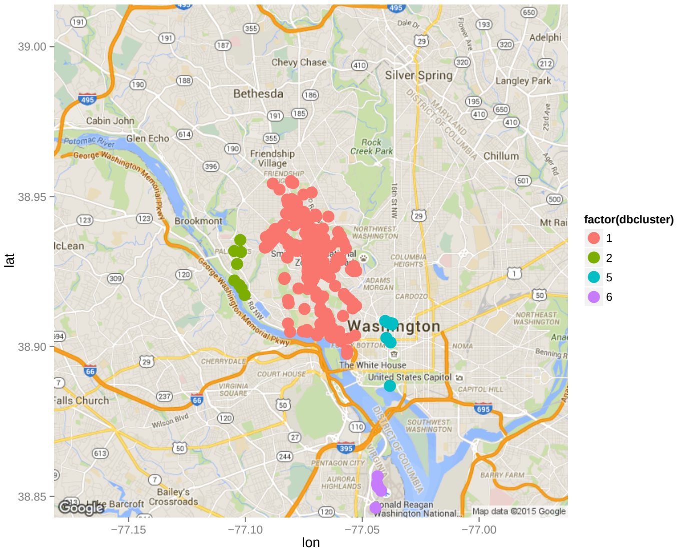


Figure 1: Clusters of Geolocations

Other applications: Regimes

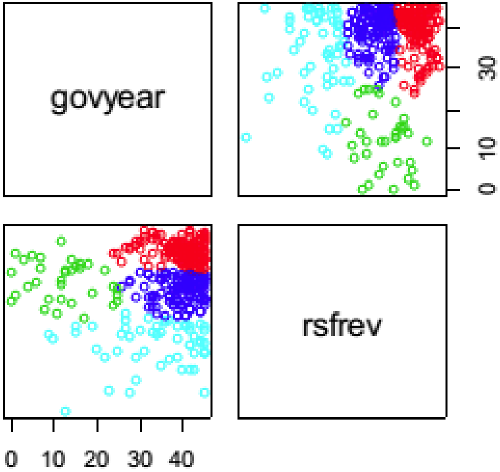


Figure 2: Comparative Regime Types

Other applications: Senate Speeches

Other applications: Senate Speeches

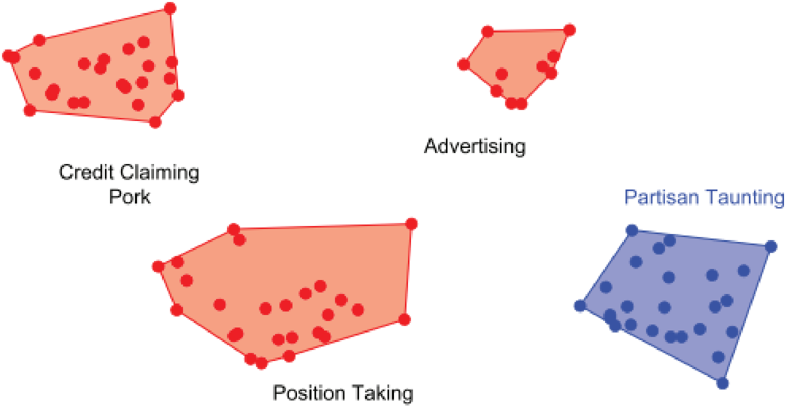


Figure 3: Senate Speeches

DBSCAN (1996)

Density-based clustering

DBSCAN (1996)

Density-based clustering

1. Find each point's neighbors

DBSCAN (1996)

Density-based clustering

1. Find each point's neighbors
2. ID *core* points with enough neighbors

DBSCAN (1996)

Density-based clustering

1. Find each point's neighbors
2. ID *core* points with enough neighbors
3. Connect nearby core points

DBSCAN (1996)

Density-based clustering

1. Find each point's neighbors
2. ID *core* points with enough neighbors
3. Connect nearby core points
4. Assign non-core points to near clusters (or noise)

Other applications:

Figure 4: CLARANS: Not Great!

Other applications:

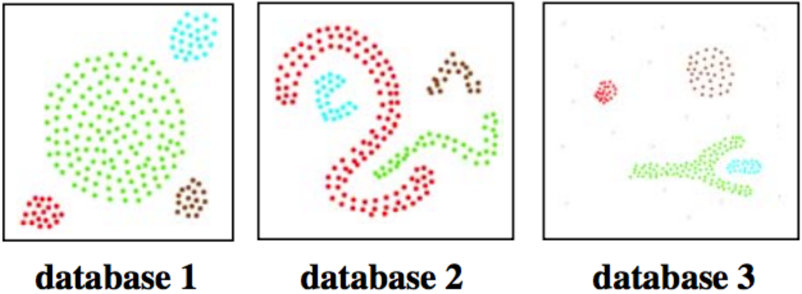
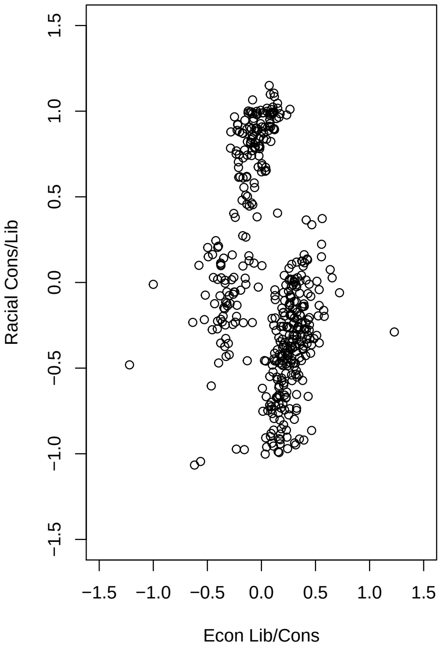


Figure 5: DBSCAN: Great!

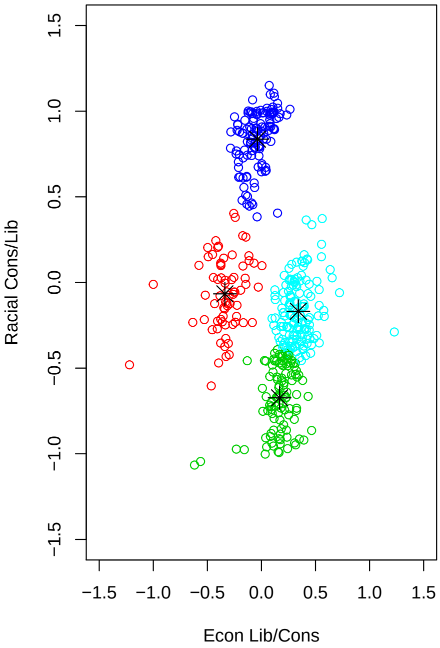
Congress Clusters

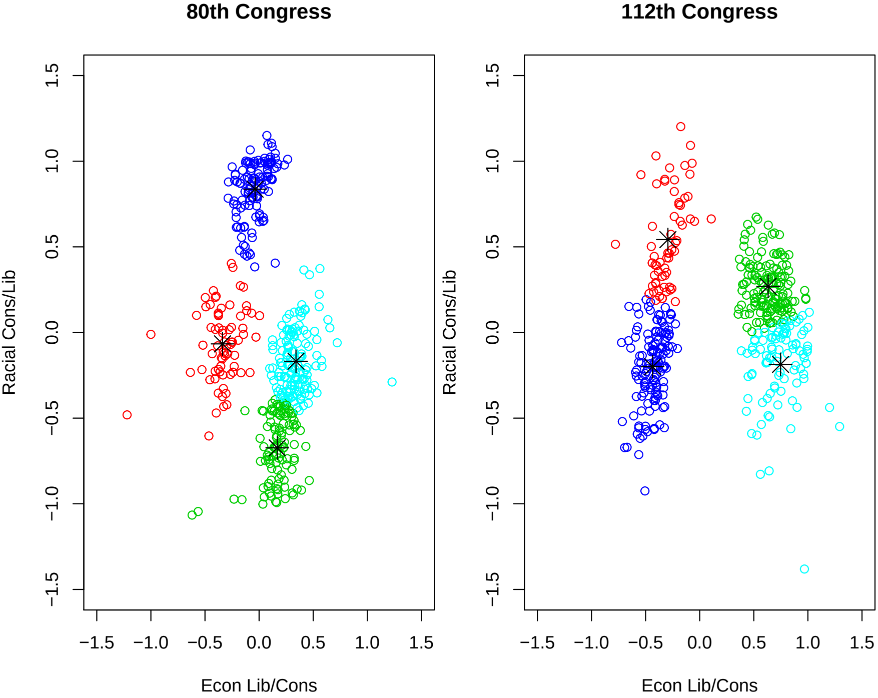
```
congress.url <- "http://j.mp/302nedz"
congress <- read_csv(congress.url)
dwnom80 <- cbind(congress$dwnom1[congress$congress == 80],
                 congress$dwnom2[congress$congress == 80])
dwnom112 <- cbind(congress$dwnom1[congress$congress == 112
                  congress$dwnom2[congress$congress == 112

k80two.out <- kmeans(dwnom80, centers = 2)
k112two.out <- kmeans(dwnom112, centers = 2)

k80four.out <- kmeans(dwnom80, centers = 4)
k112four.out <- kmeans(dwnom112, centers = 4)

lim <- c(-1.5, 1.5)
xlab <- "Econ Lib/Cons"
ylab <- "Racial Cons/Lib"
```

80th Congress



80th Congress




80th Congress
112th Congress
Racial Cons/Lib
Econ Lib/Cons
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5

Exercise

Exercise

1. Checkout CRAN Task View for Clustering:

https://cran.r-project.org/web/views/Cluster.html

2. Discover clusters in your final project data!

(Use 2 predictors to visualise; more predictors to discover higher-dim clusters.)