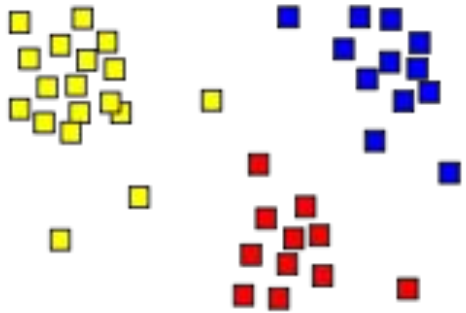
Fuzzy Clustering in R

Exploring centroid-based clustering and implementing a fuzzy clustering algorithm for NBA player classification

Author: Cole Conte

Clustering

- Grouping related objects together
- Techniques:
  - Hierarchical Clustering
  - **Centroid-Based Clustering**
  - Distribution-Based Clustering
  - Density-Based Clustering
  - Grid-Based Clustering



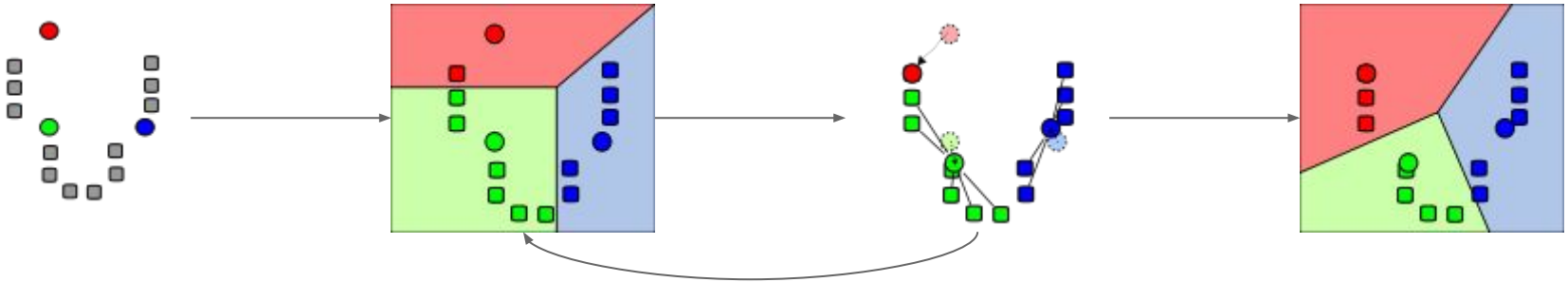
Centroid-Based Clustering

- Most well-known method
- k-means clustering is the most common version
- k groups specified beforehand
- Algorithms are quick-converging but not optimal

Initialization Methods

- Forgy: randomly choose observations from the data set as initial means
  - Preferred for standard k-means
- Random Partition: randomly assign a cluster to each observation
  - Preferred for fuzzy k-means

k-means Algorithm

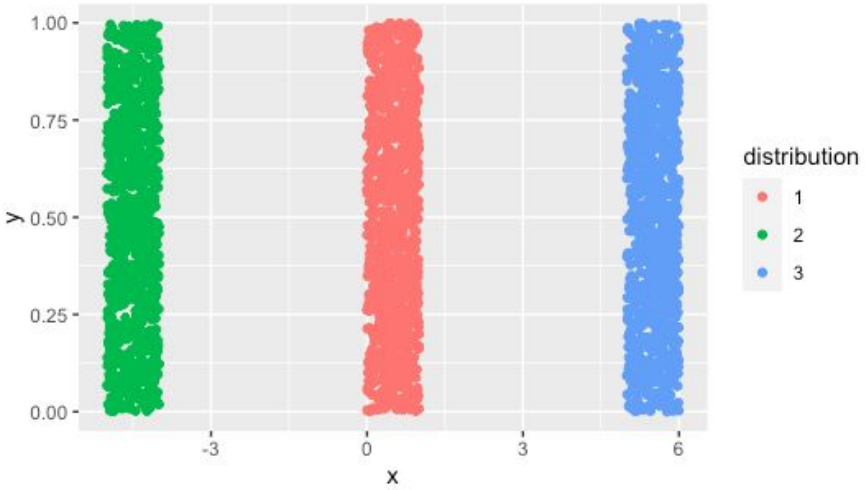


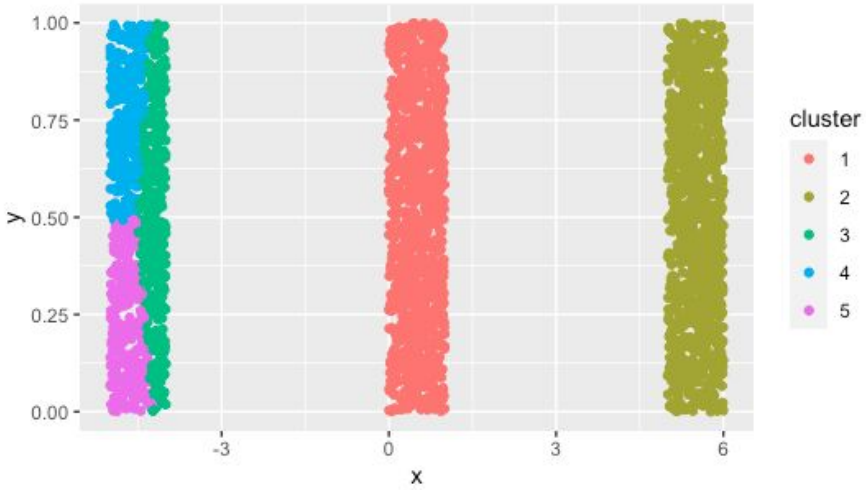
1. Randomly generate k (here k=3) initial means
2. Create k clusters by associating every observation with its nearest mean
3. Centroid of each of the clusters becomes the new mean
4. Repeat steps 2 and 3 until convergence

Problems with k-means

- Worst case run-time is super-polynomial!
- Must specify k beforehand
- Convergence depends on starting means
  - This can cause a sub-optimal clustering
  - We hope to solve this with an improved initialization algorithm
- Sub-optimal clustering
- No indication beyond graphical intuition if data is not group-based

Incorrect k Specification Example



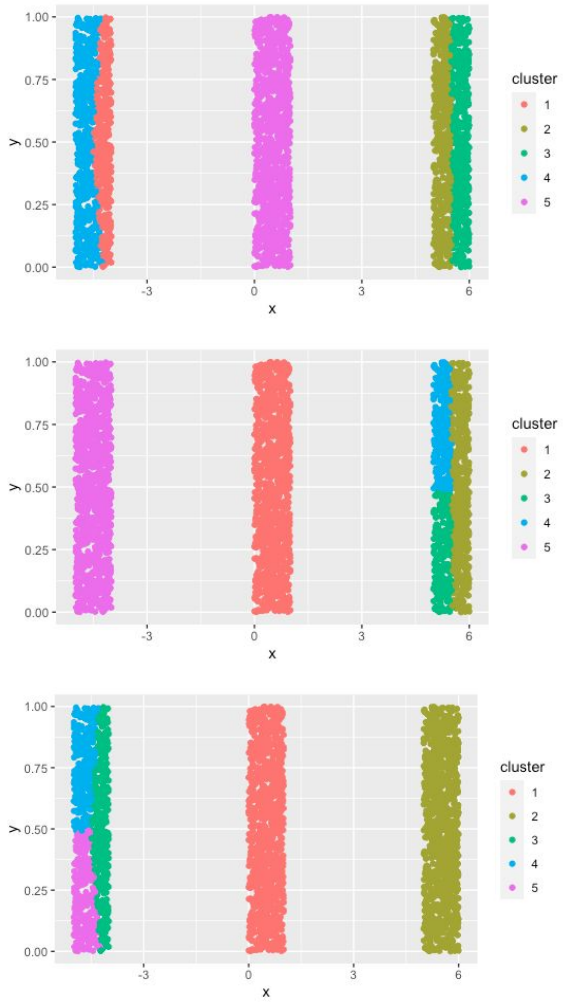


If we don't know how many groups are in the original distribution, we may choose k wrong and get an unintended result. Here it's an easy fix, but for a more complicated distribution it may not be.

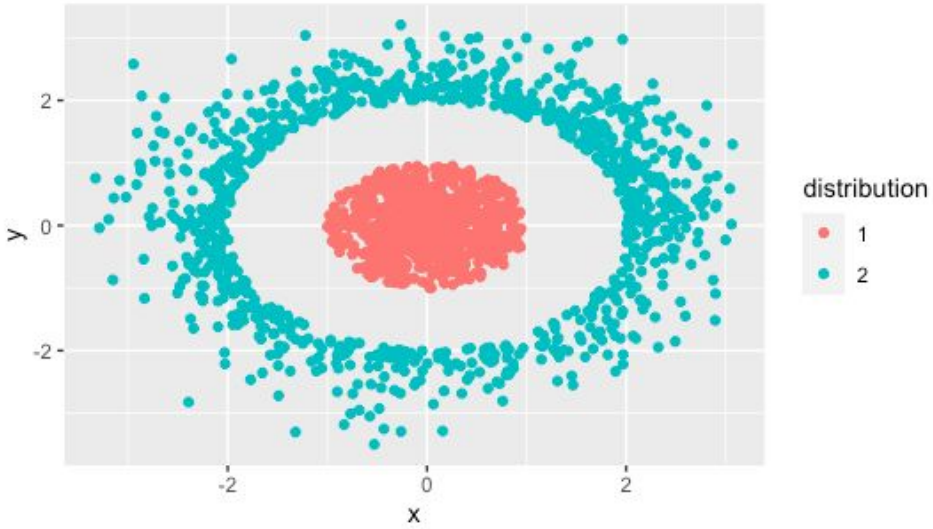
Dependence on Starting Location

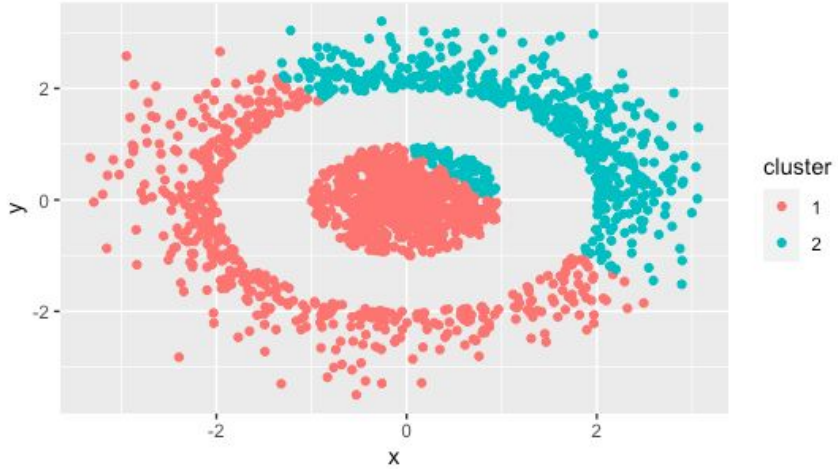
This is the result of running the k-means algorithm on 3 different times over the same data set.

This didn't take very long to generate- the results came from 3 consecutive runs!



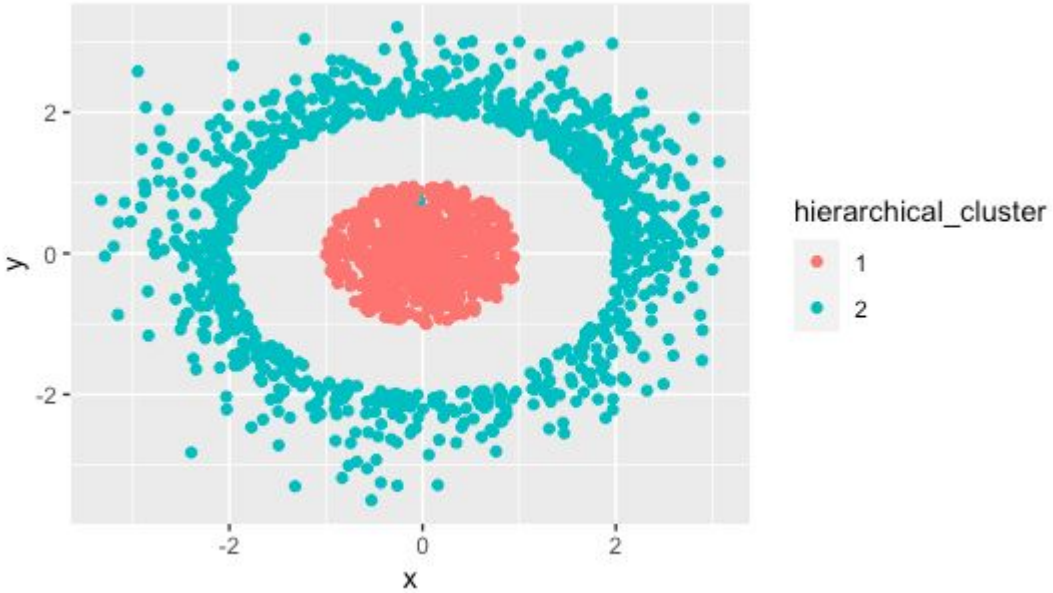
Sub-Optimal Clustering Example





Here our k-means algorithm is sub-optimal for clustering this data set.

Solving The Previous Example



While you may think that the distribution in the previous slide can not be modeled by a clustering algorithm, that's not the case. The problem just calls for a different class of clustering algorithm, namely single linkage hierarchical clustering, which here comes very close to the correct clustering.

Variation: k-means++

- Some of k-means' issues can be solved with a better initialization method.
- Ideally the initial centers will be spread out from each other.
- The tradeoff is that it takes longer to set up the centers, but the algorithm will often converge more quickly.
- After the initialization step, k-means++ proceeds just like k-means (steps 2-4 in the algorithm outline on slide 5).

k-means ++ Center Selection Algorithm

1. Choose a center at random from among the data points.
2. For each point x, compute the distance D(x) between the point and the nearest center that has been chosen.
3. Choose a new data point at random as a new center, this time using a weighted probability distribution where x is chosen with probability proportional to distance squared.
4. Repeat steps 2 and 3 until k centers have been chosen.

Other k-variations

- k-mediods
  - One of the data points must be the center (mediod).
  - More robust to noise and outliers than k-means.
  - Partitioning Around Mediods algorithm
- k-medians
  - Minimizes error over the 1-norm distance
  - k-means minimizes over the 2-norm (Euclidean) distance

Silhouette

- Method of validating consistency of clusters
- General purpose measurement for any clustering method
- High values indicate good matching

Silhouette Calculation

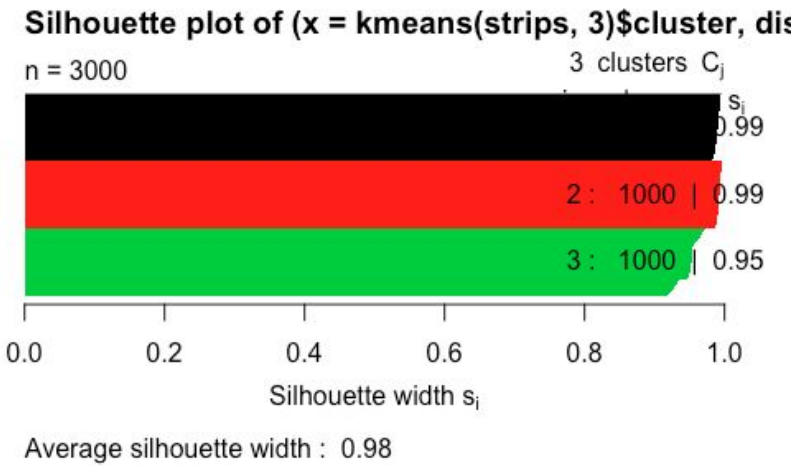
- Data point i in cluster Ci
- a(i) is mean distance between i and all other points in cluster
- b(i) is the smallest mean distance of i to all points in any cluster it doesn't belong to
- s(i) is the silhouette value (range -1 to 1)

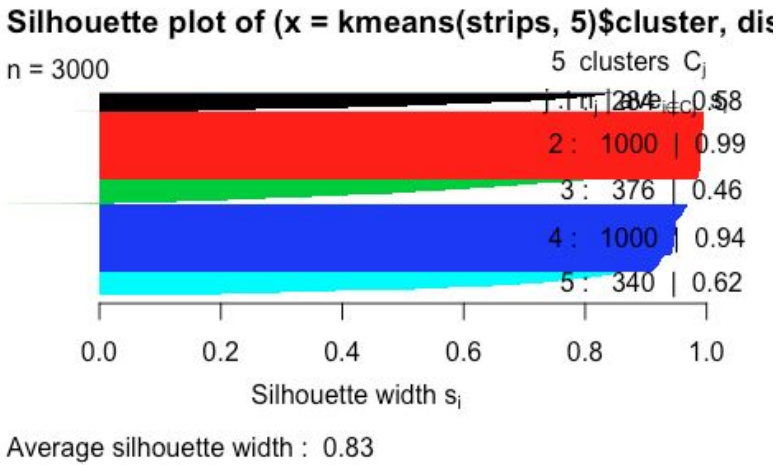
$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

Silhouette Plot Examples





Silhouette plots for the strip example from slide 7. The first plot indicates an optimal number of clusters (3). The second indicates that clusters 2 and 4 do a great job, but clusters 1, 3, and 5 have a lot of overlap.

Fuzzy Clustering

- “Soft clustering”
- Each data point belongs to each cluster to a degree (membership grade)
- Membership grades are normalized between 0 and 1
  - They do not represent probabilities
- Fuzzifier: determines the level of cluster fuzziness
- c-means, FLAME are common fuzzy clustering algorithms

Fuzzy c-means Clustering Algorithm

1. Choose a number c of clusters
2. Randomly assign start values for membership grades
3. Compute the centroid ck for each cluster
   - m is the fuzzifier hyper-parameter
   - wk(x) is the kth membership grade for point x
4. For each data point, compute its membership grades
5. Repeat steps 3 and 4 until the algorithm has converged

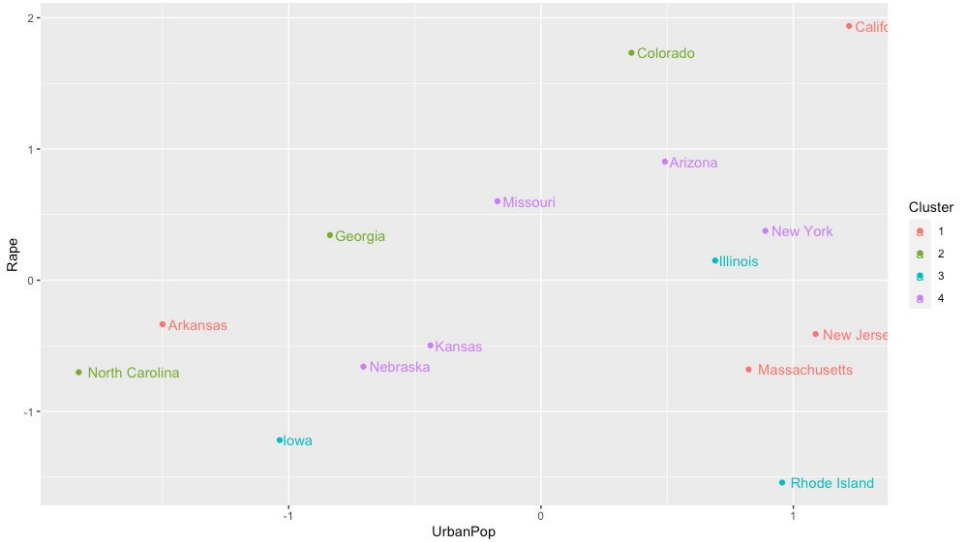
$$c_k = \frac{\sum_x w_k(x)^m x}{\sum_x w_k(x)^m},$$

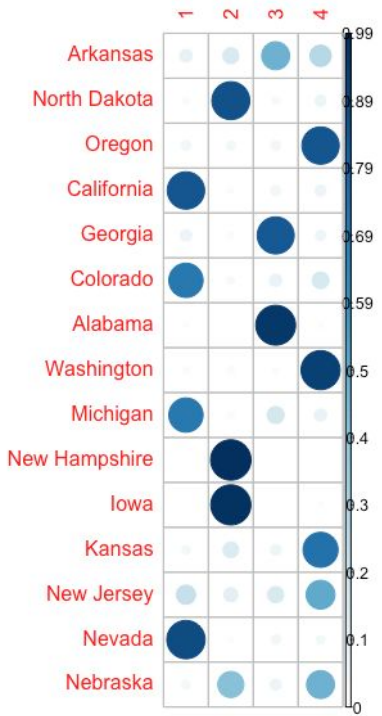
Membership grade wij of element xi in cluster cj:

$$w_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}.$$

Fuzzy c-means Example

Using data from the U.S. arrests data set, we show how different states belong to different clusters based on crime rates. The plot on the right is using fuzzy clustering, while the plot on the bottom is using hard clustering. Note that there are more than two predictors so the hard clustering plot takes into account more than just UrbanPop and Rape.





Background: A New NBA

- Until 1979, center was considered the most important position in the NBA
  - Franchises built their legacies around centers like Kareem Abdul-Jabbar, Bill Russell, and Wilt Chamberlain
- With the introduction of the three-point line in 1979, teams began turning to smaller players who could shoot from outside
  - Michael Jordan, Larry Bird
- Teams also began to use players who could play multiple positions
  - Magic Johnson

Shooting and “positionless basketball” have continued to grow popular in the 21st century, and analytics efforts by NBA teams have lent credence to their effectiveness. Even 7 foot tall centers are expected to be good outside shooters and to be able to defend multiple positions.

Problem Statement

Using a data set of NBA player data, let's explore uses of clustering.

- The NBA lists players at 5 positions, a relic from older basketball ideas
  - Point Guard, Shooting Guard, Small Forward, Power Forward, Center
- Now, NBA players traditionally fit a number of archetypes
  - Passing guard, scoring guard, scoring wing, lockdown defensive wing, big man, etc.
- We present examples for three questions:
  - How accurate is clustering in predicting listed player position?
  - What is the ideal number of archetypes (the k in k-means)? Is there fuzziness within these archetypes?
  - Do top players embody multiple archetypes? Are they "fuzzier"?

Data Set

- NBA player season data: over 20000 observations
- 40 variables
  - ID variables
  - Listed position
  - Box score statistics
  - Advanced metrics (incorporating multiple box score statistics)
- Courtesy of FiveThirtyEight

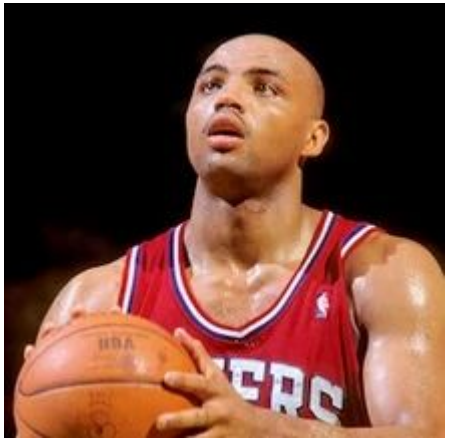
Data Cleaning

Many NBA players only play limited minutes. With such a large data set, we can afford to use only players who have played significant minutes. We'll drop players with less than 1500 minutes played over a season, leaving over 7000 observations. We'll also drop rows with NA values because of the size and nature of the analysis. Normally we would prefer to impute values, but for this task it's not necessary. This gets us to 6372 observations.

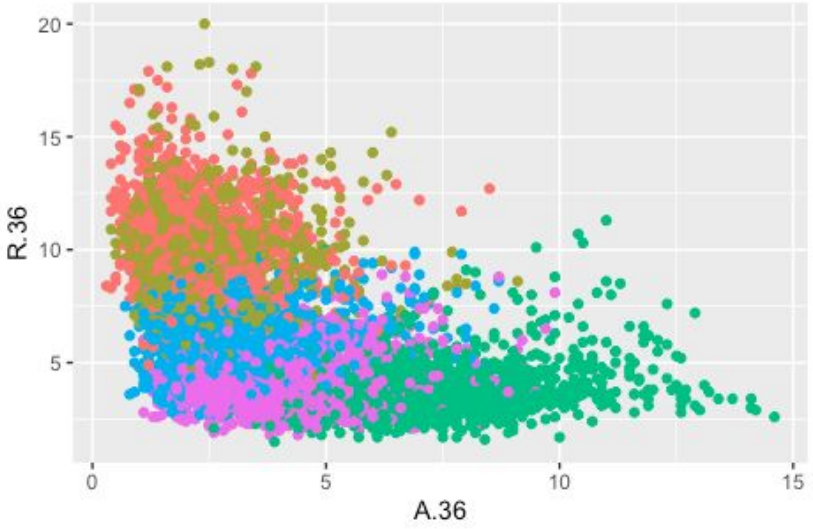
Implementing k-means in R

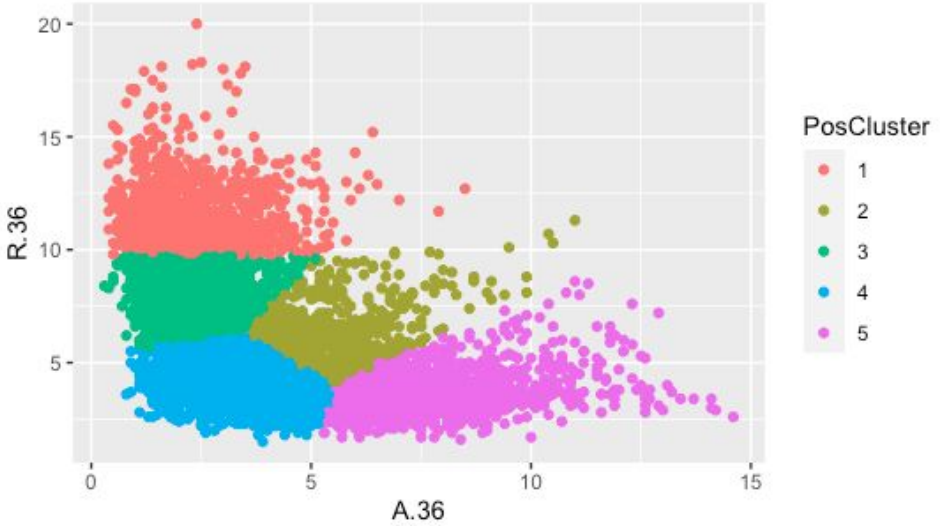
- The first call creates a k-means object with k=5. Columns 15-16 correspond to per 36 minute stats for rebounds and assists.
- Anecdotally guards tend to have more assists and centers tend to have more rebounds, so we think these will be good statistics for clustering.
- The second call adds the cluster number to our dataframe.

```
nbaData.km = kmeans(nbaData[,15:16],5)

nbaData$PosCluster = as.factor(nbaData.km$cluster)
```



Visualizing Our Clusters





	C	PF	PG	SF	SG
1	540	482	1	12	0
2	22	47	119	225	190
3	350	676	0	539	69
4	5	53	210	583	1069
5	0	1	1051	4	124

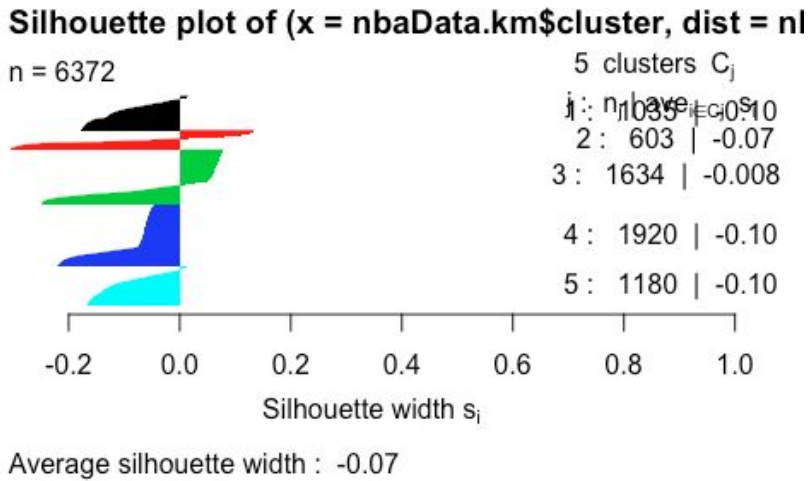
Our clusters appear somewhat similar to the actual positions, but there is a lot of category overlap between similar positions.

Using Silhouette in R

- Compute distance matrix (remove non numeric variables)
- Create silhouette
- Plot silhouette

```
nbadata.dist = dist(nbaData[,-c(1:2,5:6,8,41)],method="euclidean")

nbadata.sil = silhouette(nbaData.km$cluster,dist=nbadata.dist)

plot(nbadata.sil,col=1:5,border=NA)
```

This doesn't look nearly as good as our example silhouette plots! But real life data doesn't always fall into clean patterns. We can of course consider different combinations of predictors.



Clusters by Listed Position

Can we break a particular position apart by clusters? For instance, PG is the domain of some of the most dominant players in the NBA today. Players like Stephen Curry and Damian Lillard are considered to be “shoot-first” point guards, while players like Chris Paul and Ricky Rubio are more concerned with being top distributors. A few PGs, like Russell Westbrook and Ben Simmons, even play a lot of back to basket basketball like power forwards.

Using k-means++ in R

We'll use k-means++ in this example to help prevent sub-optimal clustering.

We'll take a subset of the data frame (PGs) and conduct analysis only on top players who have played more than 2500 minutes in a season. We'll see how well a 2 cluster split works. We choose the following stats:

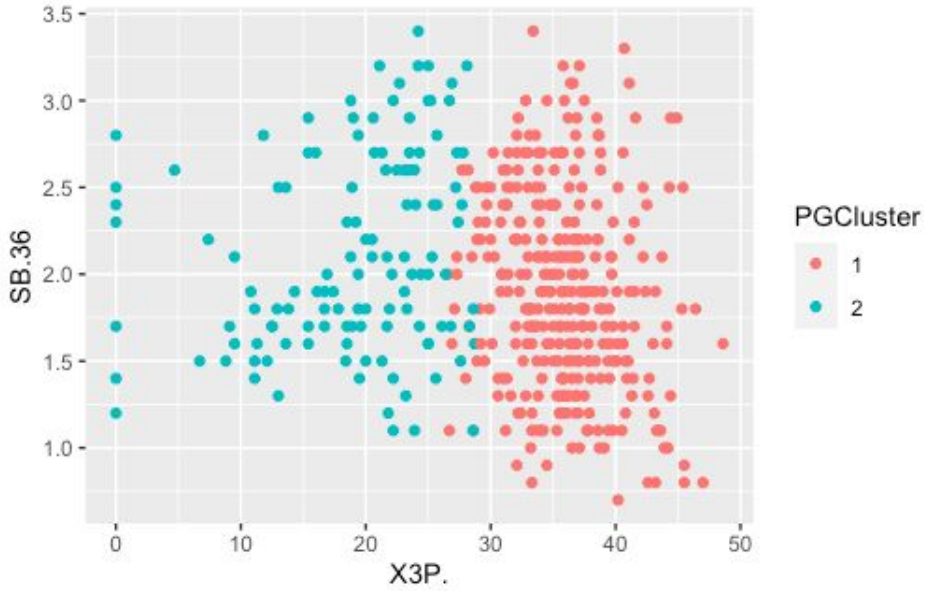
- Minutes per game, points per 36 mins, usage rate, assist, turnover, steal, block, offensive and defensive rebounding rates, 2 point, 3 point, and free throw percentages.

The syntax is then the same as k-means:

```
library(LICORS)

pgData = nbaData[nbaData$pos=="PG",]

pgData = pgData[pgData$Min>2500,]

kmeanspp(pgData[,c(12:13,25:29,31:32,36:38)],2)
```

Point Guard Clusters

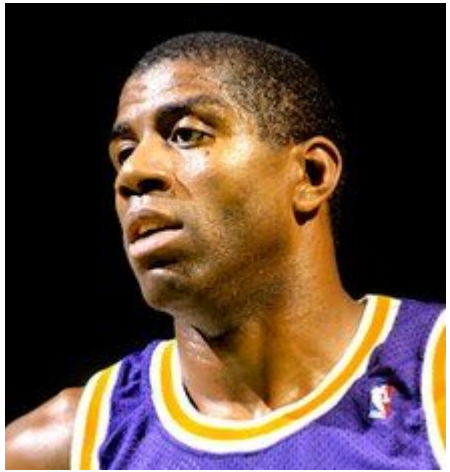
After using k-means++ to cluster top PGs into two groups using a variety of statistics, we wanted to visualize statistics that did a good job of separating them into clusters. We see here that 3-point percentage is a clear separator. In fact, we checked all of our other statistics against 3-point percentage and found similar graphs. It appears that 3-point percentage is the lone predictor that separated these clusters. Later in the presentation we note some key play-level statistics that would improve our PG clustering.



Optimal Number of Clusters

We've found that we can predict listed position fairly well using rebounds and assists per 36 minutes with k-means clustering. However, we see a lot of overlap among similar positions (particularly center and power forward). Our next task is to figure out how many statistical "roles" there are. We'll determine the optimal k by using silhouette. Here are the statistics we'll use:

- Minutes per game, points per 36 mins, usage rate, assist, turnover, steal, block, offensive and defensive rebounding rates, 2 point, 3 point, and free throw percentages.

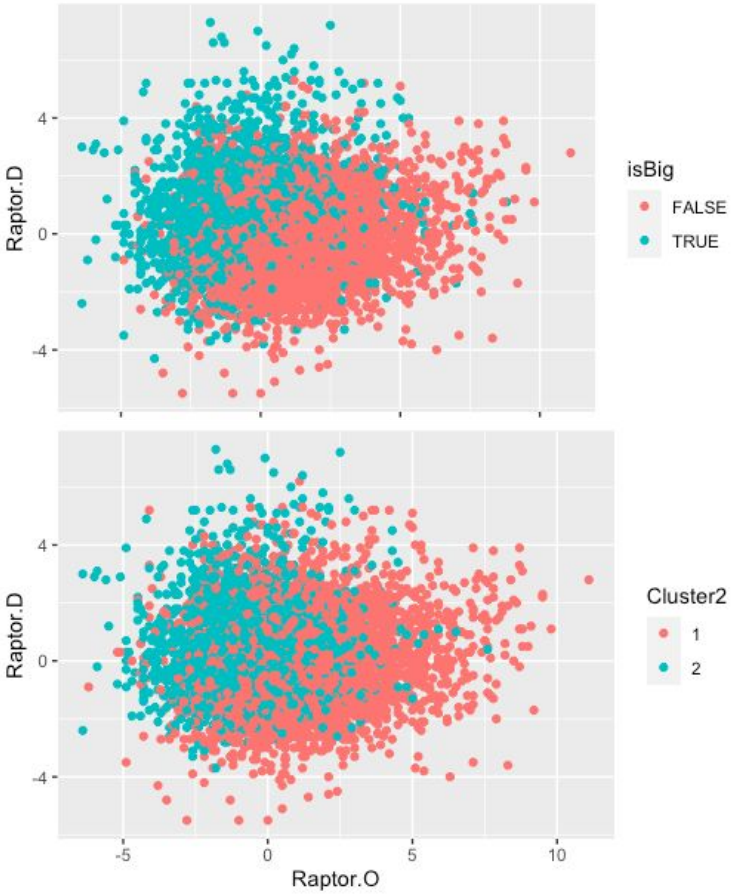


Earvin "Magic" Johnson, the original "positionless" player. Magic was a listed point guard, but started a game at center in the NBA Finals, the sport's biggest stage. At 6'9", his blend of speed, strength, height, and vision made him capable everywhere on the court.

Testing Different ks

We use our previous code to run the k-means algorithm and use silhouette to test the efficacy of our groupings. The k=2 cluster performed best, which is perhaps no surprise as we only have surface level data. 2 clusters will likely split our players into “bigs” (nominally PFs and Cs) and “smalls” (SGs and PGs). SFs often occupy both roles statistically. Including SFs in “smalls,” we tested group membership against isBig, a boolean indicating if the player’s listed position is PF or C. This clustering correctly identifies listed position for 78% of players.

	FALSE	TRUE
1	3763	974
2	433	1202



Applying Fuzzy Clustering

Expanding on our previous analysis, we want to apply c-means clustering with 2 clusters and see which positions have the most fuzziness. Based on our previous analysis, SF seems to be a good bet.

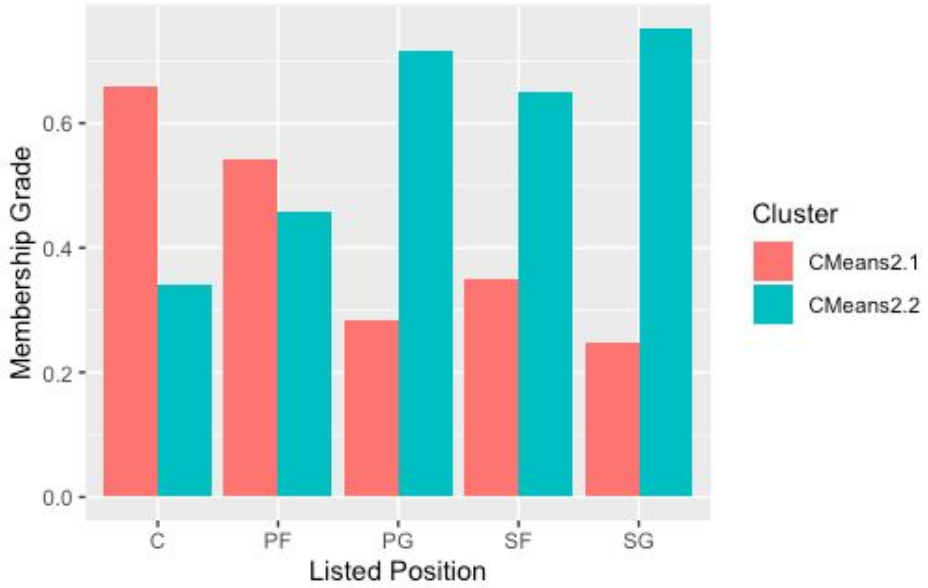
The cmeans syntax looks just like kmeans.

The difference is in the cmeans object: we must access the membership table and get the membership grades for each cluster.

```
nbaData.cm2 =
cmeans(nbaData[,c(12:13,25:29,31:32,36:38)],2)

nbaData$CMeans2.1 =
nbaData.cm2$membership[,1]

nbaData$CMeans2.2 =
nbaData.cm2$membership[,2]
```

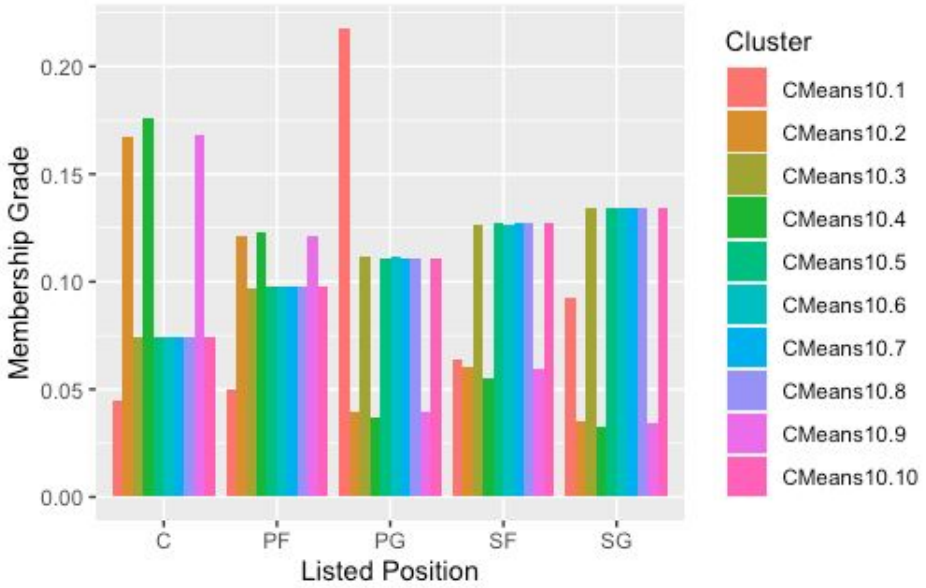
Visualizing Fuzzy Clustering

In comparing the average membership grades by position, we see that PG, SG, and C all appear to be well set in their clusters. Somewhat surprisingly, PF appears to have more crossover than SF. We can likely attribute this to the different cluster sizes, as cluster 2 is much larger.



Visualizing Fuzzy Clustering: c=10

We ran the same analysis with 10 clusters. Although 10 clusters were earlier shown to not perform particularly well under our silhouette, they are a useful tool for exploratory data analysis. We notice that PGs are easily identifiable and that Cs fall in a few groups. We can also affirm that it's difficult to separate some positions, particularly PF.



Introducing R.A.P.T.OR.

- Complex NBA player rating statistic that incorporates player tracking and play by play data
- Measures contribution to a team's offense and defense per 100 possessions
- Combines box score statistics with on/off evaluation of different player combinations
- Included in our data set

			RAPTOR			
NAME	SEASON	MIN. PLAYED	OFF.	DEF.	TOTAL	WAR
Stephen Curry	2016	3,314	+10.4	+2.1	+12.5	26.7
Chris Paul	2014	2,643	+7.7	+3.7	+11.4	19.3
Stephen Curry	2015	3,439	+8.6	+2.4	+11.0	25.1
James Harden	2019	3,291	+9.6	+1.1	+10.7	22.8
Chris Paul	2015	3,302	+8.6	+2.1	+10.7	22.6
James Harden	2018	3,172	+8.8	+1.3	+10.1	20.9
Kawhi Leonard	2016	2,719	+5.1	+4.7	+9.9	17.5

Does Fuzziness Increase with RAPTOR?

- R.A.P.T.O.R. is a measure of the “best” players.
- Is being good at multiple things a prerequisite for stardom in the NBA? Are the best players fuzzier? We’ll use 5 clusters, to represent the 5 listed positions, to test this theory.
- Let’s find the difference between the two highest cluster membership scores for each observation. We’ll see if that is correlated to overall R.A.P.T.O.R. +/-.

Results of our Analysis

- R.A.P.T.O.R. and difference between two highest cluster membership scores appear to be slightly negatively correlated (-0.07 with the c-means model being used, though this will vary slightly each time c-means is run). This means that better players tend to have lower differences between their two highest membership scores, which is equivalent to the skill overlap we postulated about earlier!

Significant?

With our large sample, even a small correlation can be statistically significant. Here we show through cor.test that we can reject the null hypothesis that the correlation between Raptor +/- and difference between top two membership scores is 0.

```
        Pearson's product-moment correlation

data:  nbaData$Raptor... and nbaData$CMeans5.Diff12
t = -5.9655, df = 6370, p-value = 2.57e-09
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.09890946 -0.05007380
sample estimates:
        cor
-0.07453632
```

Commentary On These Analyses

- This is a very informal analysis, meant to instruct about clustering methodology
- When using a variety of predictors for a cluster analysis, it might be a smart idea to perform similar model checks to a regression analysis
    - Multicollinearity in particular could have a strong effect, pushing some data points farther out from the center through its multiplicative effect

Improving This Analysis

- Though we are given R.A.P.T.O.R., which incorporates play-level data, we're not given that play-level data directly.
- Some play-level statistics that could be helpful in determining player archetype:
  - For our point guard example, percentage of shots taken from different floor locations would do a good job separating slashing point guards who get to the rim (Russell Westbrook, John Wall) and outside shooters (Stephen Curry, Damian Lillard).
  - Distance run, dribbles taken (high distance run and low dribbles taken would be a shooter who runs off screens, like Klay Thompson or J.J. Redick)
  - Post-up percentage (power forwards and centers generally play the most possessions with their back to the basket)
- Positions are not created equally in the NBA and often rotations include more depth at shooting guard and small forward. Adjusting our data set to have an equal number of players at each position might result in better clustering results.

Commentary on Cluster Analysis

- Clustering is a great tool for quick analysis
- Exploratory data analysis with clustering can be great, especially when presenting general concepts to non-technical stakeholders
- More formal statistical methods should be used to validate the ideas proposed by cluster analysis
- The clustering algorithms presented here are fairly simple compared to some of the more cutting-edge algorithms

Other Important Topics in Clustering

- Gaussian Mixture Models (distribution clustering)
- Hierarchical clustering methods
- Optimizations of the various clustering algorithms
  - We saw k-means++ as an example
- Gap statistic as a method for optimizing k

Resources: Clustering Methods

- Cluster analysis
- k-means clustering
- k-medians clustering
- k-medoids
- Fuzzy c-means clustering
- Silhouette (clustering)

Resources: Discussion and R Code

- How to understand the drawbacks of K-means
- Determining The Optimal Number Of Clusters: 3 Must Know Methods
- K Means Clustering in R
- cmeans() R function: Compute Fuzzy clustering
- Finding the column number and value the of second highest value in a row

Resources: Data Set and Basketball Background

- Introducing RAPTOR, Our New Metric For The Modern NBA
- How Our RAPTOR Metric Works
- Luka Dončić And The Mavs Are Pushing The Limits Of Offensive Efficiency
- fivethirtyeight/nba-player-advanced-metrics: Historical RAPTOR and other NBA data.
- Meet Mitch The Mop Guy | Atlanta Hawks
- The Book of Basketball