

## Answer Key: Unsupervised Learning II: Clustering

K-means clustering randomizes the initial values for optimization, so results may vary every-time it is run. This answer key uses a `set.seed(10)` to get consistent results.

**Problem statement:** A granting agency wants to identify colleges that have high numbers of low-income, and first generation college attendees to give those colleges additional funding.

1. In the clustering tutorial, we used k-means clustering to identify 3 clusters of colleges using these criteria.

A. Replicate this analysis using the code in the tutorial to generate those 3 clusters and append the cluster levels to the `college_features` dataset.

```
library(tidyverse)

### For reproducible results
set.seed(10)

### Load the colleges datasets on your machine
### colleges = read.delim("colleges.tsv", sep = '\t', header = TRUE)

college_features = colleges %>%
  select(institution_name, first_gen_share, poverty_rate, family_income_median,
         median_earnings, top_ten) %>%
  na.omit() %>%
  distinct()

kmeans_cluster = kmeans(select(college_features, -institution_name, -top_ten), 3)

college_features$cluster = kmeans_cluster$cluster
```

B. What is the median family income for each cluster (**hint:** see `kmeans_cluster$centers` from the tutorial)?

```
kmeans_cluster$centers

##   first_gen_share poverty_rate family_income_median median_earnings
## 1      0.4250142      8.579652          40221.74         41790.14
## 2      0.2808411      6.285941          75559.67         46343.79
## 3      0.5483327     12.901803          19231.73         25758.38
```

C. Subset the `colleges_features` dataset on the cluster with the lowest `family_income_median`, call this new data `grant_candidates`. **Note:** in the tutorial, `grant_candidates` were from Cluster 1, you could find that a different cluster from your analysis has the lowest `family_income_median` when you look at `kmeans_cluster$centers`.

```
grant_candidates = college_features %>% filter(cluster == 3)
```

D. How many universities are in the cluster of grant receivers?

```
dim(grant_candidates)
```

```
## [1] 2501    7
```

*2,501 are in the lowest family income cluster and would receive grants using this method.*

2. Upon review you're informed that there are too many universities receiving grants. The granting agency really likes the cluster approach but suggests you make 5 clusters instead of 3.

A. Redo the k-means analysis above but create 5 clusters instead of 3. **Note:** If you appended cluster onto your `college_features` dataset, make sure to remove it before redoing the k-means analysis.

```
### re-running this code so the cluster variable isn't included

college_features = colleges %>%
  select(institution_name, first_gen_share, poverty_rate, family_income_median,
         median_earnings, top_ten) %>%
  na.omit() %>%
  distinct()

kmeans_cluster = kmeans(select(college_features, -institution_name, -top_ten), 5)

college_features$cluster = kmeans_cluster$cluster
```

B. Again subset the data on the cluster with the lowest `family_income_median`. How many universities will receive a grant now? What is the median and range of `family_income_median` of these universities and how does it compare to your answers in Question 1?

```
### instead of printing the cluster results, you can find the minimum family income
### using programmatic methods
grant_candidates = college_features %>%
  filter(cluster == which.min(kmeans_cluster$centers[, "family_income_median"]))

dim(grant_candidates)
```

```
## [1] 31 7
```

*Compared to the 2,501 universities identified in 1D, there are only 31 that are in the cluster now.*

C. You will likely find that there were two clusters out of the five with low but similar `family_income_median`. Among these two clusters, what else determined which cluster these universities were assigned to (**hint:** look at the centers again)? Based on those other variables, do you think we made the correct decision to distribute grants considering only `family_income_median`?

```
kmeans_cluster$centers

##   first_gen_share poverty_rate family_income_median median_earnings
## 1      0.2478139      5.986602          82779.12         49646.10
## 2      0.2661290      8.406129          14966.27         126393.55
## 3      0.5632721     13.745615          17116.77         23933.89
## 4      0.4682306      9.409572          31499.25         36976.43
## 5      0.3728830      7.410169          55160.78         40199.76
```

*Despite having the lowest median family income at \$14,966, our grant cluster has the highest median earnings after graduation among all of the clusters. The cluster with the second lowest median family income (\$17,116) has the highest poverty rate and lowest post-graduation earnings, so would seemingly be a better target for a grant program.*

3. Hierarchical clustering: Part of the grant is to reformulate curriculums to better match top ten universities.

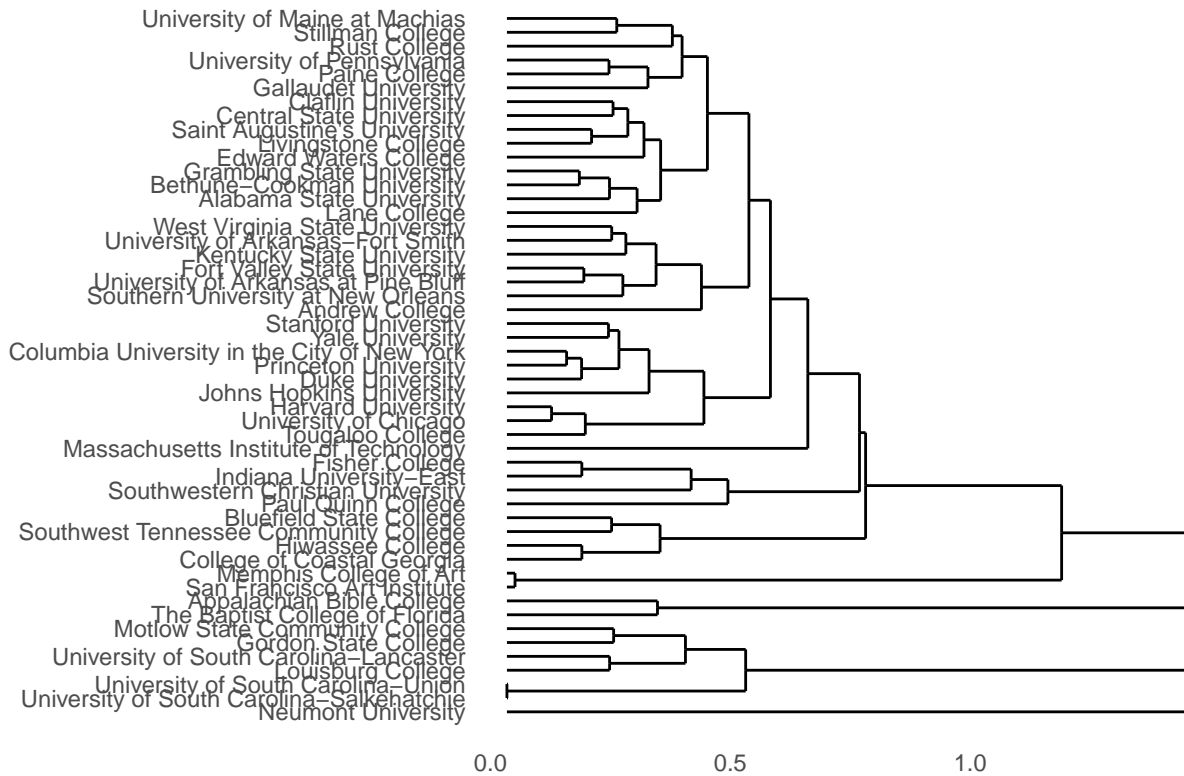
A. Subset your colleges dataset using the following code. The `!is.na(sat_verbal_quartile_1)` removes universities that do not have SAT admission criteria, so we are looking at similar degree-granting universities. What other criteria are we using to subset?

```
grant_colleges =  
  colleges %>%  
  filter(  
    !is.na(sat_verbal_quartile_1) & family_income_median < 40000 & median_earnings < 30000  
  )  
  
top_ten_schools = colleges %>% filter(top_ten == TRUE)  
  
heir_analysis_data = rbind(grant_colleges, top_ten_schools)
```

*We are also selecting colleges with median family income less than \$40,000 AND median earnings after graduation less than \$30,000.*

B. Replicate the heirarchical clustering from the tutorial comparing major percentages using `heir_analysis_data` dataset. Which universities are the most different from the top ten schools in terms of majors?

```
major_perc = heir_analysis_data %>%  
  select(institution_name, top_ten, contains("_major_perc")) %>%  
  na.omit()  
  
### Calculate distances  
euclidean = dist(select(major_perc, -institution_name, -top_ten), method = "euclidean")  
  
### hierarchical clustering  
hier = hclust(euclidean)  
  
### Relabel the nodes to be institution names  
hier$labels = major_perc$institution_name  
  
### plot using ggdendro  
library(ggdendro)  
ggdendrogram(hier, rotate = TRUE, size = 2)
```

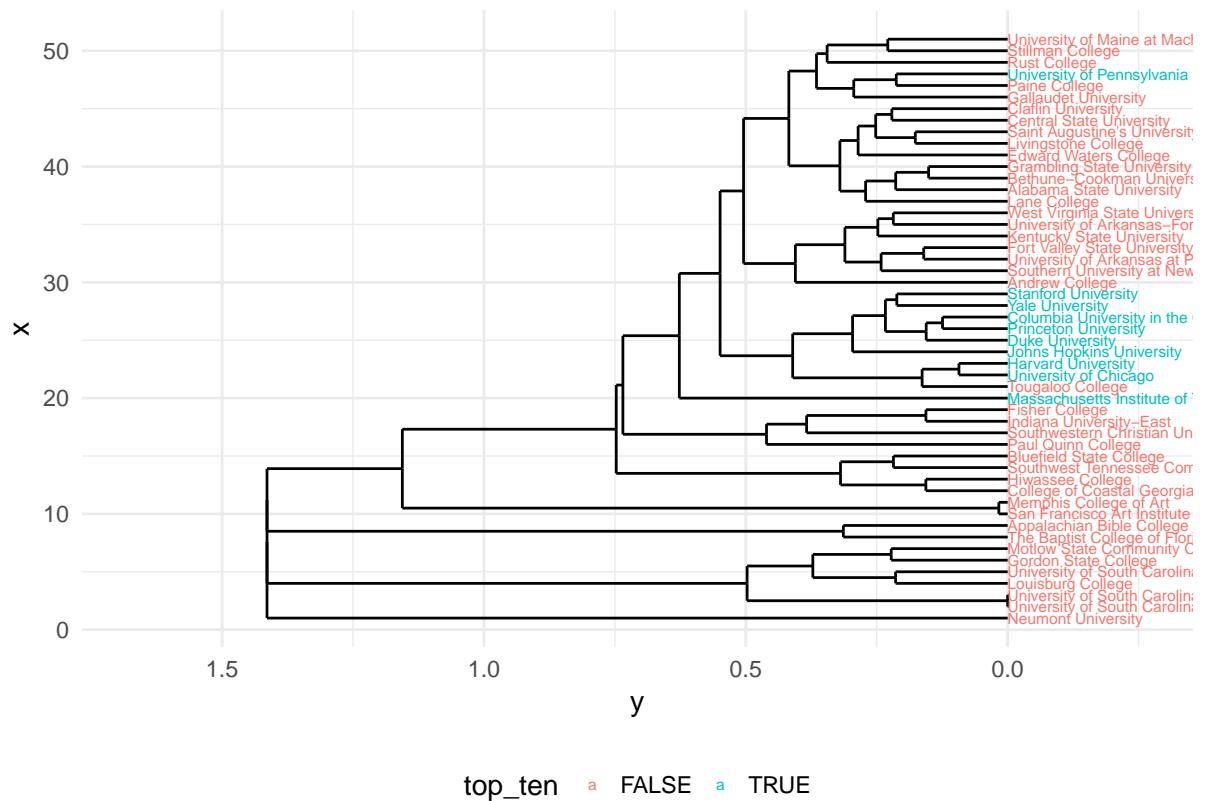


```
### plot using ggplot2

# necessary step for plotting with ggplot2
dendro_data = dendro_data(hier)

# merges on the top_ten variable for plotting
dendro_data$labels = dendro_data$labels %>%
  left_join(select(college_features, institution_name, top_ten),
    by = c("label" = "institution_name"))

ggplot(segment(dendro_data)) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_text(data = label(dendro_data),
    aes(label = label, x = x, y = 0, hjust = 0, color = top_ten), size = 2) +
  coord_flip() +
  scale_y_reverse(expand = c(0.25, 0)) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



**C.** How else can we compare the grantee schools to the top ten schools? Explore using any of the methods we learned in this class.

*Open question for exploration.*