727 HW5

Jay Kim

Github Link

```
cs_key <- read_file("API KEY.txt")

acs_il_c <- getCensus(name = "acs/acs5",
                      vintage = 2016,
                      vars = c("NAME",
                      "B01003_001E",
                      "B19013_001E",
                      "B19301_001E"),
                      region = "county:*",
                      regionin = "state:17",
                      key = cs_key) %>%
                      rename(pop = B01003_001E,
                      hh_income = B19013_001E,
                      income = B19301_001E)
head(acs_il_c)
```

```
  state county                          NAME    pop hh_income income
1    17    067   Hancock County, Illinois  18633     50077  25647
2    17    063    Grundy County, Illinois  50338     67162  30232
3    17    091  Kankakee County, Illinois 111493     54697  25111
4    17    043    DuPage County, Illinois 930514     81521  40547
5    17    003 Alexander County, Illinois   7051     29071  16067
6    17    129    Menard County, Illinois  12576     60420  31323
```

```
il_map <- map_data("county", region = "illinois")
head(il_map)
```

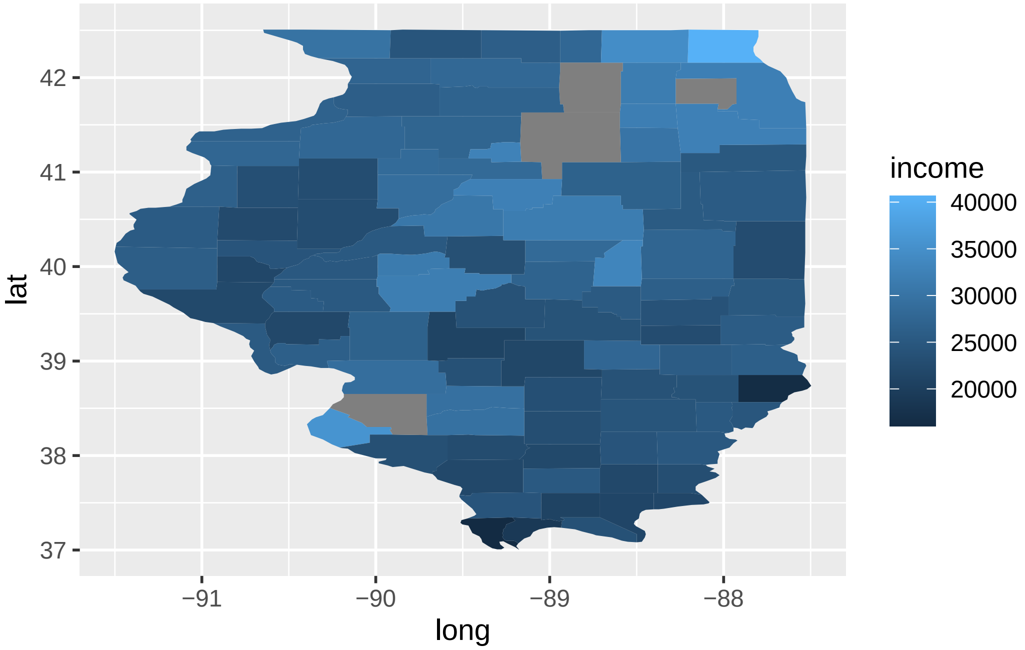
```
       long      lat group order   region subregion
```

```
1 -91.49563 40.21018     1     1 illinois     adams
2 -90.91121 40.19299     1     2 illinois     adams
3 -90.91121 40.19299     1     3 illinois     adams
4 -90.91121 40.10704     1     4 illinois     adams
5 -90.91121 39.83775     1     5 illinois     adams
6 -90.91694 39.75754     1     6 illinois     adams
```

```
acs_il_c <- acs_il_c %>%
  mutate(subregion = gsub(" County, Illinois", "", NAME) %>% tolower())

acs_map <- il_map %>%
  left_join(acs_il_c, by = "subregion")

head(acs_map)
```

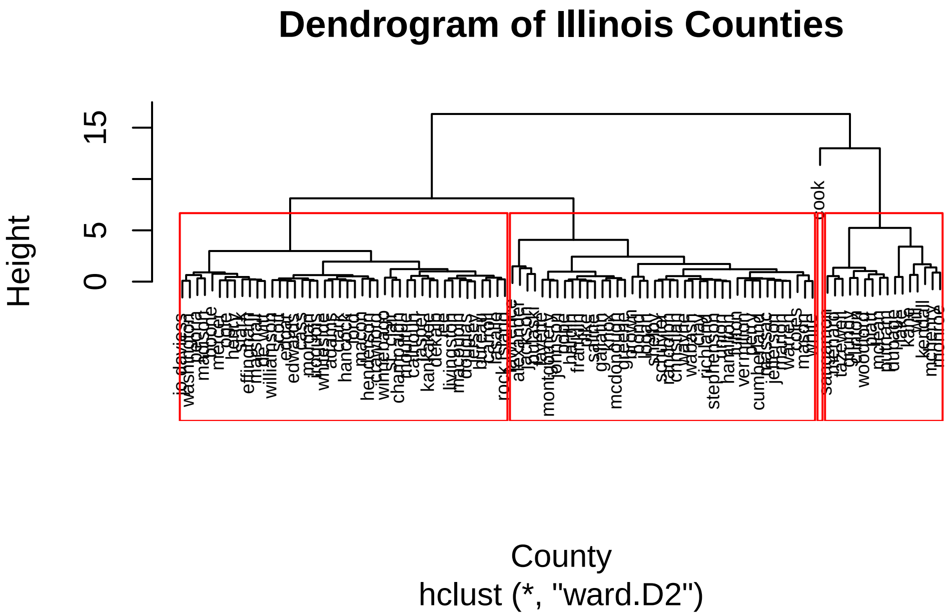
```
       long      lat group order   region subregion state county
1 -91.49563 40.21018     1     1 illinois     adams    17    001
2 -90.91121 40.19299     1     2 illinois     adams    17    001
3 -90.91121 40.19299     1     3 illinois     adams    17    001
4 -90.91121 40.10704     1     4 illinois     adams    17    001
5 -90.91121 39.83775     1     5 illinois     adams    17    001
6 -90.91694 39.75754     1     6 illinois     adams    17    001
                    NAME   pop hh_income income
1 Adams County, Illinois 66949     48065  26053
2 Adams County, Illinois 66949     48065  26053
3 Adams County, Illinois 66949     48065  26053
4 Adams County, Illinois 66949     48065  26053
5 Adams County, Illinois 66949     48065  26053
6 Adams County, Illinois 66949     48065  26053
```

```
ggplot(acs_map) +
geom_polygon(aes(x = long,
                 y = lat,
                 group = group,
                 fill = income))
```



```r
# Clean the data for clustering
# Extract unique county-level data (remove duplicate rows from map data)
acs_clean <- acs_il_c %>%
  select(subregion, pop, hh_income, income) %>%
  na.omit()  # Remove any rows with missing values

# Create a matrix with county names as row names for clustering
acs_matrix <- acs_clean %>%
  select(pop, hh_income, income) %>%
  scale()  # Standardize the variables

rownames(acs_matrix) <- acs_clean$subregion

# Create the distance matrix
dist_matrix <- dist(acs_matrix, method = "euclidean")

# Perform hierarchical clustering using Ward's method
hc_ward <- hclust(dist_matrix, method = "ward.D2")

# Plot the dendrogram
plot(hc_ward, main = "Dendrogram of Illinois Counties",
     xlab = "County", ylab = "Height", cex = 0.6)
```

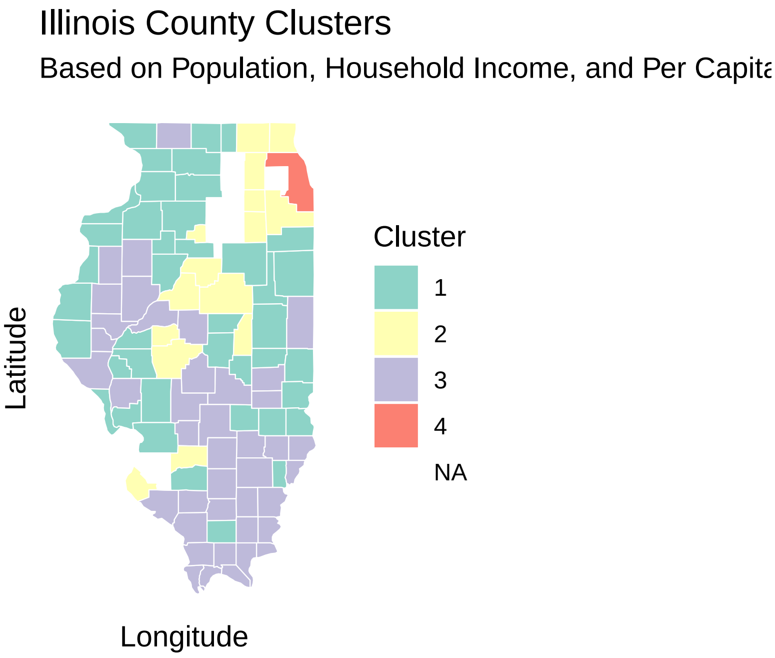
```
k <- 4

# Draw boxes around clusters
rect.hclust(hc_ward, k = k, border = "red")
```



```
# Cut the tree to create cluster assignments
clusters <- cutree(hc_ward, k = k)

# Create a data frame with cluster assignments
cluster_df <- data.frame(
  subregion = names(clusters),
  cluster = as.factor(clusters)
)

# Join cluster assignments with the original ACS data
acs_il_c <- acs_il_c %>%
  left_join(cluster_df, by = "subregion")

# Create new acs_map with cluster membership
acs_map <- il_map %>%
  left_join(acs_il_c, by = "subregion")
```

```r
# Visualize the clusters on a map
ggplot(acs_map, aes(x = long, y = lat, group = group, fill = cluster)) +
  geom_polygon(color = "white", size = 0.2) +
  coord_fixed(1.3) +
  scale_fill_brewer(palette = "Set3", name = "Cluster") +
  theme_minimal() +
  labs(title = "Illinois County Clusters",
       subtitle = "Based on Population, Household Income, and Per Capita Income",
       x = "Longitude", y = "Latitude") +
  theme(panel.grid = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank())
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

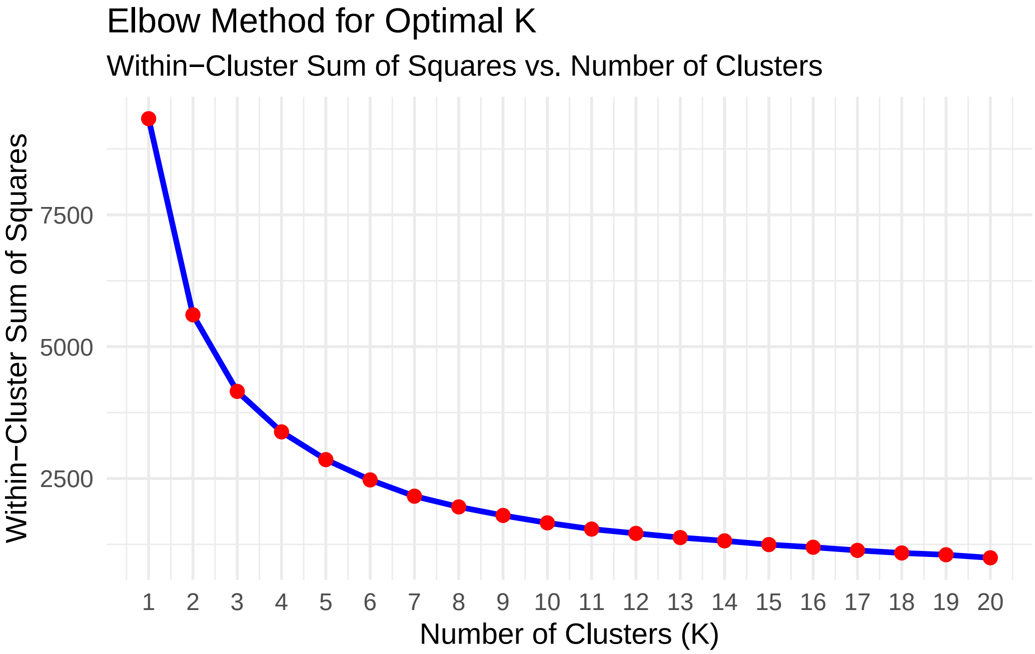


```r
acs_il_t <- getCensus(name = "acs/acs5",
                      vintage = 2016,
                      vars = c("NAME",
                               "B01003_001E",
                               "B19013_001E",
                               "B19301_001E"),
                      region = "tract:*",
                      regionin = "state:17",
                      key = cs_key) %>%
  mutate(across(everything(), ~ifelse(. == -666666666, NA, .))) %>%
  rename(pop = B01003_001E,
         hh_income = B19013_001E,
         income = B19301_001E)

head(acs_il_t)
```

```
  state county  tract                                     NAME  pop
1    17    031 806002 Census Tract 8060.02, Cook County, Illinois 7304
2    17    031 806003 Census Tract 8060.03, Cook County, Illinois 7577
3    17    031 806400    Census Tract 8064, Cook County, Illinois 2684
4    17    031 806501 Census Tract 8065.01, Cook County, Illinois 2590
5    17    031 750600    Census Tract 7506, Cook County, Illinois 3594
6    17    031 310200    Census Tract 3102, Cook County, Illinois 1521
  hh_income income
1     56975  23750
2     53769  25016
3     62750  30154
4     53583  20282
5     40125  18347
6     63250  31403
```

```
# Clean the data for clustering
acs_il_t_clean <- acs_il_t %>%
  select(NAME, state, county, tract, pop, hh_income, income) %>%
  na.omit()  # Remove rows with missing values

# Create a matrix for clustering (standardized)
acs_matrix_t <- acs_il_t_clean %>%
  select(pop, hh_income, income) %>%
  scale()

# Determine optimal K using within-cluster sum of squares
# Calculate WCSS for K = 1 to 20
set.seed(123)
wcss <- sapply(1:20, function(k) {
  kmeans(acs_matrix_t, centers = k, nstart = 25)$tot.withinss
})
```

```
# Plot the elbow curve
wcss_df <- data.frame(K = 1:20, WCSS = wcss)

ggplot(wcss_df, aes(x = K, y = WCSS)) +
  geom_line(color = "blue", size = 1) +
  geom_point(color = "red", size = 2) +
  labs(title = "Elbow Method for Optimal K",
       subtitle = "Within-Cluster Sum of Squares vs. Number of Clusters",
       x = "Number of Clusters (K)",
       y = "Within-Cluster Sum of Squares") +
  theme_minimal() +
  scale_x_continuous(breaks = 1:20)
```



```
# Run K-means with optimal K
# Adjust based on elbow plot
# 6 seems to be optimal
optimal_k <- 6
set.seed(123)
kmeans_result <- kmeans(acs_matrix_t, centers = optimal_k, nstart = 25)

# Create a temporary data frame with cluster membership for analysis
temp_clustered <- acs_il_t_clean %>%
  mutate(cluster = as.factor(kmeans_result$cluster))

# Find mean statistics and most frequent county by cluster
cluster_summary <- temp_clustered %>%
  group_by(cluster) %>%
  summarise(
    mean_pop = mean(pop, na.rm = TRUE),
    mean_hh_income = mean(hh_income, na.rm = TRUE),
    mean_income = mean(income, na.rm = TRUE),
    n_tracts = n()
  )

print("Cluster Summary Statistics:")
```

```
[1] "Cluster Summary Statistics:"
```

```
print(cluster_summary)
```

```
# A tibble: 6 x 5
  cluster mean_pop mean_hh_income mean_income n_tracts
  <fct>      <dbl>          <dbl>       <dbl>    <int>
1 1          4519.         92963.      45055.      527
2 2          3812.        135623.      77010.      154
3 3          5965.         53872.      24940.      690
4 4          2689.         32061.      17260.      764
5 5          3306.         58016.      29402.      914
6 6         11340.         93651.      39361.       60
```

```
# Find most frequent county in each cluster
most_frequent_county <- temp_clustered %>%
  group_by(cluster, county) %>%
  summarise(count = n(), .groups = "drop") %>%
  group_by(cluster) %>%
  slice_max(count, n = 1) %>%
  select(cluster, most_frequent_county = county, count)

print(most_frequent_county)
```

```
# A tibble: 6 x 3
# Groups:   cluster [6]
```

```
  cluster most_frequent_county count
  <fct>   <chr>                <int>
1 1       031                    220
2 2       031                     97
3 3       031                    326
4 4       031                    379
5 5       031                    282
6 6       197                     12
```

```r
# Combine summaries
full_summary <- cluster_summary %>%
  left_join(most_frequent_county, by = "cluster")

print(full_summary)
```

```
# A tibble: 6 x 7
  cluster mean_pop mean_hh_income mean_income n_tracts most_frequent_county
  <fct>      <dbl>          <dbl>       <dbl>    <int> <chr>
1 1          4519.         92963.      45055.      527 031
2 2          3812.        135623.      77010.      154 031
3 3          5965.         53872.      24940.      690 031
4 4          2689.         32061.      17260.      764 031
5 5          3306.         58016.      29402.      914 031
6 6         11340.         93651.      39361.       60 197
# i 1 more variable: count <int>
```

```r
# Create a function for K-means clustering
kmeans_function <- function(k, data = acs_matrix_t, seed = 123) {
  set.seed(seed)
  result <- kmeans(data, centers = k, nstart = 25)
  return(result$cluster)
}

# Iterate over multiple K values (K = 2 to 10)
k_values <- 2:10

# Apply the function for each K and create new columns
# Names is cluster_i for each iteration
for (k in k_values) {
  col_name <- paste0("cluster_", k)
  acs_il_t_clean[[col_name]] <- as.factor(kmeans_function(k))
}
```

```
# Display the first rows of the updated dataset
head(acs_il_t_clean)
```

```
                                         NAME state county  tract  pop
1 Census Tract 8060.02, Cook County, Illinois    17    031 806002 7304
2 Census Tract 8060.03, Cook County, Illinois    17    031 806003 7577
3    Census Tract 8064, Cook County, Illinois    17    031 806400 2684
4 Census Tract 8065.01, Cook County, Illinois    17    031 806501 2590
5    Census Tract 7506, Cook County, Illinois    17    031 750600 3594
6    Census Tract 3102, Cook County, Illinois    17    031 310200 1521
  hh_income income cluster_2 cluster_3 cluster_4 cluster_5 cluster_6 cluster_7
1     56975  23750         2         3         1         4         3         6
2     53769  25016         2         3         1         4         3         6
3     62750  30154         2         2         3         5         5         3
4     53583  20282         2         2         2         2         5         2
5     40125  18347         2         2         2         2         4         2
6     63250  31403         2         2         3         5         5         3
  cluster_8 cluster_9 cluster_10
1         8         7          8
2         8         7          8
3         3         6          7
4         3         6          7
5         1         4         10
6         3         6          7
```