Segmentation Example

Fareeza Khurshed

4/13/2021

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
##
##  ************************************************************
##           Loading standardize package version 0.2.2
##      Call standardize.news() to see new features/changes
##  ************************************************************
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   geo_code = col_character(),
##   FSA = col_character(),
##   alt_geo_code = col_character()
## )
## i Use `spec()` for the full column specifications.
```

Segmentation Examples

Examine the data and then do some data cleaning.

- Removes redundant variables from a contextually basis
- Filters data to Alberta to ensure relevance
- Ensure that no data is missing

This data is Census 2016 data

```
#remove columns that will not be used for modeling
dataP1 <- rawData %>%
  select(-c(year, geo_code, geo_level, GNR, GNR_LF, data_quality_flag, alt_geo_code,
            pct_age_0_14, pct_unemployed, pct_cdn_citizens, pct_immigrants, avg_census_family_size,
            median_income_recipients, pct_participation_rate, pct_home_owners, pct_education_none,
            num_dwellings_occupied ),
         -starts_with("pct_age_"),
         -starts_with("pct_LIM_AT_")
```

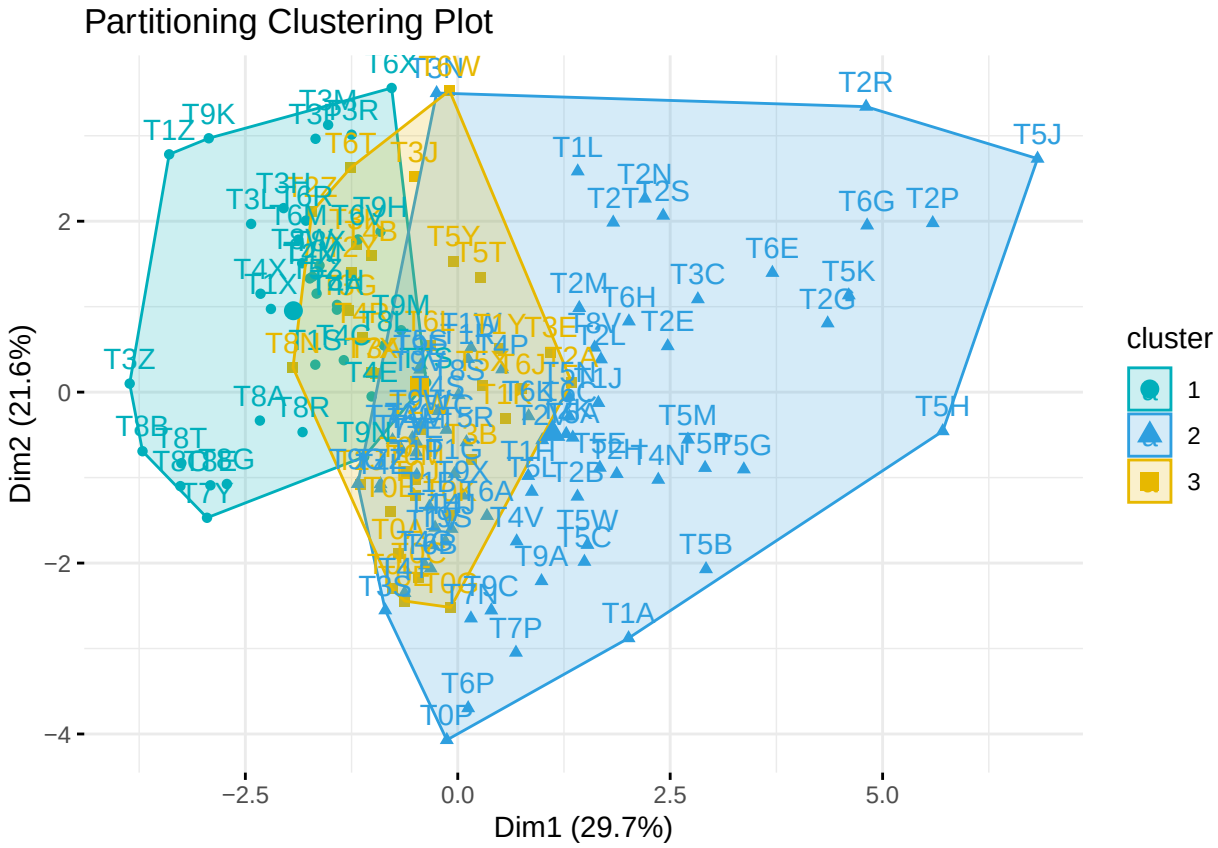
```
  ) %>%
  filter(substr(FSA, 1, 1) == "T") %>%
  column_to_rownames(var="FSA")

#remove all columns where missing values and convert to data frame
dataP2 <- dataP1 %>% filter(complete.cases(.))

summary(dataP2)
```

```
##     pop_2016     num_private_dwellings   median_age    avg_household_size
##  Min.   :  303   Min.   :  125         Min.   :28.20   Min.   :1.50
##  1st Qu.:12696   1st Qu.: 5433         1st Qu.:35.30   1st Qu.:2.40
##  Median :22496   Median :10001         Median :37.20   Median :2.60
##  Mean   :27270   Mean   :11089         Mean   :37.67   Mean   :2.59
##  3rd Qu.:37348   3rd Qu.:15770         3rd Qu.:39.90   3rd Qu.:2.80
##  Max.   :84336   Max.   :33105         Max.   :50.20   Max.   :4.90
##  median_income_household   pct_LIM_AT      pct_employed   pct_aboriginal
##  Min.   : 49981          Min.   : 3.200   Min.   :54.20   Min.   :0.00000
##  1st Qu.: 79574          1st Qu.: 6.100   1st Qu.:61.90   1st Qu.:0.03427
##  Median : 91351          Median : 8.900   Median :66.30   Median :0.05003
##  Mean   : 97418          Mean   : 9.417   Mean   :65.73   Mean   :0.06692
##  3rd Qu.:110569          3rd Qu.:11.900   3rd Qu.:68.60   3rd Qu.:0.07863
##  Max.   :216260          Max.   :24.000   Max.   :79.60   Max.   :0.69252
##  pct_visible_minority pct_home_rent    pct_education_uni_higher
##  Min.   :0.02091      Min.   :0.0000   Min.   : 0.2093
##  1st Qu.:0.06290      1st Qu.:0.1653   1st Qu.: 1.0111
##  Median :0.13726      Median :0.2237   Median : 1.9198
##  Mean   :0.20338      Mean   :0.2689   Mean   : 4.5690
##  3rd Qu.:0.30923      3rd Qu.:0.3535   3rd Qu.: 5.2376
##  Max.   :0.84685      Max.   :0.7626   Max.   :27.8310
##  pct_non_movers
##  Min.   :0.6456
##  1st Qu.:0.8168
##  Median :0.8439
##  Mean   :0.8395
##  3rd Qu.:0.8746
##  Max.   :0.9701
```

K Means Model

Trying it out without further processing:

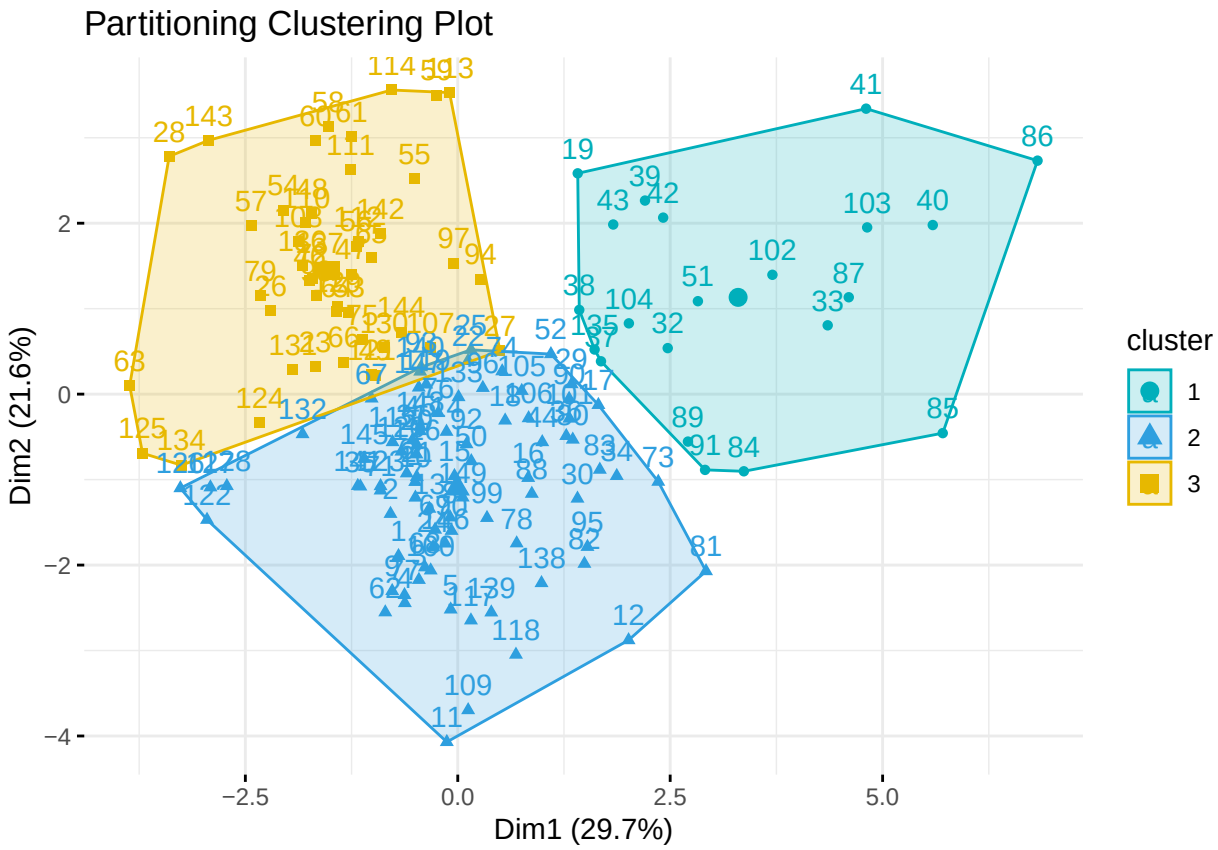
```
## Registered S3 methods overwritten by 'car':
##   method                          from
##   influence.merMod                lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod         lme4
##   dfbetas.influence.merMod        lme4
```



Standardization!

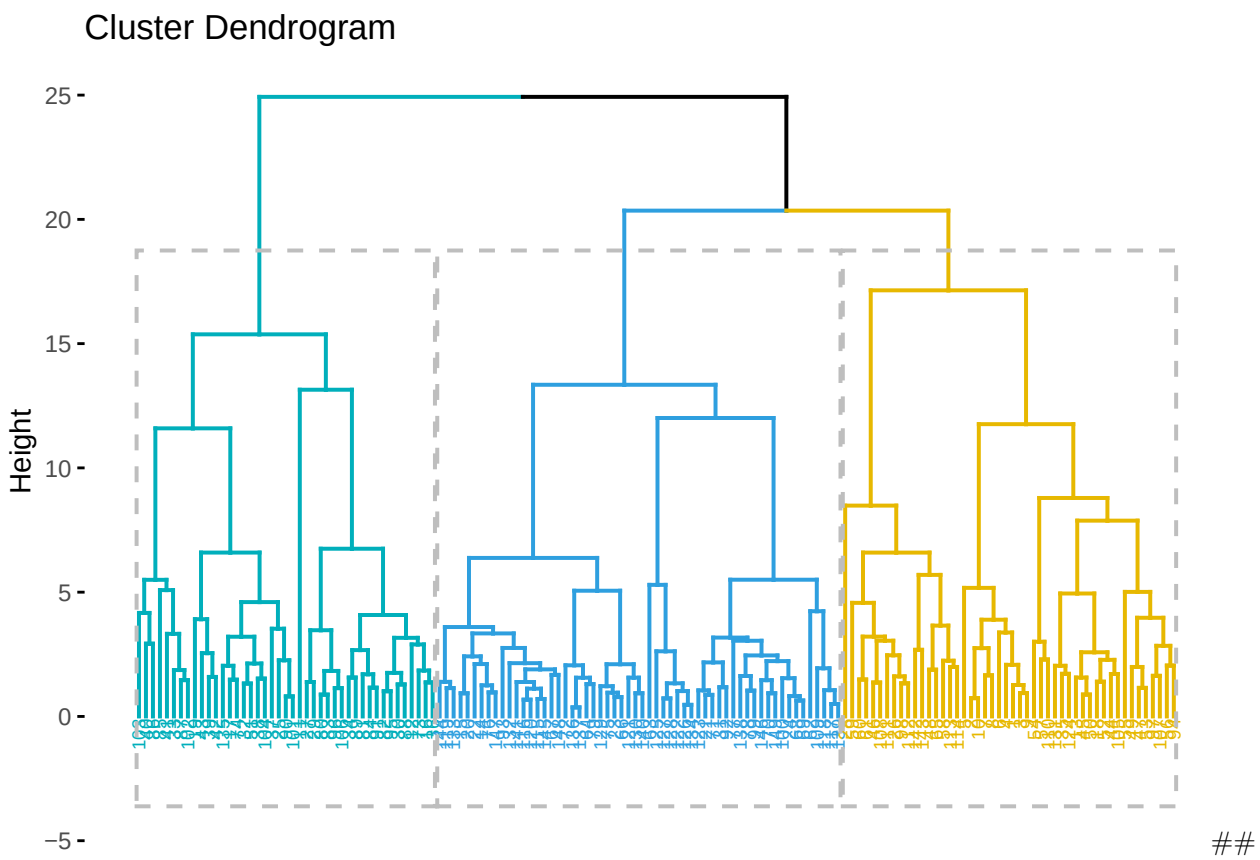
- Standardize variables and repeat

```
dataP3 <- as_tibble(scale(dataP2))

demo2 <- kmeans(dataP3, 3)

fviz_cluster(demo2, data = dataP3,
             palette = c("#00AFBB","#2E9FDF", "#E7B800"),
             ggtheme = theme_minimal(),
             main = "Partitioning Clustering Plot"
)
```



Hierarchical Modeling

```
# Compute hierarchical clustering and cut into 3 clusters
ha1_model <- hcut(dataP3, k = 3, stand = TRUE)

# Visualize
fviz_dend(ha1_model, rect = TRUE, cex = 0.5,
          k_colors = c("#00AFBB","#2E9FDF", "#E7B800"))
```



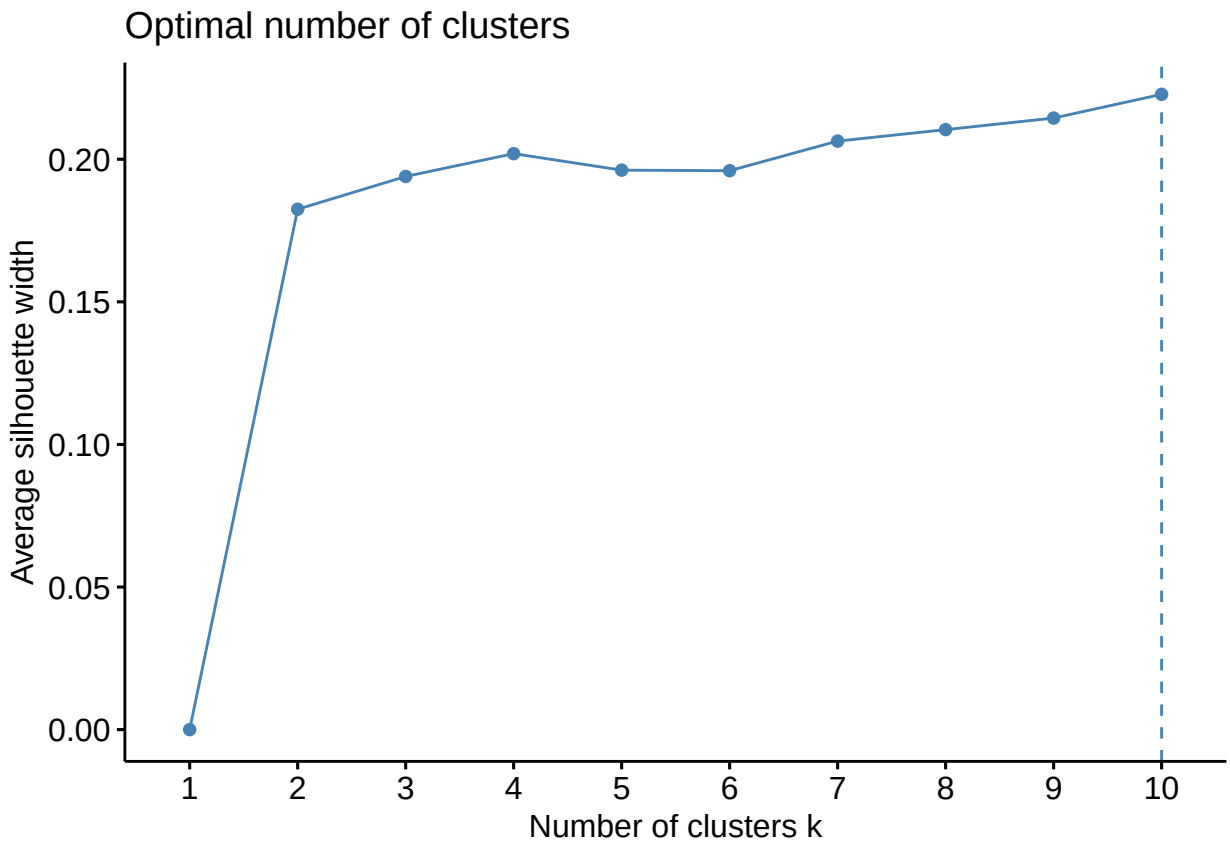
Results comparison

```
#add cluster to data
dataP4 <- dataP3
dataP4$cluster_kmeans <- demo2$cluster
dataP4$cluster_hier <- ha1_model$cluster

table(dataP4$cluster_kmeans, dataP4$cluster_hier)
```

```
##
##      1  2  3
##   1  0 21  0
##   2 18 21 43
##   3 30  1 15
```

Optimal Number of clusters?

```
#optimal number of clusters

fviz_nbclust(dataP3, kmeans, method = "silhouette")
```



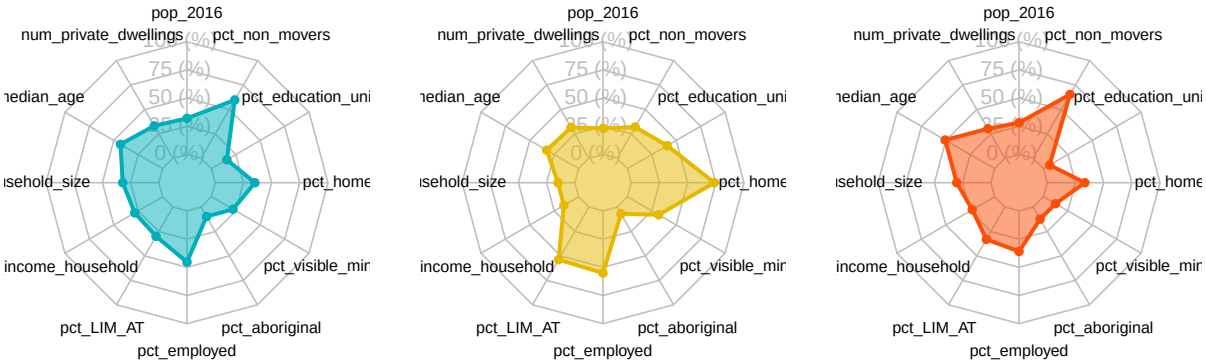
Now what?

- Visual your data

```
#source: https://www.datanovia.com/en/blog/beautiful-radar-chart-in-r-using-fmsb-and-ggplot-packages/

#visualize using a radar chart
df_scaled <- dataP3

# Variables summary
# Get the minimum and the max of every column
col_max <- apply(df_scaled, 2, max)
col_min <- apply(df_scaled, 2, min)
# Calculate the average profile
col_mean <- apply(df_scaled, 2, mean)

#cluster1

summary<-dataP4 %>%
  group_by(cluster_kmeans) %>%
  summarise(across(pop_2016:pct_non_movers, ~ mean(.x, na.rm = TRUE))) %>%
  select(-cluster_kmeans)

# Put together the summary of columns
col_summary <- t(data.frame(Max = col_max, Min = col_min, Average = col_mean))


# Bind variables summary to the data
```

```r
df_scaled2 <- as.data.frame(rbind(col_summary, summary))


# Define colors and titles
colors <- c("#00AFBB", "#E7B800", "#FC4E07")
titles <- c("Cluster 1", "Cluster 2", "Cluster 3")

# Reduce plot margin using par()
# Split the screen in 3 parts
op <- par(mar = c(1, 1, 1, 1))
par(mfrow = c(1,3))

# Create the radar chart
for(i in 1:3){
  create_beautiful_radarchart(
    data = df_scaled2[c(1, 2, i+2), ],
    color = colors[i], title = titles[i]
  )
}
```

Cluster 1 **Cluster 2** **Cluster 3**




```r
par(op)
```