APSTA-GE 2011. Project #2

January 2025: Yael Beshaw

```r
# likely useful libraries and initial seed set.
library(caret)
library(cluster)
library(NbClust)
library(klaR)
library(gtools)
library(ggplot2)
library(ggdendro)
library(GGally)
library(e1071)
library(knitr)
library(foreign)
library(fpc)
library(gridExtra)
library(LiblineaR)
library(factoextra)
library(tibble)
library(phyclust)
library(haven)
library(dplyr)
library(tidyverse)
library(tidyr)

set.seed(2011)
```

Introduction and Data Description

Health information refers to the data regarding your personal health including information about symptoms and/or outcomes that may be relevant to you. This data can come in the form of Electronic Health Records (EHR), results for lab tests and more. While physicians and clinicians are being trained in how to utilize these technologies, there is an increasing need for patients to be able to access this information in order to make personal health decisions. The COVID-19 pandemic highlighted this urgency as many were forced to utilize telehealth in lieu of seeing their physicians in person and relied on information from the news and social media to assess their risk of getting COVID-19. Now more than ever, it is imperative that healthcare, public health, and public policy professionals are able to create and enact interventions that address widespread and personal health concerns and outcomes. In order to do so, we require an understanding of how people interact with health information- both personal and general.

For this project, I utilize the Health Information National Trends Survey (HINTS) by the National Cancer Institute. This dataset collects survey data regarding how people access and utilize health information, utilizing a repeated cross-sectional survey. I utilized the HINTS6 dataset which collected data between March through November of 2022, in a two-stage design stratification method. The first stage required a stratified sample of residential addresses; stratified by rural versus urban and low minority versus high minority. Then, one adult (U.S. citizen, 18+, non-incarcerated) was randomly selected from each sampled household. Section

B of this dataset specifically focuses on the use of the internet for finding information. Variables in this section are measured on various scales, providing insights into patterns of health-related internet usage. Independent of the clustering we aim to do with Section B of this dataset, we also measure demographics such as race, sex, income, education, and health status for comparison. These variables allow exploration of whether the clusters differ significantly based on these demographics and assess potential associations between them and their cluster assignments. As such, our question of interest becomes whether unsupervised machine learning methods can identify distinct patterns in how individuals use the internet for health-related information?

Data Exploration and Transformation

Import the Dataset of Interest

```
hints6 <- read_sav("/Users/yaelbeshaw/R Scripts and Projects/NYU-APSTA-GE-2011/HINTS6_SPSS/hints6_publi
```

Extract HHID and Section B

```
project_data <- hints6 |>
    select(HHID, UseInternet, Internet_DialUp, Internet_HighSpeed,
        Electronic2_HealthInfo, Electronic2_MessageDoc, Electronic2_TestResults,
        Electronic2_MadeAppts, InternetConnection, ConfidentInternetHealth,
        HaveDevice_Tablet, HaveDevice_SmartPh, HaveDevice_CellPh,
        HaveDevice_None, UsedHealthWellnessApps2, WearableDevTrackHealth,
        FreqWearDevTrackHealth, WillingShareData_HCP, WillingShareData_Fam,
        SharedHealthDeviceInfo, SocMed_Visited, SocMed_SharedPers,
        SocMed_SharedGen, SocMed_Interacted, SocMed_WatchedVid,
        MisleadingHealthInfo, SocMed_MakeDecisions, SocMed_DiscussHCP,
        SocMed_TrueFalse, SocMed_SameViews)
```

Explore the Data

```
summaries <- summary(project_data[, c(2:30)])
```

Based on this no transformations are necessary as the majority of these variables are binary or ordinal and these scales are between 1 to 2 and 1 to 5. Standardizing would not be ideal as the standardized values would not be helpful in our understanding and interpretation of the data/results. The main concern is omitting NAs and only including complete cases as clustering is difficult to achieve with missing data.

Complete Cases Only

```
clean_data <- project_data |>
    mutate(across(everything(), ~na_if(., -9))) |>
    mutate(across(everything(), ~na_if(., -7))) |>
    mutate(across(everything(), ~na_if(., -6))) |>
    mutate(across(everything(), ~na_if(., -5))) |>
    mutate(across(everything(), ~na_if(., -4))) |>
    mutate(across(everything(), ~na_if(., -2))) |>
    mutate(across(everything(), ~na_if(., -1)))

data <- clean_data[complete.cases(clean_data), ]
```

Method

Since the data I am utilizing is binary and ordinal, it may be difficult to use kmeans clustering as it requires distance between points. This is a tool that especially useful for continuous data as there is a meaningful distance between each data point. However, in order to utilize this method, I utilize the binary and ordinal variables as psuedo-continuous variables as there is a meaningful distance between each increase in unit. For the binary variables, we have 1 = selected and 2 = not selected, every unit increase in my binary variables indicates a decreasing in technological utilization. The same can be said for our ordinal variables scaled 1-4 or 1-5, where 1 is always, every day, very confident, etc… whereas 5 reflects never or no utilization.Thus, I turn these variables into numeric variables in order to continue with the kmeans clustering methods.

However, after doing additional research, I found that Gower's distance would be helpful in cluster analysis for mixed-type objects which we have with the binary and ordinal variables. Therefore, in this project we will compare utilizing kmeans with our psuedo-continuous variables and hierarchical clustering using Gower's distance with our original variable types (binary/numeric and ordinal). Based on the sources provided below, I opted to utilize daisy() from the cluster package for "flexibility" in calculating this distance.

For hierarchical clustering, I will compare results between complete and single linkage across different k values. Next, for Kmeans, I will utilize the NbClust() package to search for the optimal number of cluster utilizing C(g). Lastly, I aim to assess the agreement between these two methods and ultimately choose the best method to evaluate the clusters againsttheir demographics.

Sources:

- https://crispinagar.github.io/blogs/gower-distance.html
- https://www.rdocumentation.org/packages/StatMatch/versions/1.4.3/topics/gower.dist
- https://stats.stackexchange.com/questions/349591/how-to-use-gowers-distance-with-clustering-algorithms-in-python
- https://stats.stackexchange.com/questions/123624/gower-distance-with-r-functions-gower-dist-and-daisy

Method Application and Results

Hierarchial Clustering with Gower's Distance

Adjust Variable Type

```
# Identify the Ordinal Variables
ordinal_columns <- c("InternetConnection", "ConfidentInternetHealth",
    "FreqWearDevTrackHealth", "SocMed_Visited", "SocMed_SharedPers",
    "SocMed_SharedGen", "SocMed_Interacted", "SocMed_WatchedVid",
    "MisleadingHealthInfo", "SocMed_MakeDecisions", "SocMed_DiscussHCP",
    "SocMed_TrueFalse", "SocMed_SameViews")

# Convert Binary Variables to numeric except the specified
# ordinal ones
data_new <- data[, -1] |>
    mutate(across(.cols = -all_of(ordinal_columns), .fns = ~as.numeric(.)))

# Ensure that the ordinal variables remain ordinal
data_new <- data_new |>
    mutate(across(.cols = all_of(ordinal_columns), .fns = ~as.factor(.)))
```

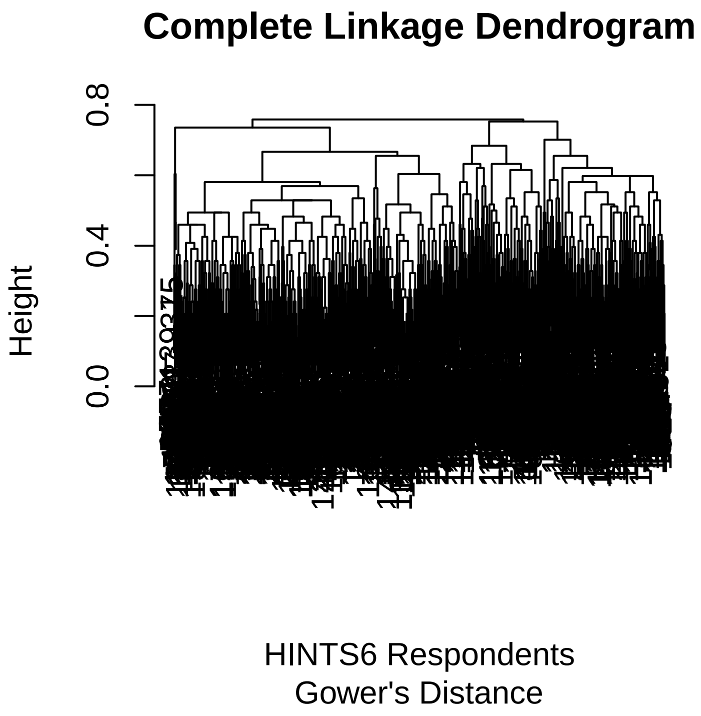
Distance Calculation

```
gower_dist <- daisy(data_new, metric = "gower")
```

Comparision of Complete and Single Linkage

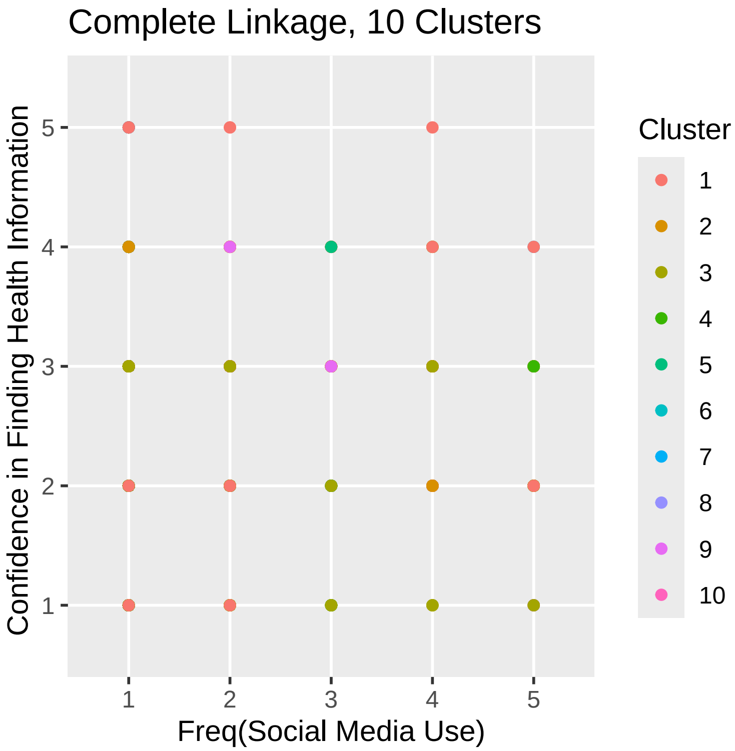
Complete Linkage Create Dendrogram to Estimate # of Clusters

```
# complete linkage hierarchical clustering
hcl_complete <- hclust(gower_dist, meth = "complete")

# dendrogram
plot(hcl_complete, main = "Complete Linkage Dendrogram", xlab = "HINTS6 Respondents",
    sub = "Gower's Distance")
```

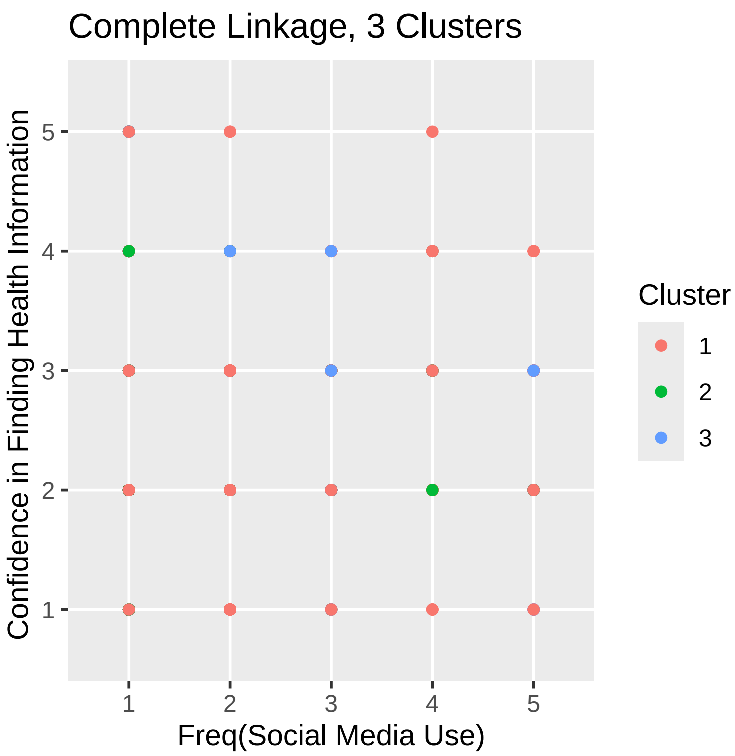


Based on the dendrogram it seems that k=10 would be reasonable in this situation given that we also have about 29 variables.

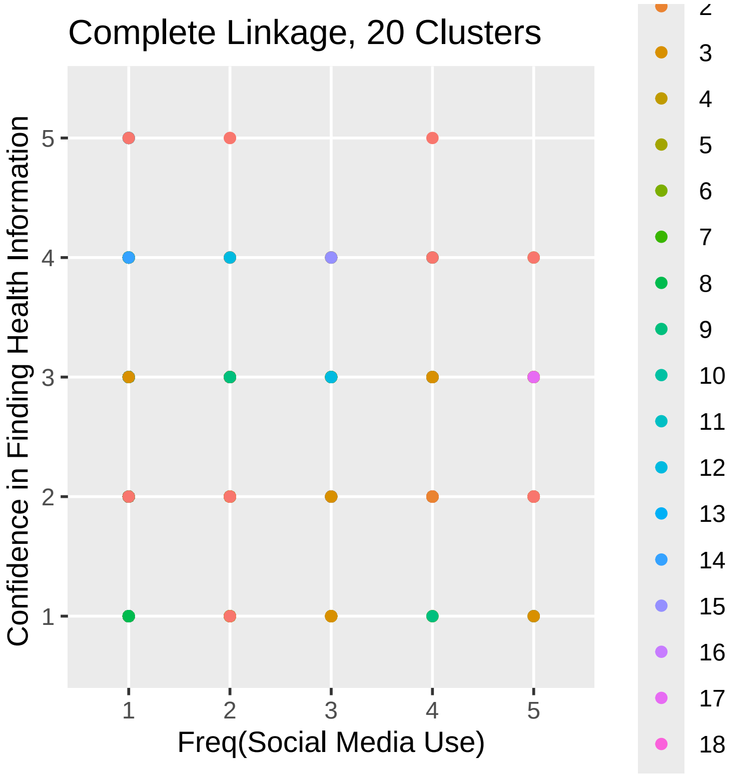
```
# k = 10
cluster10 <- cutree(hcl_complete, 10)
data_new$cluster10 <- cluster10

ggplot(data = data_new, aes(x = SocMed_Visited, y = ConfidentInternetHealth,
    color = factor(cluster10))) + geom_point() + labs(title = "Complete Linkage, 10 Clusters",
    x = "Freq(Social Media Use)", y = "Confidence in Finding Health Information ",
    color = "Cluster")
```



```
# k = 3 for comparision of very small k
cluster3 <- cutree(hcl_complete, 3)
data_new$cluster3 <- cluster3

ggplot(data = data_new, aes(x = SocMed_Visited, y = ConfidentInternetHealth,
    color = factor(cluster3))) + geom_point() + labs(title = "Complete Linkage, 3 Clusters",
    x = "Freq(Social Media Use)", y = "Confidence in Finding Health Information ",
    color = "Cluster")
```

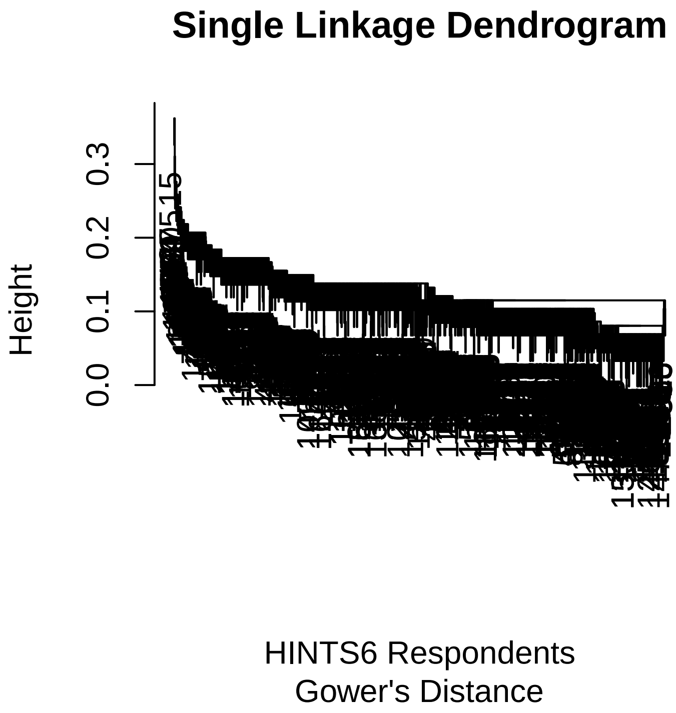


```r
# k = 20 for comparision of very large k
cluster20 <- cutree(hcl_complete, 20)
data_new$cluster20 <- cluster20

ggplot(data = data_new, aes(x = SocMed_Visited, y = ConfidentInternetHealth,
    color = factor(cluster20))) + geom_point() + labs(title = "Complete Linkage, 20 Clusters",
    x = "Freq(Social Media Use)", y = "Confidence in Finding Health Information ",
    color = "Cluster")
```



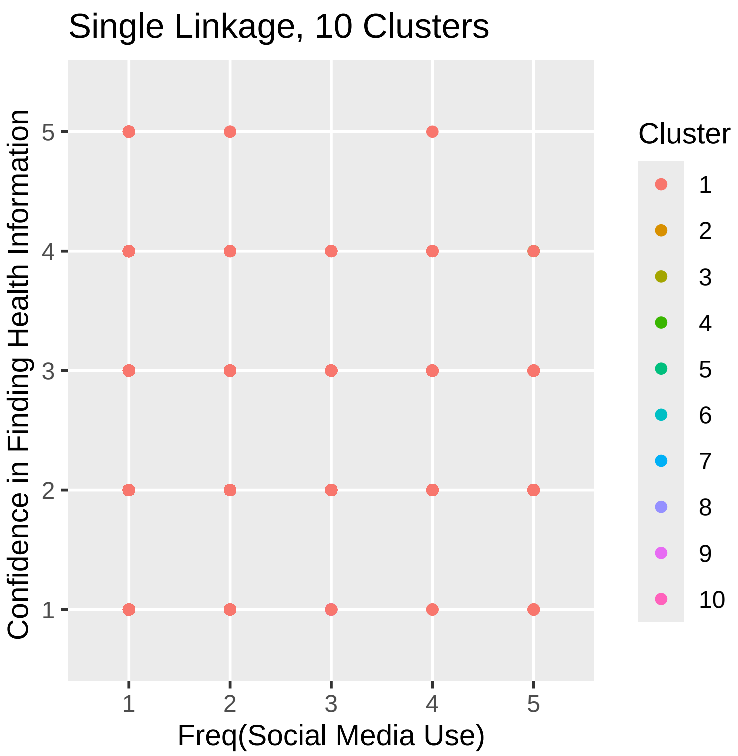
Single Linkage Create Dendrogram to Estimate # of Clusters

```
# complete linkage hierarchical clustering
hcl_single <- hclust(gower_dist, meth = "single")

# dendrogram
plot(hcl_single, main = "Single Linkage Dendrogram", xlab = "HINTS6 Respondents",
    sub = "Gower's Distance")
```

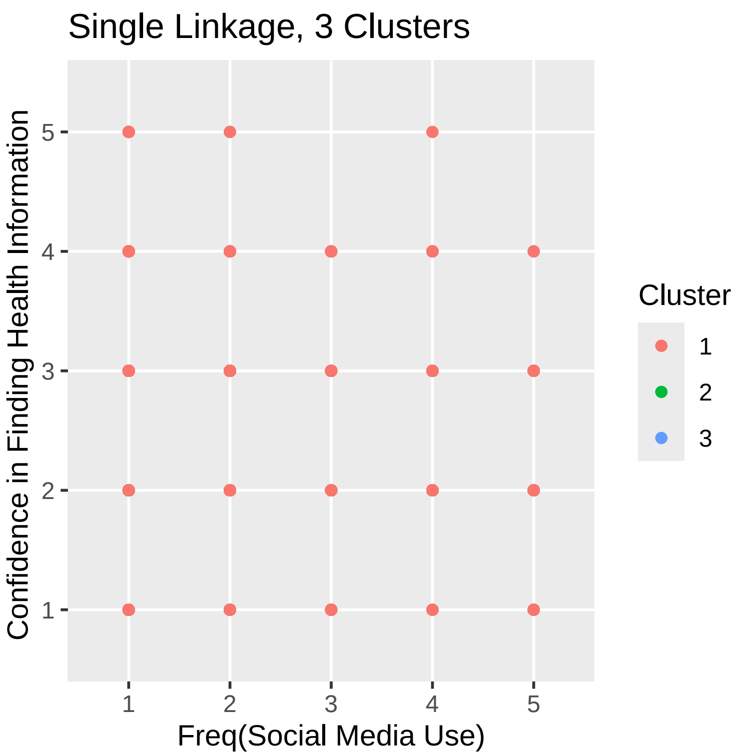


The dendrogram for the single linkage hierarchical clustering does not provide us with clear information about how many clusters to use compared to complete linakge clustering. However, based on the dendrogram, about 9-10 clusters seems reasonable.

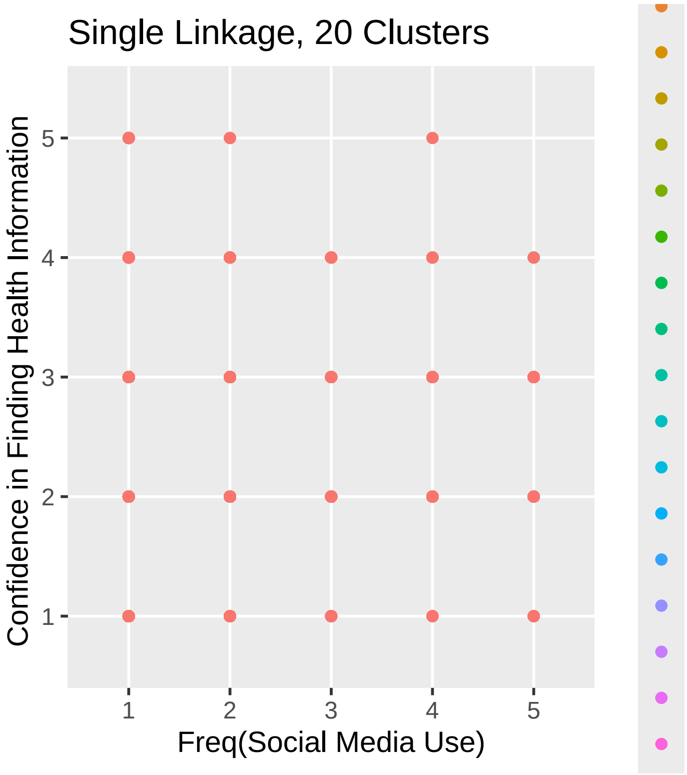
```
# k= 10
cluster10.single <- cutree(hcl_single, 10)
data_new$cluster10.single <- cluster10.single

ggplot(data = data_new, aes(x = SocMed_Visited, y = ConfidentInternetHealth,
    color = factor(cluster10.single))) + geom_point() + labs(title = "Single Linkage, 10 Clusters",
    x = "Freq(Social Media Use)", y = "Confidence in Finding Health Information ",
    color = "Cluster")
```



```r
# k= 3
cluster3.single <- cutree(hcl_single, 3)
data_new$cluster3.single <- cluster3.single

ggplot(data = data_new, aes(x = SocMed_Visited, y = ConfidentInternetHealth,
    color = factor(cluster3.single))) + geom_point() + labs(title = "Single Linkage, 3 Clusters",
    x = "Freq(Social Media Use)", y = "Confidence in Finding Health Information ",
    color = "Cluster")
```



```
# k= 20
cluster20.single <- cutree(hcl_single, 20)
data_new$cluster20.single <- cluster20.single

ggplot(data = data_new, aes(x = SocMed_Visited, y = ConfidentInternetHealth,
    color = factor(cluster20.single))) + geom_point() + labs(title = "Single Linkage, 20 Clusters",
    x = "Freq(Social Media Use)", y = "Confidence in Finding Health Information ",
    color = "Cluster")
```



Based on the results of the Confidence in Finding Health Information vs Freq(Social Media Visits) plots across different K's and between single vs complete linkage, it seems as though single linkage assigns more respondents in the same cluster compared to complete linkage which is much more diverse in its cluster assignment. Again due to the ordinal and binary nature of the variables in this dataset, the visualizations are difficult to interpret. Therefore, we look at the tables and crosstables between these methods. We see in the tables below that none of the k's I selected produce approximately equal-sized clusters. However, we see that single linkage clustering provides the most skewed results as observed in the plots as well. When analyzing the the complete linkage clustering method, k = 10 is the solution that produces the most similarly sized clusters but with the majority being distributed between across the first 5 clusters.

```
table(data_new$cluster3)
```

```
##
##   1   2   3
## 965 413 284
```

```
table(data_new$cluster10)
```

```
##
##   1   2   3   4   5   6   7   8   9  10
## 260 348 669 101 131  56   7   9  52  29
```

```
table(data_new$cluster20)
```

```
##
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
## 154 206 448  24  56  34   2 106 221   9  80  52  23  60  75  29  54   5   2  22
```

```
table(data_new$cluster3.single)
```

```
##
##    1    2    3
## 1660    1    1
```

```r
table(data_new$cluster10.single)
```

```
## 
##    1    2    3    4    5    6    7    8    9   10 
## 1653    1    1    1    1    1    1    1    1    1
```

```r
table(data_new$cluster20.single)
```

```
## 
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16 
## 1643    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##   17   18   19   20 
##    1    1    1    1
```

Therefore, I assess k= 5 and 6 for complete linkage instead. We see below that overall, k = 6 is the closest to producing the most similarly sized clusters here. Thus, we can conclude that for hclust() using Gower's Distance, k = 6 is the most optimal.

```r
cluster5 <- cutree(hcl_complete, 5)
data_new$cluster5 <- cluster5
table(data_new$cluster5)
```

```
## 
##   1   2   3   4   5 
## 958 404 284   7   9
```

```r
cluster6 <- cutree(hcl_complete, 6)
data_new$cluster6 <- cluster6
table(data_new$cluster6)
```

```
## 
##   1   2   3   4   5   6 
## 958 404 101 183   7   9
```

Kmeans

```r
data_num <- data |>
    mutate(across(.cols = -1, .fns = ~as.numeric(.)))
```

Adjust the Variable Type to all Numeric

Redo HClust with Numeric Variables Based on the dendrograms and assessments of the hierarchical clustering methods, it seems that k=6 is the most optimal number of clusters. However, because I have converted the variables into all numeric for this analysis, we repeat the above steps but with Eucledian distance instead.

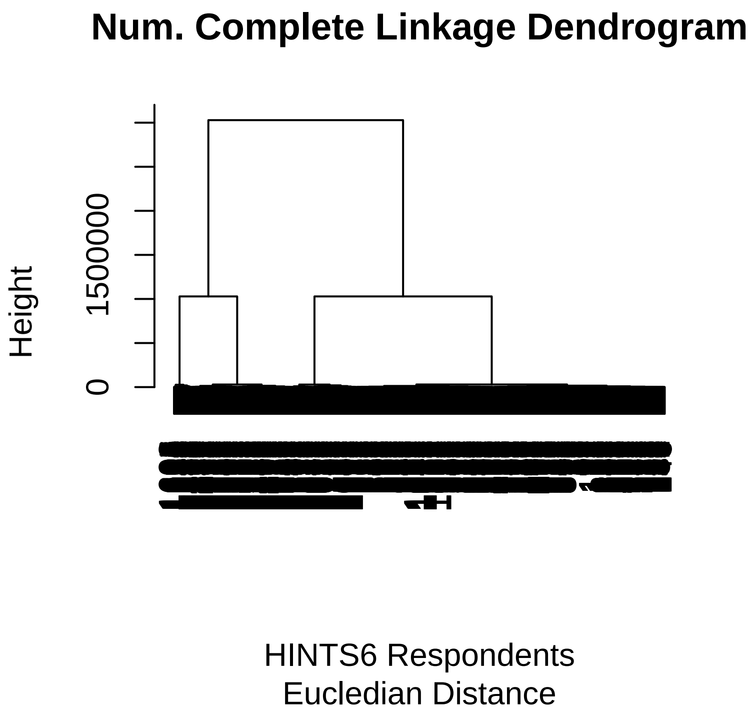
```r
num_dist <- dist(data_num, method = "euclidean")
```

Eucledian Distance Calculation

Comparision of Complete and Single Linkage

Complete Linkage Create Dendrogram to Estimate # of Clusters

```
# complete linkage hierarchical clustering
num_complete <- hclust(num_dist, meth = "complete")

# dendrogram
plot(num_complete, main = "Num. Complete Linkage Dendrogram",
    xlab = "HINTS6 Respondents", sub = "Eucledian Distance")
```

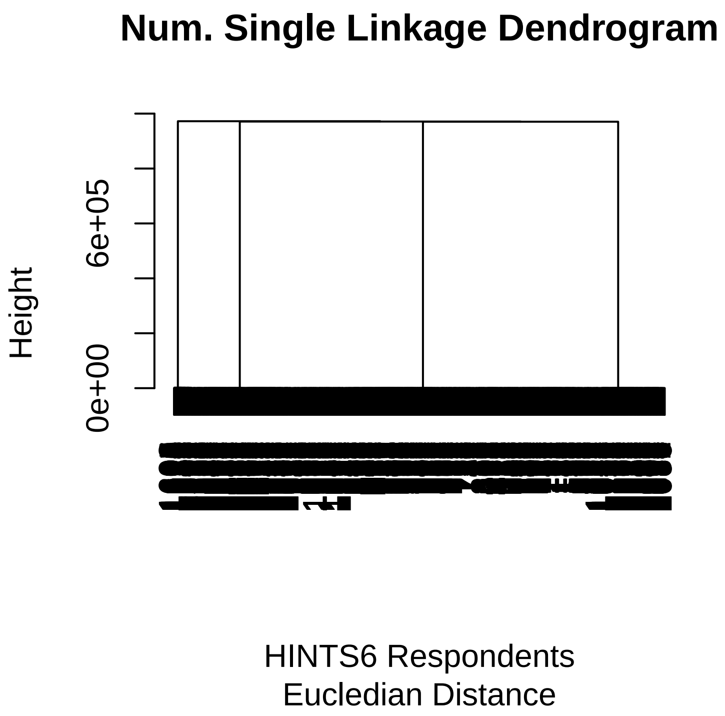


Based on the dendrogram it seems that k= 5 would be reasonable.

```
# k = 5
cluster5 <- cutree(num_complete, 5)
data_num$cluster5 <- cluster5

# k = 3 for comparison of very small k
cluster3 <- cutree(num_complete, 3)
data_num$cluster3 <- cluster3


# k = 10 for comparision of very large k
cluster10 <- cutree(num_complete, 10)
data_num$cluster10 <- cluster10
```

Single Linkage Create Dendrogram to Estimate # of Clusters

```
# single linkage hierarchical clustering
num_single <- hclust(num_dist, meth = "single")

# dendrogram
plot(num_single, main = "Num. Single Linkage Dendrogram", xlab = "HINTS6 Respondents",
     sub = "Eucledian Distance")
```



The dendrogram for the single linkage hierarchical clustering does not provide us with clear information about how many clusters to use compared to complete linakge clustering. However, based on the dendrogram, about 2 clusters seems reasonable.

```
# k = 2
cluster2.single <- cutree(num_single, 2)
data_num$cluster2.single <- cluster2.single

# k = 5 for comparison of very bigger k
cluster5.single <- cutree(num_single, 5)
data_num$cluster5.single <- cluster5.single

# k = 10 for comparision of large k
cluster10.single <- cutree(num_single, 10)
data_num$cluster10.single <- cluster10.single
```

```
table(data_num$cluster5)
```

Compare

```
##
##   1   2   3   4   5
## 580 465 218 339  60
```

```
table(data_num$cluster3)
```

```
##
##    1    2    3
## 1045  218  399
```

```
table(data_num$cluster10)
```

```
##
##   1   2   3   4   5   6   7   8   9  10
## 580 465  51  95  72 184 155  14  26  20
```

```
table(data_num$cluster2.single)
```

```
##
##    1    2
## 1602   60
```

```
table(data_num$cluster5.single)
```

```
##
##    1    2    3    4    5
## 1045  218  339    6   54
```

```
table(data_num$cluster10.single)
```

```
##
##    1    2    3    4    5    6    7    8    9   10
## 1045  218  339    6    3    5   10    7   26    3
```

Based on the table above , 3 to 10 clusters utilizing complete linkage would be reasonable to assess in order to find the optimal k-means.

Kmeans analysis using C(g) and Complete Linakge

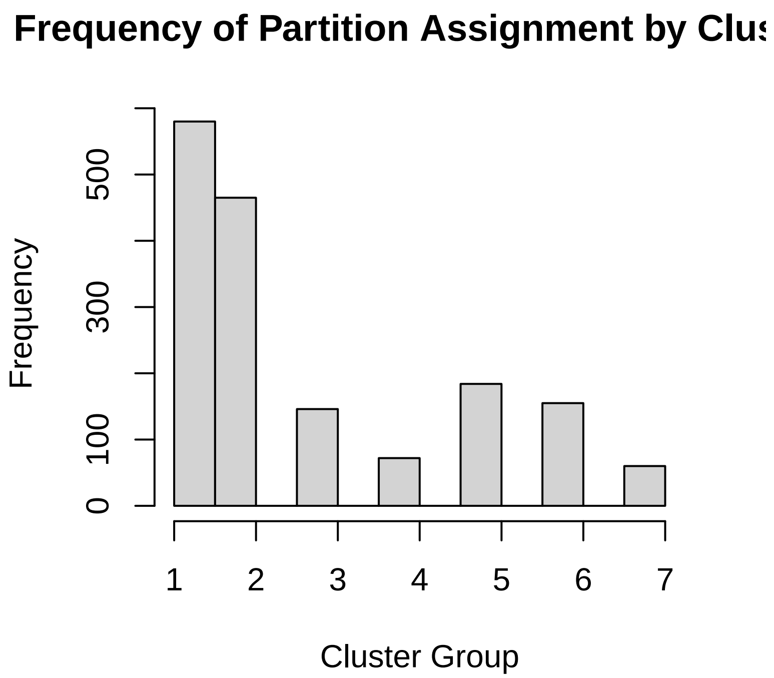
```
# compute the ratio
optimal.hints <- NbClust(data_num, min.nc = 3, max.nc = 10, method = "complete",
    index = "ch")
# table of C(g) results
cg_index <- optimal.hints[["All.index"]]
cg_table <- data.frame(Cluster = 3:10, Value = cg_index)
cg_table <- as_tibble(cg_table)
cg_table
```

```
## # A tibble: 8 x 2
##   Cluster     Value
##     <int>     <dbl>
## 1       3    22111.
## 2       4  6825969.
## 3       5  9863712.
## 4       6 10811523.
## 5       7 11416380.
## 6       8 10422330.
## 7       9 10079034.
## 8      10  9222629.
```

```
# table of Parition Results
partition_complete <- optimal.hints[["Best.partition"]]
table(partition_complete)
```

```
## partition_complete
##   1   2   3   4   5   6   7
## 580 465 146  72 184 155  60
```

```
# plot Partition Results as Frequency
hist(partition_complete, main = "Frequency of Partition Assignment by Cluster",
    xlab = "Cluster Group", )
```



Based on the criterion and utilizing kmeans clustering, the optimal number of clusters is 7 as it has the greatest index value of 11416379.74.

Comparing Hclust with Gower's and Kmeans using C(g)

```
# optimal gower's
new_data.6 <- cutree(hcl_complete, 6)

# optimal kmeans
num_data.7 <- kmeans(data_num, 7)

# table comparison
table(new_data.6)
```

```
## new_data.6
##   1   2   3   4   5   6 
## 958 404 101 183   7   9
```

```
table(num_data.7$cluster)
```

```
## 
##   1   2   3   4   5   6   7 
## 217 202 399 216 218 205 205
```

```r
# compare
compare <- xtabs(~new_data.6 + num_data.7$cluster)
compare
```

```
##           num_data.7$cluster
## new_data.6   1   2   3   4   5   6   7
##          1 127 120 226 124 131 109 121
##          2  44  48 106  52  52  49  53
##          3  15  11  19  13  12  14  17
##          4  28  23  46  25  21  27  13
##          5   1   0   0   1   1   3   1
##          6   2   0   2   1   1   3   0
```

```r
# randindex
rand_index <- phyclust::RRand(new_data.6, num_data.7$cluster)
rand_index
```

```
##       Rand    adjRand     Eindex
##  0.5643136 -0.0005659  0.1248004
```

There is a large amount of agreement between these two methods, however, our adjusted Rand is significantly lower. This indicates that the agreement is non-existent once we make adjustments. As a result, for comparing the clusters against our demographic statistics, I continue on with the kmeans method as it is more robust to the numeric variables and the difference between the outliers in the cluster groups is smaller.

Demographics

We want to assess if we observe the distributions of our selected demographics (race, sex, income, education, and health status) divide in a way that is consistent with the datasets descriptive statistics. We can do this utilizing kmeans.

Adjust the Dataset

```r
# add kmeans cluster assignments
data$kmeans7 <- num_data.7$cluster


# select demographics and clean for analysis
project_demographics <- hints6 |>
    select(HHID, RaceEthn5, BirthGender, HHInc, EducA, EverHadCancer,
        MedConditions_Diabetes, MedConditions_HighBP, MedConditions_HeartCondition,
        MedConditions_LungDisease, MedConditions_Depression)

# remove any NA's
project_demographics <- project_demographics |>
    mutate(across(everything(), ~na_if(., -9))) |>
    mutate(across(everything(), ~na_if(., -7))) |>
    mutate(across(everything(), ~na_if(., -6))) |>
    mutate(across(everything(), ~na_if(., -5))) |>
    mutate(across(everything(), ~na_if(., -4))) |>
    mutate(across(everything(), ~na_if(., -2))) |>
    mutate(across(everything(), ~na_if(., -1)))

# turn health status into dummy (yes/no)
project_demographics$HealthStatus <- ifelse(project_demographics$EverHadCancer ==
    1 | project_demographics$MedConditions_Diabetes == 1 | project_demographics$MedConditions_HighBP ==
    1 | project_demographics$MedConditions_HeartCondition ==
    1 | project_demographics$MedConditions_LungDisease == 1 |
    project_demographics$MedConditions_Depression == 1, 1, 0)

data_final <- inner_join(data, project_demographics, by = "HHID")
```

Analysis of Demographics

```
# Race

race_compare <- xtabs(~RaceEthn5 + kmeans7, data = data_final)
race_compare
```

```
##          kmeans7
## RaceEthn5   1   2   3   4   5   6   7
##         1 131 122 210 124  50 130 126
##         2  31  27  51  30   9  29  23
##         3  19  13  56  23 131  14  15
##         4  13  18  28  13   9  14   9
##         5   6   9  13   6   4   6   9
```

```
# Gender

gender_compare <- xtabs(~BirthGender + kmeans7, data = data_final)
gender_compare
```

```
##            kmeans7
## BirthGender   1   2   3   4   5   6   7
##           1  56  75 140  72  75  62  73
##           2 146 118 221 129 129 134 111
```

```
# Income

income_compare <- xtabs(~HHInc + kmeans7, data = data_final)
income_compare
```

```
##      kmeans7
## HHInc   1   2   3   4   5   6   7
##     1  13   5  18  12  12  11  12
##     2  21  12  31  13  24  15  10
##     3  25  13  28  24  31  23  16
##     4  32  37  52  30  38  35  28
##     5 107 120 226 113  92 111 113
```

```
# Education

educ_compare <- xtabs(~EducA + kmeans7, data = data_final)
educ_compare
```

```
##      kmeans7
## EducA   1   2   3   4   5   6   7
##     1   5   2   6   3   8   5   3
##     2  20  11  32  19  29  23  22
##     3  59  48 105  47  58  40  50
##     4 118 131 219 130 109 128 109
```

```r
# Health Status

health_compare <- xtabs(~HealthStatus + kmeans7, data = data_final)
health_compare
```

```
##             kmeans7
## HealthStatus   1   2   3   4   5   6   7
##            0  73  74 128  84  93  69  71
##            1 134 119 244 117 116 128 118
```

Discussion of Results and Summary

Our sample is predominantly white, female,and with almost half holding a college degree or above. The majority of our sample either has cancer and/or another chronic condition with a median household income at or above $75,000 USD. This demo-graphic breakdown is reflected in our cross tabulations as we see that the majority are White (RaceEthn5 == 1) and remain as such across all clusters except 1 and 2. Females maintain their majority in all clusters of almost two times that of males. The highest household income group ($75k+) also remains the majority in each cluster and so does educational attainment for the exception of cluster 3. Lastly, we see that there is a simillar split of healthy vs chronic condition, where the majority have a chronic condition in each cluster.

This verifies the accuracy of our clusters, maintaining the intial demographic distributions as expected in the beginning of this study. The kmeans method was the best method compared to hclust with Gower's Distance or that of Euclidean distance. This project aimed to assess the best unsupervised machine learning method that would allow us to maintain the structure of our original data and receive the most information. If possible, it would be best to convert variables into their numeric form but more projects utilizing Gower's Distance for mixed datasets provides an avenue for further analysis of survey data.