Final Project EDA

```r
library(mltools)
library(data.table)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(stringr)
library(klaR)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(gapminder)
library(ggplot2)
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.15.2
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------

##
```

```
## Attaching package: 'dendextend'

## The following object is masked from 'package:data.table':
##
##     set

## The following object is masked from 'package:stats':
##
##     cutree
```

```r
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
library(mlbench)
library(caret)
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster
```

```r
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(NbClust)
library(fossil)
```

```
## Loading required package: sp

## Loading required package: maps

## Loading required package: shapefiles

## Loading required package: foreign

##
## Attaching package: 'shapefiles'

## The following objects are masked from 'package:foreign':
##
##     read.dbf, write.dbf
```

```r
library(countrycode)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.6     v purrr   0.3.4
## v tidyr   1.2.0     v forcats 0.5.1
## v readr   2.1.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()    masks data.table::between()
## x dplyr::filter()     masks stats::filter()
## x dplyr::first()      masks data.table::first()
## x dplyr::lag()        masks stats::lag()
## x dplyr::last()       masks data.table::last()
## x purrr::lift()       masks caret::lift()
## x purrr::map()        masks maps::map()
## x tidyr::replace_na() masks mltools::replace_na()
## x MASS::select()      masks dplyr::select()
## x Hmisc::src()        masks dplyr::src()
## x Hmisc::summarize()  masks dplyr::summarize()
## x purrr::transpose()  masks data.table::transpose()
```

```r
library(ggrepel)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```r
data <- read.csv('data/immigration_policies/policy_list.csv')
# summary(data)
```

```r
colSums(is.na(data))[colSums(is.na(data)) != 0]
```

```
##               ISO2           AIR_TYPE        TARGETS_AIR          LAND_TYPE
##                  7               1073               1169               1511
##       TARGETS_LAND           SEA_TYPE        TARGETS_SEA       CITIZEN_LIST
##               1571               1534               1554               1568
##   HISTORY_BAN_LIST       REFUGEE_LIST      VISA_BAN_TYPE      VISA_BAN_LIST
##               1492               1760               1699               1741
## CITIZEN_EXCEP_LIST COUNTRY_EXCEP_LIST
##               1390               1625
```

```r
mod_df <- data.frame(data)

# dropping columns that will not affect our data analysis in any way
mod_df <- mod_df[, -c(32:44)]
colSums(is.na(mod_df))[colSums(is.na(mod_df)) != 0]
```

```
##               ISO2           AIR_TYPE        TARGETS_AIR          LAND_TYPE
##                  7               1073               1169               1511
##       TARGETS_LAND           SEA_TYPE        TARGETS_SEA       CITIZEN_LIST
##               1571               1534               1554               1568
##   HISTORY_BAN_LIST       REFUGEE_LIST      VISA_BAN_TYPE      VISA_BAN_LIST
##               1492               1760               1699               1741
## CITIZEN_EXCEP_LIST COUNTRY_EXCEP_LIST
##               1390               1625
```

```r
colSums(is.na(mod_df))[colSums(is.na(mod_df)) == 0]
```

```
##             ID   COUNTRY_NAME           ISO3    POLICY_TYPE POLICY_SUBTYPE
##              0              0              0              0              0
##     START_DATE       END_DATE            AIR           LAND            SEA
##              0              0              0              0              0
##        CITIZEN    HISTORY_BAN        REFUGEE       VISA_BAN  CITIZEN_EXCEP
##              0              0              0              0              0
##  COUNTRY_EXCEP     WORK_EXCEP
##              0              0
```

```r
# tables to summarize data
# find twelve variables that most interested in, and do correlatin matrix
# if certain variables are very highly correlated, then only use one of the two

# geom jitter -- points won't be laying on top of each other

for (i in 1:length(colnames(mod_df))) {
  column = colnames(mod_df)[i]
  if (sum(is.na(mod_df[, column])) == 0) {
    if (!(column %in% c("ID", "COUNTRY_NAME", "ISO2", "ID", "START_DATE",
                        "END_DATE", "ISO3"))) {
      print(column)
      print(table(mod_df[, column]))
    }
  }
}
```

```
## [1] "POLICY_TYPE"
##
##            COMPLETE NOPOLICYIMPLEMENTED             PARTIAL
##                 422                   7                1333
## [1] "POLICY_SUBTYPE"
##
##   BORDER_CLOSURE    CITIZEN_EXCEP  CITIZENSHIP_BAN   ESSENTIAL_ONLY
##              828              177              194               36
##      HISTORY_BAN             NONE      REFUGEE_BAN SPECIFIC_COUNTRY
##              245                7                3               79
##         VISA_BAN       WORK_EXCEP
##               63              130
## [1] "AIR"
##
##    0    1
## 1073  689
## [1] "LAND"
##
##    0    1
## 1511  251
## [1] "SEA"
##
##    0    1
## 1534  228
## [1] "CITIZEN"
##
##    0    1
```

```
## 1568  194
## [1] "HISTORY_BAN"
##
##    0    1
## 1492  270
## [1] "REFUGEE"
##
##    0    1
## 1759    3
## [1] "VISA_BAN"
##
##    0    1
## 1699   63
## [1] "CITIZEN_EXCEP"
##
##    0    1
## 1390  372
## [1] "COUNTRY_EXCEP"
##
##    0    1
## 1625  137
## [1] "WORK_EXCEP"
##
##    0    1
## 1632  130
```

we know that there are 1762 observations total. we substitute out visa_ban (0 or 1 values) with visa_ban_type, which encapsulates all, specific, or none – we will need to one-hot encode this! other ones to explore: history_ban_list and citizen_list. If I use these, then eliminate history_ban and citizen from consideration (these are values that don't have N/As)

```
# data cleaning for NA values

## VISA_BAN_LIST

colSums(is.na(mod_df))[colSums(is.na(mod_df)) != 0]
```

```
##               ISO2           AIR_TYPE        TARGETS_AIR          LAND_TYPE
##                  7               1073               1169               1511
##       TARGETS_LAND           SEA_TYPE        TARGETS_SEA       CITIZEN_LIST
##               1571               1534               1554               1568
##   HISTORY_BAN_LIST       REFUGEE_LIST      VISA_BAN_TYPE      VISA_BAN_LIST
##               1492               1760               1699               1741
## CITIZEN_EXCEP_LIST COUNTRY_EXCEP_LIST
##               1390               1625
```

```
mod_df$VISA_BAN_NONE <- rep(0, nrow(mod_df))
mod_df[is.na(mod_df$VISA_BAN_TYPE), ]$VISA_BAN_NONE <- 1

mod_df$VISA_BAN_ALL <- rep(0, nrow(mod_df))
mod_df[mod_df$VISA_BAN_TYPE == "All"
       & !is.na(mod_df$VISA_BAN_TYPE), ]$VISA_BAN_ALL <- 1

mod_df$VISA_BAN_SPECIFIC <- rep(0, nrow(mod_df))
mod_df[mod_df$VISA_BAN_TYPE == "specific"
       & !is.na(mod_df$VISA_BAN_TYPE), ]$VISA_BAN_SPECIFIC <- 1
```

```r
mod_df$POLICY_TYPE_COMPLETE <- rep(0, nrow(mod_df))
mod_df[mod_df$POLICY_TYPE ==  "COMPLETE"
       & !is.na(mod_df$POLICY_TYPE), ]$POLICY_TYPE_COMPLETE <- 1

mod_df$POLICY_TYPE_PARTIAL <- rep(0, nrow(mod_df))
mod_df[mod_df$POLICY_TYPE ==  "PARTIAL"
       & !is.na(mod_df$POLICY_TYPE), ]$POLICY_TYPE_PARTIAL <- 1

mod_df$POLICY_TYPE_NON <- rep(0, nrow(mod_df))
mod_df[mod_df$POLICY_TYPE ==  "NOPOLICYIMPLEMENTED"
       & !is.na(mod_df$POLICY_TYPE), ]$POLICY_TYPE_NON <- 1
```

```r
## HISTORY_BAN_LIST

# for now, will count the number of commas
# it would be interesting to explore whether certain countries are banned more often than others, but I

# helper function to determine the number of countries
# i.e., number of commas plus one

country_counter <- function(obj) {
  if (is.na(obj)) {
    return(0)
  }
  return ((str_count(obj, ','))[1] + 1)
}

mod_df$HISTORY_BAN_CLEANED <- unlist(lapply(mod_df$HISTORY_BAN_LIST, country_counter))
mod_df$CITIZEN_LIST_CLEANED <- unlist(lapply(mod_df$CITIZEN_LIST, country_counter))
```

for clustering, will use - policy__type, (maybe policy__subtype?) – need to one-hot-encode - length of policy (end__date - start__date) - air, land, sea, refugee, country__excep, work__excep - visa__ban, citizen__list, and history__ban are already covered by the "list" values we are including

```r
# data cleaning for non-NA values
colSums(is.na(mod_df))[colSums(is.na(mod_df)) == 0]
```

```
##                   ID         COUNTRY_NAME                 ISO3
##                    0                    0                    0
##          POLICY_TYPE       POLICY_SUBTYPE           START_DATE
##                    0                    0                    0
##             END_DATE                  AIR                 LAND
##                    0                    0                    0
##                  SEA              CITIZEN          HISTORY_BAN
##                    0                    0                    0
##              REFUGEE             VISA_BAN        CITIZEN_EXCEP
##                    0                    0                    0
##        COUNTRY_EXCEP           WORK_EXCEP        VISA_BAN_NONE
##                    0                    0                    0
##         VISA_BAN_ALL    VISA_BAN_SPECIFIC POLICY_TYPE_COMPLETE
##                    0                    0                    0
##  POLICY_TYPE_PARTIAL      POLICY_TYPE_NON  HISTORY_BAN_CLEANED
##                    0                    0                    0
## CITIZEN_LIST_CLEANED
```

```
##                      0
```

```r
## DATES
mod_df$START_DATE_CLEANED <- as.Date(mod_df$START_DATE, tryFormats = "%m_%d_%y")
mod_df$END_DATE_CLEANED <- as.Date(mod_df$END_DATE, tryFormats = "%m_%d_%y")
# making assumption that "NA" end date means the policy is still in place
# na values --> setting them equal to today's date
mod_df[is.na(mod_df$END_DATE_CLEANED), ]$END_DATE_CLEANED <- Sys.Date()

# making (possibly faulty assumption) that the ``negative" policy lengths were never in place
# set these values equal to zero
mod_df$POLICY_LENGTH <- difftime(mod_df$END_DATE_CLEANED, mod_df$START_DATE_CLEANED, units = c("days"))
mod_df[mod_df$POLICY_LENGTH < 0 & !is.na(mod_df$POLICY_LENGTH), ]$POLICY_LENGTH <- 0
# no policy implemented will have start date of none --> need to set this to zero as well
mod_df[mod_df$POLICY_TYPE == "NOPOLICYIMPLEMENTED", ]$POLICY_LENGTH <- 0
mod_df$POLICY_LENGTH <- as.numeric(mod_df$POLICY_LENGTH)
```

```r
## one-hot encoding the policy type

# 0 --> not implemented, 1 --> partially implemented, 2 --> complete
mod_df$POLICY_TYPE_CLEANED <- rep(0, nrow(mod_df))
mod_df[mod_df$POLICY_TYPE == "PARTIAL", ]$POLICY_TYPE_CLEANED <- 1
mod_df[mod_df$POLICY_TYPE == "COMPLETE", ]$POLICY_TYPE_CLEANED <- 2
```

AT THIS POINT, WE ARE DONE WITH CLEANING. THESE ARE THE VARIABLE NAMES WE WANT TO USE:

ones we've cleaned:

VISA_BAN_NONE, VISA_BAN_SPECIFIC, VISA_BAN_ALL, HISTORY_BAN_CLEANED, CITIZEN_LIST_CLEANED, POLICY_LENGTH, POLICY_TYPE_CLEANED

ones we've left alone:

AIR, LAND, SEA, REFUGEE, COUNTRY_EXCEP, WORK_EXCEP

```r
# post data cleaning -- need to aggregate by country
vars <- c("COUNTRY_NAME", "ISO3", "VISA_BAN_NONE", "VISA_BAN_SPECIFIC", "VISA_BAN_ALL",
          "HISTORY_BAN_CLEANED", "CITIZEN_LIST_CLEANED", "POLICY_LENGTH",
          "POLICY_TYPE_COMPLETE", "POLICY_TYPE_PARTIAL", "AIR", "LAND", "SEA",
          "POLICY_TYPE_NON", "REFUGEE", "COUNTRY_EXCEP", "WORK_EXCEP")

standardize <- function(col) {
  return((col - mean(col)) / sd(col))
}

cleaned_df <- subset(mod_df, select=vars)
ind <- sapply(cleaned_df, is.numeric)
cleaned_df[ind] <- lapply(cleaned_df[ind], standardize)
```

```r
flattenCorrMatrix <- function(cormat, pmat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    cor  =(cormat)[ut],
    p = pmat[ut]
    )
```

```
}

data_cor <- rcorr(as.matrix(cleaned_df[, 3:ncol(cleaned_df)]))
flattenCorrMatrix(data_cor$r, data_cor$P)
```

```
##                   row               column          cor            p
## 1         VISA_BAN_NONE    VISA_BAN_SPECIFIC -0.570343737 0.000000e+00
## 2         VISA_BAN_NONE         VISA_BAN_ALL -0.811496846 0.000000e+00
## 3     VISA_BAN_SPECIFIC         VISA_BAN_ALL -0.017162110 4.715609e-01
## 4         VISA_BAN_NONE  HISTORY_BAN_CLEANED  0.040107265 9.236883e-02
## 5     VISA_BAN_SPECIFIC  HISTORY_BAN_CLEANED -0.022874927 3.372333e-01
## 6          VISA_BAN_ALL  HISTORY_BAN_CLEANED -0.032546919 1.720684e-01
## 7         VISA_BAN_NONE CITIZEN_LIST_CLEANED  0.049409942 3.809458e-02
## 8     VISA_BAN_SPECIFIC CITIZEN_LIST_CLEANED -0.028180651 2.370821e-01
## 9          VISA_BAN_ALL CITIZEN_LIST_CLEANED -0.040096012 9.246035e-02
## 10  HISTORY_BAN_CLEANED CITIZEN_LIST_CLEANED -0.050974141 3.238934e-02
## 11        VISA_BAN_NONE        POLICY_LENGTH -0.168089253 1.235678e-12
## 12    VISA_BAN_SPECIFIC        POLICY_LENGTH  0.091825347 1.134337e-04
## 13         VISA_BAN_ALL        POLICY_LENGTH  0.139280348 4.333172e-09
## 14  HISTORY_BAN_CLEANED        POLICY_LENGTH -0.085629914 3.200860e-04
## 15 CITIZEN_LIST_CLEANED        POLICY_LENGTH -0.093463498 8.527867e-05
## 16        VISA_BAN_NONE POLICY_TYPE_COMPLETE  0.108063097 5.462945e-06
## 17    VISA_BAN_SPECIFIC POLICY_TYPE_COMPLETE -0.061633111 9.660587e-03
## 18         VISA_BAN_ALL POLICY_TYPE_COMPLETE -0.087692863 2.282530e-04
## 19  HISTORY_BAN_CLEANED POLICY_TYPE_COMPLETE -0.116883522 8.667093e-07
## 20 CITIZEN_LIST_CLEANED POLICY_TYPE_COMPLETE -0.143994061 1.265641e-09
## 21        POLICY_LENGTH POLICY_TYPE_COMPLETE  0.066741130 5.068070e-03
## 22        VISA_BAN_NONE  POLICY_TYPE_PARTIAL -0.109241375 4.305909e-06
## 23    VISA_BAN_SPECIFIC  POLICY_TYPE_PARTIAL  0.062305134 8.896198e-03
## 24         VISA_BAN_ALL  POLICY_TYPE_PARTIAL  0.088649031 1.946624e-04
## 25  HISTORY_BAN_CLEANED  POLICY_TYPE_PARTIAL  0.118157973 6.567573e-07
## 26 CITIZEN_LIST_CLEANED  POLICY_TYPE_PARTIAL  0.145564116 8.324541e-10
## 27        POLICY_LENGTH  POLICY_TYPE_PARTIAL -0.060234957 1.144090e-02
## 28 POLICY_TYPE_COMPLETE  POLICY_TYPE_PARTIAL -0.989214002 0.000000e+00
## 29        VISA_BAN_NONE                  AIR  0.154306185 7.434275e-11
## 30    VISA_BAN_SPECIFIC                  AIR -0.088007566 2.166393e-04
## 31         VISA_BAN_ALL                  AIR -0.125218983 1.340272e-07
## 32  HISTORY_BAN_CLEANED                  AIR -0.166901105 1.783906e-12
## 33 CITIZEN_LIST_CLEANED                  AIR -0.205612969 0.000000e+00
## 34        POLICY_LENGTH                  AIR -0.139599139 3.992206e-09
## 35 POLICY_TYPE_COMPLETE                  AIR -0.449690359 0.000000e+00
## 36  POLICY_TYPE_PARTIAL                  AIR  0.454593604 0.000000e+00
## 37        VISA_BAN_NONE                 LAND  0.078483473 9.765896e-04
## 38    VISA_BAN_SPECIFIC                 LAND -0.044762557 6.030329e-02
## 39         VISA_BAN_ALL                 LAND -0.063689091 7.489816e-03
## 40  HISTORY_BAN_CLEANED                 LAND -0.084889522 3.607474e-04
## 41 CITIZEN_LIST_CLEANED                 LAND -0.104579216 1.088574e-05
## 42        POLICY_LENGTH                 LAND  0.068638610 3.944803e-03
## 43 POLICY_TYPE_COMPLETE                 LAND -0.228722271 0.000000e+00
## 44  POLICY_TYPE_PARTIAL                 LAND  0.231216168 0.000000e+00
## 45                  AIR                 LAND  0.072709883 2.258420e-03
## 46        VISA_BAN_NONE                  SEA  0.074238353 1.818760e-03
## 47    VISA_BAN_SPECIFIC                  SEA -0.042341380 7.559044e-02
## 48         VISA_BAN_ALL                  SEA -0.060244189 1.142824e-02
```

```
## 49   HISTORY_BAN_CLEANED                    SEA -0.080297903 7.417694e-04
## 50  CITIZEN_LIST_CLEANED                    SEA -0.098922595 3.187326e-05
## 51         POLICY_LENGTH                    SEA -0.074383267 1.781445e-03
## 52  POLICY_TYPE_COMPLETE                    SEA -0.216350832 0.000000e+00
## 53   POLICY_TYPE_PARTIAL                    SEA  0.218709836 0.000000e+00
## 54                   AIR                    SEA  0.415273691 0.000000e+00
## 55                  LAND                    SEA  0.287956521 0.000000e+00
## 56         VISA_BAN_NONE        POLICY_TYPE_NON  0.012161413 6.099490e-01
## 57     VISA_BAN_SPECIFIC        POLICY_TYPE_NON -0.006936186 7.710884e-01
## 58          VISA_BAN_ALL        POLICY_TYPE_NON -0.009868948 6.788910e-01
## 59   HISTORY_BAN_CLEANED        POLICY_TYPE_NON -0.013154063 5.810926e-01
## 60  CITIZEN_LIST_CLEANED        POLICY_TYPE_NON -0.016205081 4.966378e-01
## 61         POLICY_LENGTH        POLICY_TYPE_NON -0.041843613 7.909562e-02
## 62  POLICY_TYPE_COMPLETE        POLICY_TYPE_NON -0.035441679 1.369837e-01
## 63   POLICY_TYPE_PARTIAL        POLICY_TYPE_NON -0.111326073 2.809257e-06
## 64                   AIR        POLICY_TYPE_NON -0.050608121 3.365431e-02
## 65                  LAND        POLICY_TYPE_NON -0.025740388 2.801888e-01
## 66                   SEA        POLICY_TYPE_NON -0.024348107 3.070334e-01
## 67         VISA_BAN_NONE                REFUGEE  0.007952456 7.386949e-01
## 68     VISA_BAN_SPECIFIC                REFUGEE -0.004535634 8.491097e-01
## 69          VISA_BAN_ALL                REFUGEE -0.006453393 7.866222e-01
## 70   HISTORY_BAN_CLEANED                REFUGEE -0.008601559 7.182407e-01
## 71  CITIZEN_LIST_CLEANED                REFUGEE -0.010596647 6.566785e-01
## 72         POLICY_LENGTH                REFUGEE  0.045312569 5.721357e-02
## 73  POLICY_TYPE_COMPLETE                REFUGEE -0.023175629 3.309189e-01
## 74   POLICY_TYPE_PARTIAL                REFUGEE  0.023428327 3.256720e-01
## 75                   AIR                REFUGEE -0.033093100 1.649798e-01
## 76                  LAND                REFUGEE -0.016831869 4.801345e-01
## 77                   SEA                REFUGEE -0.015921444 5.042041e-01
## 78       POLICY_TYPE_NON                REFUGEE -0.002608184 9.128820e-01
## 79         VISA_BAN_NONE          COUNTRY_EXCEP  0.055912278 1.891740e-02
## 80     VISA_BAN_SPECIFIC          COUNTRY_EXCEP -0.031889218 1.809036e-01
## 81          VISA_BAN_ALL          COUNTRY_EXCEP -0.045372637 5.688424e-02
## 82   HISTORY_BAN_CLEANED          COUNTRY_EXCEP -0.060476001 1.111462e-02
## 83  CITIZEN_LIST_CLEANED          COUNTRY_EXCEP -0.074503102 1.751121e-03
## 84         POLICY_LENGTH          COUNTRY_EXCEP -0.017251171 4.692633e-01
## 85  POLICY_TYPE_COMPLETE          COUNTRY_EXCEP  0.517403991 0.000000e+00
## 86   POLICY_TYPE_PARTIAL          COUNTRY_EXCEP -0.511823273 0.000000e+00
## 87                   AIR          COUNTRY_EXCEP -0.232671586 0.000000e+00
## 88                  LAND          COUNTRY_EXCEP -0.118341816 6.308464e-07
## 89                   SEA          COUNTRY_EXCEP -0.111940784 2.473241e-06
## 90       POLICY_TYPE_NON          COUNTRY_EXCEP -0.018337666 4.417368e-01
## 91               REFUGEE          COUNTRY_EXCEP -0.011991163 6.149614e-01
## 92         VISA_BAN_NONE             WORK_EXCEP  0.054348202 2.252494e-02
## 93     VISA_BAN_SPECIFIC             WORK_EXCEP -0.030997157 1.934189e-01
## 94          VISA_BAN_ALL             WORK_EXCEP -0.044103395 6.418696e-02
## 95   HISTORY_BAN_CLEANED             WORK_EXCEP -0.058784261 1.358999e-02
## 96  CITIZEN_LIST_CLEANED             WORK_EXCEP -0.072418972 2.352391e-03
## 97         POLICY_LENGTH             WORK_EXCEP  0.005453561 8.190563e-01
## 98  POLICY_TYPE_COMPLETE             WORK_EXCEP  0.502930267 0.000000e+00
## 99   POLICY_TYPE_PARTIAL             WORK_EXCEP -0.497505662 0.000000e+00
## 100                  AIR             WORK_EXCEP -0.226162892 0.000000e+00
## 101                 LAND             WORK_EXCEP -0.115031353 1.290388e-06
## 102                  SEA             WORK_EXCEP -0.108809382 4.699845e-06
```

```
## 103      POLICY_TYPE_NON           WORK_EXCEP -0.017824693 4.546164e-01
## 104              REFUGEE           WORK_EXCEP -0.011655725 6.248896e-01
## 105        COUNTRY_EXCEP           WORK_EXCEP  0.388285955 0.000000e+00
```

```
set.seed(98)
# load the library
# calculate correlation matrix
correlationMatrix <- cor(cleaned_df[, 3:ncol(cleaned_df)])
# summarize the correlation matrix
# find attributes that are highly corrected (ideally >0.75)
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.60)
# print indexes of highly correlated attributes
print(highlyCorrelated)
```

```
## [1] 8 1
```

```
# hopefully ISO3 can be easily matched with other data sets
by_country <- aggregate(cbind(VISA_BAN_NONE, VISA_BAN_SPECIFIC, VISA_BAN_ALL,
                              HISTORY_BAN_CLEANED,
                              CITIZEN_LIST_CLEANED, POLICY_LENGTH, POLICY_TYPE_NON,
                              POLICY_TYPE_COMPLETE, POLICY_TYPE_PARTIAL,
                              AIR, LAND,
                              SEA, REFUGEE, COUNTRY_EXCEP, WORK_EXCEP)~ISO3, data = cleaned_df, mean)
```

NOW, we can work with the by_country data frame!!!

```
summary(cleaned_df)
```

```
##  COUNTRY_NAME           ISO3           VISA_BAN_NONE     VISA_BAN_SPECIFIC
##  Length:1762        Length:1762        Min.   :-5.1916   Min.   :-0.1098
##  Class :character   Class :character   1st Qu.: 0.1925   1st Qu.:-0.1098
##  Mode  :character   Mode  :character   Median : 0.1925   Median :-0.1098
##                                        Mean   : 0.0000   Mean   : 0.0000
##                                        3rd Qu.: 0.1925   3rd Qu.:-0.1098
##                                        Max.   : 0.1925   Max.   : 9.1026
##   VISA_BAN_ALL     HISTORY_BAN_CLEANED CITIZEN_LIST_CLEANED POLICY_LENGTH
##  Min.   :-0.1562   Min.   :-0.2082     Min.   :-0.2565      Min.   :-0.66236
##  1st Qu.:-0.1562   1st Qu.:-0.2082     1st Qu.:-0.2565      1st Qu.:-0.58507
##  Median :-0.1562   Median :-0.2082     Median :-0.2565      Median :-0.43049
##  Mean   : 0.0000   Mean   : 0.0000     Mean   : 0.0000      Mean   : 0.00000
##  3rd Qu.:-0.1562   3rd Qu.:-0.2082     3rd Qu.:-0.2565      3rd Qu.: 0.08293
##  Max.   : 6.3976   Max.   : 6.7225     Max.   : 4.7614      Max.   : 3.96951
##  POLICY_TYPE_COMPLETE POLICY_TYPE_PARTIAL      AIR               LAND
##  Min.   :-0.561       Min.   :-1.7622     Min.   :-0.8011   Min.   :-0.4075
##  1st Qu.:-0.561       1st Qu.: 0.5671     1st Qu.:-0.8011   1st Qu.:-0.4075
##  Median :-0.561       Median : 0.5671     Median :-0.8011   Median :-0.4075
##  Mean   : 0.000       Mean   : 0.0000     Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:-0.561       3rd Qu.: 0.5671     3rd Qu.: 1.2476   3rd Qu.:-0.4075
##  Max.   : 1.781       Max.   : 0.5671     Max.   : 1.2476   Max.   : 2.4529
##       SEA           POLICY_TYPE_NON       REFUGEE          COUNTRY_EXCEP
##  Min.   :-0.3854   Min.   :-0.06314   Min.   :-0.04129   Min.   :-0.2903
##  1st Qu.:-0.3854   1st Qu.:-0.06314   1st Qu.:-0.04129   1st Qu.:-0.2903
##  Median :-0.3854   Median :-0.06314   Median :-0.04129   Median :-0.2903
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.:-0.3854   3rd Qu.:-0.06314   3rd Qu.:-0.04129   3rd Qu.:-0.2903
##  Max.   : 2.5931   Max.   :15.82947   Max.   :24.20745   Max.   : 3.4430
```

```
##    WORK_EXCEP
##  Min.   :-0.2822
##  1st Qu.:-0.2822
##  Median :-0.2822
##  Mean   : 0.0000
##  3rd Qu.:-0.2822
##  Max.   : 3.5421
```

```
new_vars <- c("VISA_BAN_NONE", "VISA_BAN_SPECIFIC", "VISA_BAN_ALL",
          "HISTORY_BAN_CLEANED", "CITIZEN_LIST_CLEANED", "POLICY_LENGTH",
          "POLICY_TYPE_COMPLETE", "POLICY_TYPE_PARTIAL", "AIR", "LAND", "SEA",
          "POLICY_TYPE_NON", "REFUGEE", "COUNTRY_EXCEP", "WORK_EXCEP")
```

goals by next Wednesday: - kMeans cluster on selected variables - hierarchical cluster - (not needed by next Wednesday, but we can vary the number of clusters and where you stop on the dendrogram) – can talk about this as next steps - plot two variables from demographics – then plot the clusters we previously generated (for immigration policies) – this can be a wednesday goal! - can also run the cluster algorithm on the demographics data – does not need to be a wednesday goal - WorldBank, Gap Minder (may have an R package!) – other potential data sets for the demographic - try different distance metrics to see how much the answer changes (how robust is it to that choice?) - k-modes clustering – better suited for categorical data

- see how clusters change with inclusion of different variables

FEEDBACK FROM PRESENTATION:

- log of GDP, population to adjust the scale
- formal tests: 2-sample means on a metric between clusters
- PCA on demographic factors for ease of visualization
- formal test to determine how many clusters there are
- some way to score the different policies, and then see if there is a correlation between that and certain demographic covariates
- find some indicator of "natural" clustering – do we see patterns among certain continents, developed vs developing, etc – then adjust number of clusters based on the number of natural clusters, and see whether the contents of those clusters are the same
- try running PCA on immigration policies (???)

FOR MEETING WITH KELLY: - have decided not to cluster countries based on their demographic factors, and to instead use that as a more informal way to investigate the clusters based on immigration policies

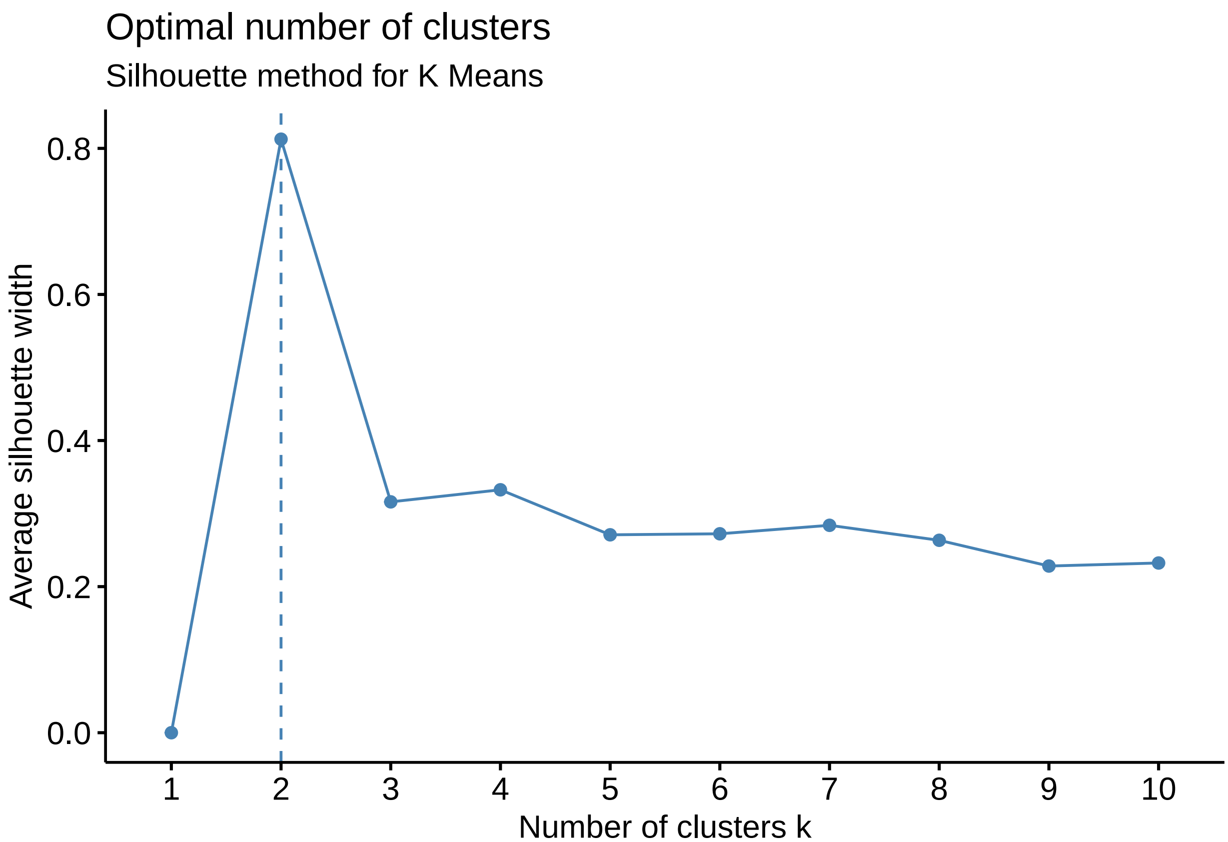
To do before meeting: - download data on GDP, population, life expectancy, education rate, and fertility rate DONE - officially decide on the number of clusters and linkage for HAC and K-means DONE - the results section will consist of some visuals (probably PCA to get two dimensions – but is this interpretable?), ANOVA test on those same factors (GDP, population, life expectancy, education rate, fertility rate) across different clusters (for both methods) - look at the natural way of clustering (by continent, development level)

DETERMINING THE NUMBER OF CLUSTERS:

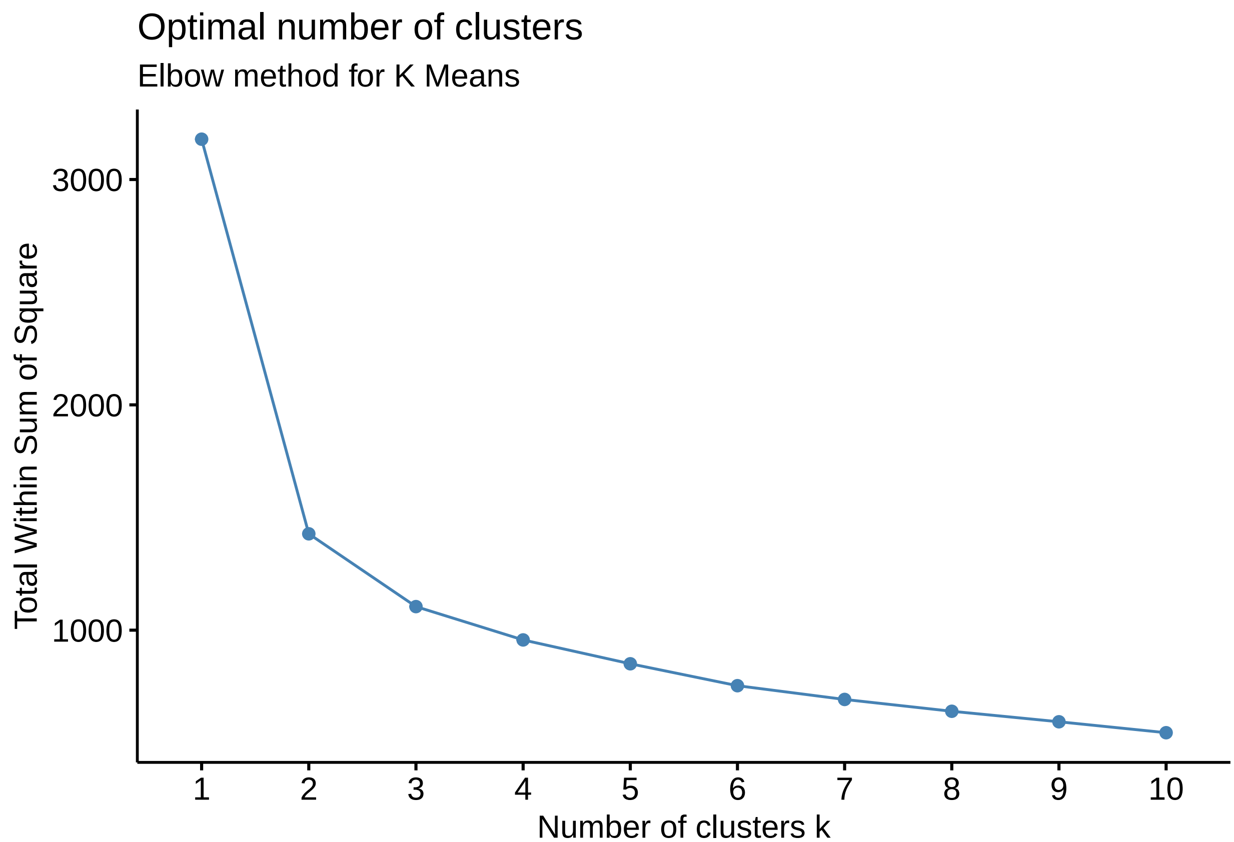
```
# explain how number of clusters is very much impacted by choice of heuristics
jpeg(file="gap_k.jpg")
fviz_nbclust(by_country[,2:ncol(by_country)], kmeans, nstart = 25,  method = "gap_stat",
             nboot =50)+ labs(subtitle = "Gap statistic method for K Means")
dev.off()
```

```
## pdf
##   2
```

```
fviz_nbclust(by_country[,2:ncol(by_country)], kmeans, nstart = 25,  method = "silhouette",
             nboot =50)+ labs(subtitle = "Silhouette method for K Means")
```



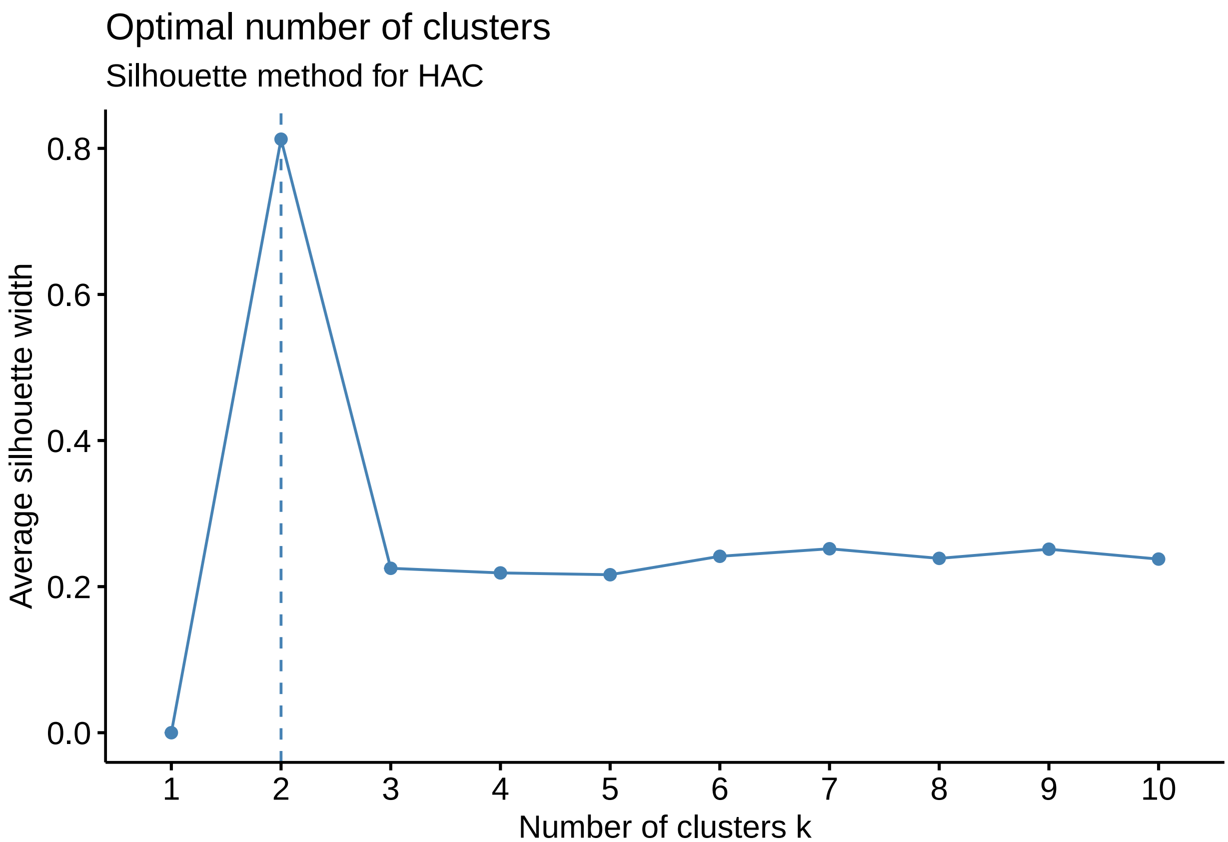

```r
fviz_nbclust(by_country[,2:ncol(by_country)], kmeans, nstart = 25,  method = "wss",
             nboot =50)+ labs(subtitle = "Elbow method for K Means")
```



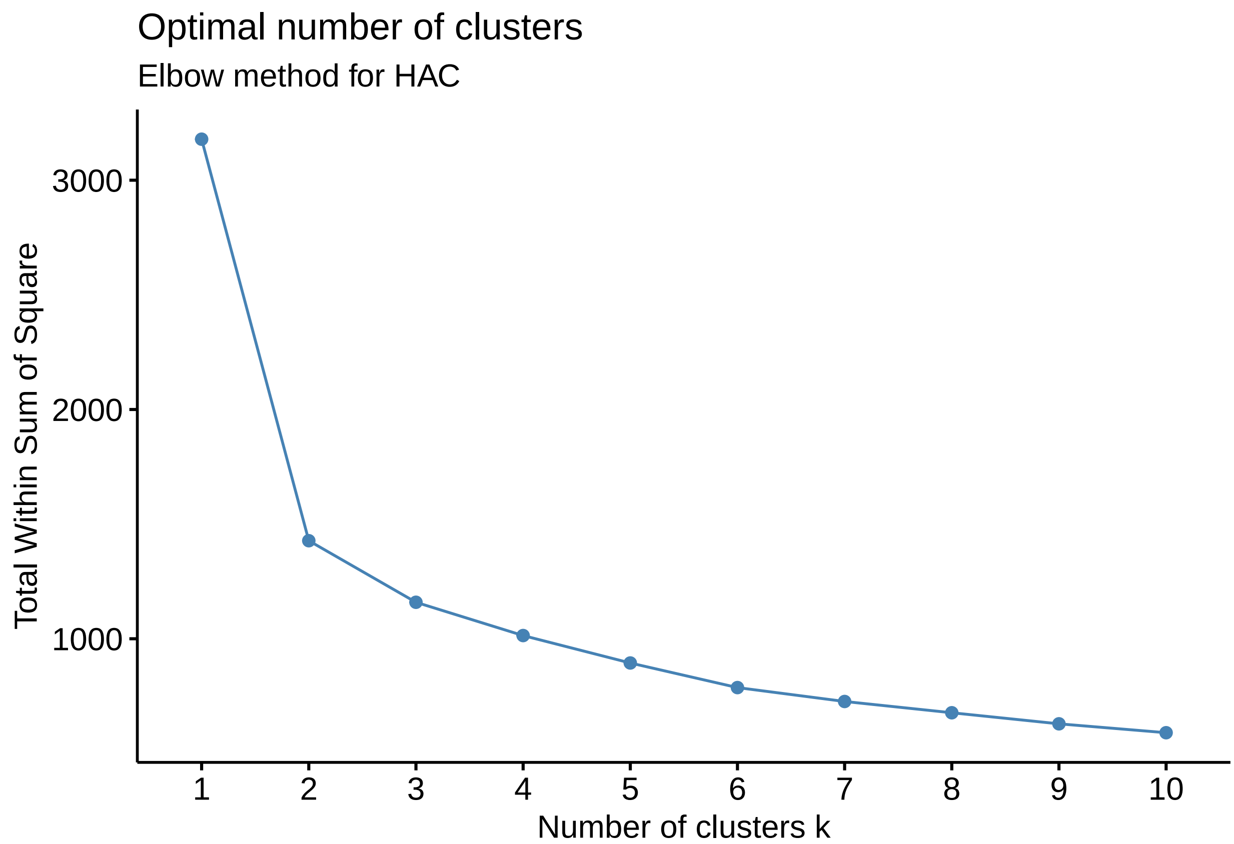

```
jpeg(file="gap_hac.jpg")
fviz_nbclust(by_country[,2:ncol(by_country)], hcut, nstart = 25,  method = "gap_stat",
             nboot = 50)+labs(subtitle = "Gap statistic method for HAC")
dev.off()
```

```
## pdf
##   2
```

```
fviz_nbclust(by_country[,2:ncol(by_country)], hcut, nstart = 25,  method = "silhouette",
             nboot = 50)+labs(subtitle = "Silhouette method for HAC")
```




```
fviz_nbclust(by_country[,2:ncol(by_country)], hcut, nstart = 25,  method = "wss",
             nboot = 50)+labs(subtitle = "Elbow method for HAC")
```




```r
# what to do if the two don't agree?
```

DETERMINING THE LINKAGE CRITERIA:

```r
# Data
dist_mat <- dist(by_country[,2:ncol(by_country)], method = 'euclidean')

# Hierarchical Agglomerative Clustering
h1=hclust(dist_mat,method='average')
h2=hclust(dist_mat,method='complete')
h4=hclust(dist_mat,method='single')

# Cophenetic Distances, for each linkage
c1=cophenetic(h1)
c2=cophenetic(h2)
c4=cophenetic(h4)

# Correlations
cor(dist_mat,c1)
```

```
## [1] 0.9733684
```

```r
cor(dist_mat,c2)
```

```
## [1] 0.8841364
```

```r
cor(dist_mat,c4)
```

```
## [1] 0.9513063
```

```r
# average is the best linkage method
```

for now, use 3 clusters (for HAC and k-means)

```r
# kmeans clustering
set.seed(98)
cluster.results.10 <- kmeans(by_country[,2:ncol(by_country)], 10,
                            iter.max = 10, nstart = 1)

# cluster.results.6 <- kmeans(by_country[,2:ncol(by_country)], 6,
#                             iter.max = 10, nstart = 1)

kcluster_by_country = data.frame(by_country)
kcluster_by_country$cluster10 <- as.factor(cluster.results.10$cluster)
# kcluster_by_country$cluster10 <- as.factor(cluster.results.6$cluster)
```

```r
# hierarchical clustering

dist_mat <- dist(by_country[,2:ncol(by_country)], method = 'euclidean')
hclust_avg <- hclust(dist_mat, method = 'average')

jpeg(file="cluster_den.jpg")
plot(hclust_avg)
dev.off()
```

```
## pdf
##   2
```

```r
cut_avg10 <- cutree(hclust_avg, k = 10)

avg_dend_obj <- as.dendrogram(hclust_avg, h = 10, leaflab = "none")
labels(avg_dend_obj) <- rep(NA, nrow(by_country))
avg_col_dend10 <- color_branches(avg_dend_obj, k = 10)

jpeg(file="cluster_den10.jpg")
plot(avg_col_dend10)
dev.off()
```

```
## pdf
##   2
```

```r
# jpeg(file="cluster_den6.jpg")
# plot(avg_col_dend6)
# dev.off()

hcluster_by_country10 <- mutate(by_country, cluster = cut_avg10)
# hcluster_by_country6 <- mutate(by_country, cluster = cut_avg6)

hcluster_by_country <- data.frame(by_country)

hcluster_by_country$cluster10 <- as.factor(hcluster_by_country10$cluster)
# hcluster_by_country$cluster10 <- as.factor(hcluster_by_country6$cluster)
```

```r
# bringing in demographic data; need life expectancy, literacy rate, and fertility rate
gdp <- read.csv('data/demographic/gdp.csv')
population <- read.csv('data/demographic/population.csv')
life_expectancy <- read.csv('data/demographic/life_expectancy.csv')
```

```r
fertility_rate <- read.csv('data/demographic/fertility_rate.csv')
literacy_rate <- read.csv('data/demographic/literacy_rate.csv')
iso3 <- read.csv('data/demographic/iso3.csv')
```

```r
gdp[gdp$Code == "ABW", ]$GDP = 3202 * 10^6
gdp[gdp$Code == "AND", ]$GDP = 3155 * 10^6
gdp[gdp$Code == "ERI", ]$GDP = 2.07 * 10^9
gdp[gdp$Code == "GIB", ]$GDP = 2885810912.00
gdp[gdp$Code == "GRL", ]$GDP = 3052 * 10^6
gdp[gdp$Code == "LIE", ]$GDP = 6839 * 10^6
gdp[gdp$Code == "MNP", ]$GDP = 1182 * 10^6
gdp[gdp$Code == "NCL", ]$GDP = 10 * 10^9
gdp[gdp$Code == "PYF", ]$GDP = 3.45 * 10^9
gdp[gdp$Code == "SMR", ]$GDP = 1616 * 10^6
gdp[gdp$Code == "SSD", ]$GDP = 1119.7 * 10^6
gdp[gdp$Code == "TKM", ]$GDP = 45231 * 10^6
gdp[gdp$Code == "VEN", ]$GDP = 47.26 * 10^9
gdp[gdp$Code == "YEM", ]$GDP = 23486 * 10^6
```

```r
life_expectancy[life_expectancy$Code == "AND", ]$Expectancy = 84.5
life_expectancy[life_expectancy$Code == "ASM", ]$Expectancy = 73.32
life_expectancy[life_expectancy$Code == "CYM", ]$Expectancy = 82.19
life_expectancy[life_expectancy$Code == "DMA", ]$Expectancy = 76.6
life_expectancy[life_expectancy$Code == "GIB", ]$Expectancy = 78.7
life_expectancy[life_expectancy$Code == "KNA", ]$Expectancy = 71.34
life_expectancy[life_expectancy$Code == "MCO", ]$Expectancy = 89.4
life_expectancy[life_expectancy$Code == "MHL", ]$Expectancy = 65.24
life_expectancy[life_expectancy$Code == "MNP", ]$Expectancy = 77.1
life_expectancy[life_expectancy$Code == "PLW", ]$Expectancy = 69.13
life_expectancy[life_expectancy$Code == "SMR", ]$Expectancy = 85.42
life_expectancy[life_expectancy$Code == "TCA", ]$Expectancy = 80.6
```

```r
fertility_rate[fertility_rate$Code == "AND", ]$Fertility = 1.3
fertility_rate[fertility_rate$Code == "ASM", ]$Fertility = 2.28
fertility_rate[fertility_rate$Code == "CYM", ]$Fertility = 1.83
fertility_rate[fertility_rate$Code == "DMA", ]$Fertility = 1.9
fertility_rate[fertility_rate$Code == "GIB", ]$Fertility = 1.91
fertility_rate[fertility_rate$Code == "KNA", ]$Fertility = 2.1
fertility_rate[fertility_rate$Code == "MCO", ]$Fertility = 1.52
fertility_rate[fertility_rate$Code == "MHL", ]$Fertility = 4.5
fertility_rate[fertility_rate$Code == "MNP", ]$Fertility = 2.66
fertility_rate[fertility_rate$Code == "PLW", ]$Fertility = 2.21
fertility_rate[fertility_rate$Code == "SMR", ]$Fertility = 1.3
fertility_rate[fertility_rate$Code == "TCA", ]$Fertility = 1.7
```

```r
# some data cleaning on literacy rate -- need to note how not all of them were pulled from
# 2020
literacy_rate <- merge(literacy_rate, iso3, by.x = "country", by.y = "Country")
literacy_rate <- subset(literacy_rate, select = c(latestRate, Alpha.3.code))
colnames(literacy_rate) <- c('literacy', 'Code')
literacy_rate$Code <- trimws(literacy_rate$Code)
```

```r
master_df_k <- merge(kcluster_by_country, gdp, by.x = "ISO3", by.y = "Code")
master_df_k <- merge(master_df_k, population, by.x = "ISO3", by.y = "Code")
master_df_k <- merge(master_df_k, life_expectancy, by.x = "ISO3", by.y = "Code")
```

```r
master_df_k <- merge(master_df_k, fertility_rate, by.x = "ISO3", by.y = "Code")
master_df_k <- merge(master_df_k, literacy_rate, by.x = "ISO3", by.y = "Code")
master_df_k <- subset(master_df_k, select = -c(Name.x, X.x, Name.y, X.y))

master_df_h <- merge(hcluster_by_country, gdp, by.x = "ISO3", by.y = "Code")
master_df_h <- merge(master_df_h, population, by.x = "ISO3", by.y = "Code")
master_df_h <- merge(master_df_h, life_expectancy, by.x = "ISO3", by.y = "Code")
master_df_h <- merge(master_df_h, fertility_rate, by.x = "ISO3", by.y = "Code")
master_df_h <- merge(master_df_h, literacy_rate, by.x = "ISO3", by.y = "Code")
master_df_h <- subset(master_df_h, select = -c(Name.x, X.x, Name.y, X.y))
```

```r
# anova stuff
# histograms or boxplots -- distribution of these variables across these clusters change
# for ones that are significant -- look for outliers!

k_gdp <- aov(GDP ~ cluster10, data = master_df_k)
k_pop <- aov(Pop ~ cluster10, data = master_df_k)
k_exp <- aov(Expectancy ~ cluster10, data = master_df_k)
k_fert <- aov(Fertility ~ cluster10, data = master_df_k)
k_lit <- aov(literacy ~ cluster10, data = master_df_k)

summary(k_gdp)
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## cluster10     9 4.984e+25 5.538e+24   1.509  0.148
## Residuals   179 6.569e+26 3.670e+24
```

```r
summary(k_pop)
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## cluster10     9 2.762e+17 3.069e+16   1.423  0.181
## Residuals   179 3.861e+18 2.157e+16
```

```r
summary(k_exp)
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)
## cluster10     9   2221  246.79   5.405 1.5e-06 ***
## Residuals   179   8174   45.66
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(k_fert)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## cluster10     9  40.66   4.517   3.429 0.000644 ***
## Residuals   179 235.84   1.318
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(k_lit)
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)
## cluster10     9   8491   943.4    3.11 0.00169 **
## Residuals   179  54299   303.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
h_gdp <- aov(GDP ~ cluster10, data = master_df_h)
h_pop <- aov(Pop ~ cluster10, data = master_df_h)
h_exp <- aov(Expectancy ~ cluster10, data = master_df_h)
h_fert <- aov(Fertility ~ cluster10, data = master_df_h)
h_lit <- aov(literacy ~ cluster10, data = master_df_h)

summary(h_gdp)
```

```
##              Df    Sum Sq   Mean Sq F value   Pr(>F)
## cluster10     8 1.104e+26 1.380e+25   4.166 0.000136 ***
## Residuals   180 5.964e+26 3.313e+24
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(h_pop)
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## cluster10     8 3.348e+16 4.184e+15   0.184  0.993
## Residuals   180 4.104e+18 2.280e+16
```

```r
summary(h_exp)
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)
## cluster10     8   1146  143.24   2.788 0.00623 **
## Residuals   180   9249   51.38
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(h_fert)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## cluster10     8  18.05   2.256   1.571  0.136
## Residuals   180 258.44   1.436
```

```r
summary(h_lit)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## cluster10     8   3473   434.2   1.318  0.237
## Residuals   180  59317   329.5
```

https://gist.github.com/tadast/8827699 https://worldpopulationreview.com/country-rankings/literacy-rate-by-country https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/

```r
# need to merge continent and development level into the data
continents <- read.csv('data/demographic/continent.csv')
continents <- subset(continents, select = c(continent, code_3))
old <- c("Asia", "Europe", "Africa", "Oceania", "Americas")
new <- 1:length(old)
continents$continent[continents$continent %in% old] <- new[match(continents$continent,
                                                                  old, nomatch = 0)]
continents$continent <- as.numeric(continents$continent)
master_df_k_continent <- merge(master_df_k, continents, by.x = "ISO3", by.y = "code_3")
master_df_h_continent <- merge(master_df_h, continents, by.x = "ISO3", by.y = "code_3")
```

HDI classifications are based on HDI fixed cutoff points, which are derived from the quartiles of dis- tributions of the component indicators. The cutoff-points are HDI of less than 0.550 for low human development, 0.550–0.699 for medium human development, 0.700–0.799 for high human development and 0.800 or greater for very high human development.

https://hdr.undp.org/en/content/human-development-report-2020-readers-guide

```
hdi <- read.csv('data/demographic/hdi.csv')
hdi <- merge(hdi, iso3, by.x = "country", by.y = "Country")
hdi <- subset(hdi, select = c(hdi, Alpha.3.code))
colnames(hdi) <- c('hdi', 'Code')
hdi$development <- rep(1, nrow(hdi))
hdi[hdi$hdi >= 0.55 & hdi$hdi <= 0.699, ]$development <- 2
hdi[hdi$hdi >= 0.7 & hdi$hdi <= 0.799, ]$development <- 3
hdi[hdi$hdi >= 0.8, ]$development <- 4
hdi$Code <- trimws(hdi$Code)

master_df_k_hdi <- merge(master_df_k, hdi, by.x = "ISO3", by.y = "Code")
master_df_h_hdi <- merge(master_df_h, hdi, by.x = "ISO3", by.y = "Code")
```

```
rand.index(as.numeric(levels(master_df_k_continent$cluster10))[master_df_k_continent$cluster10],
           master_df_k_continent$continent)
```

```
## [1] 0.6787684
```

```
rand.index(as.numeric(levels(master_df_h_continent$cluster10))[master_df_h_continent$cluster10],
           master_df_k_continent$continent)
```

```
## [1] 0.3761117
```

```
rand.index(as.numeric(levels(master_df_h_continent$cluster10))[master_df_h_continent$cluster10],
           as.numeric(levels(master_df_k_continent$cluster10))[master_df_k_continent$cluster10])
```

```
## [1] 0.4467522
```

```
rand.index(as.numeric(levels(master_df_k_hdi$cluster10))[master_df_k_hdi$cluster10],
           master_df_k_hdi$development)
```

```
## [1] 0.6513932
```

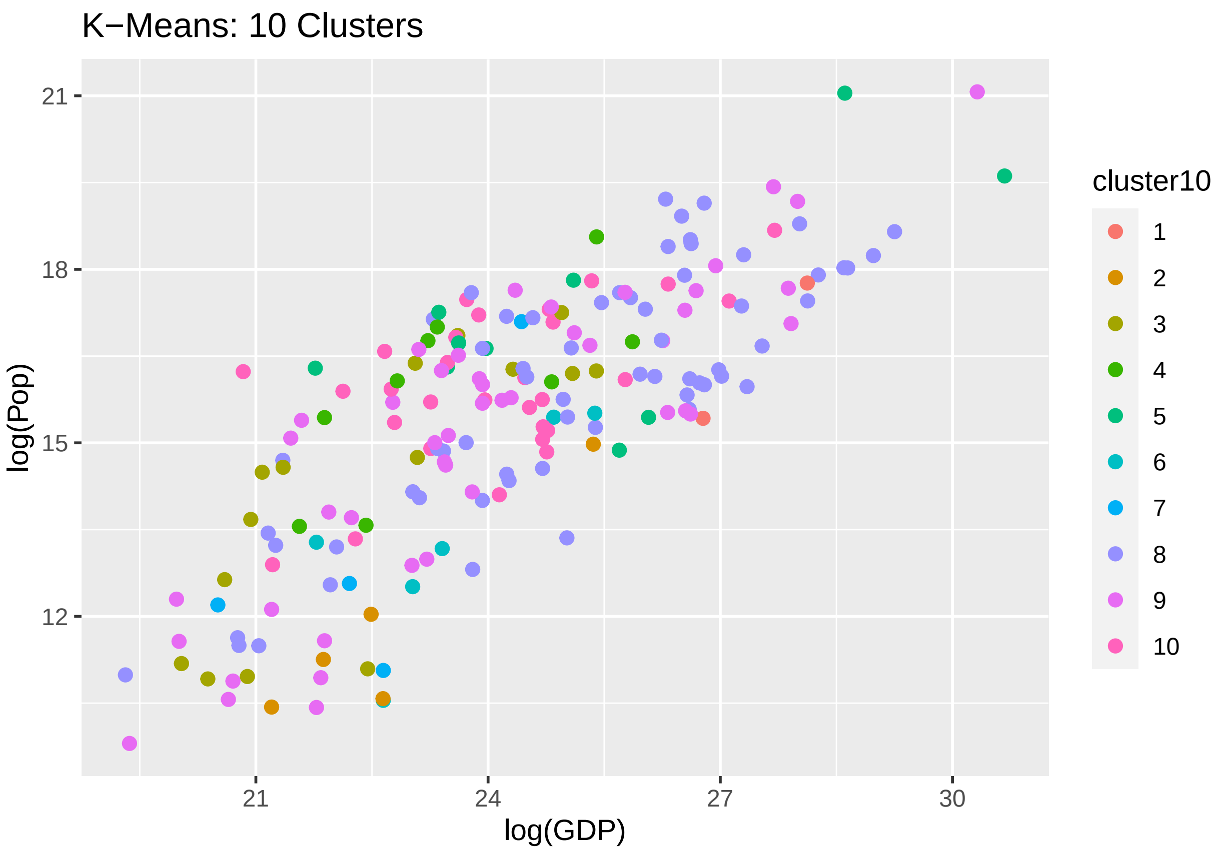
```
rand.index(as.numeric(levels(master_df_h_hdi$cluster10))[master_df_h_hdi$cluster10],
           master_df_h_hdi$development)
```

```
## [1] 0.3738562
```

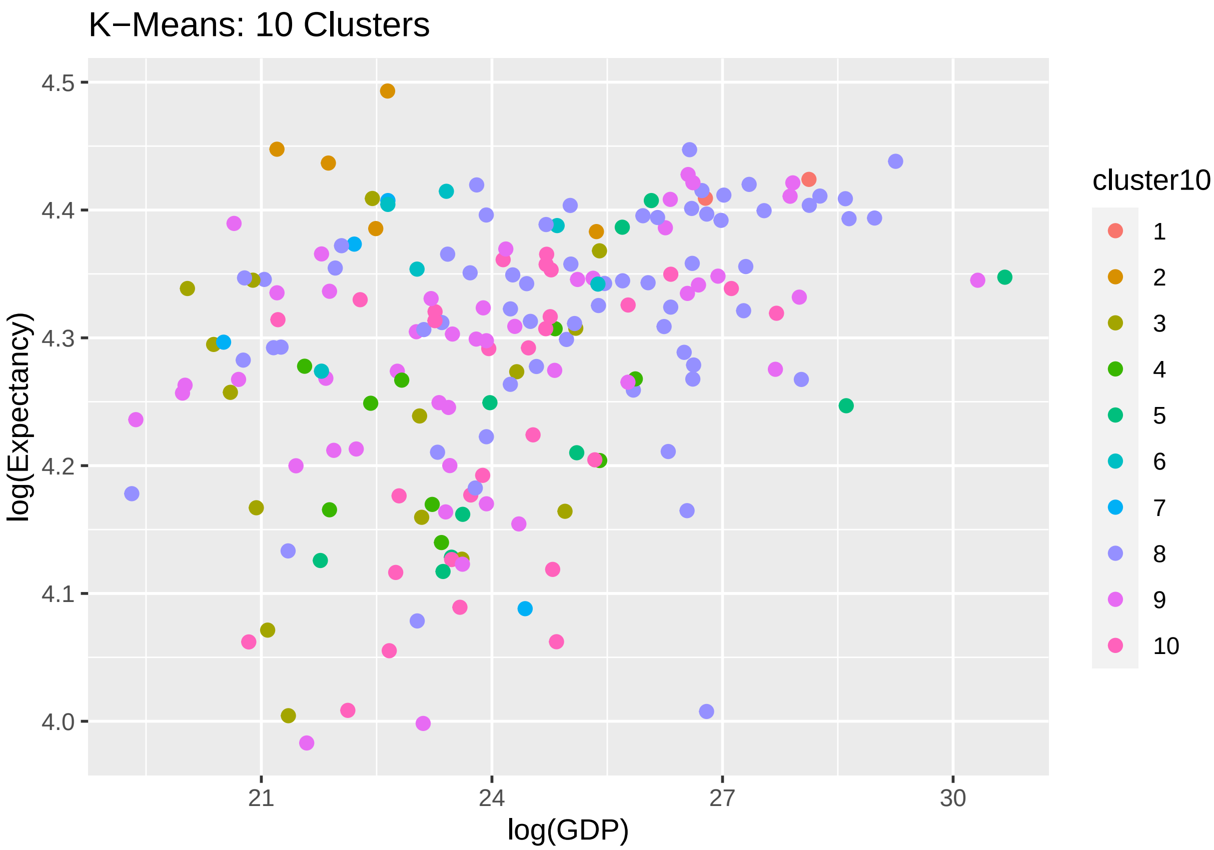
```
# trying to explain these results? need similar data for developed vs undeveloped
# be prepared to justify why!!
```

```
# generating graphs: possible pairs

#       [,1]         [,2]
#  [1,] "GDP"        "Pop"
#  [2,] "GDP"        "Expectancy"
#  [3,] "GDP"        "Fertility"
#  [4,] "GDP"        "literacy"
#  [5,] "Pop"        "Expectancy"
#  [6,] "Pop"        "Fertility"
#  [7,] "Pop"        "literacy"
#  [8,] "Expectancy" "Fertility"
#  [9,] "Expectancy" "literacy"
# [10,] "Fertility"  "literacy"
```

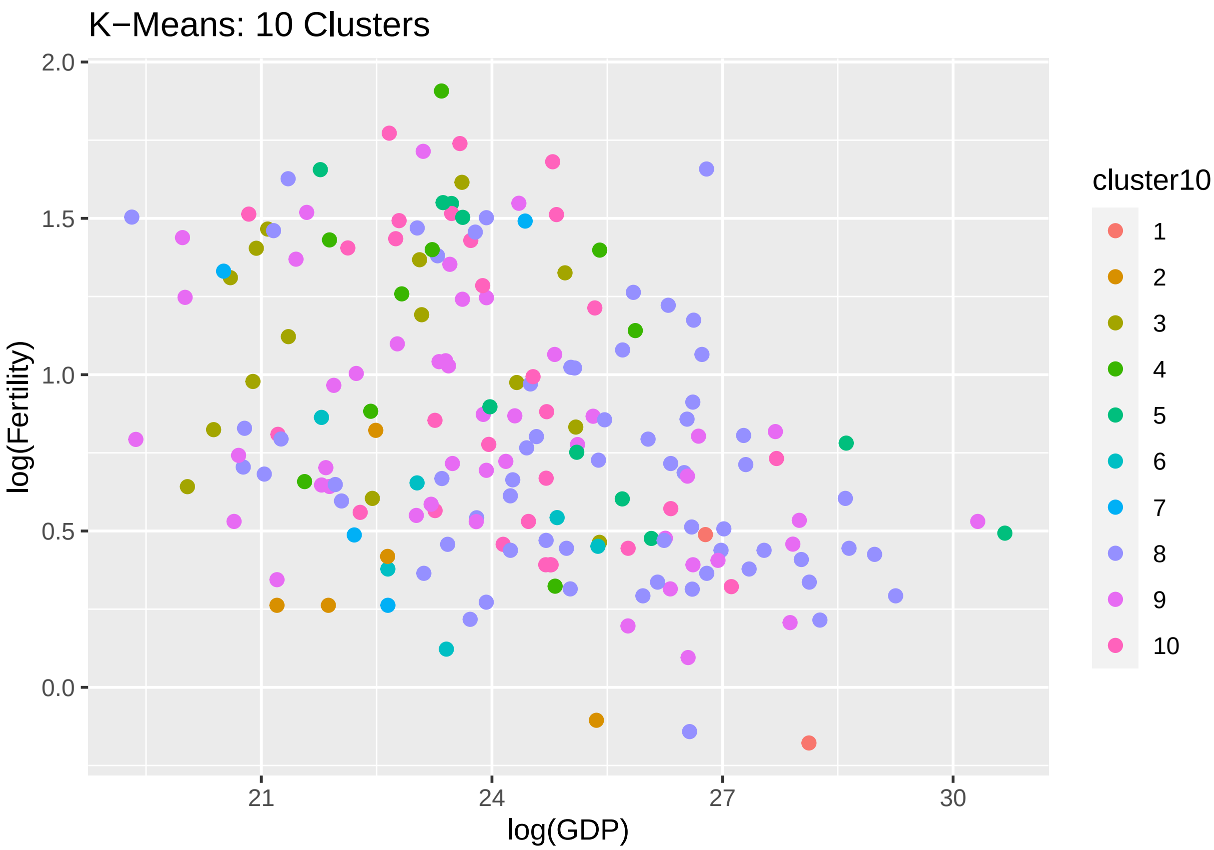
```
vars <- c("GDP", "Pop", "Expectancy", "Fertility", "literacy")
pairs <- t(combn(vars, 2))

p2 <- ggplot(master_df_k, aes(x = log(GDP), y = log(Pop), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("K-Means: 10 Clusters") + scale_fill_brewer(palette="Set3")
```



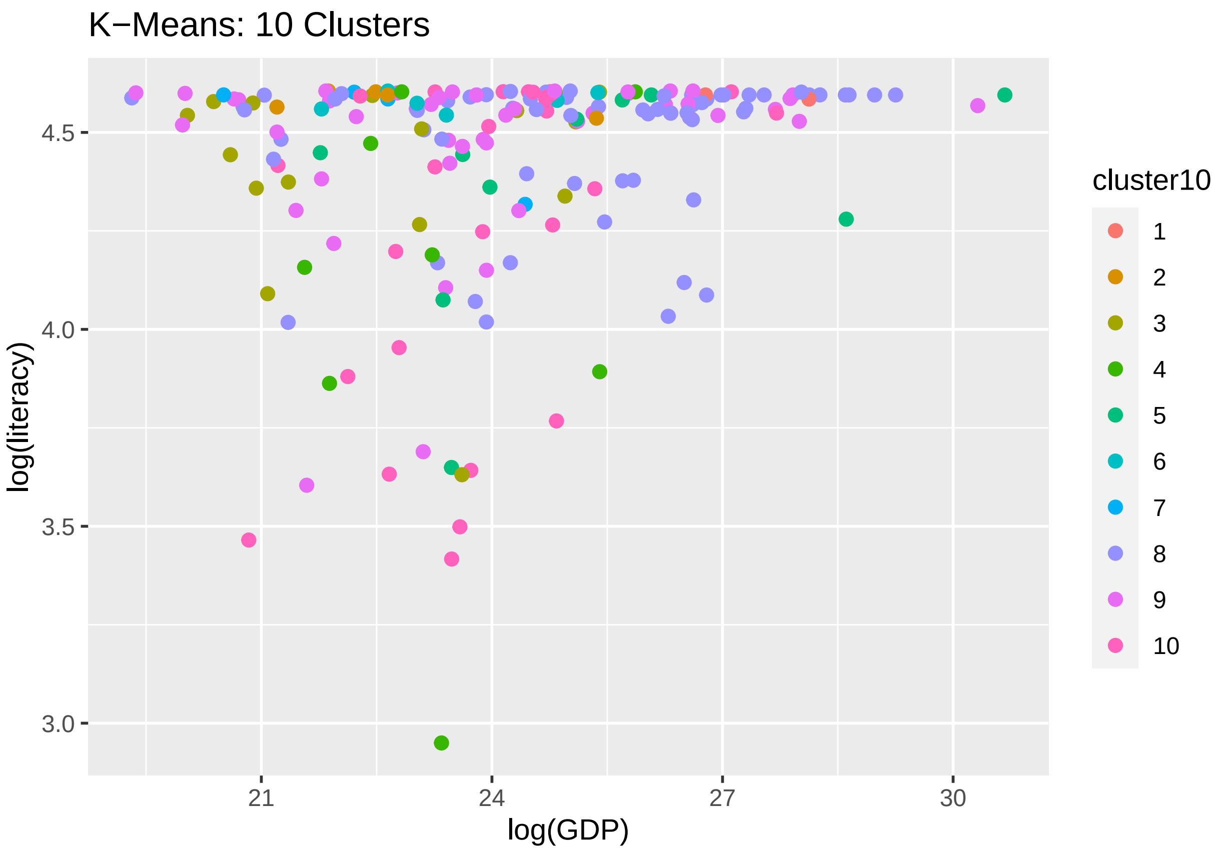

```
p2 <- ggplot(master_df_k, aes(x = log(GDP), y = log(Expectancy), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("K-Means: 10 Clusters") + scale_fill_brewer(palette="Set3")
```



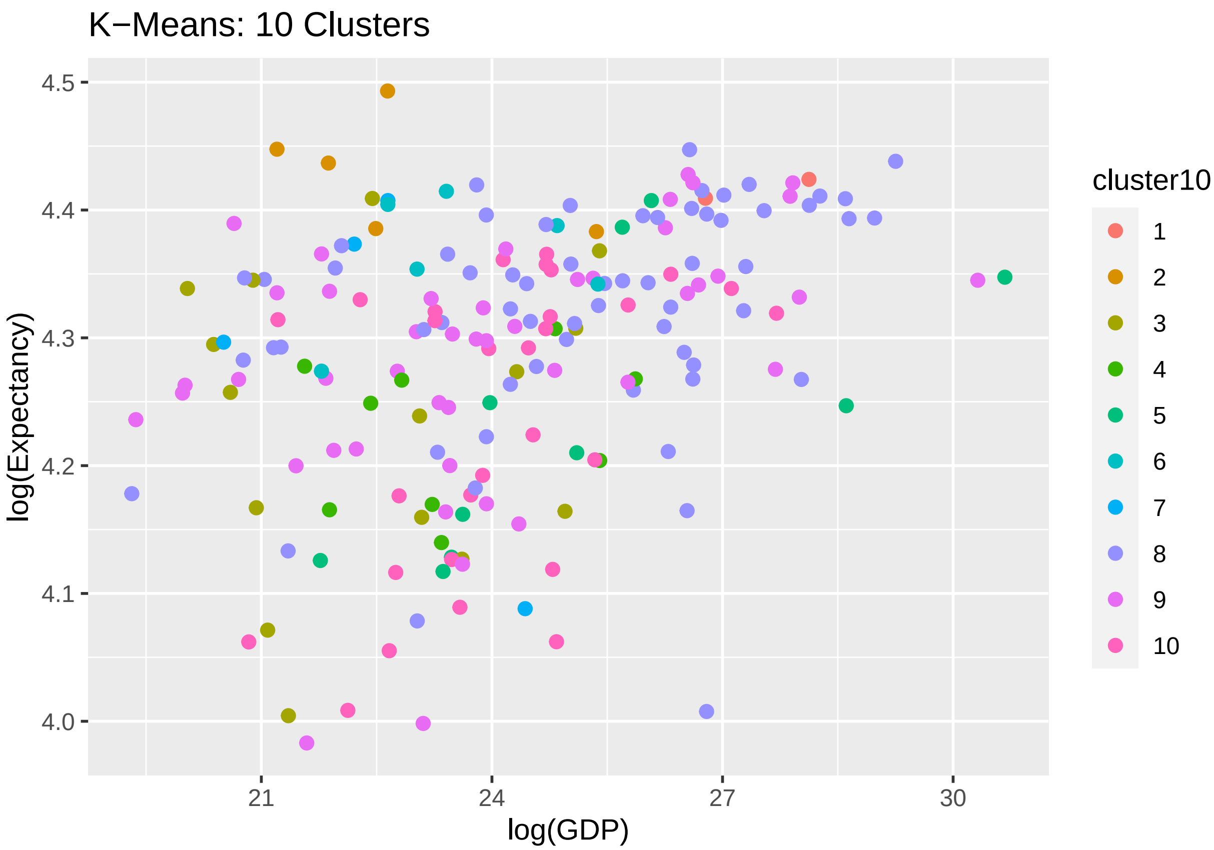

```r
p2 <- ggplot(master_df_k, aes(x = log(GDP), y = log(Fertility), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("K-Means: 10 Clusters") + scale_fill_brewer(palette="Set3")
```



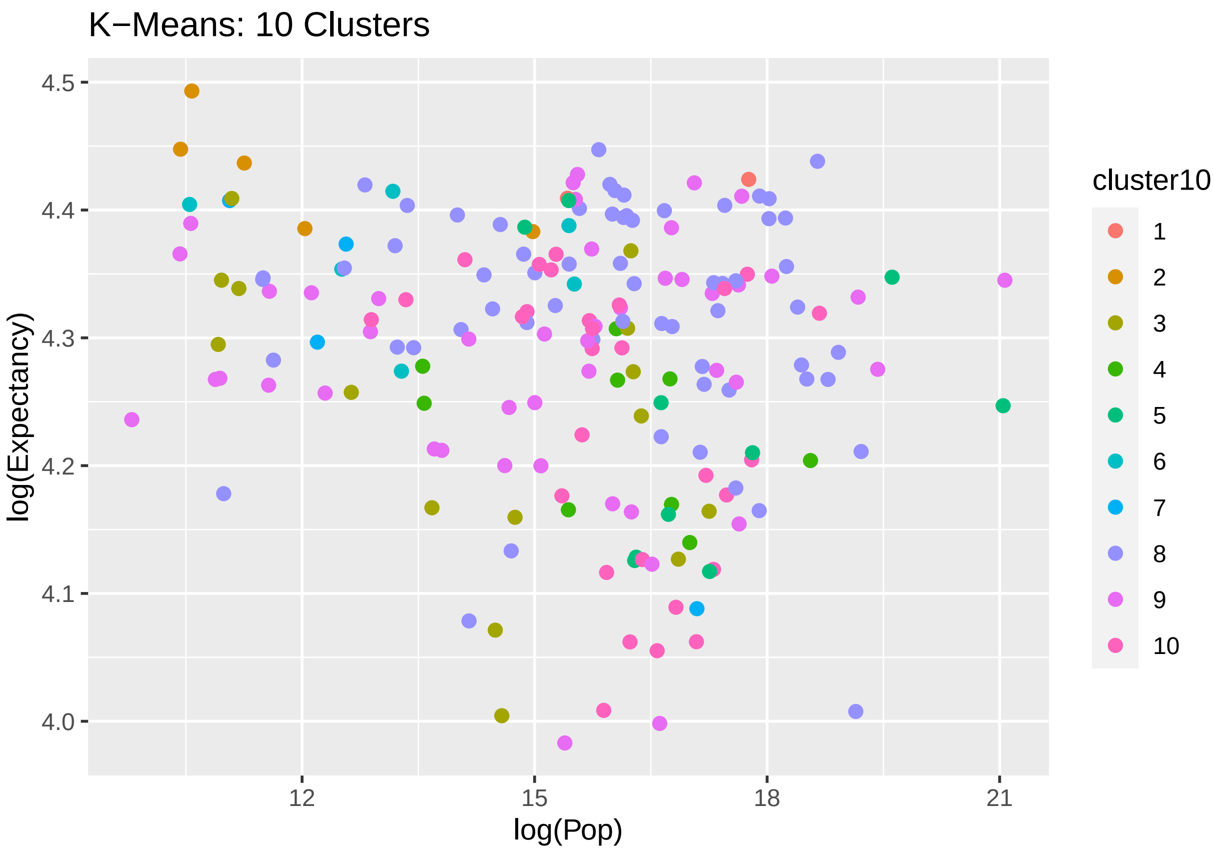

```r
p2 <- ggplot(master_df_k, aes(x = log(GDP), y = log(literacy), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("K-Means: 10 Clusters") + scale_fill_brewer(palette="Set3")
```



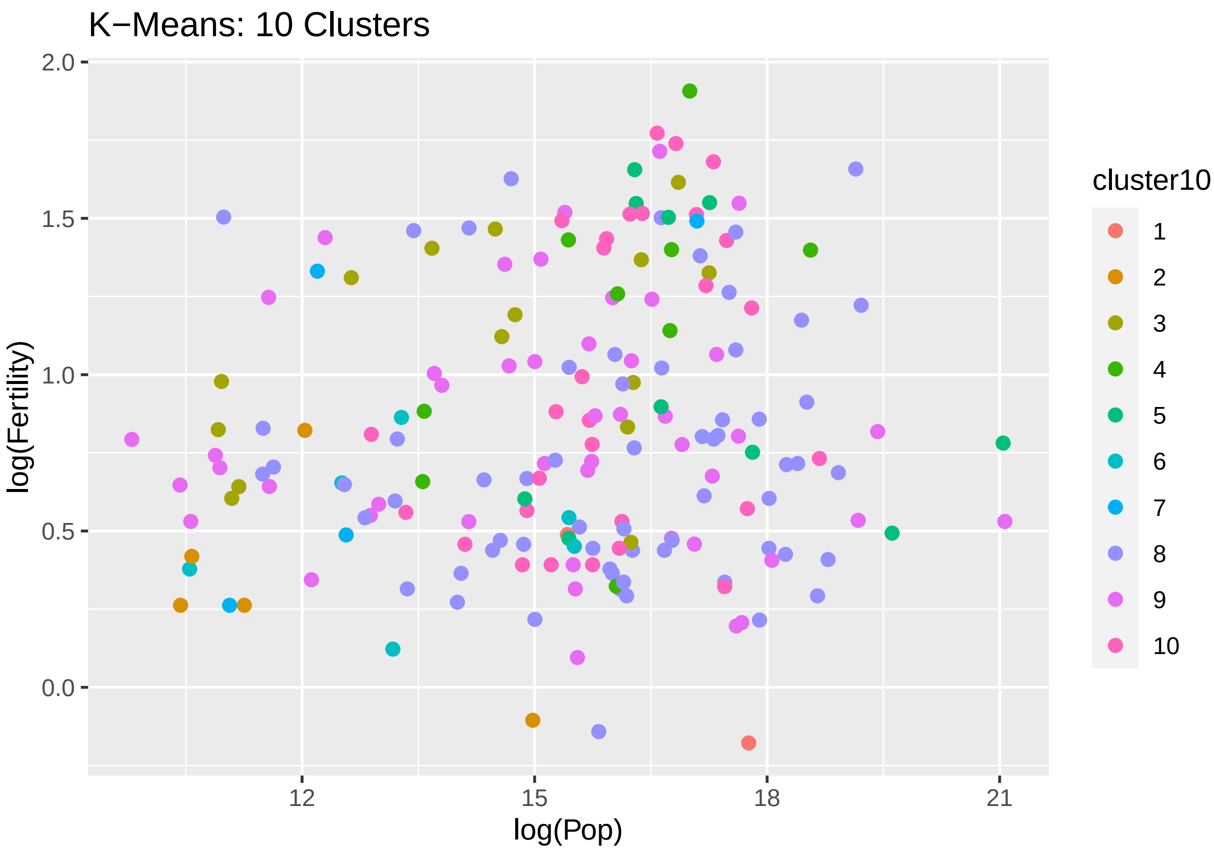

```r
p2 <- ggplot(master_df_k, aes(x = log(GDP), y = log(Expectancy), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("K-Means: 10 Clusters") + scale_fill_brewer(palette="Set3")
```



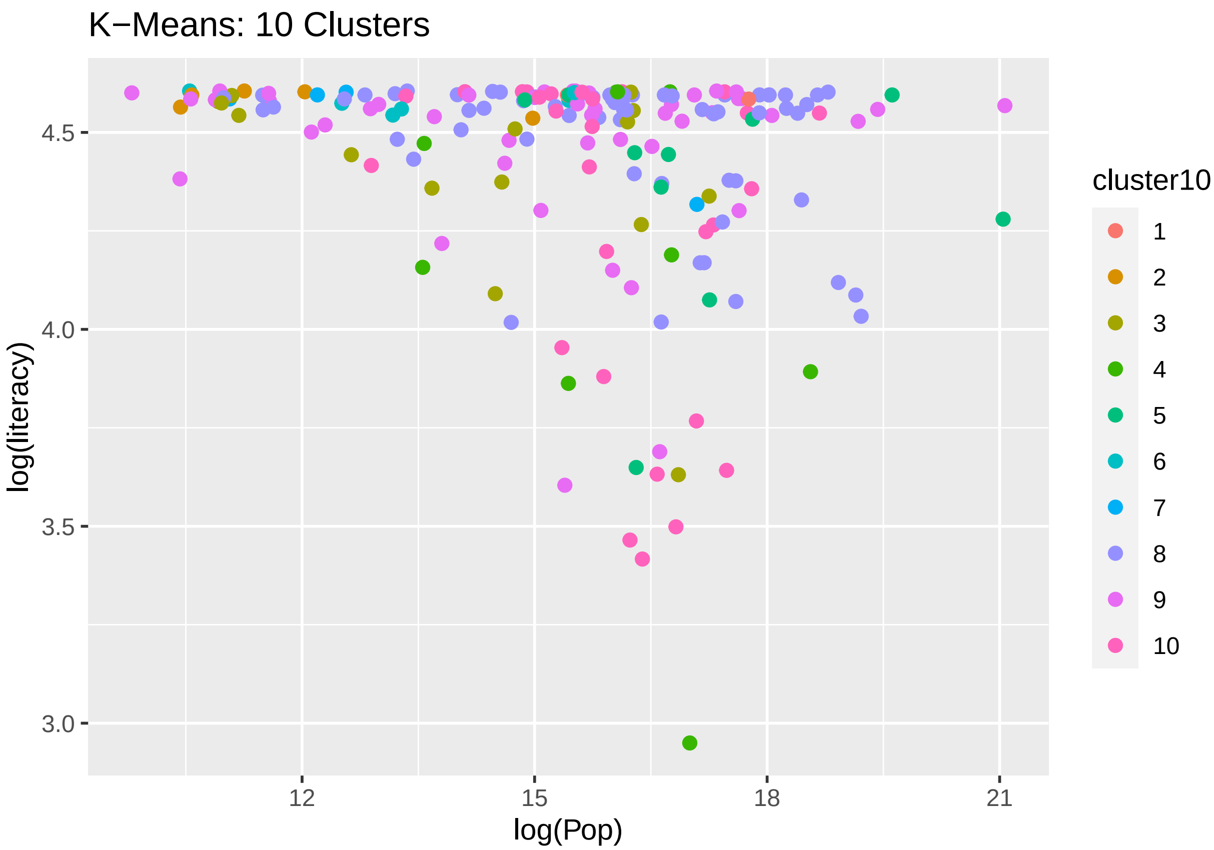

```
p2 <- ggplot(master_df_k, aes(x = log(Pop), y = log(Expectancy), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("K-Means: 10 Clusters") + scale_fill_brewer(palette="Set3")
```



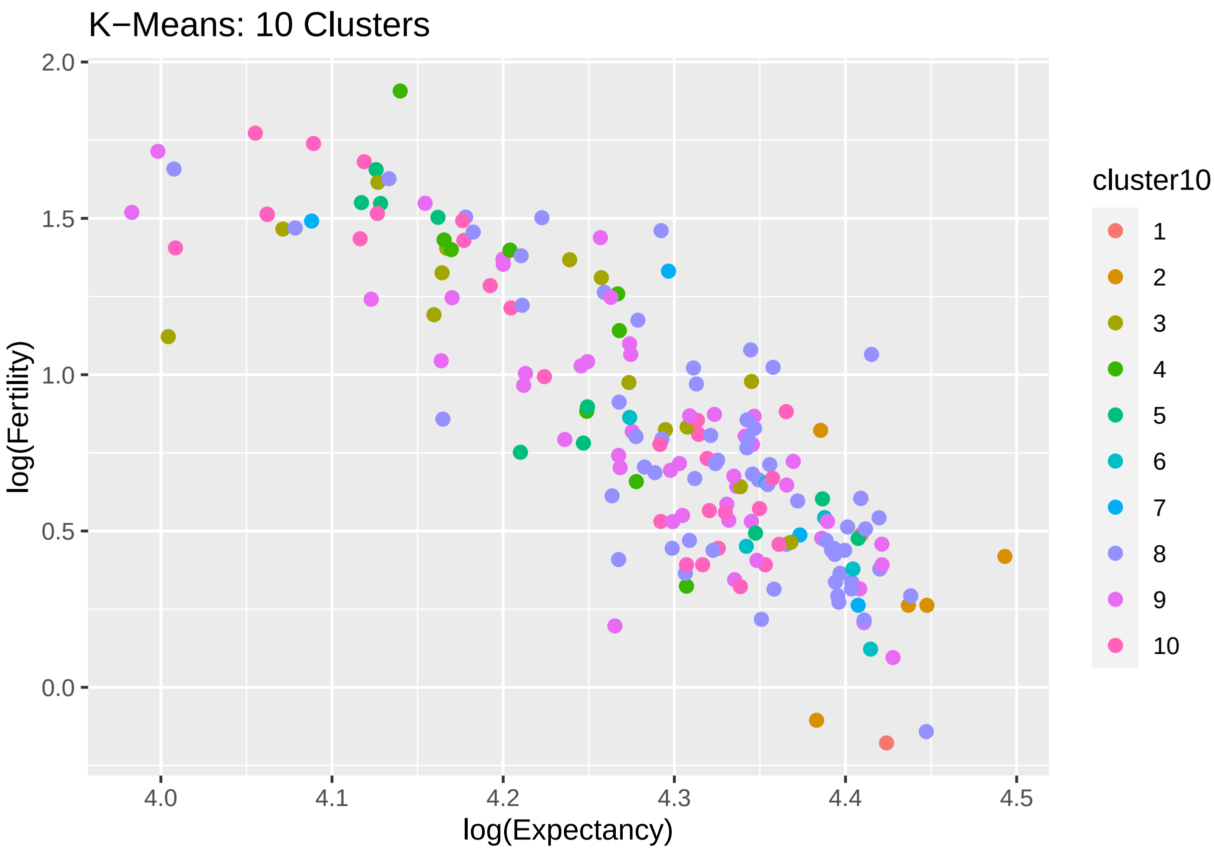

```
p2 <- ggplot(master_df_k, aes(x = log(Pop), y = log(Fertility), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("K-Means: 10 Clusters") + scale_fill_brewer(palette="Set3")
```



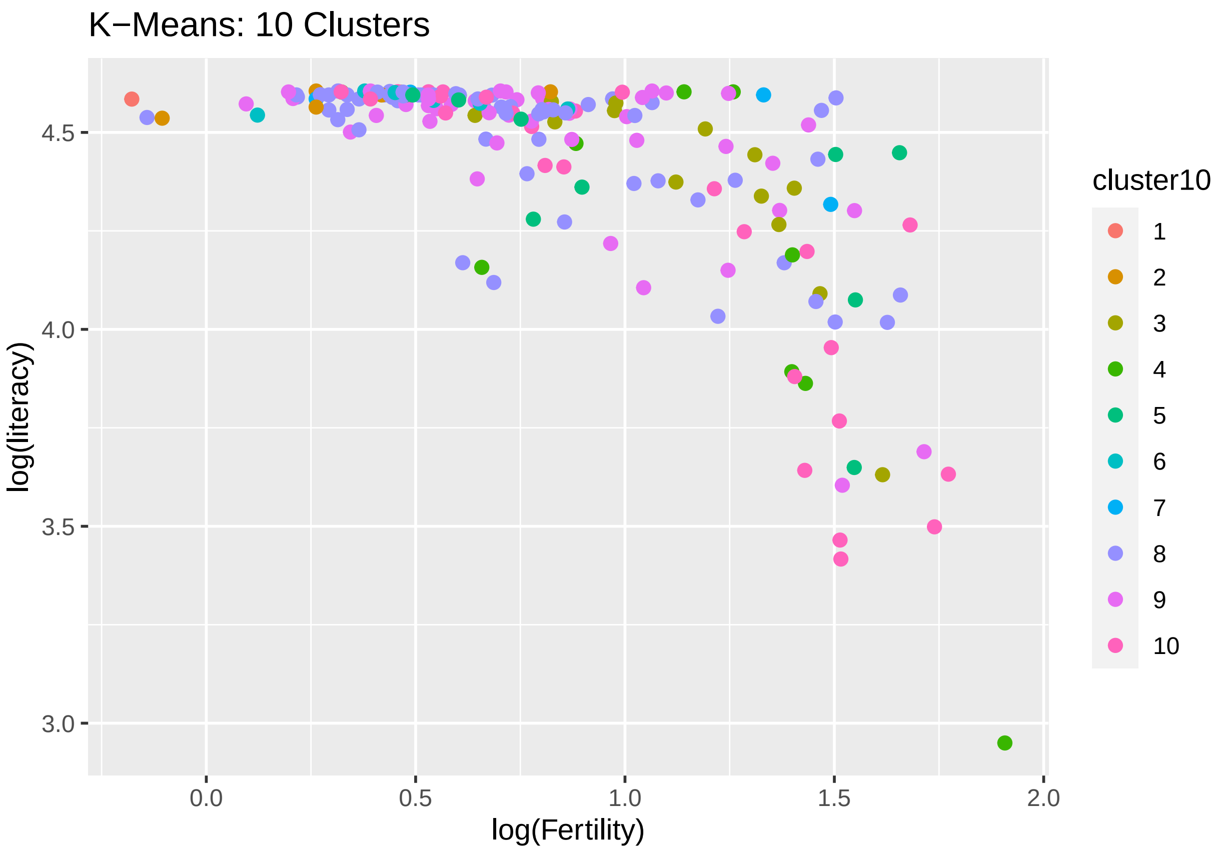

```
p2 <- ggplot(master_df_k, aes(x = log(Pop), y = log(literacy), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("K-Means: 10 Clusters") + scale_fill_brewer(palette="Set3")
```




```r
p2 <- ggplot(master_df_k, aes(x = log(Expectancy), y = log(Fertility), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("K-Means: 10 Clusters") + scale_fill_brewer(palette="Set3")
```



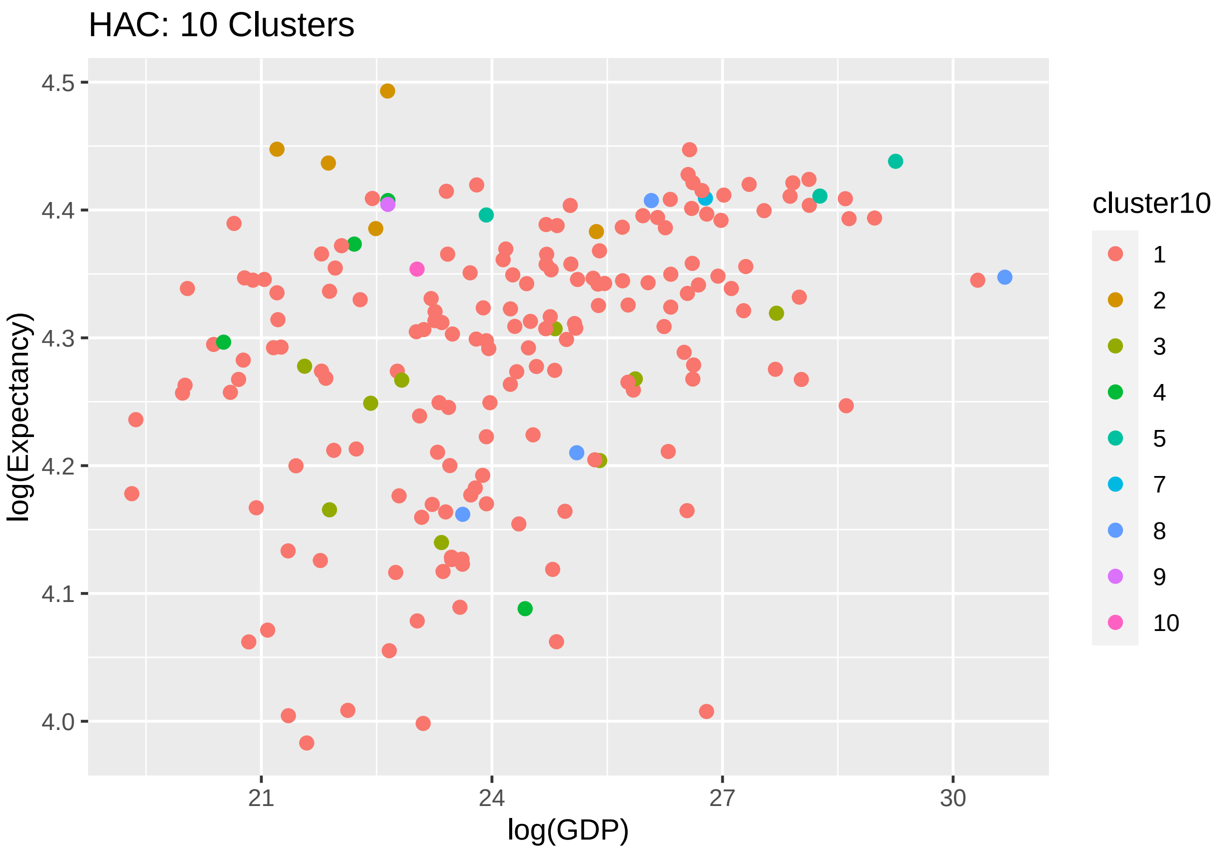

```r
p2 <- ggplot(master_df_k, aes(x = log(Fertility), y = log(literacy), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("K-Means: 10 Clusters") + scale_fill_brewer(palette="Set3")
```



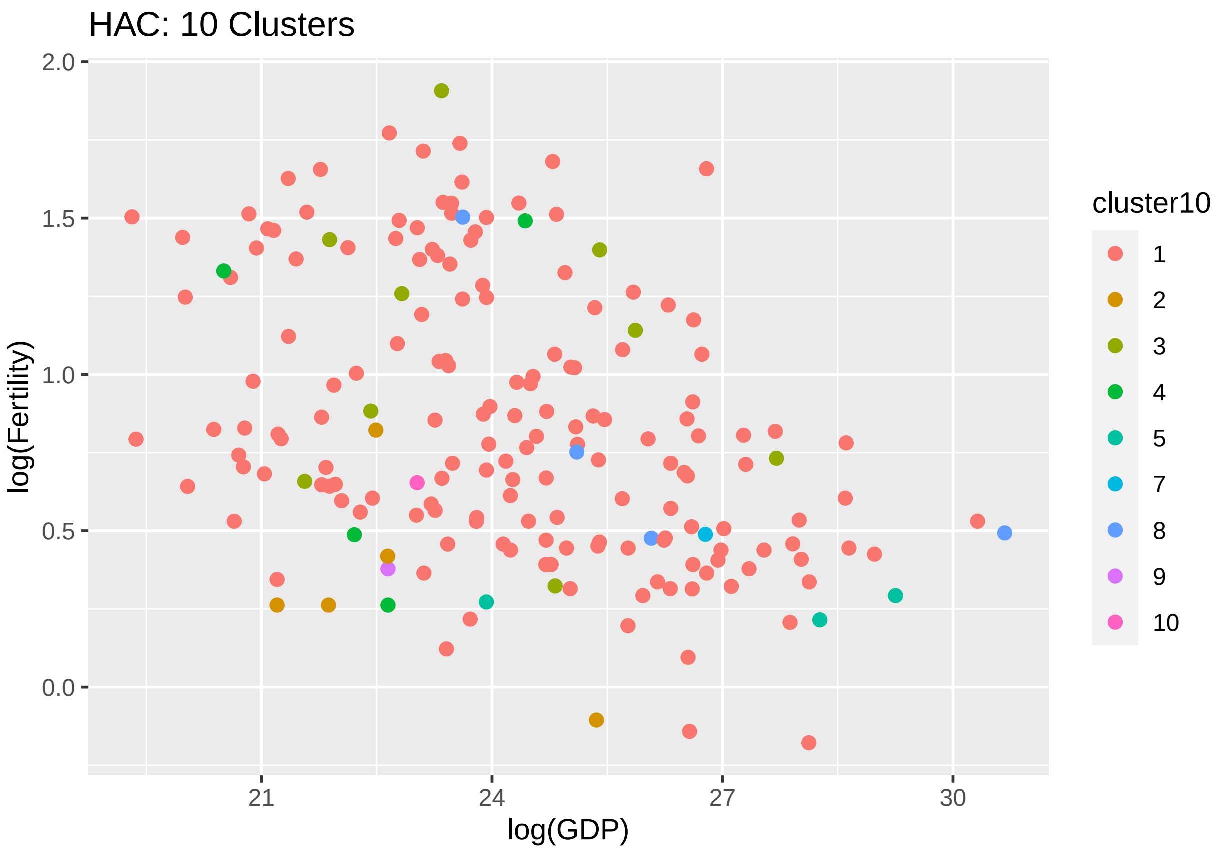

```r
#jpeg(file="kmeans_6.jpg")
#dev.off()

#jpeg(file="hac_6.jpg")
# p3 <- ggplot(master_df_h, aes(x = log(Pop), y = log(Expectancy), color = cluster10)) + geom_point(siz
# p3 + ggtitle("HAC: 6 Clusters") + scale_fill_brewer(palette="Set3")
#dev.off()
```

```r
p2 <- ggplot(master_df_h, aes(x = log(GDP), y = log(Pop), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("HAC: 10 Clusters") + scale_fill_brewer(palette="Set3")
```


```
p2 <- ggplot(master_df_h, aes(x = log(GDP), y = log(Expectancy), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("HAC: 10 Clusters") + scale_fill_brewer(palette="Set3")
```

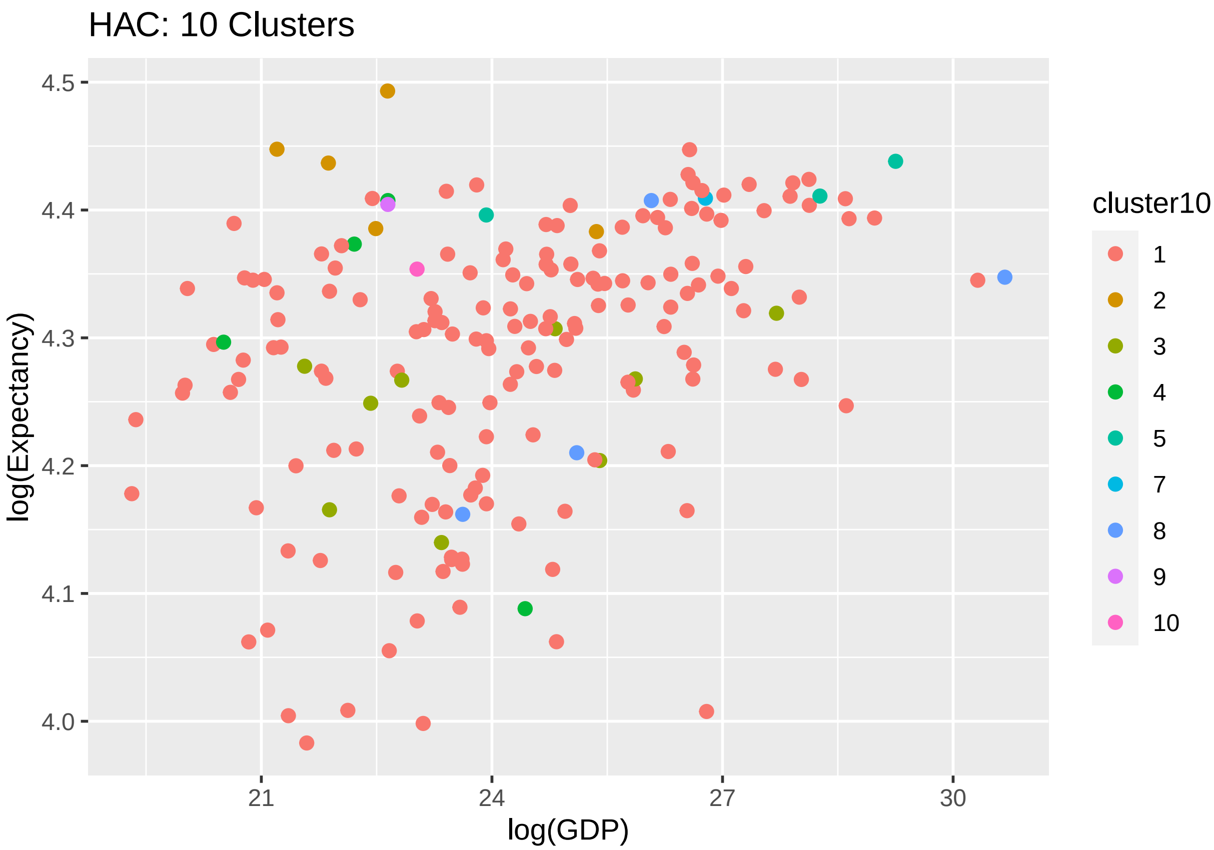



```r
p2 <- ggplot(master_df_h, aes(x = log(GDP), y = log(Fertility), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("HAC: 10 Clusters") + scale_fill_brewer(palette="Set3")
```

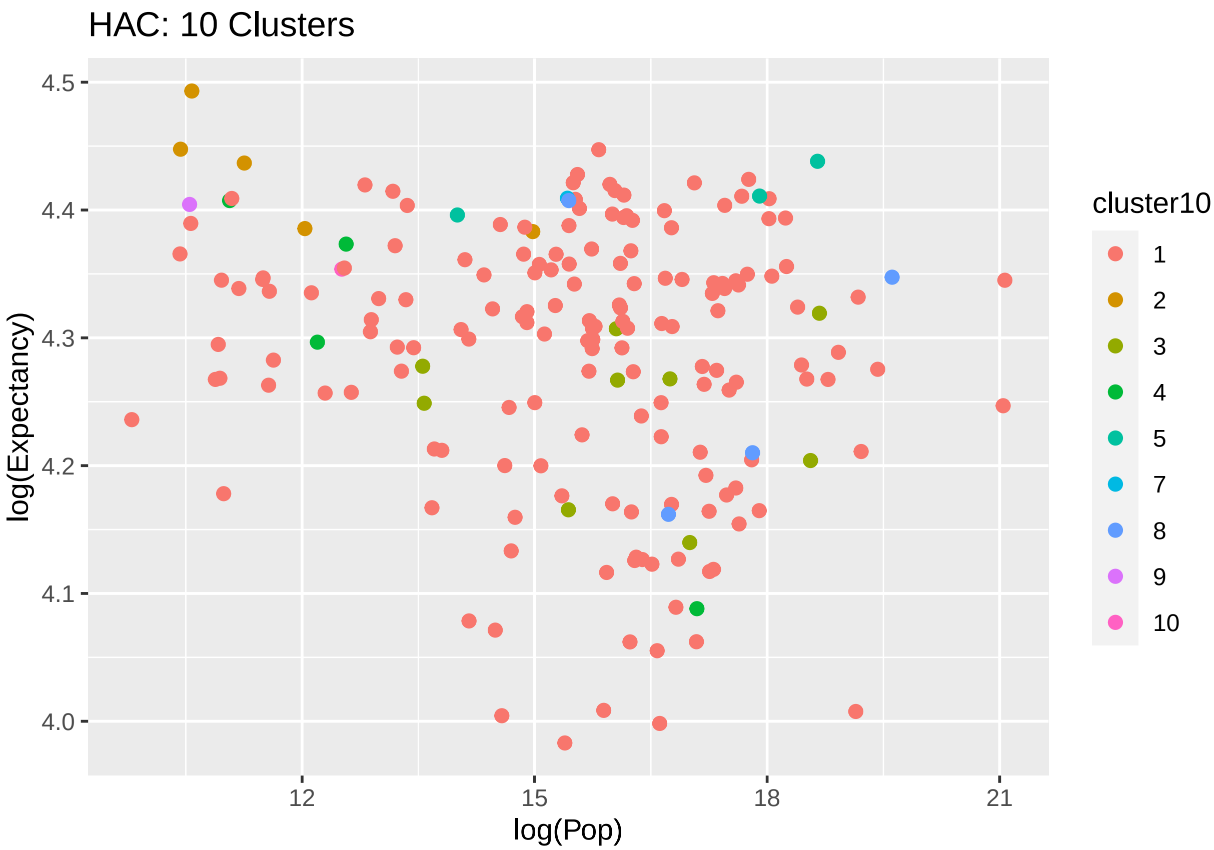



```
p2 <- ggplot(master_df_h, aes(x = log(GDP), y = log(literacy), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("HAC: 10 Clusters") + scale_fill_brewer(palette="Set3")
```


```
p2 <- ggplot(master_df_h, aes(x = log(GDP), y = log(Expectancy), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("HAC: 10 Clusters") + scale_fill_brewer(palette="Set3")
```

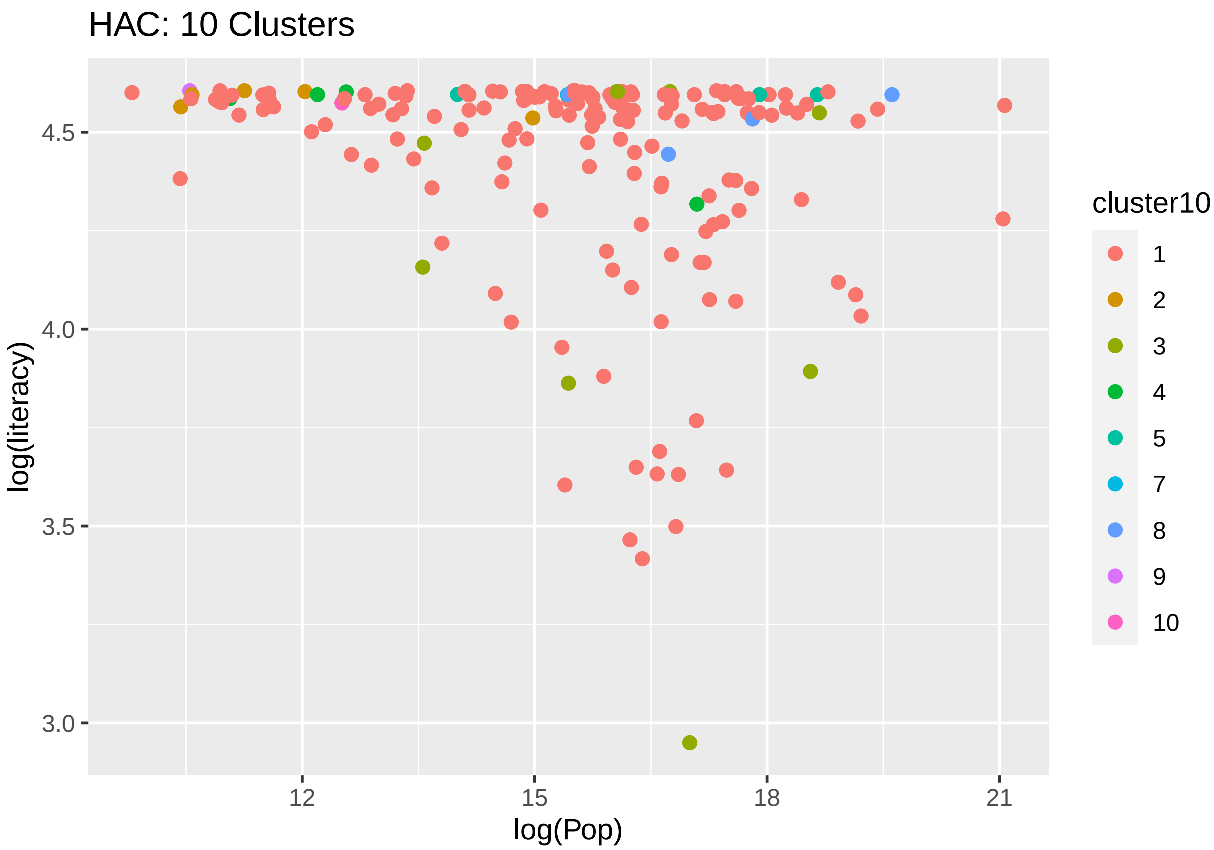



```
p2 <- ggplot(master_df_h, aes(x = log(Pop), y = log(Expectancy), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("HAC: 10 Clusters") + scale_fill_brewer(palette="Set3")
```



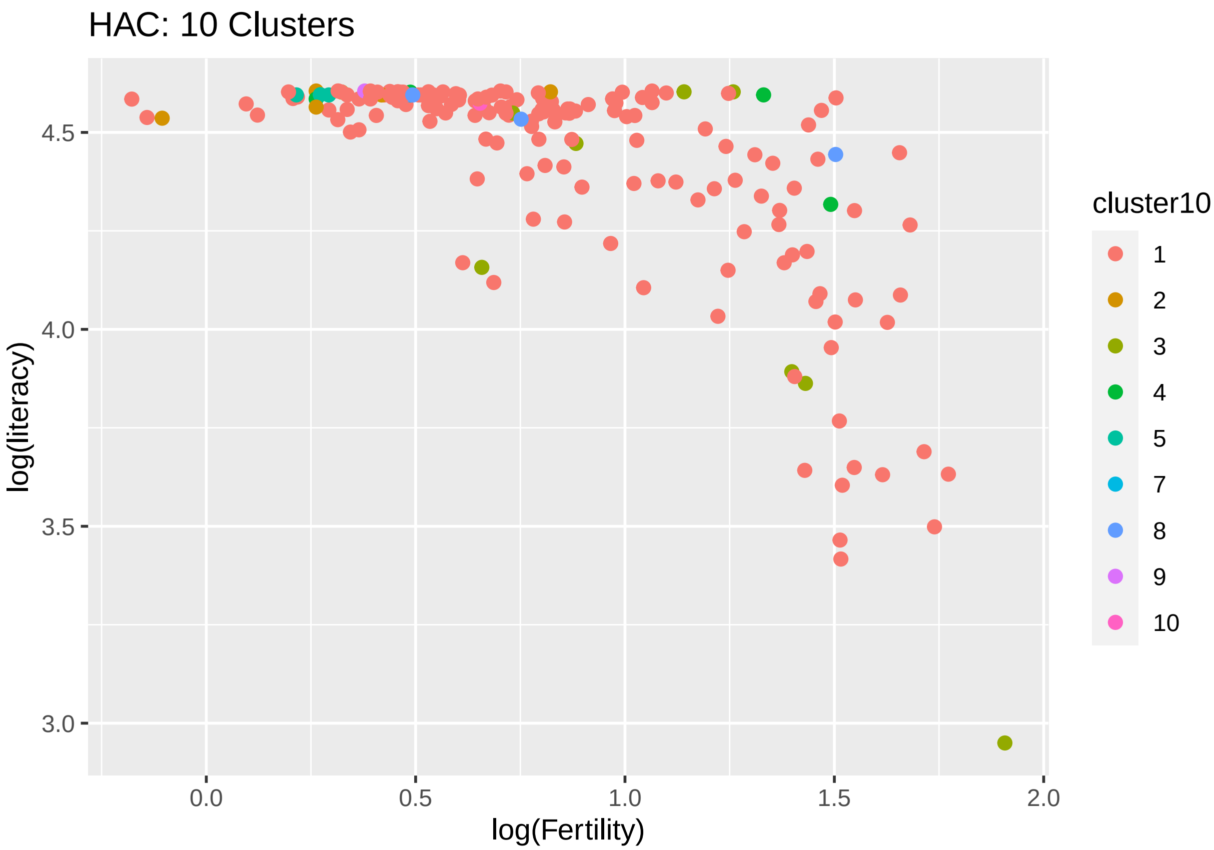

```
p2 <- ggplot(master_df_h, aes(x = log(Pop), y = log(Fertility), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("HAC: 10 Clusters") + scale_fill_brewer(palette="Set3")
```


```
p2 <- ggplot(master_df_h, aes(x = log(Pop), y = log(literacy), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("HAC: 10 Clusters") + scale_fill_brewer(palette="Set3")
```




```r
p2 <- ggplot(master_df_h, aes(x = log(Expectancy), y = log(Fertility), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("HAC: 10 Clusters") + scale_fill_brewer(palette="Set3")
```


```
p2 <- ggplot(master_df_h, aes(x = log(Fertility), y = log(literacy), color = cluster10)) +
  geom_point(size=2)
p2 + ggtitle("HAC: 10 Clusters") + scale_fill_brewer(palette="Set3")
```




```
# not sure how meaningful something like this would be? definitely needs some cleaning up
# source of code: https://stackoverflow.com/questions/47842646/labelling-outliers-with-ggplot
jpeg(file="boxplot_k_literacy.jpg")
ggplot(master_df_k, aes(x = cluster10, y = literacy, fill = cluster10)) +
  geom_boxplot(alpha = 0.3) +
  geom_point(aes(color = cluster10, group = cluster10), position = position_dodge(width=0.75)) +
  geom_text_repel(aes(group = cluster10,
              label = ifelse(test = literacy > median(literacy) + 1.5*IQR(literacy)
                             | literacy < median(literacy) - 1.5*IQR(literacy),
                yes = ISO3,
                no = '')),
          position = position_dodge(width=0.75),
          hjust = "left", size = 3) + ggtitle("HAC GDP Boxplot") + xlab("Policy Cluster") +
  theme(legend.position = "none")
dev.off()
```

```
## pdf
##   2
```

```
jpeg(file="boxplot_k_fertility.jpg")
ggplot(master_df_k, aes(x = cluster10, y = Fertility, fill = cluster10)) +
  geom_boxplot(alpha = 0.3) +
  geom_point(aes(color = cluster10, group = cluster10), position = position_dodge(width=0.75)) +
  geom_text_repel(aes(group = cluster10,
              label = ifelse(test = Fertility > median(Fertility) + 1.5*IQR(Fertility)
                             | Fertility < median(Fertility) - 1.5*IQR(Fertility),
                yes = ISO3,
                no = '')),
```

```
            position = position_dodge(width=0.75),
            hjust = "left", size = 3) + ggtitle("HAC GDP Boxplot") + xlab("Policy Cluster") +
  theme(legend.position = "none")

dev.off()
```

```
## pdf
##   2
```

```
jpeg(file="boxplot_h_gdp.jpg")
ggplot(master_df_h[!is.na(master_df_h$GDP), ], aes(x = cluster10, y = log(GDP), fill =
                                                    cluster10)) +
  geom_boxplot(alpha = 0.3) +
  geom_point(aes(color = cluster10, group = cluster10), position = position_dodge(width=0.75)) +
  geom_text_repel(aes(group = cluster10,
                label = ifelse(test = log(GDP) > median(log(GDP)) + 1.5*IQR(log(GDP))
                               | log(GDP) < median(log(GDP)) - 1.5*IQR(log(GDP)),
                  yes = ISO3,
                  no = '')),
            position = position_dodge(width=0.75),
            hjust = "left", size = 3) +
  ggtitle("HAC GDP Boxplot") + xlab("Policy Cluster") + theme(legend.position = "none")
dev.off()
```

```
## pdf
##   2
```

```
# ggrepel
# deal with missing data
```

```
jpeg(file = "boxplot_k_expectancy.jpg")
ggplot(master_df_k,
       aes(x = cluster10, y = Expectancy, fill = cluster10)) +
  geom_boxplot(alpha = 0.3) +
  geom_point(aes(color = cluster10, group = cluster10), position = position_dodge(width=0.75)) +
  geom_text_repel(aes(group = cluster10,
                label = ifelse(test = Expectancy > median(Expectancy)
                               + 1.5*IQR(Expectancy) | Expectancy <
                                 median(Expectancy) - 1.5*IQR(Expectancy),
                  yes = ISO3,
                  no = '')),
            position = position_dodge(width=0.75),
            hjust = "left", size = 3) + ggtitle("HAC GDP Boxplot") +
  xlab("Policy Cluster") + theme(legend.position = "none")
dev.off()
```

```
## pdf
##   2
```

```
## BOXPLOT QUESTIONS
jpeg(file = "boxplot_h_expectancy.jpg")
ggplot(master_df_h,
       aes(x = cluster10, y = Expectancy, fill = cluster10)) +
  geom_boxplot(alpha = 0.3) +
  geom_point(aes(color = cluster10, group = cluster10), position = position_dodge(width=0.75)) +
  geom_text_repel(aes(group = cluster10,
```

```
                label = ifelse(test = Expectancy > median(Expectancy)
                               + 1.5*IQR(Expectancy) | Expectancy <
                                 median(Expectancy) - 1.5*IQR(Expectancy),
                  yes = ISO3,
                  no = '')),
            position = position_dodge(width=0.75),
            hjust = "left", size = 3) + ggtitle("HAC GDP Boxplot") + xlab("Policy Cluster") +
  theme(legend.position = "none")
dev.off()
```

```
## pdf
##   2
```

```
iso3$Alpha.3.code <- trimws(iso3$Alpha.3.code)
names_h <- merge(master_df_h, iso3, by.x = "ISO3", by.y = "Alpha.3.code")
names_k <- merge(master_df_k, iso3, by.x = "ISO3", by.y = "Alpha.3.code")
countries_h <- data.frame()
countries_k <- rep(NA, 10)
for (i in 1:10) {
  names_h[names_h$cluster10 == i, ]$Country
  names_h[names_k$cluster10 == i, ]$Country
}
```

```
#appendix (use xtable to format into Latex)
# kableExtra
tb <- split(master_df_k, master_df_k$cluster10)
```

FINAL OFFICE HOURS: - moving the tables in the data section to the appendix? - standardized or original data summary statistics in the table? both? - telling a meaningful story with the scatter plot? - difficult drawing general conclusions – don't seem like I have a ton of meaningful results :(– or are there ways to find general, practical results that don't involve scoring / evaluating the different policies on "strictness?"

- classification tree, multinomial
- top two principal components, color code (or focusing on two variables of interest)

```
cluster_means <- aggregate(cbind(VISA_BAN_NONE, VISA_BAN_SPECIFIC, VISA_BAN_ALL,
                HISTORY_BAN_CLEANED,
                CITIZEN_LIST_CLEANED, POLICY_LENGTH, POLICY_TYPE_NON,
                POLICY_TYPE_COMPLETE, POLICY_TYPE_PARTIAL,
                AIR, LAND,
                SEA, REFUGEE, COUNTRY_EXCEP, WORK_EXCEP)~cluster10, data = master_df_k, mean)
cluster_means <- lapply(cluster_means, function(x) if(is.numeric(x)) round(x, 3) else x)
cluster_means <- data.frame(cluster_means)
kable(cluster_means, format = "latex")
```

cluster10	VISA_BAN_NONE	VISA_BAN_SPECIFIC	VISA_BAN_ALL	HISTORY_BAN_CLEANED	CITI
1	-3.038	4.496	0.499	-0.205	
2	0.193	-0.110	-0.156	-0.208	
3	0.193	-0.110	-0.156	-0.199	
4	0.073	0.095	-0.156	-0.204	
5	-1.735	0.129	2.021	-0.192	
6	0.193	-0.110	-0.156	-0.205	
7	0.193	-0.110	-0.156	-0.208	
8	0.039	-0.013	-0.039	0.009	
9	0.176	-0.110	-0.136	-0.165	
10	0.135	-0.085	-0.103	-0.154	

```r
# cluster 4 has longest policies
# cluster 5 has strictest policies against refugees
# cluster 6 has highest country exception list (and second highest work exception)
```

```r
master_df_k[is.na(master_df_k$GDP),]
```

```
##  [1] ISO3                VISA_BAN_NONE       VISA_BAN_SPECIFIC
##  [4] VISA_BAN_ALL        HISTORY_BAN_CLEANED CITIZEN_LIST_CLEANED
##  [7] POLICY_LENGTH       POLICY_TYPE_NON     POLICY_TYPE_COMPLETE
## [10] POLICY_TYPE_PARTIAL AIR                 LAND
## [13] SEA                 REFUGEE             COUNTRY_EXCEP
## [16] WORK_EXCEP          cluster10           GDP
## [19] Pop                 Expectancy          Fertility
## [22] literacy
## <0 rows> (or 0-length row.names)
```

```r
# prob add into the original GDP data frame (so we have it in HAC too)
# ABW -- 3202 x 10^6
# AND -- 3155 x 10^6
# ERI -- 2.07 bil (2011)
# GIB -- 2,885,810,912.00
# GRL -- 3052 x 10^6
# LIE -- 6,839 x 10^6
# MNP -- 1,182 x 10^6
# NCL -- 10 bil
# PYF -- 3.45 bil
# SMR -- 1616 mil
# SSD -- 1,119.7 mil
# TKM -- 45231 mil
# VEN -- 47.26 bil
# YEM -- 23,486 mil
```

```r
master_df_k[is.na(master_df_k$Expectancy),]
```

```
##  [1] ISO3                VISA_BAN_NONE       VISA_BAN_SPECIFIC
##  [4] VISA_BAN_ALL        HISTORY_BAN_CLEANED CITIZEN_LIST_CLEANED
##  [7] POLICY_LENGTH       POLICY_TYPE_NON     POLICY_TYPE_COMPLETE
## [10] POLICY_TYPE_PARTIAL AIR                 LAND
## [13] SEA                 REFUGEE             COUNTRY_EXCEP
## [16] WORK_EXCEP          cluster10           GDP
## [19] Pop                 Expectancy          Fertility
## [22] literacy
## <0 rows> (or 0-length row.names)
```

```r
# AND - 84.5
# ASM - 73.32
# CYM - 82.19
# DMA - 76.6
# GIB - 78.7
# KNA - 71.34
# MCO - 89.4
# MHL - 65.24
# MNP - 77.1
# PLW - 69.13
# SMR - 85.42
```

```r
# TCA - 80.6
```

```r
master_df_k[is.na(master_df_k$Fertility),]
```

```
##  [1] ISO3                 VISA_BAN_NONE        VISA_BAN_SPECIFIC
##  [4] VISA_BAN_ALL         HISTORY_BAN_CLEANED  CITIZEN_LIST_CLEANED
##  [7] POLICY_LENGTH        POLICY_TYPE_NON      POLICY_TYPE_COMPLETE
## [10] POLICY_TYPE_PARTIAL  AIR                  LAND
## [13] SEA                  REFUGEE              COUNTRY_EXCEP
## [16] WORK_EXCEP           cluster10            GDP
## [19] Pop                  Expectancy           Fertility
## [22] literacy
## <0 rows> (or 0-length row.names)
```

```r
# AND - 1.3
# ASM - 2.28
# CYM - 1.83
# DMA - 1.9
# GIB - 1.91
# KNA - 2.1
# MCO - 1.52
# MHL - 4.5
# MNP - 2.66
# PLW - 2.21
# SMR - 1.3
# TCA - 1.7
```