# Jane-Downer-Homework10

### Jane Downer

### 12/12/2020

```
#install.packages("lsa")
library(lsa)
```

```
## Loading required package: SnowballC
```

```
rm(list=ls())
setwd("/Users/user/Desktop/CS_422/HW10")
```

## Part 2.1-a-i

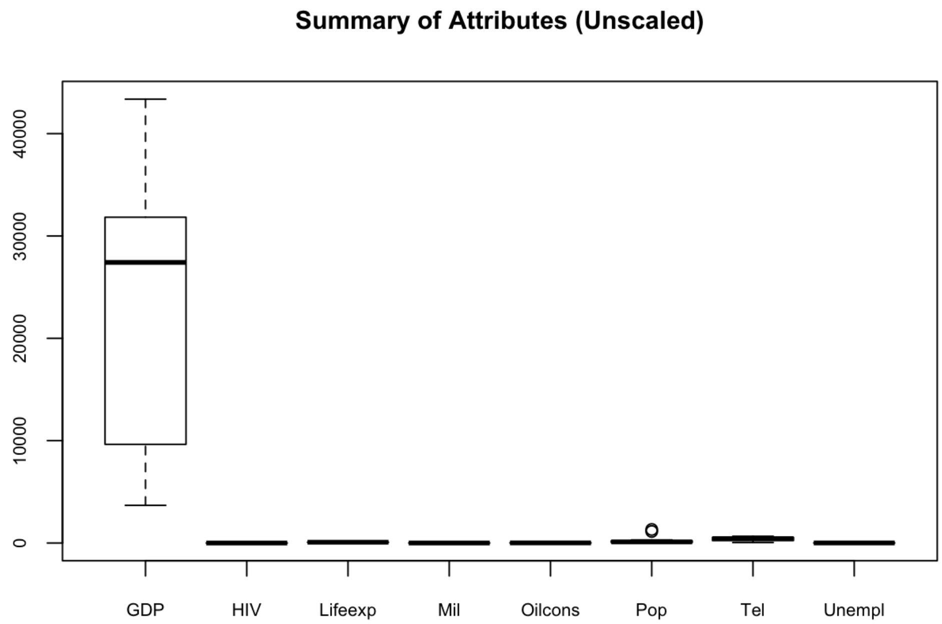
```
countries_csv <- read.csv('countries.csv')
countries <- data.frame(countries_csv[,-1], row.names = countries_csv[,1])
countries_scaled = scale(countries)

# Summary of unscaled data
summary(countries)
```

```
##       GDP             HIV            Lifeexp          Mil
##  Min.   : 3685   Min.   :0.1000   Min.   :65.90   Min.   :0.500
##  1st Qu.: 9640   1st Qu.:0.1000   1st Qu.:72.55   1st Qu.:1.350
##  Median :27418   Median :0.3000   Median :78.00   Median :2.500
##  Mean   :22296   Mean   :0.4133   Mean   :76.06   Mean   :2.253
##  3rd Qu.:31832   3rd Qu.:0.6500   3rd Qu.:79.85   3rd Qu.:2.700
##  Max.   :43369   Max.   :1.1000   Max.   :82.00   Max.   :4.300
##     Oilcons           Pop              Tel            Unempl
##  Min.   : 0.80   Min.   :  33.0   Min.   : 44.0   Min.   : 2.90
##  1st Qu.: 5.25   1st Qu.:  59.5   1st Qu.:241.3   1st Qu.: 4.15
##  Median :11.30   Median : 109.0   Median :432.8   Median : 6.60
##  Mean   :10.91   Mean   : 262.8   Mean   :391.3   Mean   : 6.42
##  3rd Qu.:15.10   3rd Qu.: 212.5   3rd Qu.:550.4   3rd Qu.: 7.95
##  Max.   :25.10   Max.   :1322.0   Max.   :657.8   Max.   :12.50
```

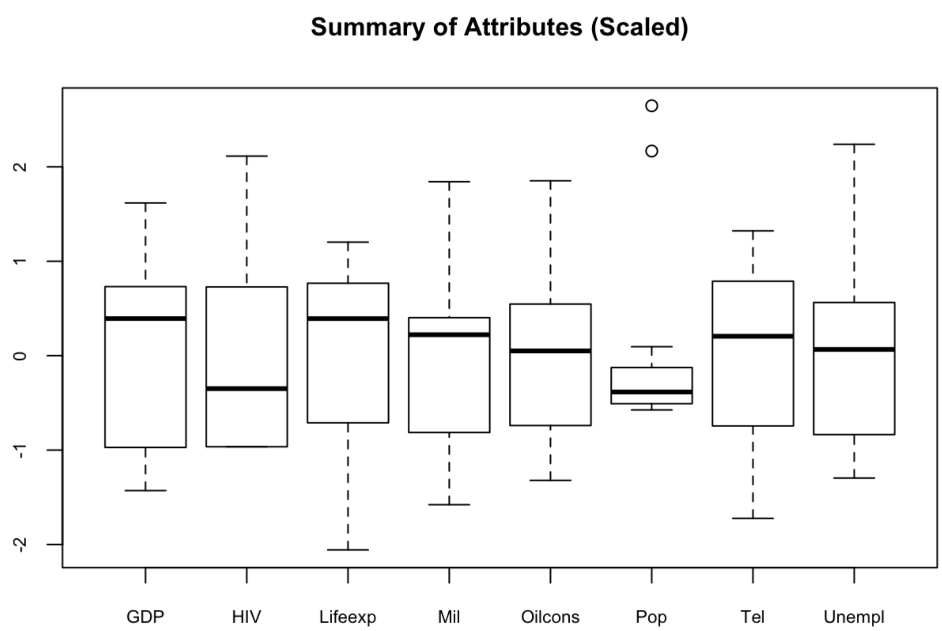
## Part 2.1-a-ii

I've included two boxplots here: one unscaled, and one scaled (which the instructions do not ask for but which I found is much more helpful.)

```
boxplot(countries,  cex.axis=0.7, main = "Summary of Attributes (Unscaled)", cex.main = 1)
```




```
boxplot(countries_scaled,  cex.axis=0.7, main = "Summary of Attributes (Scaled)", cex.main = 1)
```




The two outliers in the population category represent India and China, which are by far the most populous countries represented in the data.

## Part 2.1-b

```
#e <- eigen(cov(countries_scaled))
#row.names(e$vectors) <- c("GDP", "HIV", "Lifeexp", "Mil", "Oilcons", "Pop", "Tel","Unempl")
#colnames(e$vectors) <- c("PC1", "PC2", "PC3", "PC4", "PC5", "PC6", "PC7", "PC8")
#e

#phi <- -e$vectors
#phi

#phi.1 <- as.matrix(phi[,1])
#PC1.score <- apply(X, 1, function(x) t(phi.1) %*% x)
#as.matrix(head(PC1.score))

#phi.2 <- as.matrix(phi[,2])
#PC2.score <- apply(X, 1, function(x) t(phi.2) %*% x)
#as.matrix(head(PC2.score))

pca <- prcomp(countries_scaled)
```

## Part 2.1-c-i

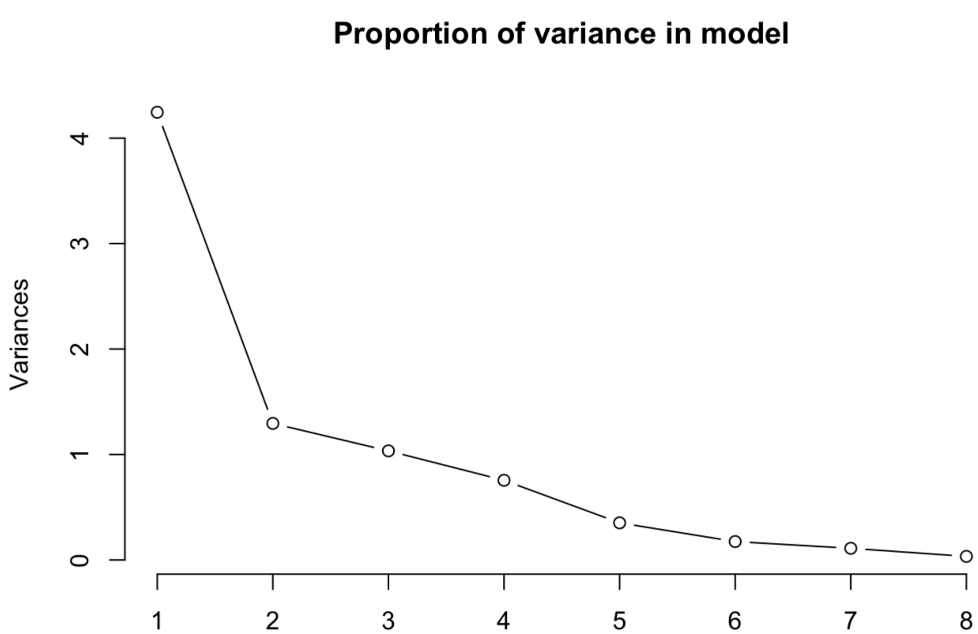
```
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     2.0605 1.1379 1.0170 0.8690 0.59319 0.41733 0.33233
## Proportion of Variance 0.5307 0.1618 0.1293 0.0944 0.04398 0.02177 0.01381
## Cumulative Proportion  0.5307 0.6925 0.8218 0.9162 0.96022 0.98199 0.99580
##                           PC8
## Standard deviation     0.1833
## Proportion of Variance 0.0042
## Cumulative Proportion  1.0000
```

4 components explain at least 90% of the data.

## Part 2.1-c-ii

```
screeplot(pca, type = 'l', main = "Proportion of variance in model")
```




## Part 2.1-c-iii

The “elbow” of the screeplot suggests that we should select two components for modeling if we were to engage in a feature reduction task.

## Part 2.1-d

```
#pca$rotation <- -pca$rotation
#pca$rotation
pca
```

```
## Standard deviations (1, .., p=8):
## [1] 2.0604984 1.1378752 1.0170179 0.8690036 0.5931903 0.4173294 0.3323291
## [8] 0.1833373
##
## Rotation (n x k) = (8 x 8):
##                PC1         PC2          PC3         PC4         PC5         PC6
## GDP      0.4560268  0.06362084 -0.271093835 -0.05442480  0.14947678 -0.08002969
## HIV     -0.1934368 -0.32800239 -0.694218852  0.48404674  0.19548002 -0.23804695
## Lifeexp  0.4407804  0.01452102  0.187653505 -0.14486052  0.44812657 -0.24828417
## Mil     -0.1964834  0.56706336 -0.480082473 -0.41625346 -0.32517176 -0.05002767
## Oilcons  0.4275287  0.03644676 -0.320454603  0.04650622  0.07839373  0.78889952
## Pop     -0.3150143  0.54189871  0.006639875  0.07981396  0.72635243  0.05264696
## Tel      0.4423016  0.14093180 -0.200094388 -0.11206139 -0.07267202 -0.49436031
## Unempl  -0.2099599 -0.50174771 -0.190006038 -0.73986102  0.30639110  0.05705115
##                PC7         PC8
## GDP      0.1466324  0.81319670
## HIV      0.1456698 -0.15052276
## Lifeexp  0.5683508 -0.40474473
## Mil      0.3340639 -0.12747214
## Oilcons -0.1184137 -0.26173851
## Pop     -0.2583224  0.06006144
## Tel     -0.6471483 -0.25186858
## Unempl  -0.1520024  0.02084195
```

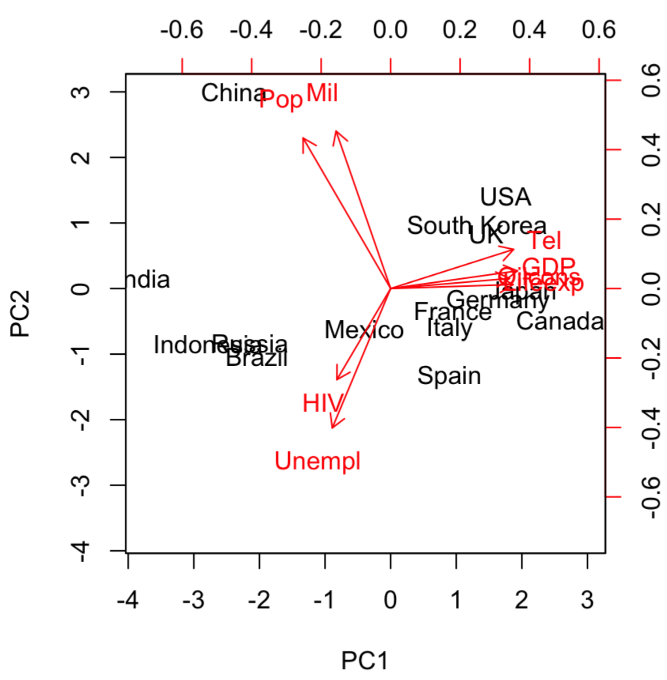
## Part 2.1-d-i

PC1 is positively correlated with GDP, Lifeexp, Oilcons, and Tel. It is negatively correlated with HIV, Mil, Pop, and Unempl. This suggests that smaller first-world countries explain much of the variance in the data.

## Part 2.1-d-ii

PC2 is positively correlated with GDP, Lifeexp, Mil, Oilcons, Pop, and Tel. It is negatively correlated with HIV and Unempl. This suggests that larger, wealthier countries contribute the second-largest component of variance to the dataset.

```
pca <- prcomp(countries_scaled)
biplot(pca, scale=0)
```



```
pca$x[c(1,9,14),c(1,2)]
```

```
##              PC1         PC2
## Brazil -2.037017 -1.03998472
## Japan   2.013116 -0.05735491
## UK      1.456991  0.83445072
```

Brazil has negative values for both PC1 and PC2, suggesting the country has higher rates of HIV and unemployment, and below-average levels of wellness measures like GDP and life-expectancy.

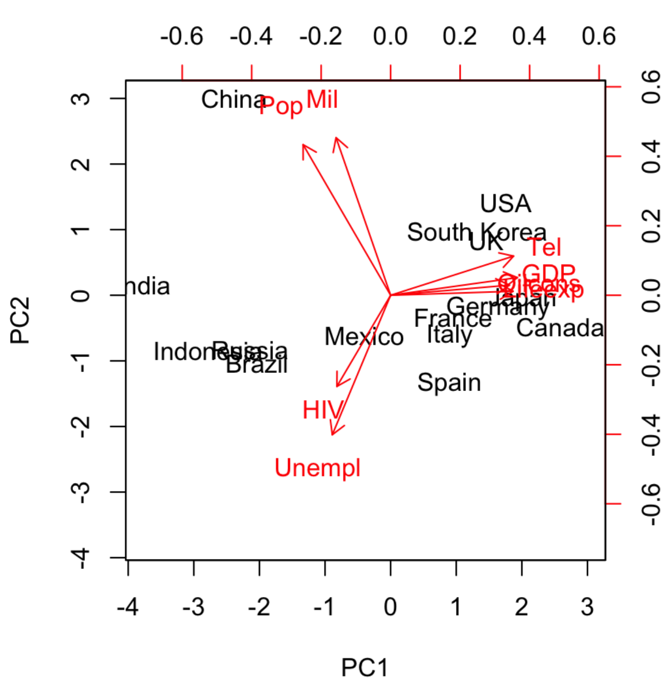
Japan has a positive value for PC1 and a slightly negative value for PC2. This suggests higher values for attributes positively associated with PC1, including GDP, Lifeexp, Oilcons, and Tel. The slightly negative value for PC2 suggests either slightly above average rates of HIV and unemployment (which PC2 is negatively correlated with and heavily influenced by) or slightly above average levels of population and military spending (which PC2 is positively correlated with and heavily influenced by).

The UK has positive values for both PC1 and PC2. It certainly belongs in PC1, which is comprised of first-world countries. Again, the opposing effects of HIV/unemployment and Pop/Mil on PC2 make it difficult to assess whether its slightly positive score is due to high population, high military spending, or low HIV levels and low levels of unemployment. I would guess the latter.

Overall, based on my knowledge of these countries, these categorizations make sense.

## Part 2.2

```
#pca$rotation <- -pca$rotation
biplot(pca, scale=0)
```




## Part 2.2

```
ratings <- read.csv('ratings.csv')
head(ratings)
```

```
##   userId movieId rating  timestamp
## 1      1      31    2.5 1260759144
## 2      1    1029    3.0 1260759179
## 3      1    1061    3.0 1260759182
## 4      1    1129    2.0 1260759185
## 5      1    1172    4.0 1260759205
## 6      1    1263    2.0 1260759151
```

```
movies <- read.csv('movies.csv')
head(movies)
```

```
##   movieId                              title
## 1       1                   Toy Story (1995)
## 2       2                     Jumanji (1995)
## 3       3            Grumpier Old Men (1995)
## 4       4           Waiting to Exhale (1995)
## 5       5 Father of the Bride Part II (1995)
## 6       6                        Heat (1995)
##                                        genres
## 1 Adventure|Animation|Children|Comedy|Fantasy
## 2                  Adventure|Children|Fantasy
## 3                               Comedy|Romance
## 4                         Comedy|Drama|Romance
## 5                                       Comedy
## 6                        Action|Crime|Thriller
```

```
genres <-  1:20
names(genres) <- c("Action", "Adventure", "Animation", "Children", "Comedy", "Crime", "Documentary", "Drama", "Fa
ntasy",
"Film-Noir", "Horror", "IMAX", "Musical", "Mystery", "Romance", "Sci-Fi", "Thriller", "War", "Western", "(no genr
es listed)")
```

```
# get user profile matrix
user_profile <- function(A_number) {
  user_ID <- A_number%%671
  user_subset <- subset(ratings, userId == user_ID)
  # get number of movies watched to create dimensions of dataset
  movies_watched <- unique(user_subset$movieId)
  # create empty user profile to fill out
  rows = length(movies_watched)
  empty_profile <- data.frame(matrix(NA, nrow = rows, ncol = 20), row.names = movies_watched)
  names(empty_profile) <- names(genres)
  # find information about movies watched
  count = 1
  for (mid in movies_watched) {
    genre_entry = subset(movies, movieId == mid)$genres
    for(g in names(genres)) {
      genre_idx <- genres[[g]]
      if (grepl(tolower(g), tolower(genre_entry), fixed = TRUE)) {
        empty_profile[count, genre_idx] <- 1
      } else {
        empty_profile[count, genre_idx] <- 0
      }
      #genres_in_movie <- unique(genres_in_movie)
    }
    count = count + 1
  }
  return(empty_profile)
}

# Get user profile vector
user_profile_vector <- function(profile_df) {
  vector_values <- unname(colMeans(profile_df))
  return(vector_values)
}
```

```r
# get movie profile vector
movie_profile <- function(movie_id) {
  genre_entry = subset(movies, movieId == movie_id)$genres
  genre_vector <- c()
  for(g in names(genres)) {
    genre_idx <- genres[[g]]
    if (grepl(tolower(g), tolower(genre_entry), fixed = TRUE)) {
      genre_vector <- append(genre_vector, 1)
    } else {
      genre_vector <- append(genre_vector, 0)
    }
  }
  return(genre_vector)
}
```

```r
my_profile <- user_profile(20452471)
my_vector <- user_profile_vector(my_profile)
```

```r
top_five_from_ten <- function(user_ID, movie_sample) {
  ID <- user_ID%%671
  my_profile <- user_profile(ID)
  my_vector <- user_profile_vector(my_profile)

  movie_profiles <- c()
  list_idx <- 0
  for (m in movie_sample) {
    profile <- movie_profile(m)
    movie_profiles <- append(movie_profiles, list(profile))
  }
  mid_list <- c()
  name_list <- c()
  similarity_list <- c()
  for (i in 1:length(movie_sample)) {
    mid_list <- append(mid_list, movie_sample[i])
    name_list <- append(name_list, as.character(movies$title[movies$movieId == movie_sample[i]]))
    similarity_list <- append(similarity_list, as.numeric(cosine(movie_profiles[[i]], my_vector)))
  }

  indices <- sort(similarity_list, index.return=TRUE, decreasing=TRUE)[[2]][1:5]
  top_5_mid <- c()
  top_5_names <- c()
  top_5_similarity <- c()
  for (i in indices) {
    top_5_mid <- append(top_5_mid, mid_list[i])
    top_5_names <- append(top_5_names, name_list[i])
    top_5_similarity <- append(top_5_similarity, similarity_list[i])
  }
  column_names <- c("MovieId", "MovieName", "Similarity")
  recs <- data.frame(top_5_mid, top_5_names, top_5_similarity)
  names(recs) <- column_names

  cat(paste0("User ID ", user_ID, " chose the following 10 movies: ", movie_sample[1], ", ", movie_sample[2], ",
 ",
             movie_sample[3], ", ", movie_sample[4], ", ", movie_sample[5], ", ",movie_sample[6], ", ",
             movie_sample[7], ", ", movie_sample[8], ", ", movie_sample[9], ", ",movie_sample[10],
             ". Of these, the following 5 movies are recommended: "))
  return(recs)
}
```

```r
unique_movies <- movies$movieId
movie_sample <- sample(unique_movies, 10)
user_ID <- 20452471
top_five_from_ten(user_ID, movie_sample)
```

```
## User ID 20452471 chose the following 10 movies: 63540, 39416, 5523, 63393, 65552, 897, 8129, 66686, 1020, 487.
Of these, the following 5 movies are recommended:
```

```
##   MovieId                           MovieName Similarity
## 1   39416               Kids in America (2005)  0.7891827
## 2     897       For Whom the Bell Tolls (1943)  0.6825569
## 3   66686              Unsuspected, The (1947)  0.6283889
## 4   63393                     Camp Rock (2008)  0.3248451
## 5    5523 Adventures of Pluto Nash, The (2002)  0.2628767
```