ML homework 4

# Part I: Implementing a Simple Prediction Pipeline

### Preprocessing

#### Tidying

Here we read-in the data, convert categorical variables to factors and remove missing data

nycpat <-  
 read\_csv("data/class4\_p1.csv") %>%   
 mutate(  
 chronic1 = factor(chronic1),  
 chronic3 = factor(chronic3),  
 chronic4 = factor(chronic4),  
 tobacco1 = factor(tobacco1),  
 alcohol1 = factor(alcohol1),  
 habits5 = factor(habits5),  
 habits7 = factor(habits7),  
 agegroup = factor(agegroup),  
 dem3 = factor(dem3),  
 dem4 = factor(dem4),  
 dem8 = factor(dem8),  
 povertygroup = factor(povertygroup)  
 ) %>%   
 select(- ...1) %>%   
 drop\_na()

## New names:  
## Rows: 3811 Columns: 17  
## ── Column specification  
## ──────────────────────────────────────────────────────── Delimiter: "," dbl  
## (17): ...1, chronic1, chronic3, chronic4, bmi, tobacco1, alcohol1, gpaq8...  
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ  
## Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
## • `` -> `...1`

The New York City Department of Health general health and physical activity has 2195 observations and 16 features. The variables chronic1, chronic3, chronic4, tobaco1, alcohol1, habits5, habits7, agegroup, dem3, dem4, dem8, povertygroup, and healthydaysare categorical variables but were read-in as continuous. While the variables bmi, gpaq11days, gpaq8totmin, and healthydays are continuous and were read-in correctly.

#### Finding correlations

Then we examine the data for feature with high correlations. No high correlations was found.

cor\_pat <-  
 nycpat %>%   
 select(where(is.numeric)) %>%   
 cor(use = "complete.obs") %>%   
 findCorrelation(cutoff=0.4)

#### Creating balanced partitions in the data

Next we partition data into training and testing (use a 70/30 split)

set.seed(123)  
train.index<-createDataPartition(nycpat$healthydays, p=0.7, list=FALSE)  
  
pat\_train<-nycpat[train.index,]  
pat\_test<-nycpat[-train.index,]

#### Building and fitting the model

Here we build two linear regression models using the training data.

In model 1 we use: \* chronic4: asthma status \* bmi: body mass index \* alcohol1: alcohol use status \* agegroup: age group \* dem3: sex \* dem4: Hispanic or Latino \* povertygroup: poverty status by household income

In model 2 we use: \* chronic1: hypertension status \* chronic3: diabetes status \* bmi: body mass index \* alcohol1: alcohol use status \* gpaq8totmin: minutes/day doing activitiy \* gpaq11days days active/week \* agegroup: age group \* habits5: General activity status \* habits7: General healthy eating status \* dem3: sex \* dem8: birthplace in US or outside \* povertygroup: poverty status by household income

set.seed(123)  
  
regresscontrol <- trainControl(method ="repeatedcv", number = 10, repeats = 5)  
   
lm1 <- train(healthydays ~ chronic4 + bmi + alcohol1 + agegroup + dem3 + dem4 + povertygroup, data = pat\_train, method = "lm", preProc=c("center", "scale"), trControl = regresscontrol)  
  
lm2 <- train(healthydays ~ chronic1 + chronic3 + bmi + alcohol1 + gpaq8totmin + gpaq11days + agegroup + habits5 + habits7+ dem3 + dem8 + povertygroup, data = pat\_train, method = "lm", preProc=c("center", "scale"), trControl = regresscontrol)

#### evaluating performance

We evaluate the two models performance using testing data based on the root mean square error (RMSE) for each as an evaluation metric.

set.seed(123)  
  
predictions1 <- predict(lm1, pat\_test)  
RMSE(predictions1, pat\_test$healthydays)

## [1] 7.330525

predictions2 <- predict(lm2, pat\_test)  
RMSE(predictions2, pat\_test$healthydays)

## [1] 7.233054

With a lower (RMSE = 7.23) model 2 performs slightly better compared to model 1 (RMSE = 7.33). This model could be used to evaluate where it may be best to allocate some recreational and wellness resources like funding certain wellness programs on community levels as well establishing preventive health programs among New York City communities.

## Part II: Conducting an Unsupervised Analysis

Using the dataset from the Group assignment Part 3 (USArrests), identify clusters using hierarchical analysis. Use an agglomerative algorithm for hierarchical clustering. Use a Euclidian distance measure to construct your dissimilarity matrix.

### Preprocessing

#### Tidying

data("USArrests")  
  
usarrests <-  
 USArrests %>%   
 as\_tibble(USArrests) %>%   
 janitor::clean\_names() %>%   
 drop\_na()   
  
# checking if scaling is necessary  
colMeans(usarrests)

## murder assault urban\_pop rape   
## 7.788 170.760 65.540 21.232

apply(usarrests, 2, sd)

## murder assault urban\_pop rape   
## 4.355510 83.337661 14.474763 9.366385

#scale data  
arrestsdf <- scale(usarrests)

### Conduct Principle Component Analysis

PC1 explains 62% of variance, PC2 explains 24% of variance. Together they explain 87% of variance

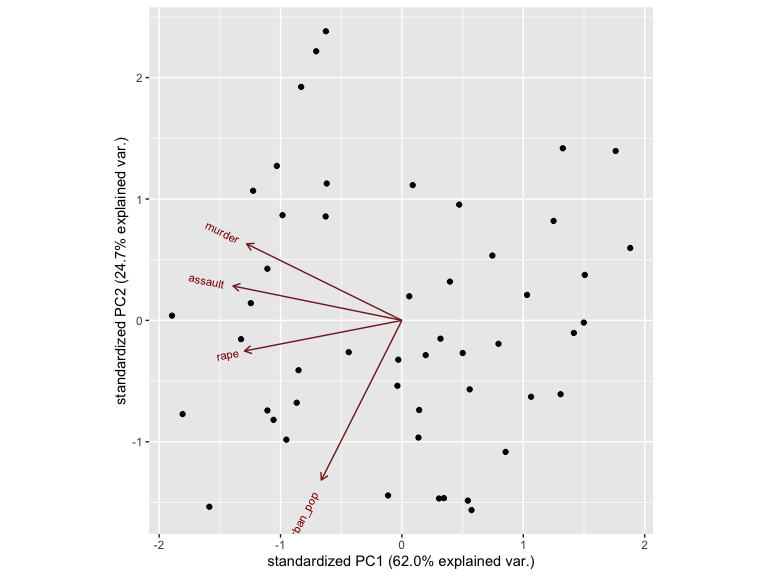
set.seed(123)  
  
usarrests.pca <- prcomp( ~., data = usarrests, center = TRUE, scale = TRUE)  
  
#view results of pca  
summary(usarrests.pca)

## Importance of components:  
## PC1 PC2 PC3 PC4  
## Standard deviation 1.5749 0.9949 0.59713 0.41645  
## Proportion of Variance 0.6201 0.2474 0.08914 0.04336  
## Cumulative Proportion 0.6201 0.8675 0.95664 1.00000

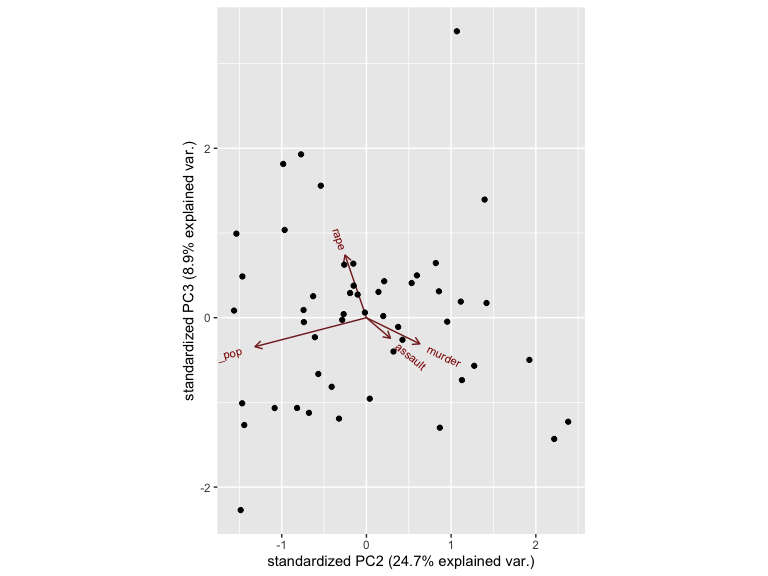
#Identify how features loaded on the different components  
usarrests.pca$rotation

## PC1 PC2 PC3 PC4  
## murder -0.5358995 0.4181809 -0.3412327 0.64922780  
## assault -0.5831836 0.1879856 -0.2681484 -0.74340748  
## urban\_pop -0.2781909 -0.8728062 -0.3780158 0.13387773  
## rape -0.5434321 -0.1673186 0.8177779 0.08902432

ggbiplot(usarrests.pca)



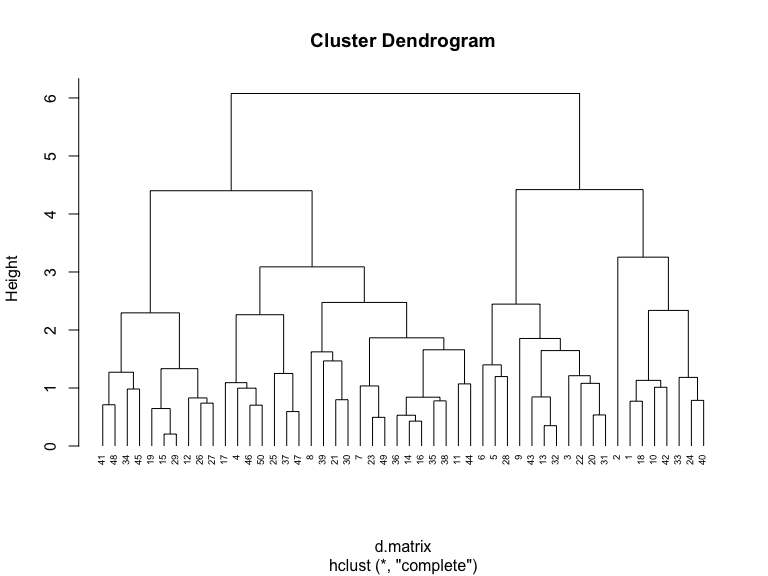
ggbiplot(usarrests.pca, choices=c(2,3))



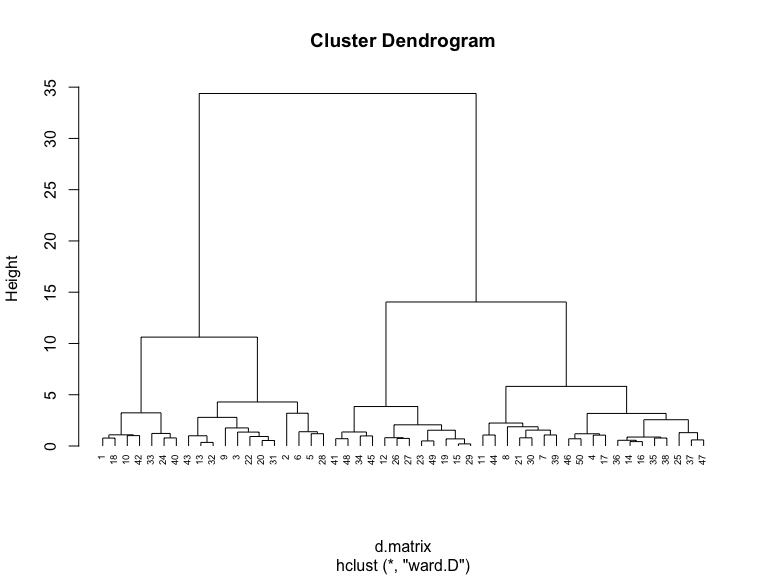
### Hierarchical clustering analysis

Here we conduct a hierarchical analysis using Euclidian distance measure to construct your dissimilarity matrix and multiple both Complete and Ward methods of the agglomerative algorithm to obtain the optimal number of clusters.

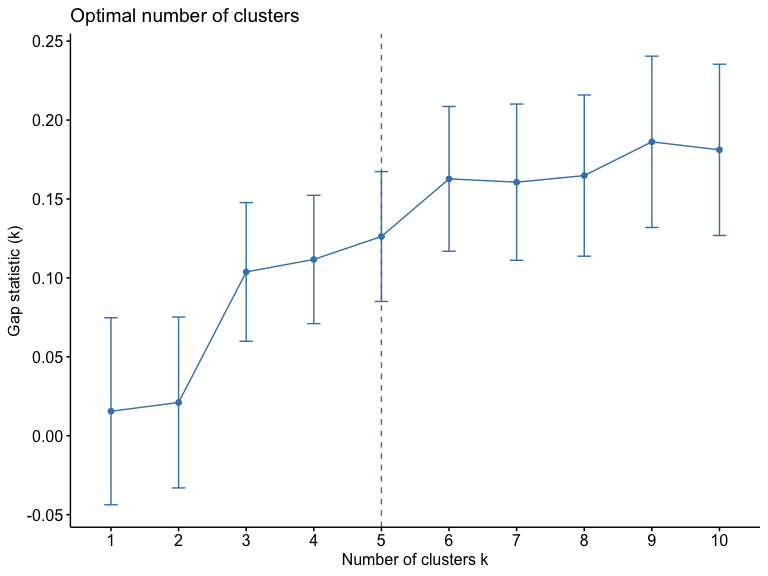
set.seed(123)  
# Create Dissimilarity matrix  
d.matrix <- dist(arrestsdf, method = "euclidean")  
  
# Hierarchical clustering using Complete Linkage  
ch1<- hclust(d.matrix, method = "complete" )  
  
# Plot the obtained dendrogram  
plot(ch1, cex = 0.6, hang = -1)



# Hierarchical clustering using Ward Linkage  
ch2<- hclust(d.matrix, method = "ward.D" )  
  
# Plot the obtained dendrogram   
plot(ch2, cex = 0.6, hang = -1)



gap\_stat <- clusGap(usarrests, FUN = hcut, nstart = 25, K.max = 10, B = 50)  
fviz\_gap\_stat(gap\_stat)



The optimal number of clusters is 3 as based on plotting the gap statistic. However, based on the cluster dendogram (Ward linkage), the optimal number of clusters is 4 if the cut off line is above 5 and below 10 on the height axis. This is more reasonable as the group within each of the 4 clusters appear to share more similarities. We expect that states with higher urban populations are likely to experience higher crime rates (murder, assault, rape)

One research question that can be addressed using the newly identified clusters is: Can falling in a certain cluster of crime prevalence in 2020 be used to predict future urban population growth in US states?

Ethical consideration to consider before using this data to answer the research question is how representative is the data of the US population and what implications could it lead to in terms of safety, economic growth, state-funded social and economic programs, policing laws, real-state prices and other socioeconomic factors.