P8451\_HW4\_ML

Ruixi Li

2024-02-07

# Part I: Implementing a Simple Prediction Pipeline

The New York City Department of Health administered a questionnaire on general health and physical activity among residents. Using the dataset class4\_p1.csv, fit and evaluate two prediction models using **linear regression**. The aim of the models are to predict the number of days in a month an individual reported having good physical health (feature name: **healthydays**). A codebook is provided so you can look-up the meaning and values of each feature in the dataset. (Note the codebook lists information on features that are not included in your dataset).

Your analytic pipeline should include the following:

## 1. Data Cleaning

library(ggbiplot)  
library(caret)  
library(gtsummary)

library(tidyverse)  
# Read data  
hw4data = read\_csv("class4\_p1.csv")  
set.seed(123)  
  
# View the datatypes and summary statistics of the whole dataset  
skimr::skim(hw4data)

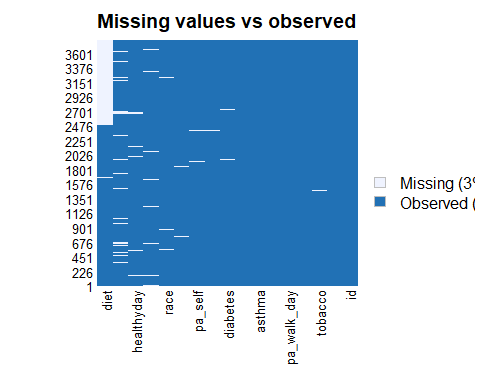
Data summary

|  |  |
| --- | --- |
| Name | hw4data |
| Number of rows | 3811 |
| Number of columns | 17 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| numeric | 17 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

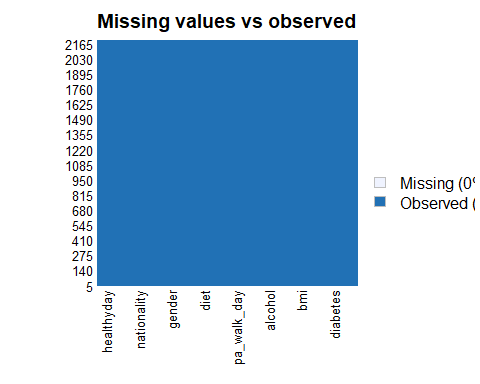
**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| …1 | 0 | 1.00 | 1906.00 | 1100.29 | 1.00 | 953.50 | 1906.00 | 2858.50 | 3811.00 | ▇▇▇▇▇ |
| chronic1 | 28 | 0.99 | 1.71 | 0.45 | 1.00 | 1.00 | 2.00 | 2.00 | 2.00 | ▃▁▁▁▇ |
| chronic3 | 17 | 1.00 | 1.89 | 0.32 | 1.00 | 2.00 | 2.00 | 2.00 | 2.00 | ▁▁▁▁▇ |
| chronic4 | 8 | 1.00 | 1.94 | 0.24 | 1.00 | 2.00 | 2.00 | 2.00 | 2.00 | ▁▁▁▁▇ |
| bmi | 73 | 0.98 | 27.32 | 6.27 | 9.09 | 23.05 | 26.47 | 30.23 | 99.83 | ▇▆▁▁▁ |
| tobacco1 | 3 | 1.00 | 2.74 | 0.64 | 1.00 | 3.00 | 3.00 | 3.00 | 3.00 | ▁▁▁▁▇ |
| alcohol1 | 17 | 1.00 | 2.64 | 0.54 | 1.00 | 2.00 | 3.00 | 3.00 | 3.00 | ▁▁▃▁▇ |
| gpaq8totmin | 5 | 1.00 | 43.77 | 89.60 | 0.00 | 0.00 | 0.00 | 60.00 | 960.00 | ▇▁▁▁▁ |
| gpaq11days | 5 | 1.00 | 4.27 | 2.68 | 0.00 | 2.00 | 5.00 | 7.00 | 7.00 | ▅▂▃▃▇ |
| habits5 | 17 | 1.00 | 1.97 | 0.83 | 1.00 | 1.00 | 2.00 | 2.00 | 4.00 | ▅▇▁▃▁ |
| habits7 | 1335 | 0.65 | 2.73 | 1.05 | 1.00 | 2.00 | 3.00 | 3.00 | 5.00 | ▃▆▇▅▁ |
| agegroup | 8 | 1.00 | 2.78 | 0.89 | 1.00 | 2.00 | 3.00 | 3.00 | 4.00 | ▂▆▁▇▅ |
| dem3 | 0 | 1.00 | 1.59 | 0.49 | 1.00 | 1.00 | 2.00 | 2.00 | 2.00 | ▆▁▁▁▇ |
| dem4 | 30 | 0.99 | 1.78 | 0.42 | 1.00 | 2.00 | 2.00 | 2.00 | 2.00 | ▂▁▁▁▇ |
| dem8 | 6 | 1.00 | 1.39 | 0.49 | 1.00 | 1.00 | 1.00 | 2.00 | 2.00 | ▇▁▁▁▅ |
| povertygroup | 244 | 0.94 | 3.30 | 1.64 | 1.00 | 2.00 | 3.00 | 5.00 | 6.00 | ▇▃▃▅▂ |
| healthydays | 88 | 0.98 | 26.34 | 7.89 | 0.00 | 27.00 | 30.00 | 30.00 | 30.00 | ▁▁▁▁▇ |

# Rename and datatype conversion  
var.names = c("id", "hypertension", "diabetes", "asthma", "bmi", "tobacco", "alcohol", "pa\_chores\_min", "pa\_walk\_day", "pa\_self", "diet", "agegrp", "gender", "race","nationality", "fincome", "healthyday")  
  
hw4data = hw4data |>  
 set\_names(var.names) |>  
 mutate(across(-c(5,8,9,17), function(x) as.factor(x)))  
  
hw4data |> Amelia::missmap(main = "Missing values vs observed")



hw4data = hw4data|>   
 select(-id) |>   
 drop\_na()  
  
hw4data |> Amelia::missmap(main = "Missing values vs observed")



# it's necessary to make sure that directly drop NAs won't cause bias

## 2. Data partition

# Creating balanced partitions in the data  
train.index = createDataPartition(hw4data$healthyday, p = 0.7, list = FALSE)  
  
train = hw4data[train.index, ]  
test = hw4data[-train.index, ]  
  
# I should check the independence of each variable

## 3. Fit two models

There are lots of methods to fit my models. I have thought about using lasso regression to do the auto-selection of predictors and handle multicollinearity. But lasso regression doesn’t work well with categorical variables in terms of both feature selection and prediction accuracy. It needs One-hot encoding of the categorical data and ignore the grouping effect. However, group lasso may be a good alternative for robust feature selection although at the cost of prediction accuracy. (Lack of Robustness of Lasso and Group Lasso with Categorical Predictors: Impact of Coding Strategy on Variable Selection and Prediction <https://escholarship.org/content/qt40b200z6/qt40b200z6_noSplash_c3819f1c49cdc6380c6ae5b0ac0af41d.pdf?t=qaj586>)

Here, I would build the two models based on my knowledge. I built my model2 from scratch manually. But actually, model builded using the whole dataset might lead to increased risk of overfitting of the trained model.

# Conducting the bivariate analysis for model building  
  
outcome <- "healthyday"  
predictors <- c("hypertension", "diabetes", "asthma", "bmi", "tobacco", "alcohol", "pa\_chores\_min", "pa\_walk\_day", "pa\_self", "diet", "agegrp", "gender", "race", "nationality", "fincome")  
  
bivariate\_results <- list()  
  
for (predictor in predictors) {  
 formula <- as.formula(paste(outcome, "~", predictor))  
 model <- lm(formula, data=hw4data)   
 bivariate\_results[[predictor]] <- summary(model)$coefficients  
}  
  
  
# Start with an intercept-only model  
current\_model <- lm(healthyday ~ 1, data=hw4data)   
  
# Iteratively add variables based on LRT  
for (predictor in predictors) {  
 new\_model\_formula <- update(formula(current\_model), paste(". ~ . +", predictor))  
 new\_model <- lm(new\_model\_formula, data=hw4data)   
   
 # Perform Likelihood Ratio Test  
 lr\_test <- anova(current\_model, new\_model)  
   
 # Extract the last p-value (corresponding to the addition of the new predictor)  
 p\_value <- lr\_test$`Pr(>F)`[nrow(lr\_test)]  
   
 # Check if the p-value exists and is significant  
 if (!is.na(p\_value) && p\_value < 0.05) {  
 current\_model <- new\_model # Update the model to include the new predictor  
 cat("Added", predictor, "to the model\n")  
 } else {  
 cat(predictor, "not added\n")  
 }  
}

## Added hypertension to the model  
## Added diabetes to the model  
## Added asthma to the model  
## bmi not added  
## tobacco not added  
## Added alcohol to the model  
## pa\_chores\_min not added  
## Added pa\_walk\_day to the model  
## Added pa\_self to the model  
## Added diet to the model  
## Added agegrp to the model  
## gender not added  
## Added race to the model  
## nationality not added  
## Added fincome to the model

# Final model after adding all significant predictors  
 model1 = lm(healthyday ~ hypertension + pa\_walk\_day, data = hw4data)  
 model2 = current\_model

# Avoid overfitting through cross validation  
control = trainControl(method="repeatedcv", number=10, repeats=10, summaryFunction=defaultSummary)  
  
# Train models  
model1\_train = train(healthyday ~ hypertension + pa\_walk\_day, data = train, method = "lm", trControl = control)  
model2\_train = train(healthyday ~ hypertension + diabetes + asthma + alcohol + pa\_walk\_day + pa\_self + diet + agegrp + race + fincome, data = train, method = "lm", trControl = control)  
  
model1\_train$finalModel |> tbl\_regression()

## Table printed with {flextable}, not {gt}. Learn why at  
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

| **Characteristic** | **Beta** | **95% CI**1 | **p-value** |
| --- | --- | --- | --- |
| hypertension2 | 3.7 | 2.9, 4.5 | <0.001 |
| pa\_walk\_day | 0.24 | 0.10, 0.39 | <0.001 |
| 1CI = Confidence Interval | | | |

model2\_train$finalModel |> tbl\_regression()

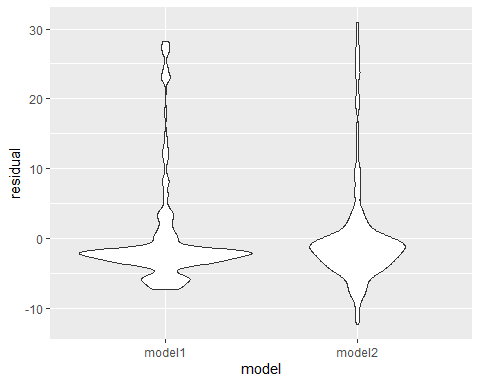
## Table printed with {flextable}, not {gt}. Learn why at  
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

| **Characteristic** | **Beta** | **95% CI**1 | **p-value** |
| --- | --- | --- | --- |
| hypertension2 | 2.1 | 1.2, 3.0 | <0.001 |
| diabetes2 | 0.21 | -0.99, 1.4 | 0.7 |
| asthma2 | 3.2 | 1.7, 4.7 | <0.001 |
| alcohol2 | -1.3 | -3.5, 0.81 | 0.2 |
| alcohol3 | -1.8 | -3.9, 0.24 | 0.083 |
| pa\_walk\_day | 0.11 | -0.04, 0.25 | 0.14 |
| pa\_self2 | 0.12 | -0.75, 0.98 | 0.8 |
| pa\_self3 | -1.5 | -2.6, -0.37 | 0.010 |
| pa\_self4 | -3.4 | -5.3, -1.5 | <0.001 |
| diet2 | -0.38 | -1.6, 0.84 | 0.5 |
| diet3 | -0.93 | -2.1, 0.26 | 0.13 |
| diet4 | -2.8 | -4.2, -1.4 | <0.001 |
| diet5 | -5.2 | -7.3, -3.2 | <0.001 |
| agegrp2 | -1.1 | -2.7, 0.36 | 0.14 |
| agegrp3 | -2.5 | -4.0, -0.95 | 0.001 |
| agegrp4 | -3.6 | -5.2, -1.9 | <0.001 |
| race2 | 1.1 | 0.22, 2.1 | 0.015 |
| fincome2 | 0.09 | -1.1, 1.3 | 0.9 |
| fincome3 | 2.0 | 0.74, 3.2 | 0.002 |
| fincome4 | 1.4 | 0.12, 2.7 | 0.033 |
| fincome5 | 1.5 | 0.27, 2.7 | 0.017 |
| fincome6 | 0.18 | -1.4, 1.7 | 0.8 |
| 1CI = Confidence Interval | | | |

* model1: healthyday ~ hypertension + pa\_walk\_day
* model2: healthyday ~ hypertension + diabetes + asthma + alcohol + pa\_walk\_day + pa\_self + diet + agegrp + race + fincome

## 4. Model Evaluation

# Predictions for models  
predictions1 = predict(model1\_train, test)  
predictions2 = predict(model2\_train, test)  
  
# Evaluation using Mean Squared Error (MSE)  
mse1 = mean((predictions1 - test$healthyday)\*\*2/nrow(test))   
mse2 = mean((predictions2 - test$healthyday)\*\*2/nrow(test))   
  
model1\_result = bind\_rows(train = predictions1, test = test$healthyday, .id = "split")  
model2\_result = bind\_rows(train = predictions2, test = test$healthyday, .id = "split")  
result = bind\_rows(model1 = model1\_result, model2 = model2\_result, .id = "model")  
  
  
result = result |>   
 mutate(residual = train - test)  
  
result |>   
 ggplot(aes(x = model, y = residual)) + geom\_violin()



* Mean squared error (MSE) for model1(0.0853517)< model2(0.0781096), and the residual plot shows that both models have residuals that center around zero, which is good. The spread of the residuals (the range from top to bottom of the violin) seems similar for both models, suggesting that they have a similar variance in their predictions. However, it looks like Model 2 might have a slightly tighter distribution, indicating more consistent predictions. So, model2 is the preferred prediction model using MSE evaluation.

## 5. Implementing setting

My final model: healthyday ~ hypertension + diabetes + asthma + alcohol + pa\_walk\_day + pa\_self + diet + agegrp + race + fincome can be used to predict the number of days in a month an individual reported having good physical health through a comprehensive evaluation of medical conditions, life habits and socioeconomic factors. This insight can guide targeted health interventions and resource allocation to improve the general well-being of the population in NYC.

# Part II: Conducting an Unsupervised Analysis

Using the dataset from the Group assignment Part 3 (USArrests), identify clusters using hierarchical analysis.

## 6. Hierarchical clustering analysis

library(factoextra)  
library(cluster)  
  
# Load data  
data("USArrests")  
  
# look at the structure and summary  
skimr::skim(USArrests)

Data summary

|  |  |
| --- | --- |
| Name | USArrests |
| Number of rows | 50 |
| Number of columns | 4 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| numeric | 4 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Murder | 0 | 1 | 7.79 | 4.36 | 0.8 | 4.08 | 7.25 | 11.25 | 17.4 | ▇▇▅▅▃ |
| Assault | 0 | 1 | 170.76 | 83.34 | 45.0 | 109.00 | 159.00 | 249.00 | 337.0 | ▆▇▃▅▃ |
| UrbanPop | 0 | 1 | 65.54 | 14.47 | 32.0 | 54.50 | 66.00 | 77.75 | 91.0 | ▁▆▇▅▆ |
| Rape | 0 | 1 | 21.23 | 9.37 | 7.3 | 15.08 | 20.10 | 26.17 | 46.0 | ▆▇▅▂▂ |

# Check means and SDs to determine if scaling is necessary  
USArrests |>  
 summarise\_all(mean, na.rm = TRUE) |>  
 print()

## Murder Assault UrbanPop Rape  
## 1 7.788 170.76 65.54 21.232

USArrests |>  
 summarise\_all(sd, na.rm = TRUE) |>  
 print()

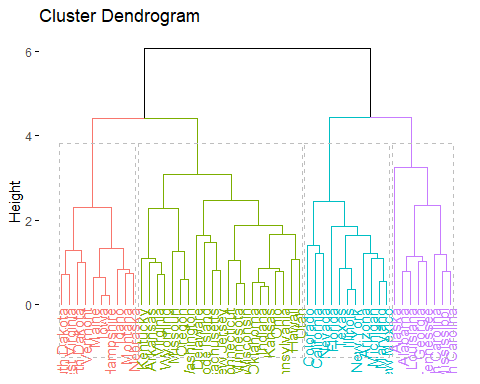
## Murder Assault UrbanPop Rape  
## 1 4.35551 83.33766 14.47476 9.366385

# need standardization  
  
# Centering and Scaling  
set.up.preprocess = preProcess(USArrests, method = c("center", "scale"))  
  
# Output pre-processed values  
transformed.vals = predict(set.up.preprocess, USArrests)  
  
  
  
  
set.seed(123)  
  
# Hierarchical clustering using Complete Linkage,   
clusters.hcut=hcut(transformed.vals, k=4, hc\_func="hclust", hc\_method="complete", hc\_metric="euclidian")# k=2 or k=4  
  
clusters.hcut$size

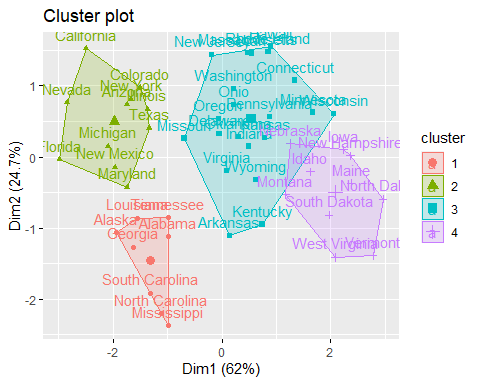
## [1] 8 11 21 10

# Plot the obtained dendrogram  
fviz\_dend(clusters.hcut, rect=TRUE)

## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as  
## of ggplot2 3.3.4.  
## ℹ The deprecated feature was likely used in the factoextra package.  
## Please report the issue at <https://github.com/kassambara/factoextra/issues>.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

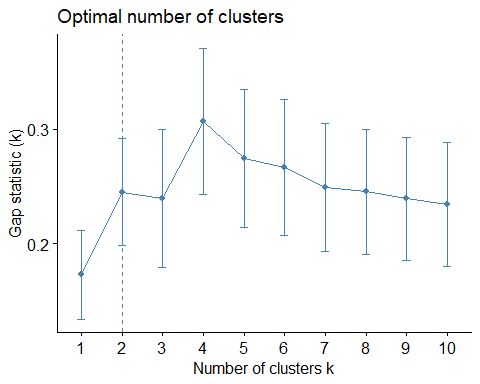


fviz\_cluster(clusters.hcut)



## 7. Determine the optimal number of clusters

# Plot gap statistic graph  
gap\_stat = clusGap(transformed.vals, FUN = hcut, hc\_method="complete", K.max = 10, B = 50)  
fviz\_gap\_stat(gap\_stat)



* Although the graph chose the optimal cluster number as 2, the gap statistics increases when moving k=3 to k=4, and constantly decreases when k becomes bigger,despite a slight decrease, indicating that k=4 is the optimal number.

## 8. Describe the composition of each cluster

input.feature.vals=cbind(transformed.vals,cluster=clusters.hcut$cluster)  
  
input.feature.vals |>  
 group\_by(cluster) |>  
 summarise\_all(mean) |>  
 knitr::kable()

| cluster | Murder | Assault | UrbanPop | Rape |
| --- | --- | --- | --- | --- |
| 1 | 1.4463290 | 0.9838289 | -0.8317925 | 0.3529110 |
| 2 | 0.7499801 | 1.1199128 | 0.9361748 | 1.2156432 |
| 3 | -0.4400338 | -0.4353831 | 0.3607592 | -0.2830385 |
| 4 | -1.0579703 | -1.1046626 | -1.1219527 | -1.0251554 |

* Cluster 1: This cluster is marked by above-average values in all categories except UrbanPop, which is close to average (z-score of -0.832). It suggests states with higher crime rates and moderately smaller urban populations.
* Cluster 2: States in this cluster have all features above the average,especially Assault and Rape rates. This cluster might represent states with high crime rates and large urban populations.
* Cluster 3: The negative z-scores for Murder, Assault, and Rape indicate that these areas have lower than average rates of these crimes. The UrbanPop is slightly above average, meaning these areas might have moderately sized urban populations but high crime rates.
* Cluster 4: This cluster is characterized by below-average values for all features (Murder, Assault, UrbanPop, Rape), as indicated by the negative z-scores. This suggests that states falling into this cluster have lower than average crime rates and urban population.

## 9. Research Questions

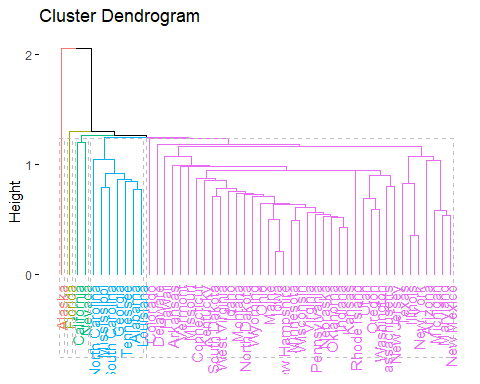
* Research question: Is the level of utilization(represented by ‘UrbanPop’ feature in this dataset) associated with the crime rate(measured by ‘Murder’, ‘Assualt’ and ‘Rape’ features in this dataset) across state?
* Scientific consideration: We should figure out which linkage method to use and how many clusters to have based on the nature of data.
* Ethical consideration:We should avoid drawing conclusions that could stigmatize certain regions or communities.

## 10. Repeat analysis with different parameters

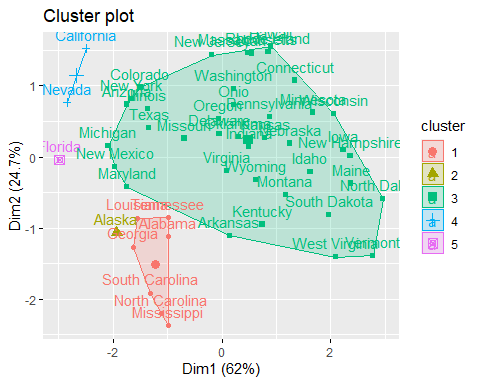
clusters.hcut<-hcut(transformed.vals, k=5, hc\_func="hclust", hc\_method="single", hc\_metric="euclidian")  
  
clusters.hcut$size

## [1] 7 1 39 2 1

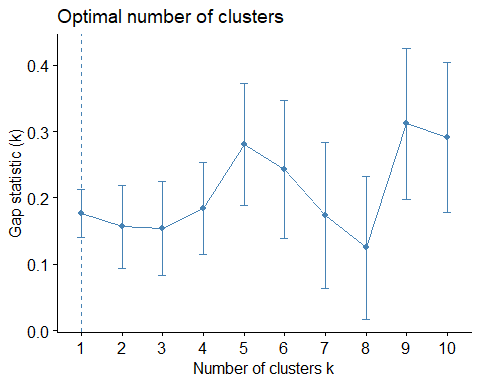
fviz\_dend(clusters.hcut, rect=TRUE)



fviz\_cluster(clusters.hcut)



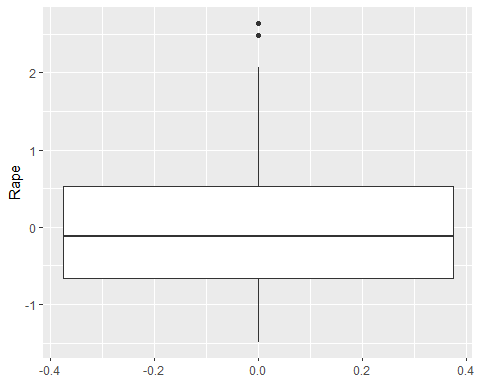
gap\_stat <- clusGap(transformed.vals, FUN = hcut, hc\_method="single", K.max = 10, B = 50)  
fviz\_gap\_stat(gap\_stat)



input.feature.vals<-cbind(transformed.vals,cluster=clusters.hcut$cluster)  
  
input.feature.vals %>%  
 group\_by(cluster) %>%  
 summarise\_all(mean)

## # A tibble: 5 × 5  
## cluster Murder Assault UrbanPop Rape  
## <int> <dbl> <dbl> <dbl> <dbl>  
## 1 1 1.58 0.966 -0.778 0.0484  
## 2 2 0.508 1.11 -1.21 2.48   
## 3 3 -0.375 -0.310 0.0725 -0.222   
## 4 4 0.646 1.12 1.41 2.36   
## 5 5 1.75 1.97 0.999 1.14

transformed.vals|>  
 ggplot(aes(y=Rape))+ geom\_boxplot()



* Yes, the clusters changed. For my research question, I want to find tightly knit clusters and the data contains outliers(in Rape), so compared with ‘single’ I think ‘complete’ is better(using single linkage, some cluster has only one state). The gap statistic graph also shows an obvious ‘elbow’ at k=4(compared with using ‘single’ method). Changing set seeds didn’t change the clusters a lot,indicating that the data has a strong inherent tighter clustering structure.