4. Can Living Sustainably Bring You Happiness - R Script

```
library(haven)
library(ggplot2)
library(dplyr)
library(GPArotation)
library(stats)
library(psych)
library(factoextra)
library(corrplot)

data <- read.csv('./Documents/Dimentionality Reduction/Group Project/Happiness-Sustainable-
Behaviour.csv')
data$X <- NULL
head(data)
str(data)

#Total Null Values
sum(is.na(data))

#Data Dimensions
dim(data)

#Number of missing values in each row
NAcol <- which(colSums(is.na(data)) > 0);NAcol
sort(colSums(sapply(data[NAcol], is.na)), decreasing = TRUE)

#Removing SC_10 column because of unclear question and a lot of missing values
data$SC_10 <- NULL


#Number of missing values per row for part 1 and part 2 quiz only
sort(rowSums(is.na(data[,3:54])), decreasing = T)


#Columns 21 to 54 belongs to part2 questions
###Replacing missing values in the part2 questions with the neutral value
data[21:54] <- lapply(data[21:54], function(X) {
  X <- ifelse(is.na(X), 4, X)
  return(X)
})
```

```
#So, No NAs in part 2 questions
sum(is.na(data[21:54]))


#Replace missing values for each column in part 1 with maximum repeated values
replace_with_max_value <- function(x) {
  ux <- unique(x)
  return(ux[which.max(tabulate(match(x, ux)))])
}

getEachColumn <- function(X) {
  X <- ifelse(is.na(X), replace_with_max_value(X), X)
  return(X)
}

##Part2 values are already replaces
data[,c(3:54)] <- lapply(data[3:54], getEachColumn)

#No missing values for part 1 and part 2
sum(is.na(data[,c(3:54)]))

#Replacing the missing values with 0 becaue those homes don't have Hybrid car
#4 value is out of range, will replace that with 0 as well because most of the homes don't have Hybrid car
#table(data$III.9.8)
#data$III.9.8 <- ifelse(is.na(data$III.9.8), 0, data$III.9.8)
#data$III.9.8 <- ifelse(data$III.9.8 != 1, 0, data$III.9.8)

#Replacing the NA in flights with 0 becasue NAs means people haven't taken any flight this year
#table(data$flights)
#data$flights <- ifelse(is.na(data$flights), 0, data$flights)

Not_attempted_q9 <- which(
  is.na(data$III.9.2)
  & is.na(data$III.9.3)
  & is.na(data$III.9.4)
  & is.na(data$III.9.5)
  & is.na(data$III.9.6)
  & is.na(data$III.9.1)
  & is.na(data$III.9.7)
  & is.na(data$III.9.8)
)
```

```
#Means 28 people completly skipped this questions
length(Not_attempted_q9)

#data[c(Not_attempted_q9),"income"]
#table(data$income)

#Checking out of range values
outOfRange <- lapply(data[3:54], function(X) {
  isInRange <- ifelse(!X %in% c(1:7), 'YES', 'NO')
  if ('YES' %in% isInRange) {
    return(1)
  }
  return(0)
})

#Column M05 and E04 have out of range values
names(which(outOfRange == 1))

table(data$M05);table(data$E04)

data$M05 <- ifelse(data$M05 == 4.5, 5, data$M05)

data$E04 <- ifelse(data$E04 == 6.5, 6, data$E04)


################Removing Outliers#######################

#Part 1
#Mahalanobis distance
distances <-
  mahalanobis(x = data[3:20],
              center = colMeans(data[3:20]) ,
              cov = cov(data[3:20]))
cutoff <-
  qchisq(0.999, ncol(data[3:20]))
cat("cutoff = ", cutoff)
cat("Number of outliers = ", dim(data[3:20][distances > cutoff, ])[1])

data <- data[distances < cutoff, ]
cat("Number of rows left after removing outliers = ", dim(data)[1], " ")

#Part 2
```

```
#Mahalanobis distance
distances <-
   mahalanobis(x = data[21:54],
                        center = colMeans(data[21:54]) ,
                        cov = cov(data[21:54]))
cutoff <-
   qchisq(0.999, ncol(data[21:54]))
cat("cutoff = ", cutoff)
cat("Number of outliers = ", dim(data[21:54][distances > cutoff, ])[1])

data <- data[distances < cutoff, ]
cat("Number of rows left after removing outliers = ", dim(data)[1], " ")

#Export Cleaned DataSet
write.csv(data, "./Documents/Dimentionality Reduction/Group Project/CleanedDataFile.csv",
row.names=FALSE)

###################SummaryStatistics####################
lapply(data[3:54], function(X) {
   return(mean(X))
})
#min(X); max(X); sd(X)

lapply(data[3:54], function(X) {
   v <- paste("Mean = ", mean(X),
                      "Min = ", min(X),
                      "Max = ", max(X),
                      "SD = ", sd(X))
   return(v)
})

###############################PCA######################

#PCA for part 1 quiz
pca_part1 <-
   princomp(data[3:20], cor = T, scores = T)
pca_part1

summary(pca_part1)
pca_part1$loadings
fviz_eig(pca_part1)
names(pca_part1)
```

```
pca_part1$scores
eig.val <- get_eigenvalue(pca_part1)
eig.val


#PCA for part 2 quiz
pca_part2 <-
  princomp(data[21:54], cor = T, scores = T)
pca_part2

summary(pca_part2)
pca_part2$loadings
fviz_eig(pca_part2)
pca_part2$scores
eig.val <- get_eigenvalue(pca_part2)
eig.val

######################################FA######################################
nofactors1 = fa.parallel(data[3:20], fm="ml", fa="fa")
nofactors1$fa.values#eigen values

nofactors2 = fa.parallel(data[21:54], fm="ml", fa="fa")
nofactors2$fa.values#eigen values

sum(nofactors1$fa.values > 0.7) ##new kaiser criterion
sum(nofactors2$fa.values > 0.7) ##new kaiser criterion


####FA part 1 ########
EFA.model.one <- fa(data[3:20], nfactors=2, rotate = "oblimin", fm = "ml")
fa.diagram(EFA.model.one)

EFA.model.one$scores


######FA part 2 ######
EFA.model.two <- fa(data[21:54], nfactors=3, rotate = "oblimin", fm = "ml")
fa.diagram(EFA.model.two)

efa2new <- data[, c(21:48,50:54)]
EFA.model.two.new <- fa(efa2new, nfactors=3, rotate = "oblimin", fm = "ml")
fa.diagram(EFA.model.two.new)
```

```
#Fit indices
#Comparative fix index (CFI) = 0.8934938 (<0.90, poor)
EFA.model.one
#RMSR: 0.05; <0.06 excellent
#RMSEA: 0.064; 0.06-0.08 acceptable
#NNFI/TLI: 0.868; <0.90 poor
EFA.model.one$STATISTIC
EFA.model.one$dof
EFA.model.one$null.chisq
EFA.model.one$null.dof
1 - ((279.3556-118)/(1744.852-153))
#CFI: 0.8986366; <0.90 poor

EFA.model.two.new
#RMSR: 0.04; <0.06 excellent
#RMSEA: 0.066; 0.06-0.08 acceptable
#NNFI/TLI: 0.847; <0.90 poor
EFA.model.two.new$STATISTIC
EFA.model.two.new$dof
EFA.model.two.new$null.chisq
EFA.model.two.new$null.dof
1 - ((1064.346-432)/(5601.906-528))
#CFI: 0.8753729; <0.90 poor

#Reliability
#part1
#f1 for ML1; f2 for ML2
names(data[, c(3:20)])
f1p1 = c(3:8, 11, 15:16, 18:20)
f2p1 = c(9:10, 12:14, 17)
psych::alpha(data[ , f1p1])
#raw alpha of factor 1: 0.86; >0.80 acceptable
psych::alpha(data[ , f2p1])
#raw alpha of factor 2: 0.68; <0.80 unacceptable

#part2
#efa2new <- data[, c(21:48,50:54)]
names(efa2new)
f1p2 = c(25:28, 46, 48, 51:54)
f2p2 = c(40:45, 47)
```

```
f3p2 = c(21:24, 29:39, 50)
psych::alpha(data[ , f1p2])
#raw alpha of factor 1: 0.9; >0.80 acceptable
psych::alpha(data[ , f2p2])
#raw alpha of factor 2: 0.83; >0.80 acceptable
psych::alpha(data[ , f3p2])
#raw alpha of factor 3: 0.9; >0.80 acceptable

###########Measuring factors #################

#Part 1
data$MeaningAndEngagement <- c(rowSums(data[,c("M11", "M14", "M02", "M12", "M05", "E04", "E09",
"M17", "E07", "P13", "E01", "E10")])/12)
data$Pleasure <- c(rowSums(data[,c("P15", "P03", "P18", "P16", "P08", "E06")])/6)

#Part 2
data$EnvironmentalConscious <- c(rowSums(data[, c("SC_4", "SC_13", "SC_19", "SC_18", "SC_17",
"SC_3", "SC_12", "SC_14", "SC_9", "SC_20", "SC_1", "SC_16", "SC_11", "SC_2", "SC_15",
"SC_31")])/16)
data$ThreeRs <- c(rowSums(data[,c("SC_22", "SC_26", "SC_25", "SC_21", "SC_23", "SC_28",
"SC_24")])/7)
data$EnergyConservation <- c(rowSums(data[, c("SC_33", "SC_34", "SC_35", "SC_7", "SC_6", "SC_5",
"SC_32", "SC_29", "SC_27", "SC_8")])/10)

head(data)
################Regression Analysis ################
data_reduced <- data[,c("water",
                        "MeaningAndEngagement",
                        "Pleasure",
                        "EnvironmentalConscious",
                        "ThreeRs",
                        "EnergyConservation",
                        "petrol",
                        "electricity",
                        "income",
                        "adult",
                        "home",
                        "edu",
                        "job",
                        "sex",
                        "age")]
```

```
NAcol <- which(colSums(is.na(data_reduced)) > 0);NAcol
sort(colSums(sapply(data_reduced[NAcol], is.na)), decreasing = TRUE)

#Replacing NULL values in the sex column with female, as we know most of the participants are female in
this survey
data_reduced$sex <- ifelse(is.na(data_reduced$sex), 1, data_reduced$sex)


M <- cor(data_reduced, use = "pairwise.complete.obs")
corrplot(M, method = "number", type = "upper")


#It is clear from the correlation plot that none of the demographic variables have correlation with other
#variables, which means we cannot use any of the variables from demographic data as a response variable
and cannot
#do regression analysis for part 3 on this dataset

###################################SR########################
#Part 1: Independent variable; Part 2: Dependent Variable
#Relationships between 'Orientations of Happiness' (OTH) & different categories of Sustainable Behaviors
(SBs)
#OTH: data$MeaningAndEngagement, data$Pleasure
#sb1 for data$EnvironmentalConscious; sb2 for data$ThreeRs; sb3 for data$EnergyConservation

sb1 <- lm(EnvironmentalConscious ~ MeaningAndEngagement + Pleasure, data=data); summary(sb1)
par(mfrow = c(2, 2)); plot(sb1)
#Normality, linearity, homogeneity, and homoscedasticity check
library(lmtest); bptest(sb1)
#Further homoskedasticity check
#leverage
k1 = 2 ##number of IVs in the sb1
leveragesb1 = hatvalues(sb1)
cutleveragesb1 = (2*k1+2) / nrow(data); cutleveragesb1 ##cut off = 0.01775148
badleveragesb1 = as.numeric(leveragesb1 > cutleveragesb1)
table(badleveragesb1); badleveragesb1
#influence points measured by Cook's distance
cookssb1 = cooks.distance(sb1)
cutcookssb1 = 4 / (nrow(data) - k1 - 1); cutcookssb1 ##get the cut off = 0.0119403
badcookssb1 = as.numeric(cookssb1 > cutcookssb1)
table(badcookssb1); badcookssb1
#overall outliers; add them up and get rid of them
totaloutsb1 = badleveragesb1 + badcookssb1
```

```
table(totaloutsb1); totaloutsb1
inlinersb1 = subset(data, totaloutsb1 < 2) #330 observations
#inspect assumptions
sb1.clean <- lm(EnvironmentalConscious ~ MeaningAndEngagement + Pleasure, data=inlinersb1); summary(sb1.clean)
par(mfrow = c(2, 2)); plot(sb1.clean); par(mfrow = c(1, 1))
#assumption set up
standardizedsb1 = rstudent(sb1.clean) #Create the standardized residuals
fittedsb1 = scale(sb1.clean$fitted.values); fittedsb1 #Create the fitted values
#normality
hist(standardizedsb1)
#linearity
qqnorm(standardizedsb1); abline(0,1)
#homogeneity and homoscedasticity
plot(fittedsb1, standardizedsb1); abline(0,0); abline(v=0); abline(v=-2); abline(v=2); abline(h=-2); abline(h=2)
library(lmtest); bptest(sb1.clean)
#stepwise
intercept.only.model.sb1 <- lm(EnvironmentalConscious ~ 1, data = inlinersb1); summary(intercept.only.model.sb1)
full.model.clean.sb1 <- lm(EnvironmentalConscious ~ MeaningAndEngagement + Pleasure, data = inlinersb1)
lm.step.sb1 <- step(intercept.only.model.sb1, direction = 'both', scope = formula(full.model.clean.sb1))
lm.step.one.sb1 <- lm(EnvironmentalConscious ~ MeaningAndEngagement, data = inlinersb1); summary(lm.step.one.sb1)
library(QuantPsyc); lm.beta(lm.step.sb1)
#MeaningAndEngagement = 0.6636543; Pleasure is removed

sb2 <- lm(ThreeRs ~ MeaningAndEngagement + Pleasure, data=data); summary(sb2)
par(mfrow = c(2, 2)); plot(sb2); par(mfrow = c(1, 1))
library(lmtest); bptest(sb2)
#Further homoskedasticity check
#leverage
k2 = 2 ##number of IVs in the sb2
leveragesb2 = hatvalues(sb2)
cutleveragesb2 = (2*k2+2) / nrow(data); cutleveragesb2 ##cut off = 0.01775148
badleveragesb2 = as.numeric(leveragesb2 > cutleveragesb2)
table(badleveragesb2); badleveragesb2
#influence points measured by Cook's distance
cookssb2 = cooks.distance(sb2)
cutcookssb2 = 4 / (nrow(data) - k2 - 1); cutcookssb2 ##get the cut off = 0.0119403
badcookssb2 = as.numeric(cookssb2 > cutcookssb2)
```

```
table(badcookssb2); badcookssb2
#overall outliers; add them up and get rid of them
totaloutsb2 = badleveragesb2 + badcookssb2
table(totaloutsb2); totaloutsb2
inlinersb2 = subset(data, totaloutsb2 < 2) #329 observations
#inspect assumptions
sb2.clean <- lm(ThreeRs ~ MeaningAndEngagement + Pleasure, data=inlinersb2); summary(sb2.clean)
par(mfrow = c(2, 2)); plot(sb2.clean); par(mfrow = c(1, 1))
#assumption set up
standardizedsb2 = rstudent(sb2.clean) #Create the standardized residuals
fittedsb2 = scale(sb2.clean$fitted.values); fittedsb2 #Create the fitted values
#normality
hist(standardizedsb2)
#linearity
qqnorm(standardizedsb2); abline(0,1)
#homogeneity and homoscedasticity
plot(fittedsb2, standardizedsb2); abline(0,0); abline(v=0); abline(v=-2); abline(v=2); abline(h=-2);
abline(h=2)
library(lmtest); bptest(sb2.clean)
#stepwise
intercept.only.model.sb2 <- lm(EnvironmentalConscious ~ 1, data = inlinersb2);
summary(intercept.only.model.sb2)
full.model.clean.sb2 <- lm(EnvironmentalConscious ~ MeaningAndEngagement + Pleasure, data =
inlinersb2)
lm.step.sb2 <- step(intercept.only.model.sb2, direction = 'both', scope = formula(full.model.clean.sb2))
lm.step.one.sb2 <- lm(EnvironmentalConscious ~ MeaningAndEngagement, data = inlinersb2);
summary(lm.step.one.sb2)
library(QuantPsyc); lm.beta(lm.step.sb2)
#MeaningAndEngagement = 0.6316773; Pleasure is removed

sb3 <- lm(EnergyConservation ~ MeaningAndEngagement + Pleasure, data=data); summary(sb3)
par(mfrow = c(2, 2)); plot(sb3); par(mfrow = c(1, 1))
library(lmtest); bptest(sb3)
#Further homoskedasticity check
#leverage
k3 = 2 ##number of IVs in the sb3
leveragesb3 = hatvalues(sb3)
cutleveragesb3 = (2*k3+2) / nrow(data); cutleveragesb3 ##cut off = 0.01775148
badleveragesb3 = as.numeric(leveragesb3 > cutleveragesb3)
table(badleveragesb3); badleveragesb3
#influence points measured by Cook's distance
cookssb3 = cooks.distance(sb3)
```

```
cutcookssb3 = 4 / (nrow(data) - k3 - 1); cutcookssb3 ##get the cut off = 0.0119403
badcookssb3 = as.numeric(cookssb3 > cutcookssb3)
table(badcookssb3); badcookssb3
#overall outliers; add them up and get rid of them
totaloutsb3 = badleveragesb3 + badcookssb3
table(totaloutsb3); totaloutsb3
inlinersb3 = subset(data, totaloutsb3 < 2) #333 observations
#inspect assumptions
sb3.clean <- lm(EnergyConservation ~ MeaningAndEngagement + Pleasure, data=inlinersb3);
summary(sb3.clean)
par(mfrow = c(2, 2)); plot(sb3.clean); par(mfrow = c(1, 1))
#assumption set up
standardizedsb3 = rstudent(sb3.clean) #Create the standardized residuals
fittedsb3 = scale(sb3.clean$fitted.values); fittedsb3 #Create the fitted values
#normality
hist(standardizedsb3)
#linearity
qqnorm(standardizedsb3); abline(0,1)
#homogeneity and homoscedasticity
plot(fittedsb3, standardizedsb3); abline(0,0); abline(v=0); abline(v=-2); abline(v=2); abline(h=-2);
abline(h=2)
library(lmtest); bptest(sb3.clean)
#stepwise
intercept.only.model.sb3 <- lm(EnergyConservation ~ 1, data = inlinersb3);
summary(intercept.only.model.sb3)
full.model.clean.sb3 <- lm(EnergyConservation ~ MeaningAndEngagement + Pleasure, data = inlinersb3)
lm.step.sb3 <- step(intercept.only.model.sb3, direction = 'both', scope = formula(full.model.clean.sb3))
lm.step.one.sb3 <- lm(EnergyConservation ~ MeaningAndEngagement, data = inlinersb3);
summary(lm.step.one.sb3)
library(QuantPsyc); lm.beta(lm.step.sb3)
#MeaningAndEngagement = 0.58524790; Pleasure is removed

##################DiscriminantAnalysis########################
library(tidyverse)
library(MASS)    #load the package for lda functions
library(DiscriMiner) #load the package for lda functions
library(ggplot2)    #visualization
library(dplyr) #data manipulation
library(gridExtra) #visualization
library(car) #multvariate test
library(psych)
library(corrplot) #visualization for correlation
```

```
library(Hmisc)
### Data preparation
data<-read.csv('/Users/zhangzhixuan/Desktop/DANA4830/Project/CleanedDataFile.csv')
data$MeaningAndEngagement <- c(rowSums(data[,c("M11", "M14", "M02", "M12", "M05", "E04", "E09",
"M17", "E07", "P13", "E01", "E10")])/12)
data$Pleasure <- c(rowSums(data[,c("P15", "P03", "P18", "P16", "P08", "E06")])/6)


data$EnvironmentalConscious <- c(rowSums(data[, c("SC_4", "SC_13", "SC_19", "SC_18", "SC_17",
"SC_3", "SC_12", "SC_14", "SC_9", "SC_20", "SC_1", "SC_16", "SC_11", "SC_2", "SC_15",
"SC_31")])/16)
data$ThreeRs <- c(rowSums(data[,c("SC_22", "SC_26", "SC_25", "SC_21", "SC_23", "SC_28",
"SC_24")])/7)
data$EnergyConservation <- c(rowSums(data[, c("SC_33", "SC_34", "SC_35", "SC_7", "SC_6", "SC_5",
"SC_32", "SC_29", "SC_27", "SC_8")])/10)
#####-------------DA using 5 factors from Part1 & Part2---------------
sex<-data$sex
v1<-data$MeaningAndEngagement;v1
v2<-data$Pleasure
v3<-data$EnvironmentalConscious
v4<-data$ThreeRs
v5<-data$EnergyConservation
DA <- data_frame(sex,v1,v2,v3,v4,v5)
DA$sex=factor(DA$sex)
DA <- na.omit(DA)
summary(DA)
###--Assumption--Check--------------------------------
qqPlot(DA$v1)
qqPlot(DA$v2)
qqPlot(DA$v3)
qqPlot(DA$v4)
qqPlot(DA$v5)
shapiro.test(DA$v1)
shapiro.test(DA$v2)
shapiro.test(DA$v3)
shapiro.test(DA$v4)
shapiro.test(DA$v5) ## most of the variables failed the normality test.
## Equal variance test
X=as.matrix(DA[,2:5])
Y=as.matrix(DA[,1])
M=manova(X~Y)
summary(M) ## P-value <0.05, we reject the Null hypothesis that our data is equal variance.
```

```
#Plot Checking the Assumption of Equal Variance
plot <- list()
box_variables <- c("sex","v1","v2","v3","v4","v5")
for(i in box_variables) {
  plot[[i]] <- ggplot(DA, aes_string(x = "sex", y = i, col = "sex", fill = "sex")) +
    geom_boxplot(alpha = 0.2) +
    theme(legend.position = "none") +
    scale_color_manual(values = c("blue", "red", "green"))
  scale_fill_manual(values = c("blue", "red", "green"))
}
do.call(grid.arrange, c(plot, nrow = 1))
##--------Data partition with the ratio of 7:3-----------------------
set.seed(105)
DAdiv <- sample(2, nrow(DA),
                replace = TRUE,
                prob = c(0.7, 0.3))
trainingset <- DA[DAdiv == 1,]
testingset <- DA[DAdiv == 2,]
#----------------------------------------------
#variable selections
library(klaR)
daforward <- greedy.wilks(sex~., data = trainingset, method = "lda")
daforward
da.fwd <- lda(daforward$formula, data = trainingset)
da.fwd
## training dataset
prediction1 <- predict(da.fwd, trainingset)
prediction1$class
confusiontab.one <- table(Predicted = prediction1$class, Actual = trainingset$sex)
confusiontab.one
sum(diag(confusiontab.one))/sum(confusiontab.one)
## testing dataset
prediction2 <- predict(da.fwd, testingset)
prediction2$class
confusiontab2 <- table(Predicted = prediction2$class, Actual = testingset$sex)
confusiontab2
sum(diag(confusiontab2))/sum(confusiontab2)

##--DA--For--Factors--for-Jobs
job<-data$job
v1<-data$MeaningAndEngagement
v2<-data$Pleasure
```

```
v3<-data$EnvironmentalConscious
v4<-data$ThreeRs
v5<-data$EnergyConservation
DA <- data_frame(job,v1,v2,v3,v4,v5)
DA$job=factor(DA$job)
DA <- na.omit(DA)
summary(DA)
##--------Data partition with the ratio of 7:3-----------------------
set.seed(125)
DAdiv <- sample(2, nrow(DA),
                replace = TRUE,
                prob = c(0.7, 0.3))
trainingset <- DA[DAdiv == 1,]
testingset <- DA[DAdiv == 2,]
#---------------------------------------------
#variable selections
library(klaR)
daforward <- greedy.wilks(job~., data = trainingset, method = "lda")
daforward
da.fwd <- lda(daforward$formula, data = trainingset)
da.fwd
## training dataset
prediction1 <- predict(da.fwd, trainingset)
prediction1$class
confusiontab.one <- table(Predicted = prediction1$class, Actual = trainingset$job)
confusiontab.one
sum(diag(confusiontab.one))/sum(confusiontab.one)
## testing dataset
prediction2 <- predict(da.fwd, testingset)
prediction2$class
confusiontab2 <- table(Predicted = prediction2$class, Actual = testingset$job)
confusiontab2
sum(diag(confusiontab2))/sum(confusiontab2)

##--DA--For--Factors--for-Edu
edu<-data$edu
v1<-data$MeaningAndEngagement
v2<-data$Pleasure
v3<-data$EnvironmentalConscious
v4<-data$ThreeRs
v5<-data$EnergyConservation
DA <- data_frame(edu,v1,v2,v3,v4,v5)
```

```
DA$edu=factor(DA$edu)
DA <- na.omit(DA)
summary(DA)
##--------Data partition with the ratio of 7:3------------------------
set.seed(205)
DAdiv <- sample(2, nrow(DA),
                replace = TRUE,
                prob = c(0.7, 0.3))
trainingset <- DA[DAdiv == 1,]
testingset <- DA[DAdiv == 2,]
#----------------------------------------------
#variable selections
library(klaR)
daforward <- greedy.wilks(edu~., data = trainingset, method = "lda")
daforward
da.fwd <- lda(daforward$formula, data = trainingset)
da.fwd
## training dataset
prediction1 <- predict(da.fwd, trainingset)
prediction1$class
confusiontab.one <- table(Predicted = prediction1$class, Actual = trainingset$edu)
confusiontab.one
sum(diag(confusiontab.one))/sum(confusiontab.one)
## testing dataset
prediction2 <- predict(da.fwd, testingset)
prediction2$class
confusiontab2 <- table(Predicted = prediction2$class, Actual = testingset$edu)
confusiontab2
sum(diag(confusiontab2))/sum(confusiontab2)

##--DA--For sex--Using Entire Part1
sex<-data$sex
part1<-data[3:20]
DA <- data_frame(sex,part1)
DA$sex=factor(DA$sex)
DA <- na.omit(DA)
summary(DA)
##--------Data partition with the ratio of 7:3------------------------
set.seed(230)
DAdiv <- sample(2, nrow(DA),
                replace = TRUE,
                prob = c(0.7, 0.3))
```

```
trainingset <- DA[DAdiv == 1,]
testingset <- DA[DAdiv == 2,]
#-----------------------------------------------
#variable selections
library(klaR)
daforward <- greedy.wilks(sex~., data = trainingset, method = "lda")
daforward
da.fwd <- lda(daforward$formula, data = trainingset)
da.fwd
## training dataset
prediction1 <- predict(da.fwd, trainingset)
prediction1$class
confusiontab.one <- table(Predicted = prediction1$class, Actual = trainingset$sex)
confusiontab.one
sum(diag(confusiontab.one))/sum(confusiontab.one)
## testing dataset
prediction2 <- predict(da.fwd, testingset)
prediction2$class
confusiontab2 <- table(Predicted = prediction2$class, Actual = testingset$sex)
confusiontab2
sum(diag(confusiontab2))/sum(confusiontab2)

##--DA--For sex--Using Entire Part1
sex<-data$sex
part1<-data[3:54]
DA <- data_frame(sex,part1)
DA$sex=factor(DA$sex)
DA <- na.omit(DA)
summary(DA)
##--------Data partition with the ratio of 7:3------------------------
set.seed(333)
DAdiv <- sample(2, nrow(DA),
                replace = TRUE,
                prob = c(0.7, 0.3))
trainingset <- DA[DAdiv == 1,]
testingset <- DA[DAdiv == 2,]
#-----------------------------------------------
#variable selections
library(klaR)
daforward <- greedy.wilks(sex~., data = trainingset, method = "lda")
daforward
da.fwd <- lda(daforward$formula, data = trainingset)
```

```
da.fwd
## training dataset
prediction1 <- predict(da.fwd, trainingset)
prediction1$class
confusiontab.one <- table(Predicted = prediction1$class, Actual = trainingset$sex)
confusiontab.one
sum(diag(confusiontab.one))/sum(confusiontab.one)
## testing dataset
prediction2 <- predict(da.fwd, testingset)
prediction2$class
confusiontab2 <- table(Predicted = prediction2$class, Actual = testingset$sex)
confusiontab2
sum(diag(confusiontab2))/sum(confusiontab2)

##################################MCA#######################
#### Data preparation
contingency <- read.csv('/Users/zhangzhixuan/Desktop/DANA4830/Project/contingency.csv')
table_contingency <- contingency[,-1]
rownames(table_contingency) <- contingency[,1]
MeaningAndEngagement <- c(colSums(table_contingency[c("M11", "M14", "M02", "M12", "M05", "E04",
"E09", "M17", "E07", "P13", "E01", "E10"),]))
Pleasure <- c(colSums(table_contingency[c("P15", "P03", "P18", "P16", "P08", "E06"),]))
EnvironmentalConscious <- c(colSums(table_contingency[c("SC_4", "SC_13", "SC_19", "SC_18",
"SC_17", "SC_3", "SC_12", "SC_14", "SC_9", "SC_20", "SC_1", "SC_16", "SC_11", "SC_2", "SC_15",
"SC_31"),]))
ThreeRs <- c(colSums(table_contingency[c("SC_22", "SC_26", "SC_25", "SC_21", "SC_23", "SC_28",
"SC_24"),]))
EnergyConservation <- c(colSums(table_contingency[c("SC_33", "SC_34", "SC_35", "SC_7", "SC_6",
"SC_5", "SC_32", "SC_29", "SC_27", "SC_8"),]))
new_table_contigency <- rbind(MeaningAndEngagement, Pleasure, EnvironmentalConscious, ThreeRs,
EnergyConservation)

### MCA
mca<-new_table_contigency
View(mca)
mca <- as.data.frame(mca)
rownames(mca) <- mca[,1]
ca.mca <- CA(mca, graph = TRUE)
print(ca.mca)

## cutoff point
1/(nrow(mca)-1) #0.25
```

```
1/(ncol(mca)-1) # 0.167
## plot without arrows
fviz_screeplot(ca.mca,addlabels=T) +
  geom_hline(yintercept=16.7,linetype=2,color="red")

### loadings for rows & columns
row <- get_ca_row(ca.mca)
row$cos2
col <- get_ca_col(ca.mca)
col$cos2
### Checking coordinates
row$coord
col$coord
#plot a standard asymetric biplot ( with arrows)
fviz_ca_biplot(ca.mca,
               map ="rowprincipal", arrow = c(TRUE, TRUE),
               repel = TRUE)
### plot columns-wise
fviz_ca_col(ca.mca)
### plot rows-wise
fviz_ca_row(ca.mca, repel = TRUE)# relationship between row points
```