```
#   FIFA - 19 Predicting Wage of Players

######################################################################
#####################   Preprocessing     ##########################

setwd('D:/College/4th Quarter/DSC 424/Final Project')

dataset1 <- read.csv(file="final.csv", header=TRUE, sep=",")

sum(is.na(dataset1))
dataset1 <- na.omit(dataset1)
sum(is.na(dataset1))

head(dataset1)


library(dplyr)
library(tidyr)
dataset1 = select(dataset1, -ID, -Name, -Photo, -Nationality, -Flag, -
Club.Logo, -Real.Face, -Jersey.Number, -Joined, -Loaned.From, -
Contract.Valid.Until, -Club, -Body.Type, -Position)
dataset1 = select(dataset1, -1)
dataset1 = select(dataset1, -ST, -RS, -CF, -RF, -RW, -CAM, -RAM, -CM, -RCM, -
RM, -CDM, -RDM, -RWB, -CB, -RCB, -RB)


dataset1$Preferred.Foot <- ifelse(dataset1$Preferred.Foot == 'Left', '0',
dataset1$Preferred.Foot)
dataset1$Preferred.Foot <- ifelse(dataset1$Preferred.Foot == 'Right', '1',
dataset1$Preferred.Foot)


dataset1$Weight <- gsub(pattern = "lbs", replacement = "", dataset1$Weight)



dataset1$Value <- gsub(pattern = "â,¬", replacement = "", dataset1$Value)
dataset1$Wage <- gsub(pattern = "â,¬", replacement = "", dataset1$Wage)
dataset1$Release.Clause <- gsub(pattern = "â,¬", replacement = "",
dataset1$Release.Clause)



head(dataset1)



dataset1$release <- gsub(pattern = "", replacement = NULL, dataset1$Weight)


##Separated Work.Rate variable into 2 Variables named as AWR, DWR.
##Low = 0
##Medium = 1
##High = 2
```

```
dataset1$AWR <- ifelse(dataset1$AWR == 'Low', '0', dataset1$AWR)
dataset1$AWR <- ifelse(dataset1$AWR == 'Medium', '1', dataset1$AWR)
dataset1$AWR <- ifelse(dataset1$AWR == 'High', '2', dataset1$AWR)


dataset1$DWR <- ifelse(dataset1$DWR == 'Low', '0', dataset1$DWR)
dataset1$DWR <- ifelse(dataset1$DWR == 'Medium', '1', dataset1$DWR)
dataset1$DWR <- ifelse(dataset1$DWR == 'High', '2', dataset1$DWR)


##Height in centimeters

write.csv(dataset1,"D:\\College\\4th Quarter\\DSC 424\\Final Project\
\final.csv", row.names = FALSE)




######################################################################
###################### Regression Analysis  ###########################




install.packages(c("ggplot2", "ggpubr", "tidyverse", "broom", "AICcmodavg"))

library(ggplot2)
library(ggpubr)
library(tidyverse)
library(broom)
library(AICcmodavg)



setwd('D:/College/4th Quarter/DSC 424/Final Project')

one <- read.csv("final.csv", header = TRUE, colClasses = c("factor", "factor",
"factor", "numeric"))


summary(one)


hist(one$Value)




log_values <- hist(log(one$Value))
```

```
model <- lm(log_values ~., data=one)
vif(model)


log_
m5 <- lm(Value ~., data=one)




m5 <- lm(responses2$Wage ~ ., data=responses2)
summary(m5)
library(MASS)
step_forward <- stepAIC(m5, direction = "both")
step_forward$anova
summary(step_forward)



#######################################################################
###################### Principal Component Analysis  ##################



#Ian Weimer
#PCA Analysis of FIFA data

library(psych)
library(REdaS)
library(dplyr)
library(ggplot2)
library(factoextra)
library("corrplot")


final <- final_fifa1

dim(final)
sum(is.na(final))

#Testing KMO Sampling Adequacy
#Tests sample size reliability
KMO(final)
#Overall MSA =  0.72


#Test Bartlett's Test of Sphericity
#testing for shared variance
bart_spher(final)
#p-value < 0.001
```

```
#Test for Reliability Analysis using Cronbach's Alpha
#Assesses consistency of each factor / component
alpha(final,check.keys=TRUE)
#initial chronbach's alpha of all data in dataset
#raw_alpha = 0.81
#Chronbach's alpha analysis showed a reliability analysis with an alpha = 0.81

#initial pca
#using prcomp
pca1 <- prcomp(final, center=T, scale=T)

#using psych package
#checked with 3-9 components
#4 components was best
p1 = psych::principal(final, rotate="varimax", nfactors=4, scores=TRUE)
p1
summary(p1)

#p1 loadings

print(p1$loadings, cutoff=.6, sort=T)

#eigenvalue method
p1$values    #eigenvalues
table(p1$values > 1)
ggplot(t)

ggplot(as.data.frame(a))


#check eigenvalues > 1

#plain scree plot
plot(pca1, main = "FIFA Scree Plot", xlab="Components")
abline(1, 0)

summary(pca1)

#enhanced scree plot

pca1 %>% fviz_eig()

#summary of scores

scores <- p1$scores
print(scores)
print(scores[,1])

#aggregate of scores representing each principal component

scores_1 <- scores[,1]
scores_2 <- scores[,2]
scores_3 <- scores[,3]
scores_4 <- scores[,4]
```

```
#summary of scores representing each pc

summary(scores_1)
summary(scores_2)
summary(scores_3)
summary(scores_4)



######################################################################
###################### Factor Analysis  ###############################


#Libraries
library(Hmisc) #Describe Function
library(psych) #Multiple Functions for Statistics and Multivariate Analysis
library(GGally) #ggpairs Function
library(ggplot2) #ggplot2 Functions
library(vioplot) #Violin Plot Function
library(corrplot) #Plot Correlations
library(REdaS) #Bartlett's Test of Sphericity
library(psych) #PCA/FA functions
library(factoextra) #PCA Visualizations
library("FactoMineR") #PCA functions
library(ade4) #PCA Visualizations


#Set Working Directory
setwd('F:/DSC 424/Project/')


#Read in Datasets
responses <- read.csv(file="final-cca.csv", header=TRUE, sep=",")
responses_try <- read.csv(file="try-kmo.csv", header=TRUE, sep=",")

#responses1 <- read.csv(file="final1.csv", header=TRUE, sep=",")
#responses_withposition <- read.csv(file="final2.csv", header=TRUE, sep=",")



dim(responses)

#hist(responses$Wage)
#hist(responses$Value)
#hist(log(responses$Value))
#hist(log(responses$Wage))


#Check for Missing Values (i.e. NAs)
sum(is.na(responses))


responses2 <- na.omit(responses)
```

```
sum(is.na(responses2))


#m5 <- lm(responses2$Wage ~ ., data=responses2)
#summary(m5)

#library(MASS)

#step_forward <- stepAIC(m5, direction = "both")
#step_forward$anova
#summary(step_forward)

#m6<- lm(responses2$Wage ~ Age + Value + International.Reputation +
#      Skill.Moves + LM + LDM + LCB + Crossing +
#      HeadingAccuracy + Dribbling + FKAccuracy + LongPassing +
#      BallControl + SprintSpeed + ShotPower +
#      Stamina + Positioning + Penalties + SlidingTackle,
#    data = responses2)

#summary(m6)

#positions <- responses[,14:38]

#abilities <- responses[,39:72]


#positions <- responses1[,13:22]
#abilities <- responses1[,23:56]
#c <-cor(responses2)
#ca <-cor(abilities)

#library(corrplot)
#corrplot(c, method="circle")
#corrplot(ca, method="circle")









####################################################################
#Conducting the PCA

alpha(responses2, check.keys=TRUE)
r = cor(responses2)
KMO(responses2)
cortest.bartlett(r)
```

```
p2 = prcomp(responses2, center=T, scale=T)


plot(p2)
abline(1, 0)

summary(p2)
print(p2)

#Conducting Factor Analysis


fit = factanal(responses2, 4,rotation = "varimax", lower = 0.01,scores =
c("regression"))
print(fit$loadings,  cutoff=.5, sort=T)
summary(fit)

scores <- fit$scores
scores_1 <- scores[,1]
scores_2 <- scores[,2]
scores_3 <- scores[,3]
scores_4 <- scores[,4]



#dim(fit$scores)

#scores_1

#new_position <- cbind(fit$scores)
#Labeling the data

#names(new_position) <- c("Defence", "Attack")
#head(new_position)

#write.csv(new_position,"F:/DSC 424/Assignment -3/position.csv", row.names =
FALSE)


#m1 <- lm(responses_withposition$Wage ~ ., data=responses_withposition)
#summary(m1)

##################################################################################

#Using Factoextra
library(factoextra)

p3 <- prcomp(responses2, scale = TRUE)
fviz_eig(p3)

#PCA Individuals
pI<-fviz_pca_ind(p3,
                 col.ind = "cos2", # Color by the quality of representation
                 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
```

```
                 repel = TRUE     # Avoid text overlapping
)

pI

#PCA Variables
pca_var<-fviz_pca_var(p3,
                      col.var = "contrib", # Color by contributions to the PC
                      gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
                      repel = TRUE     # Avoid text overlapping
)

pca_var

#Biplot
bi_plot<-fviz_pca_biplot(p3, repel = TRUE,
                         col.var = "#2E9FDF", # Variables color
                         col.ind = "#696969"  # Individuals color
)

bi_plot

library("FactoMineR")
p4 <- PCA(responses2, graph = FALSE)
#IF graph is set to true, it will provide the individual and variable maps

#Shows all the objects or functions available in PCA
print(p4)

#Options for providing screeplot
fviz_eig(p4, addlabels = TRUE, ylim = c(0, 35))
fviz_screeplot(p4, addlabels = TRUE, ylim = c(0, 35))

variables <- get_pca_var(p4)

#Which variables contibute the most to the PCs?
#there are 11 variables
head(variables$contrib, 11)

library("corrplot")
corrplot(variables$contrib, is.corr=FALSE)

# Contributions of variables to PC1
fviz_contrib(p4, choice = "var", axes = 1, top = 10)
# Contributions of variables to PC2
fviz_contrib(p4, choice = "var", axes = 2, top = 10)


library(ade4)
p5 <- dudi.pca(personality_views_opinions,
               scannf = FALSE,   # Hide scree plot
               nf = 3          # Number of components kept in the results
)
fviz_screeplot(p5, addlabels = TRUE, ylim = c(0, 35))
```

```
variables2 <- get_pca_var(p5)

#Which variables contibute the most to the PCs?
#there are 11 variables
head(variables2$contrib, 11)

library("corrplot")
corrplot(variables2$contrib, is.corr=FALSE)




#######################################################################
############### Canonical Correlation Analysis  #######################



#CCA - KOMAL

library(foreign)
library(CCA)
library(yacca)
library(MASS)
library(psych) #Multiple Functions for Statistics and Multivariate Analysis
library(corrplot) #Plot Correlations
library(DescTools) #VIF Function
library(leaps) #Best Set Linear Regression Functions

#Read in Fifa data
setwd("~/Desktop/Advanced data Analysis/final project DSC424")

fifadata= read.csv("final.csv", header = TRUE, sep = ",")


#See the first six lines of the data
head(fifadata)
names(fifadata)
dim(fifadata)
##################################################################

#Check for Missing Values (i.e. NAs)

#For All Variables
sum(is.na(fifadata))
#No missing values

##################################################################

#Show Structure of Dataset
str(fifadata, list.len=ncol(fifadata))

#Create new subsets of data (Numeric Variables Only)
```

```
posiition<- fifadata[,13:22]
skills <- fifadata[,c(23:24,27:32,36, 38:56)]
balanceandaccuracy <- fifadata[,c(25,33:35,37)]
Metrics <- fifadata[,c(2:3,6,8)]
Worth<- fifadata[,4:5] #DVs
Playersattr <- cbind(posiition,skills,balanceandaccuracy)


#Show descriptive statistics

#Normality Rule of Thumb with Skewnewss and Kurtosis (think normal bell
curve):
#Short Way:
#If skewnewss is close to 0, the distribution is normal.
#If Kurtosis is -3 or 3, the distribution is normal.

#If skewness is less than -1 or greater than 1, the distribution is highly
skewed.
#If skewness is between -1 and -0.5 or between 0.5 and 1, the distribution is
moderately skewed.
#If skewness is between -0.5 and 0.5, the distribution is approximately
symmetric.


library(psych)
describe(posiition)
describe(skills)
describe(balanceandaccuracy)
describe(Metrics)
describe(Worth)
describe(Playersattr)


##################################################################
# Exploring correlations among the player's worth and his attributes


##################################################################
# This is a nice function for computing the Wilks lambdas for
# CCA data from the CCA library's method
# It computes the wilkes lambas the degrees of freedom and te
# p-values
##################################################################

ccaWilks = function(set1, set2, cca)
{
  ev = ((1 - cca$cor^2))
  ev

  n = dim(set1)[1]
  p = length(set1)
  q = length(set2)
  k = min(p, q)
  m = n - 3/2 - (p + q)/2
```

```
  m

  w = rev(cumprod(rev(ev)))

  # initialize
  d1 = d2 = f = vector("numeric", k)

  for (i in 1:k)
  {
    s = sqrt((p^2 * q^2 - 4)/(p^2 + q^2 - 5))
    si = 1/s
    d1[i] = p * q
    d2[i] = m * s - p * q/2 + 1
    r = (1 - w[i]^si)/w[i]^si
    f[i] = r * d2[i]/d1[i]
    p = p - 1
    q = q - 1
  }

  pv = pf(f, d1, d2, lower.tail = FALSE)
  dmat = cbind(WilksL = w, F = f, df1 = d1, df2 = d2, p = pv)
}
##################################################################
# Now, lets do some computation
##################################################################


# This gives us the cannonical correlates, but no significance tests
c = cancor(balanceandaccuracy, skills)
c
# The CCA library has more extensive functionality
library(CCA)

#Breakdown of the Correlations
cmat <- matcor(balanceandaccuracy, skills)

cc_fifa= cc(balanceandaccuracy, skills)

round(cc_fifa$cor, 4)


#XCoef Correlations
round(cc_fifa$xcoef, 4)

#YCoef Correlations
round(cc_fifa$ycoef, 4)

#Calculate Scores
loadings_fifa = comput(balanceandaccuracy, skills,  cc_fifa)


#Correlation X Scores
loadings_fifa$corr.X.xscores
```

```
#Correlation Y Scores
loadings_fifa$corr.Y.yscores

#Wilk's Lambda Test
wilks_fifa = ccaWilks(balanceandaccuracy,skills, cc_fifa)
round(wilks_fifa, 2)

# Now, let's calcualte the standardized coefficients
s1 = diag(sqrt(diag(cov(balanceandaccuracy))))
s1 %*% cc_fifa$xcoef

s2 = diag(sqrt(diag(cov(skills))))
s2 %*% cc_fifa$ycoef

# A basic visualization of the cannonical correlation
plt.cc(cc_fifa, type="v")
#################################################################
# Now, let's try it with yacca
#################################################################
library(yacca)

c1 = cca( posiition, Worth)

c2 = cca(skills, Worth)
c3= cca(balanceandaccuracy, Worth)

c4 =cca(Playersattr, Metrics)
c5 =cca(Playersattr, Worth)


# Perform a chisquare test on c1, c2. c3, c4
c1
ls(c1)
c1$chisq
c1$df
summary(c1)
round(pchisq(c1$chisq, c1$df, lower.tail=F), 3)

c2
ls(c2)
c2$chisq
c2$df
summary(c2)
round(pchisq(c2$chisq, c2$df, lower.tail=F), 3)

c3
ls(c3)
c3$chisq
c3$df
summary(c3)
round(pchisq(c3$chisq, c3$df, lower.tail=F), 3)

c4
ls(c4)
```

```
c4$chisq
c4$df
round(c4$xstructcorr, 2)
summary(c4)
round(pchisq(c4$chisq, c4$df, lower.tail=F), 3)

c5
ls(c5)
c5$chisq
c5$df
round(c5$xstructcorr, 2)
summary(c5)
round(pchisq(c5$chisq, c4$df, lower.tail=F), 3)
#PLOT
plot(c1)
plot(c2)
plot(c3)
plot(c4)
plot(c5)

#helioplot1
helio.plot(c1, cv=1, x.name="Position",
           y.name="Worth", main = " Players' Postion and its Worth")
#helioplot2
helio.plot(c4, cv=1, x.name="Attributes",
           y.name="Metrics", main = "CC plot of players's Attributes and 
Metrics" )




######################################################################
################## Correspondence Analysis  #########################


library(ca)
library(factoextra)
library(gplots)
Table1 <- table(Fifa.categorical$Nationality, Fifa.categorical$Club)
Table1
fit = ca(Table1)
summary(fit)
prop.table(Table1, 1)
plot(fit, mass=T, contrib="absolute",
     map="rowgreen", arrows=c(F, T)+ labels=2)
eig.val <- get_eigenvalue(fit)
eig.val
fviz_screeplot(fit, addlabels = TRUE, ylim = c(0, 50))
fviz_ca_biplot(fit, repel = FALSE, label=2)



######################################################################
```

############################ END #####################################