```
library("ggplot2") # plotting
library(MASS) # lda, qda
library(class) # knn
library(boot)
library(leaps)
library("corrplot")
library(outliers)


# 1.0 CARICAMENTO DATASET

mydf <- read.csv("suicide-rates.csv",  sep =",")


mydf <- read.csv("C:/Users/Marco/Desktop/SL Data Set/suicide-rates.csv",  sep =",")

mydf <- read.csv("/Users/mattiaspiga/Dropbox/Marco-Mattia/Studio Universitario/04 - Metodi Statistici Data Science/03
- Esercitazione/00 - ESAME FINALE/02 - Suicides - LM KFOLD/suicide-rates.csv",  sep =",")

mydf <- read.csv("D:/spigama/Documenti/Dropbox/Marco-Mattia/Studio Universitario/04 - Metodi Statistici Data
Science/03 - Esercitazione/01 - Febbraio/Suicides/suicide-rates.csv",  sep =",")



# 1.1 ANALISI ESPLORATIVA
# Visualizzo il data set e faccio una ANALISI ESPLORATIVA con le funzioni base
# noto che il data set ha una dimensione elevata circa 28.000 osservazioni e 13 variabili
# Le osservazioni riguardano statistiche relativi ai suicidi
# In particolare trovo le seguenti variabili:
# country:
# year
# ... TODO

# Numero dei casi e delle variabili
dim(mydf)

names(mydf)

# Visualizza il dataset
View(mydf)

# Le prime osservazioni del Dataset
head(mydf)

# Le ultime osservazioni del Dataset
tail(mydf)

# Struttura e classi del Dataset
str(mydf)

# Osservazione generale
summary(mydf)




# In caso di caricamento su Windows il nome colonna risultava sporca
mydf["country"] = mydf[1]
mydf = mydf[,-1]

# rimpiazza la variabile
mydf[,"gdp_for_year2"] = mydf$gdp_for_year
mydf$gdp_for_year = gsub("," , "", mydf$gdp_for_year2)
mydf$gdp_for_year = as.numeric(mydf$gdp_for_year)


# elimino le variabili ridondanti
mydf = mydf[, -which(colnames(mydf)== "gdp_for_year2")]
mydf = mydf[, -which(colnames(mydf)== "country.year")]

# Verifica presenza di osservazioni con dati NA
sum(is.na(mydf))
sum(is.na(mydf$HDI_for_year))

# Utilizziamo table per verificare la numerositÃ  delle rilevazioni su ogni Country
table(mydf$country)


#trasformazione in factor
mydf$sex = factor(mydf$sex)
levels(mydf$sex) = c("famale", "male")


# Ordinare un factor Age
levels(mydf$age)
```

```
mydf$age = factor(mydf$age, levels = c("5-14 years", "15-24 years", "25-34 years", "35-54 years", "55-74 years", "75+
years"))

summary(mydf)

mydf$generation = factor(mydf$generation)




# calcoliamo gli outlier
# non trovo significativite motivazioni per eliminare id 40
mydf[outlier( mydf$suicides.100kpop, opposite = FALSE, logical = FALSE),]




# tolto al momento
# mydf$country = factor(mydf$country)

##### Ricostruiamo la variabile HDI per i valori mancanti.
## La ricostruzione la facciamo nella consapevolezza che i valori mancati sono 20.000 e quelli presenti 8000 circa


## Ripuliamo il Dataset da i valori Na presenti solo in HDI
mydfnona = na.omit(mydf)

mydfna = mydf[is.na(mydf$HDI_for_year),]



## VALIDATION APPROACH
## Costruisco il TRAINING SET
set.seed(1)
nperc = 0.8
training.index = sample(1:nrow(mydfnona), nperc * nrow(mydfnona), replace = F)

## Creo un modello con il TRAINING SET
model_lm = lm(formula = HDI_for_year ~. -suicides_no ,  data = mydfnona[,-which(colnames(mydfnona)=="country")],
na.action = na.omit, subset = training.index)

# Costruisco la predizione utilizzando il TEST-SET
lm_pred = predict(model_lm, mydfnona[-training.index,], se.fit = TRUE)

# Verifico l'accuratezza della mia predizione
mean((mydfnona[-training.index,]$HDI_for_year - lm_pred$fit) ^ 2)

# Plot verifico aderenza tra la predizione e i dati reali del test set
plot(mydfnona[-training.index,]$HDI_for_year ~ lm_pred$fit )

## Genero i dati
summary(mydf)

str(mydf)


lm_pred2 = predict(model_lm, mydf, se.fit = TRUE)
# Rimpiazzo con i dati predetti
mydf$HDI_for_year_NEW = ifelse(is.na(mydf$HDI_for_year), lm_pred2$fit, mydf$HDI_for_year)
mydf = mydf[, -which(colnames(mydf)== "HDI_for_year")]
str(mydf)
summary(mydf)
sum(is.na(mydf))



#GRAFICO16
# verifico le correlazione tra lv
# La funzione non puo' avere NA
# suicides_no sembra essere positivamente correlato con popolazione, il che e' accettabile. maggiore e' la
popolazione e maggiore
# sara' il numero di suicidi in assoluto.
# anche gdp_for_year e' correlato a suiceds_no. Uno sarebbe portato a pensare che maggiore e' il GDP e minore e' il
numero di suicidi
# invece tante ricerche' hanno gia' correlato l'indice gdp con fenomeni di depressione e ansia

corrplot::corrplot(cor(mydf[,sapply(mydf, is.numeric)]),
                   method = "number", type = "upper", order = "AOE",
                   title = "Matrice di correlazione lineare")



### 1.4 ANALISI GRAFICA DELLE RELAZIONI TRA LE VARIABILI
#GRAFICO17
ggplot( mydf) +
```

```
  geom_bar(position="dodge", aes(x=country, y=suicides.100kpop, fill=sex), stat = "identity") +
theme(axis.text.x = element_text(angle = 90, hjust = 1))


#GRAFICO19
# ISTOGRAMMA - Numero Suicidi (Continuo) - Sex (Categorico)
sex = mydf$sex
Suicidi = mydf$suicides_no
dt = data.frame(sex, Suicidi)
colnames(dt) = c("sex", "Suicidi")

ggplot2::ggplot(data = dt) +
  ggplot2::geom_histogram(stat = "identity",
                          mapping = aes(x = reorder(dt$sex,dt$Suicidi), y = dt$Suicidi, fill = sex) ) + coord_flip()




# Densita' suicides.100kpop
# l'eta' sembra distribuita' uniformemente
ggplot2::ggplot(data = mydf) +
  geom_density(  mapping= aes(x=suicides.100kpop), alpha=0.3, fill="Red") + xlab("Suicides 100k") + ylab("Densita'")
+ ggtitle("Grafico di densit? per l'et? delle puerpere")


#GRAFICO19
### Istogramma: Y -> Numero Suicidi  nelle X -> anni
anno = mydf$year
Suicidi = mydf$suicides_no
dt = data.frame(anno, Suicidi)

colnames(dt) = c("anno", "Suicidi")

ggplot2::ggplot(data = dt) +
  ggplot2::geom_histogram(stat = "identity",
                          mapping = aes(x = dt$anno, y = dt$Suicidi, fill = anno) )

#Dalla rilevazione notiamo che il campianemnto 2016 Ã¨ pariziale.
# Si decide di eliminare dal DS la rilevazione 2016 onde evitare distorsioni. Si eliminano circa 160 osservazioni.
mydf = mydf[mydf$year!=2016,]


### Istogramma: Y -> Suicidi ogni 100k abitanti nelle X -> anni
anno = mydf$year
Suicidi100k = mydf$suicides.100kpop
dt = data.frame(anno, Suicidi100k)

colnames(dt) = c("anno", "Suicidi100k")

ggplot2::ggplot(data = dt) +
  ggplot2::geom_histogram(stat = "identity",
                          mapping = aes(x = dt$anno, y = dt$Suicidi100k, fill = anno) )



#GRAFICO20
### PUNTI - Grafico Relazione tra GDP per capital e numero di suicidi100k colorato per SEX
ggplot2::ggplot(data = mydf) +
  geom_point(alpha = 6/10, mapping = aes( x = gdp_per_capit, y = Suicidi100k, color = sex), shape = 1, size = 2,
stroke = 0.25) +
  geom_smooth(mapping = aes(x = gdp_per_capit, y = Suicidi100k, linetype = "r2"),
              method = "lm",
              formula = y ~ splines::bs(x, 3) , se = F,
              color = "green") +
  xlab("GDP per capital") + ylab("Numero di Suicidi100k") +
  ggtitle("Relazione tra GDP per capital e numero di suicidi100k colorato per SEX")




#GRAFICO21
# l'ISTOGRAMMA Paesi (categorico) - Media Suicidi100k (continuo)
# Funzione di aggregazione di una variabile Paesi (categorico) con una variabile Suidicdi100k (continuo) in MEDIA
dt <- as.data.frame( aggregate(x=mydf$suicides.100kpop, by=list(Paesi=mydf$country), FUN=mean))
ggplot2::ggplot(data = dt) +
  ggplot2::geom_histogram(alpha = 6/10, show.legend = FALSE, stat = "identity",
                mapping = aes(x = reorder(dt$Paesi,dt$x), y = dt$x,  fill = cut(x, 20))) + coord_flip() +
  xlab("Media suicidi 100k") + ylab("Paesi") +
  ggtitle("Media dei suicidi ogni 100k abitanti suddivisa per Paesi")



#GRAFICO22
# BUBBLE - BOLLE -
dt <- as.data.frame( aggregate(x=mydf$suicides.100kpop, by=list(Age=mydf$age), FUN=sum))
```

```
ggplot(dt, aes(x=dt$Age, y=dt$x, size=dt$x)) +
  geom_point(alpha=4/10)


#GRAFICO23
### PUNTI - Grafico Relazione tra GDP per capital e numero di suicidi100k colorato per SEX
ggplot2::ggplot(data = mydf[mydf$country=="Lithuania",]) +
  geom_point(alpha = 6/10, mapping = aes( x = population, y = suicides_no, color = age, shape=sex), size = 2, stroke
= 1)

#GRAFICO24
## ISTOGRAMMA - Somma dei suicidi per fasce d'eta' suddivisi per genere
ggplot(mydf, aes(x = age, y = suicides_no, fill = sex)) +
  geom_bar( alpha=6/10, stat = "identity") +
  xlab("Fasce d'eta'") + ylab("Numero di suicidi") +
  ggtitle("Somma dei suicidi per fasce d'eta' suddivisi per genere")


#GRAFICO25
## ISTOGRAMMA - Somma dei suicidi per fasce d'eta' suddivisi per genere
ggplot(mydf, aes(x = age, y = suicides_no, fill = sex)) +
  geom_bar(position="dodge", alpha=6/10, stat = "identity") +
  xlab("Fasce d'eta'") + ylab("Numero di suicidi") +
  ggtitle("Somma dei suicidi per fasce d'eta' suddivisi per genere")



plot(mydf[mydf$country=="Lithuania",]$suicides_no ~ mydf[mydf$country=="Lithuania",]$population )



#GRAFICO25bis

dt <- as.data.frame( aggregate(x=mydf$suicides.100kpop, by=list(Generazione=mydf$generation), FUN=mean))

cxc <- ggplot(dt, aes(x=Generazione, y=x, fill=factor(Generazione))) +
  geom_bar(alpha = 4/10, width = 1,stat="identity",colour = "black") +
ggtitle("Media dei suicidi per Generazione, per 100Mila Abitanti")


cxc + coord_polar()



#GRAFICO25ter

ggplot( mydf[(mydf$country=="Italy" | mydf$country=="Serbia") & mydf$sex=="male",]) +
  geom_point(aes(x=year, y=suicides.100kpop, color=country, shape = generation, fill= country), stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))


#GRAFICO25quater
ggplot( mydf[(mydf$country=="Italy" | mydf$country=="Serbia"),]) +
  geom_point(aes(x=population, y=suicides.100kpop, color=sex, shape = generation, fill= sex), stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))



####### 2 #######
#
# Sviluppare adeguati  modelli previsionali, utilizzando come variabile di risposta Suicide_no Commentare
# i modelli che si e' sviluppato, motivando quello ritenuto migliore (considerando sia l'aspetto predittivo,
# sia quello inferenziale)
#


### Metodo Subset Forward Selection

library(leaps)

subsets = regsubsets(suicides_no~. -country , mydf, really.big=T, method = "forward", nvmax = 8)

subsets.summary = summary(subsets)

plot(subsets)

plot(subsets.summary$rss[2:dim(array(subsets.summary$rss))])

plot(subsets, scale="r2")
plot(subsets, scale="adjr2")
plot(subsets, scale="Cp")
plot(subsets, scale="bic")

plot(subsets.summary$cp,xlab="Number of Variables",ylab="Cp", type="l")
which.min(subsets.summary$cp)
```

```
coef(subsets, which.min(subsets.summary$cp))


points(which.min(subsets.summary$cp), subsets.summary$cp[which.min(subsets.summary$cp)],col="red",cex=2,pch=20)


### 2.1 Modello Regressione lineare


# per realizzare il modello distinguo tra training set e test set
# con il data set preparo il modello
# con il test set ne valuto l'accuratezza
# il numero delle osservazioni ? abbastanza contenuto, potrebbe essere un problema applicando la logistica
set.seed(1)
nperc = 0.8
index_train = sample(1:nrow(mydf), nperc * nrow(mydf), replace = F)


# Crea la regressione lineare in base all'analisi eseguita sulla forward


mod_lineare1 = glm(suicides_no ~ sex + age + population + suicides.100kpop  , data = mydf[index_train,] )

mod_linearelm = lm(suicides_no ~ + sex + age + population + suicides.100kpop , data = mydf[index_train, ] )

summary(mod_linearelm)

mod_lineare2 = glm(suicides_no ~ sex + age + population + I(suicides.100kpop^2) , data = mydf[index_train, ] )

summary(mod_linearelm)

mod_lineare3 = glm(suicides_no ~ sex * age + population + I(suicides.100kpop^2) , data = mydf[index_train, ] )


summary(mod_linearelm)

plot(mod_linearelm)



mod_pred = predict(mod_lineare, mydf[-index_train,])

summary(mod_linearelm)




###    3    ####

# validazione del modello


kfold = 10

mod_lineare1 = glm(suicides_no ~ sex + age + population + suicides.100kpop  , data = mydf )
CV1 = cv.glm(mydf,mod_lineare1,  K = kfold)


mod_lineare2 = glm(suicides_no ~ sex + age + population + I(suicides.100kpop^2) , data = mydf )

summary(mod_lineare2)
plot(mod_lineare2)

CV2 = cv.glm(mydf,mod_lineare2,  K = kfold)


mod_lineare3 = glm(suicides_no ~ sex + age + sex * age + population + I(suicides.100kpop^2) , data = mydf )


CV3 = cv.glm(mydf,mod_lineare3,  K = kfold)

# I modelli 2 e 3 che ho sviluppato hanno p-value migliori,
# ma non riescono a performare quanto sperato. l'R2 rimane comunque con pochi
# margini di miglioramento e quindi si resta con CV1.
CV1$delta
CV2$delta
CV3$delta
```

```
mydf$suicidi = mydf$suicides_no / mydf$population

q <- quantile(mydf$suicidi, probs = seq(0, 1, by = 0.20))

# q <- seq(0,1, by = 0,20)

ApplyQuintiles <- function(x) {
  cut(x, breaks=c(q),
      labels=c("BASSO","MEDIO-BASSO","MEDIO","MEDIO-ALTO","ALTO"), include.lowest=TRUE)
}
mydf$suicidi_q <- sapply(mydf$suicidi, ApplyQuintiles)


table(mydf$suicidi_q )


set.seed(1)
nperc = 0.8
training.index = sample(1:nrow(mydfnona), nperc * nrow(mydfnona), replace = F)


mod_lda <- lda(suicidi_q ~ sex * age + poly(population,3) + poly(gdp_for_year,2), data = mydf,  subset =
training.index )


pred_lda = predict(mod_lda, mydf[-training.index,] )

# per valutare la bonda' della predizione
# dovete dividere la probabilità per il numero di classi
# ci sono 5 classi? 1/5 = 20%, allora il vostro predittore fa 42, performa meglio del casuale
# se ci fossero due classi ...
mean(pred_lda$class == mydf[-training.index, which(colnames(mydf)=="suicidi_q")])


t <- table(pred_lda$class, mydf[-training.index,  which(colnames(mydf)=="suicidi_q")])

t <- matrix(t, ncol=nrow(t), nrow=nrow(t), byrow = TRUE)

sum(diag(t))/sum(t)
```