**PROJECT1**

**1.**

library(readxl)

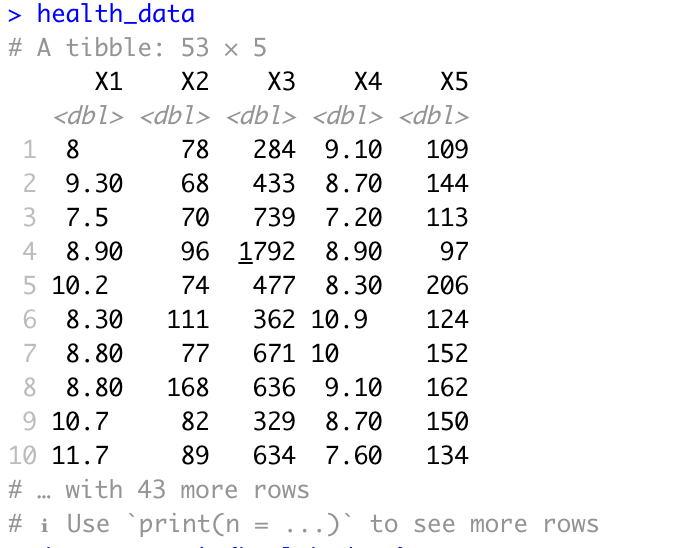
health\_data <- read\_excel("/Users/vinithavudhayagiri/Downloads/Health.xlsx")

dim(health\_data)

Graphical user interface

Description automatically generated with medium confidence

health\_data



data<-na.omit(health\_data)

dim(health\_data)

A picture containing graphical user interface

Description automatically generated

hist(health\_data$X1)

Chart, histogram

Description automatically generated

names(health\_data)

A picture containing text

Description automatically generated

par(mfrow = c(1, 1))

boxplot(`X2`~`X1`, data = health\_data)

Chart

Description automatically generated

par(mfrow = c(1, 1))

boxplot(`X3`~`X1`, data = health\_data)

Chart, box and whisker chart

Description automatically generated

par(mfrow = c(1, 1))

boxplot(`X4`~`X1`, data = health\_data)

Chart, box and whisker chart

Description automatically generated

par(mfrow = c(1, 1))

boxplot(`X5`~`X1`, data = health\_data)

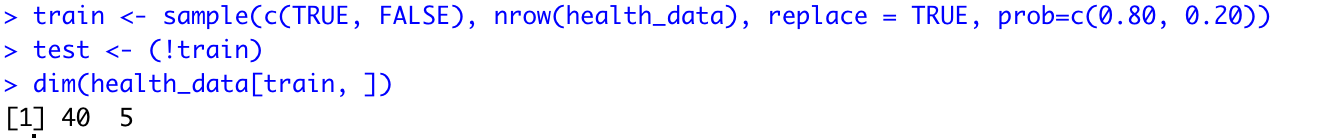
Chart, box and whisker chart

Description automatically generated

train <- sample(c(TRUE, FALSE), nrow(health\_data), replace = TRUE, prob=c(0.80, 0.20))

test <- (!train)

dim(health\_data[train, ])



lm.fit <- lm(`X1` ~ ., data = health\_data[train, ])

summary(lm.fit)

Table

Description automatically generated

lm.fit.pred <- predict(lm.fit,health\_data[test, ])

summary(lm.fit.pred)

mean((health\_data$`X1`[test] - lm.fit.pred)^2)

Text

Description automatically generated

lm.poly.fit <- lm(`X1` ~ poly(`X2`, 2) + poly(`X3`, 2) +poly(`X4`, 2) + poly(`X5`, 2), data = health\_data[train, ])

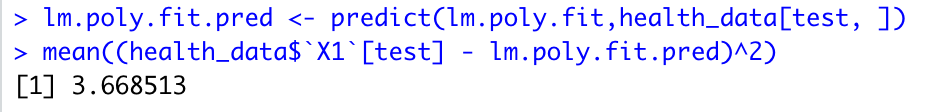
summary(lm.poly.fit)

Table

Description automatically generated

lm.poly.fit.pred <- predict(lm.poly.fit,health\_data[test, ])

mean((health\_data$`X1`[test] - lm.poly.fit.pred)^2)



library (glmnet)

x.rr <- model.matrix(`X1` ~ ., health\_data)[, -1]

y.rr <- health\_data$`X1`

cv.out.rr <- cv.glmnet(x.rr[train, ], y.rr[train], alpha = 0)

best.lambda <- cv.out.rr$lambda.min

ridge.reg <- glmnet(x.rr[train, ], y.rr[train], alpha = 0, lambda = best.lambda)

ridge.pred <- predict(ridge.reg, s = best.lambda, newx = x.rr[test, ])

mean((ridge.pred - y.rr[test])^2)

Text

Description automatically generated

cv.out.l <- cv.glmnet(x.rr[train, ], y.rr[train], alpha = 1)

best.lambda.l <- cv.out.l$lambda.min

lasso.mod <- glmnet(x.rr[train, ], y.rr[train], alpha = 1, lambda = best.lambda.l)

lasso.pred <- predict(lasso.mod, s = best.lambda.l, newx = x.rr[test, ])

mean((lasso.pred - y.rr[test])^2)

Graphical user interface, text

Description automatically generated with medium confidence

library(pls)

pcr.fit<-pcr(`X1`~ ., data = health\_data[train, ]

,scale=TRUE,validation="CV")

summary(pcr.fit)

Text, table

Description automatically generated

validationplot(pcr.fit,val.type = "MSEP")

Chart, line chart, histogram

Description automatically generated

pcr.pred<-predict(pcr.fit,health\_data[test, ],ncomp = 3)

mean((health\_data$`X1`[test]-pcr.pred)^2)

A picture containing graphical user interface

Description automatically generated

pls.fit<-plsr(`X1`~ ., data = health\_data[train, ],subset=train

,scale=TRUE,validation="CV")

summary(pls.fit)

Text, table

Description automatically generated

validationplot(pls.fit,val.type = "MSEP")

Chart, line chart

Description automatically generated

pls.pred<-predict(pls.fit,health\_data[test, ],ncomp = 1)

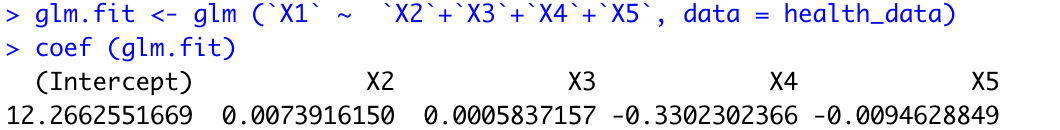
mean((health\_data$`X1`[test]-pls.pred)^2)

Text

Description automatically generated with medium confidence

glm.fit <- glm (`X1` ~ `X2`+`X3`+`X4`+`X5`, data = health\_data)

coef (glm.fit)



lm.fit <- lm (`X1` ~ `X2`+`X3`+`X4`+`X5`, data = health\_data)

coef (lm.fit)

Text

Description automatically generated with medium confidence

library (boot)

glm.fit <- glm (`X1` ~ `X2`+`X3`+`X4`+`X5`, data = health\_data)

glm.fit

Text

Description automatically generated

cv.err <- cv.glm (health\_data , glm.fit)

cv.err$delta

Text

Description automatically generated with medium confidence

cv.error <- rep(0, 10)

for (i in 1:10) {

glm.fit <- glm(`X1` ~ poly(`X2`, i)+poly(`X3`, i)+poly(`X4`, i)+poly(`X5`, i), data = health\_data)

cv.error[i] <- cv.glm(health\_data, glm.fit)$delta[1]

}

cv.error

Text

Description automatically generated

set.seed (17)

cv.error.5 <- rep (0, 5)

for (i in 1:5) {

glm.fit <- glm (`X1` ~ poly(`X2`, i) + poly(`X3`, i) +poly(`X4`, i) + poly(`X5`, i) , data = health\_data)

cv.error.5[i] <- cv.glm (health\_data , glm.fit , K = 5)$delta[1]

}

cv.error.5

Text

Description automatically generated

set.seed (17)

cv.error.10 <- rep (0, 10)

for (i in 1:10) {

glm.fit <- glm (`X1` ~ poly(`X2`, i) + poly(`X3`, i) +poly(`X4`, i) + poly(`X5`, i) , data = health\_data)

cv.error.10[i] <- cv.glm (health\_data , glm.fit , K = 10)$delta[1]

}

cv.error.10

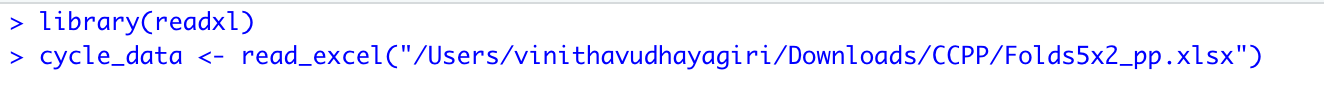
A picture containing text

Description automatically generated

**2.**

library(readxl)

cycle\_data <- read\_excel("/Users/vinithavudhayagiri/Downloads/CCPP/Folds5x2\_pp.xlsx")



dim(cycle\_data)

cycle\_data

Text

Description automatically generated

data<-na.omit(cycle\_data)

dim(cycle\_data)

A picture containing graphical user interface

Description automatically generated

hist(cycle\_data$PE)

Chart, histogram

Description automatically generated

names(cycle\_data)

A picture containing text

Description automatically generated

par(mfrow = c(1, 1))

boxplot(`AT`~`PE`, data = cycle\_data)

Chart, line chart, histogram

Description automatically generated

par(mfrow = c(1, 1))

boxplot(`V`~`PE`, data = cycle\_data)

Chart, histogram

Description automatically generated

par(mfrow = c(1, 1))

boxplot(`AP`~`PE`, data = cycle\_data)

Chart, histogram

Description automatically generated

par(mfrow = c(1, 1))

boxplot(`RH`~`PE`, data = cycle\_data)

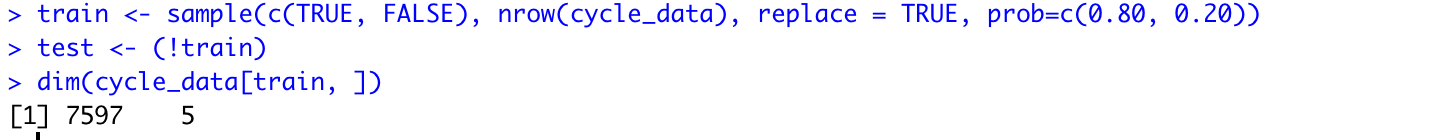
Chart

Description automatically generated

train <- sample(c(TRUE, FALSE), nrow(cycle\_data), replace = TRUE, prob=c(0.80, 0.20))

test <- (!train)

dim(cycle\_data[train, ])



lm.fit <- lm(`PE` ~ ., data = cycle\_data[train, ])

summary(lm.fit)

Table

Description automatically generated with medium confidence

lm.fit.pred <- predict(lm.fit,cycle\_data[test, ])

summary(lm.fit.pred)

mean((cycle\_data$`PE`[test] - lm.fit.pred)^2)

Text

Description automatically generated

lm.poly.fit <- lm(`PE` ~ poly(`AT`, 2) + poly(`V`, 2) +poly(`AP`, 2) + poly(`RH`, 2), data = cycle\_data[train, ])

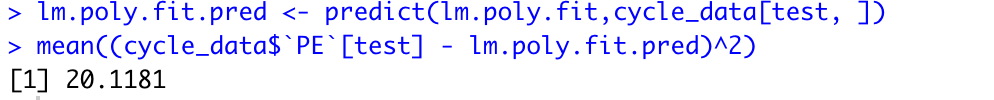
summary(lm.poly.fit)

Graphical user interface, text

Description automatically generated

lm.poly.fit.pred <- predict(lm.poly.fit,cycle\_data[test, ])

mean((cycle\_data$`PE`[test] - lm.poly.fit.pred)^2)



library (glmnet)

x.rr <- model.matrix(`PE` ~ ., cycle\_data)[, -1]

y.rr <- cycle\_data$`PE`

cv.out.rr <- cv.glmnet(x.rr[train, ], y.rr[train], alpha = 0)

best.lambda <- cv.out.rr$lambda.min

ridge.reg <- glmnet(x.rr[train, ], y.rr[train], alpha = 0, lambda = best.lambda)

ridge.pred <- predict(ridge.reg, s = best.lambda, newx = x.rr[test, ])

mean((ridge.pred - y.rr[test])^2)

Graphical user interface, text, application

Description automatically generated

cv.out.l <- cv.glmnet(x.rr[train, ], y.rr[train], alpha = 1)

best.lambda.l <- cv.out.l$lambda.min

lasso.mod <- glmnet(x.rr[train, ], y.rr[train], alpha = 1, lambda = best.lambda.l)

lasso.pred <- predict(lasso.mod, s = best.lambda.l, newx = x.rr[test, ])

mean((lasso.pred - y.rr[test])^2)

Graphical user interface, text, application

Description automatically generated

library(pls)

pcr.fit<-pcr(`PE`~ ., data = cycle\_data[train, ]

,scale=TRUE,validation="CV")

summary(pcr.fit)

Text

Description automatically generated

validationplot(pcr.fit,val.type = "MSEP")

Chart, line chart

Description automatically generated

pcr.pred<-predict(pcr.fit,cycle\_data[test, ],ncomp = 4)

mean((cycle\_data$`PE`[test]-pcr.pred)^2)

Text

Description automatically generated

pls.fit<-plsr(`PE`~ ., data = cycle\_data[train, ],subset=train

,scale=TRUE,validation="CV")

summary(pls.fit)

Table

Description automatically generated

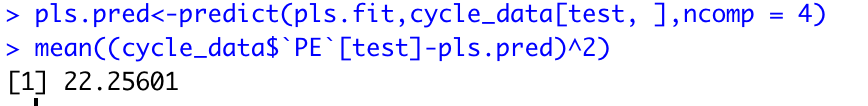
validationplot(pls.fit,val.type = "MSEP")

Chart

Description automatically generated

pls.pred<-predict(pls.fit,cycle\_data[test, ],ncomp = 4)

mean((cycle\_data$`PE`[test]-pls.pred)^2)



set.seed(1)

glm.fit <- glm (`PE` ~ `AT` + `V` + `AP` + `RH`, data = cycle\_data)

coef (glm.fit)

Chart

Description automatically generated

lm.fit <- lm (`PE` ~ `AT` + `V` + `AP` + `RH`, data = cycle\_data)

coef (lm.fit)

Text

Description automatically generated with low confidence

library (boot)

glm.fit <- glm (`PE` ~ `AT` + `V` + `AP` + `RH`, data = cycle\_data)

cv.err <- cv.glm (cycle\_data , glm.fit)

cv.err$delta

A picture containing graphical user interface

Description automatically generated

set.seed (17)

cv.error.5 <- rep (0, 5)

for (i in 1:5) {

glm.fit <- glm (`PE` ~ poly(`AT`, i) + poly(`V`, i) +poly(`AP`, i) + poly(`RH`, i) , data = cycle\_data)

cv.error.5[i] <- cv.glm (cycle\_data , glm.fit , K = 5)$delta[1]

}

cv.error.5

Text

Description automatically generated

set.seed (17)

cv.error.10 <- rep (0, 10)

for (i in 1:10) {

glm.fit <- glm (`PE` ~ poly(`AT`, i) + poly(`V`, i) +poly(`AP`, i) + poly(`RH`, i) , data = cycle\_data)

cv.error.10[i] <- cv.glm (cycle\_data , glm.fit , K = 10)$delta[1]

}

cv.error.10

Text

Description automatically generated