midterm

Christopher Huong

2023-07-14

Problem 4 (Total: 18 Points - 3 points each)

```
library(ISLR)
library(psych)
library(tidyverse)
data("Auto")
```

(a) Which of the predictors are quantitative, and which are qualitative?

```
Auto <- na.omit(Auto)
str(Auto)

## 'data.frame':    392 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : num  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36
231 14 161 141 54 223 241 2 ...

table(Auto$origin)

##
##   1   2   3
## 245  68  79
```

It seems that origin and name are qualitative, and the rest are quantitative predictors.

(b) What is the range of each quantitative predictor? You can answer this using the range() function.

```
range(Auto[,1])

## [1]  9.0 46.6

range(Auto[,2])

## [1] 3 8

range(Auto[,3])
```

```
## [1]  68 455
range(Auto[,4])
## [1]  46 230
range(Auto[,5])
## [1] 1613 5140
range(Auto[,6])
## [1]  8.0 24.8
range(Auto[,7])
## [1] 70 82
```

(c) What is the mean and standard deviation of each quantitative predictor?

```
describe(Auto[,1:7])[,3:4]

##                 mean     sd
## mpg            23.45   7.81
## cylinders       5.47   1.71
## displacement  194.41 104.64
## horsepower    104.47  38.49
## weight       2977.58 849.40
## acceleration   15.54   2.76
## year           75.98   3.68
```

(d) Now remove the 20th through 80th observations. What is the range, mean, and standard deviation of each predictor in the subset of the data that remains?

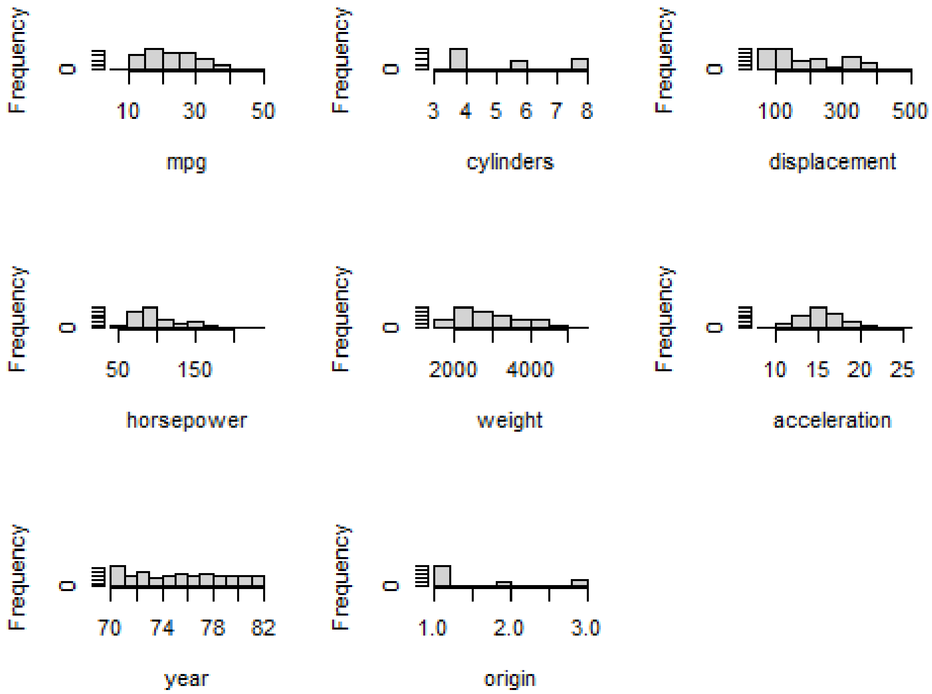
```
Auto2 <- Auto[-c(20:80),]
describe(Auto2[,1:7])[,3:4]

##                 mean     sd
## mpg            24.22   7.82
## cylinders       5.40   1.67
## displacement  189.44 101.76
## horsepower    101.86  36.60
## weight       2935.02 805.48
## acceleration   15.61   2.76
## year           76.85   3.32
```

(e) Using the full data set, investigate the predictors graphically, using scatterplots or other tools of your choice. Create some plots highlighting the relationships among the predictors. Comment on your findings.

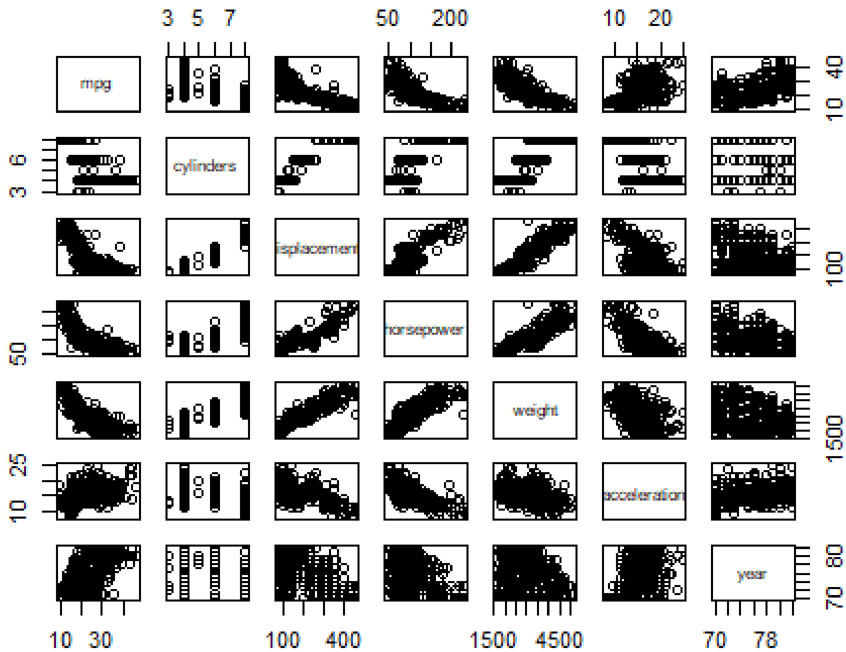
```
vars_list <- as.list(colnames(select(Auto,-name)))

par(mfrow=c(3,3))
for(i in vars_list){hist(select(Auto,-name)[,i],xlab=i,main="")}
```



mpg seems a bit right-skewed. 4-cylinders is the most common. Displacement, horsepower, and weight seems heavily right-skewed. Acceleration seems normally distributed. year seems uniformly distributed. 1 is the most common origin.

```
plot(Auto[,1:7])
```



Predictors that seem highly correlated with mpg are cylinders, displacement, horsepower, and weight. Other correlated predictors are displacement + horsepower, displacement + weight, displacement + acceleration, horsepower + weight, and horsepower + acceleration. Basically there is high colinearity in this data set.

```
cor(Auto[,1:7])

##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
##              acceleration       year
## mpg             0.4233285  0.5805410
## cylinders      -0.5046834 -0.3456474
## displacement   -0.5438005 -0.3698552
## horsepower     -0.6891955 -0.4163615
## weight         -0.4168392 -0.3091199
## acceleration    1.0000000  0.2903161
## year            0.2903161  1.0000000
```

As suspected, the predictors are highly correlated.

(f) Suppose that we wish to predict gas mileage (mpg) on the basis of the other variables. Do your plots suggest that any of the other variables might be useful in predicting mpg? Justify your answer.

Yes, the predictors that are highly related to mpg are cylinders, displacement, horsepower, and weight. These predictors are also all related to each other, so using a dimension reduction technique may be useful.

Problem 5 (Total: 22 Points)

```
library(AppliedPredictiveModeling)

## Warning: package 'AppliedPredictiveModeling' was built under R version
4.2.3

data("ChemicalManufacturingProcess")

dat <- ChemicalManufacturingProcess
```

(a) A small percentage of cells in the predictor set contain missing values. Use an appropriate imputation function to fill in these missing values. [3 points]

```
library(mice)

# dat_imp <- mice(dat, maxit=3,m=3, seed=333)
# too computationally demanding, will just drop all missing values

dat <- na.omit(dat)
```

(b) Split the data into a training and a test set, pre-process the data, and build at least four different models from Chapter 6. For those models with tuning parameters (e.g., ENET), what are the optimal values of the tuning parameter(s)? [8 points]

```
library(caret)
library(earth)

set.seed(111)

train <- createDataPartition(dat[,1], p=.80, list=F)

predicttrain <- as.data.frame(dat[train,2:58])
predicttest <- as.data.frame(dat[-train,2:58])
outcometrain <- dat[train, 1]
outcometest <- dat[-train, 1]
```

Train a linear regression model using 10-fold cross-validation, mean centering, scaling, and pca reduction

```
set.seed(111)
lm <- train(x=predicttrain,
            y=outcometrain,
```

```
            preProcess = c("center","scale","pca"),
            method='lm',
            trControl=trainControl(method="cv", number=10))

lm

## Linear Regression
##
## 124 samples
##  57 predictor
##
## Pre-processing: centered (57), scaled (57), principal component
##  signal extraction (57)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 112, 112, 112, 112, 112, 112, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   2.022703  0.5378034  1.269326
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

RMSE = 2.02, R2 = 0.54

Train elastic net model with 5-fold cross-validation, centering, scaling, and pca reduction

```
set.seed(111)
enet <- train(x=predicttrain,
             y=outcometrain,
             preProcess = c("center","scale","pca"),
             method='enet',
             tuneGrid= expand.grid(.lambda = c(0, 0.01, .1), .fraction =
seq(.05, 1, length = 10)),
             trControl=trainControl(method="cv", number=5))

enet

## Elasticnet
##
## 124 samples
##  57 predictor
##
## Pre-processing: centered (57), scaled (57), principal component
##  signal extraction (57)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 99, 99, 100, 99, 99
## Resampling results across tuning parameters:
##
##   lambda  fraction   RMSE      Rsquared   MAE
##   0.00    0.0500000  1.692612  0.2716166  1.392080
##   0.00    0.1555556  1.513049  0.4205701  1.246422
```

```
##   0.00    0.2611111  1.407004  0.4803133  1.155392
##   0.00    0.3666667  1.272371  0.5685035  1.042056
##   0.00    0.4722222  1.313006  0.5260962  1.038660
##   0.00    0.5777778  1.468802  0.4926490  1.070700
##   0.00    0.6833333  1.615362  0.4774628  1.099206
##   0.00    0.7888889  1.785173  0.4662887  1.128226
##   0.00    0.8944444  1.939039  0.4562179  1.158276
##   0.00    1.0000000  2.118992  0.4452731  1.205547
##   0.01    0.0500000  1.692612  0.2716166  1.392080
##   0.01    0.1555556  1.513049  0.4205701  1.246422
##   0.01    0.2611111  1.407004  0.4803133  1.155392
##   0.01    0.3666667  1.272371  0.5685035  1.042056
##   0.01    0.4722222  1.313006  0.5260962  1.038660
##   0.01    0.5777778  1.468802  0.4926490  1.070700
##   0.01    0.6833333  1.615362  0.4774628  1.099206
##   0.01    0.7888889  1.785173  0.4662887  1.128226
##   0.01    0.8944444  1.939039  0.4562179  1.158276
##   0.01    1.0000000  2.118992  0.4452731  1.205547
##   0.10    0.0500000  1.692612  0.2716166  1.392080
##   0.10    0.1555556  1.513049  0.4205701  1.246422
##   0.10    0.2611111  1.407004  0.4803133  1.155392
##   0.10    0.3666667  1.272371  0.5685035  1.042056
##   0.10    0.4722222  1.313006  0.5260962  1.038660
##   0.10    0.5777778  1.468802  0.4926490  1.070700
##   0.10    0.6833333  1.615362  0.4774628  1.099206
##   0.10    0.7888889  1.785173  0.4662887  1.128226
##   0.10    0.8944444  1.939039  0.4562179  1.158276
##   0.10    1.0000000  2.118992  0.4452731  1.205547
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were fraction = 0.3666667 and lambda =
0.1.
```

Best tuning parameters: fraction = 0.3666667 and lambda = 0.1. RMSE = 1.27, R2 = 0.57

Train a partial least squares model using 5-fold cross-validation, mean centering, scaling, and pca reduction

```
set.seed(111)
pls <- train(x=predicttrain,
             y=outcometrain,
             preProcess = c("center","scale","pca"),
             method='pls',
             trControl=trainControl(method="cv", number=5),
             tuneLength=10)

pls

## Partial Least Squares
##
```

```
## 124 samples
##  57 predictor
##
## Pre-processing: centered (57), scaled (57), principal component
##  signal extraction (57)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 99, 99, 100, 99, 99
## Resampling results across tuning parameters:
##
##   ncomp  RMSE      Rsquared   MAE
##    1     1.478104  0.4294234  1.153437
##    2     1.301414  0.5270782  1.036109
##    3     1.460432  0.5106712  1.058622
##    4     1.590515  0.4822214  1.086996
##    5     1.929150  0.4525423  1.170811
##    6     2.017962  0.4433364  1.189270
##    7     2.127957  0.4409885  1.209252
##    8     2.137412  0.4419235  1.211728
##    9     2.120014  0.4434193  1.206942
##   10     2.119472  0.4447445  1.206037
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was ncomp = 2.
```

Best tuning parameters: 2 principal components

RMSE = 1.30, R2 = 0.53

Train a lasso regression model using 5-fold cross-validation, mean centering, scaling, and pca reduction

```
set.seed(111)
lasso <- train(x=predicttrain,
               y=outcometrain,
               preProcess = c("center","scale","pca"),
               method='lasso',
               trControl=trainControl(method="cv", number=5),
               tuneLength=10)

lasso
```

```
## The lasso
##
## 124 samples
##  57 predictor
##
## Pre-processing: centered (57), scaled (57), principal component
##  signal extraction (57)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 99, 99, 100, 99, 99
## Resampling results across tuning parameters:
```

```
##
##   fraction   RMSE      Rsquared   MAE
##   0.1000000  1.596454  0.3654209  1.318927
##   0.1888889  1.473699  0.4409874  1.216656
##   0.2777778  1.392393  0.4907973  1.138702
##   0.3666667  1.272371  0.5685035  1.042056
##   0.4555556  1.298595  0.5328206  1.036542
##   0.5444444  1.417668  0.5004990  1.059598
##   0.6333333  1.552328  0.4829102  1.088492
##   0.7222222  1.669791  0.4735901  1.107288
##   0.8111111  1.822087  0.4637489  1.134843
##   0.9000000  1.947905  0.4557028  1.160134
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was fraction = 0.3666667.
```

Best tuning parameters: fraction = 0.3666667

RMSE = 1.27, R2 = 0.57

(c) Which model has the best predictive ability? Is any model significantly better or worse than the others? You need to conduct a hypothesis testing to justify your choice if necessary. [5 points]

The enet and lasso regression had the best predictive ability. Both were not significantly different from each other and both were significantly better than the linear regression

```
lmpred <- predict(lm, predicttest)

lmvalues  <- data.frame(obs = outcometest, pred = lmpred)

defaultSummary(lmvalues)

##      RMSE  Rsquared       MAE
## 1.4136563 0.5403592 1.1886276

enetpred <- predict(enet, predicttest)

enetvalues  <- data.frame(obs = outcometest, pred = enetpred)

defaultSummary(enetvalues)

##      RMSE  Rsquared       MAE
## 1.5893585 0.4934902 1.2136965

lassopred <- predict(lasso, predicttest)

lassovalues  <- data.frame(obs = outcometest, pred = lassopred)

defaultSummary(lassovalues)
```

```
##      RMSE  Rsquared       MAE
## 1.5893585 0.4934902 1.2136965
```

(d) Which predictors are most important in the model you have trained? Do either the biological or process predictors dominate the list [3 points]

```
set.seed(111)

varImp(lasso)

## loess r-squared variable importance
##
##   only 20 most important variables shown (out of 57)
##
##                        Overall
## ManufacturingProcess13  100.00
## ManufacturingProcess32   88.52
## BiologicalMaterial06     84.09
## ManufacturingProcess17   82.04
## BiologicalMaterial03     76.26
## ManufacturingProcess36   73.62
## ManufacturingProcess09   70.32
## BiologicalMaterial04     68.77
## BiologicalMaterial02     62.05
## BiologicalMaterial01     56.79
## BiologicalMaterial12     55.86
## ManufacturingProcess06   54.47
## BiologicalMaterial08     49.72
## ManufacturingProcess29   43.90
## BiologicalMaterial09     43.44
## ManufacturingProcess11   40.57
## ManufacturingProcess33   38.91
## ManufacturingProcess30   37.92
## BiologicalMaterial11     36.47
## ManufacturingProcess20   34.29
```

It seems that process variables are the most important predictors

(e) Explore the relationships between each of the top predictors and the response. How could this information be helpful in improving yield in future runs of the manufacturing process? [3 points]

```
cor(dat$Yield, dat$ManufacturingProcess13)

## [1] -0.5475796

cor(dat$Yield, dat$ManufacturingProcess32)

## [1] 0.5727888

cor(dat$Yield, dat$BiologicalMaterial06)

## [1] 0.4544859
```

```
cor(dat$Yield, dat$ManufacturingProcess17)

## [1] -0.4898141

cor(dat$Yield, dat$BiologicalMaterial03)

## [1] 0.4581014
```

The most important predictors tend to be more correlated with the response variable. This could be helpful because it’s a simple way to gauge how likely a variable is to contribute significantly to a predictive model.