Cross Validation_Homework_(HW5)

Kunaal Raghav

Install Packages

```
#install.packages("class")
#install.packages("boot")
#install.packages("ISLR2")
#install.packages("tidyverse")
library(class)
library(boot)
```

```
Warning: package 'boot' was built under R version 4.5.2
```

```
library(ISLR2)
```

```
Warning: package 'ISLR2' was built under R version 4.5.2
```

```
library(tidyverse)
```

```
Warning: package 'tidyverse' was built under R version 4.5.2
```

```
Warning: package 'ggplot2' was built under R version 4.5.2
```

```
Warning: package 'tibble' was built under R version 4.5.2
```

```
Warning: package 'tidyr' was built under R version 4.5.2
```

```
Warning: package 'readr' was built under R version 4.5.2
```

```
Warning: package 'purrr' was built under R version 4.5.2

Warning: package 'stringr' was built under R version 4.5.2

Warning: package 'forcats' was built under R version 4.5.2

-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.6
v forcats   1.0.1     v stringr   1.6.0
v ggplot2   4.0.1     v tibble    3.3.1
v lubridate 1.9.4     v tidyr     1.3.2
v purrr     1.2.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

Part 1: Predicting a Grade

Subsection C:

```
x <- c(90, 88, 83, 78, 85, 84)
g <- factor(c("A", "A", "A", "B", "B", "B"))
pred <- knn.cv(train = data.frame(x), cl = g, k = 3)
pred
```

```
[1] B B B B A A
Levels: A B
```

```
mean(pred != g)
```

```
[1] 0.8333333
```

Part 2: Excersize 5.4.8

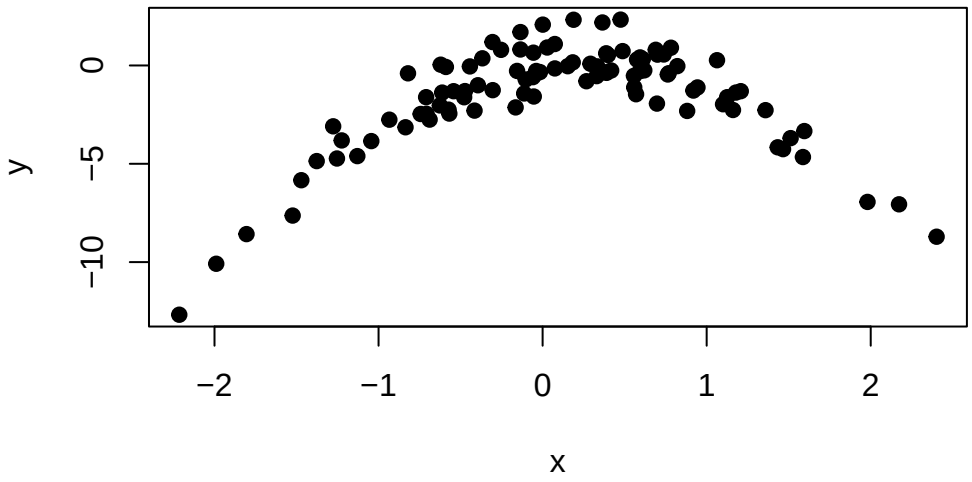
Subsection A

```
y <- x - 2*x^2 + rnorm(100)
```

```
Warning in x - 2 * x^2 + rnorm(100): longer object length is not a multiple of
shorter object length
```

```
set.seed(1)
x <- rnorm(100)
y <- x - 2*x^2 + rnorm(100)
sim.df <- data.frame(x, y)

plot(x, y, pch = 19)
```



Note that the first line of code can be rewritten as

$$Y = X - 2X^2 + \varepsilon, \varepsilon \sim \ N(0, 1)$$

n is the 00 observation, while p is the one predictor (x)

The plot is nonlinear , dominated by the quadratic term, with a downward opening parabola with noise

Subsection B

```r
set.seed(1)
x <- rnorm(100)
y <- x - 2*x^2 + rnorm(100)
sim.df <- data.frame(x, y)

#Model 1: linear
glm1 <- glm(y ~ x, data = sim.df)
set.seed(12)
cv1 <- cv.glm(sim.df, glm1)$delta[1]


# Model 2: quadratic
glm2 <- glm(y ~ x + I(x^2), data = sim.df)
set.seed(12)
cv2 <- cv.glm(sim.df, glm2)$delta[1]


# Model 3: cubic
glm3 <- glm(y ~ x + I(x^2) + I(x^3), data = sim.df)
set.seed(12)
cv3 <- cv.glm(sim.df, glm3)$delta[1]


# Model 4: quartic
glm4 <- glm(y ~ x + I(x^2) + I(x^3) + I(x^4), data = sim.df)
set.seed(12)
cv4 <- cv.glm(sim.df, glm4)$delta[1]

cv1; cv2; cv3; cv4
```

```
[1] 7.288162
```

```
[1] 0.9374236
```

```
[1] 0.9566218
```

```
[1] 0.9539049
```

Subsection C

The quadratic model (Model 2) has the smallest LOOCV Error, as the tru model is quadratic

Subsection D

```
set.seed(1)
x <- rnorm(100)
y <- x - 2*x^2 + rnorm(100)
sim.df <- data.frame(x, y)

# 10-fold CV
set.seed(12)
cv1_10 <- cv.glm(sim.df, glm1, K = 10)$delta[1]

set.seed(12)
cv2_10 <- cv.glm(sim.df, glm2, K = 10)$delta[1]

set.seed(12)
cv3_10 <- cv.glm(sim.df, glm3, K = 10)$delta[1]

set.seed(12)
cv4_10 <- cv.glm(sim.df, glm4, K = 10)$delta[1]

cv1_10; cv2_10; cv3_10; cv4_10
```

```
[1] 8.205558
```

```
[1] 0.9457335
```

```
[1] 1.032554
```

```
[1] 1.271199
```

Subsection E

LOOCV has no randomness when splitting the data, as it only leaves out one point, whereas the 10 fold CV involves splitting data RANDOMLY into 10 sets

Part 3: Excersize 5.4.5

Initial Code

```r
set.seed(123)
df <- Default


df <- df %>% mutate(default_num = if_else(default == "Yes", 1, 0))
```

Subsection A

```r
set.seed(123)
train_idx <- sample(seq_len(nrow(df)), size = 0.6 * nrow(df))
train <- df[train_idx, ]
test  <- df[-train_idx, ]

fit_full <- glm(default ~ income + balance + student,
                data = train, family = binomial)

probs <- predict(fit_full, newdata = test, type = "response")
preds <- if_else(probs > 0.5, "Yes", "No")
mean(preds != test$default)
```

```
[1] 0.0265
```

```r
fit_reduced <- glm(default ~ income + balance,
                   data = train, family = binomial)

probs2 <- predict(fit_reduced, newdata = test, type = "response")
preds2 <- if_else(probs2 > 0.5, "Yes", "No")
mean(preds2 != test$default)
```

```
[1] 0.02625
```