[Homework #6]

[Arjun Bhan]

[5/20/2021]

```r
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.0.5
```

```r
library(boot)
?Default
```

```
## starting httpd help server ... done
```

```r
names(Default)
```

```
## [1] "default" "student" "balance" "income"
```

```r
head(Default)
```

```
##   default student   balance    income
## 1      No      No  729.5265 44361.625
## 2      No     Yes  817.1804 12106.135
## 3      No      No 1073.5492 31767.139
## 4      No      No  529.2506 35704.494
## 5      No      No  785.6559 38463.496
## 6      No     Yes  919.5885  7491.559
```

```r
set.seed(100)
```

Exercise 1:

```r
mod1=glm(default~balance+income,data=Default, family="binomial")
summary(mod1)
```

```
##
## Call:
## glm(formula = default ~ balance + income, family = "binomial",
##     data = Default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4725  -0.1444  -0.0574  -0.0211   3.7245
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
## balance      5.647e-03  2.274e-04  24.836  < 2e-16 ***
## income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1579.0  on 9997  degrees of freedom
## AIC: 1585
##
## Number of Fisher Scoring iterations: 8
```

Exercise 2:

```r
cv.glm(Default,K=5,mod1)$delta[1]
```

```
## [1] 0.02155176
```

```r
cv.glm(Default,K=10,mod1)$delta[1]
```

```
## [1] 0.02146792
```

```r
#K=10 has the smallest error and therefore the most optimal for the model. Both answers are very similar showing
 that this method is very stable for the data.
```

Exercise 3:

```r
set.seed(9)
num_obs = nrow(Default)

train_index = sample(num_obs, size = trunc(0.80 * num_obs))
train_data = Default[train_index, ]
test_data = Default[-train_index, ]

moda=glm(default~balance+income,data=train_data, family="binomial")
a=predict(moda,newdata=test_data,type="response")
b=rep("No",nrow(test_data))
b[a>0.5]="Yes"
mean(b != test_data$default)
```

```
## [1] 0.0255
```

```r
#The fraction of the observation in the validation set that are incorrectly classified is 0.0255
```

Exercise 4:

```r
set.seed(90)
num_obs = nrow(Default)

train_index = sample(num_obs, size = trunc(0.95 * num_obs))
train_data = Default[train_index, ]
test_data = Default[-train_index, ]

moda=glm(default~balance+income,data=train_data, family="binomial")
a=predict(moda,newdata=test_data,type="response")
b=rep("No",nrow(test_data))
b[a>0.5]="Yes"
mean(b != test_data$default)
```

```
## [1] 0.02
```

```r
set.seed(995)
num_obs = nrow(Default)

train_index = sample(num_obs, size = trunc(0.5 * num_obs))
train_data = Default[train_index, ]
test_data = Default[-train_index, ]

moda=glm(default~balance+income,data=train_data, family="binomial")
a=predict(moda,newdata=test_data,type="response")
b=rep("No",nrow(test_data))
b[a>0.5]="Yes"
mean(b != test_data$default)
```

```
## [1] 0.0284
```

```r
set.seed(445)
num_obs = nrow(Default)

train_index = sample(num_obs, size = trunc(0.75 * num_obs))
train_data = Default[train_index, ]
test_data = Default[-train_index, ]

moda=glm(default~balance+income,data=train_data, family="binomial")
a=predict(moda,newdata=test_data,type="response")
b=rep("No",nrow(test_data))
b[a>0.5]="Yes"
mean(b != test_data$default)
```

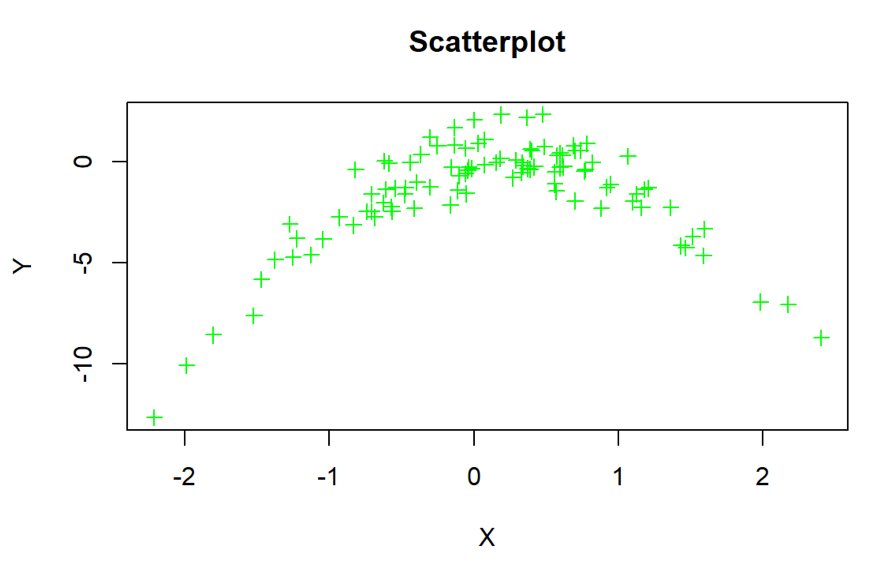
```
## [1] 0.0288
```

```r
#The outcome of the previous model is not the same as those for the current models. This shows that the validatio
n method is not stable for the data as it gives out different results. This is unlike question number 2 which gav
e out very similar results.
```

Exercise 5:

```r
set.seed(1)
X=rnorm(100) #get 100 random numbers from the standard normal distribution N(0,1)
Y=X-2*X^2+rnorm(100)
```

```r
plot(X,Y,main = "Scatterplot", col="green",pch=3)
```




```r
#The data looks like a parabola. A quadratic formula would likely be the best model to use for this data.
```

Exercise 6:

```r
library(boot)
set.seed(2)
d=data.frame(X=X,Y=Y)

mod1=glm(Y~X,data=d)
mod2=glm(Y~X+I(X^2),data=d)
mod3=glm(Y~X+I(X^2)+I(X^3),data=d)
mod4=glm(Y~X+I(X^2)+I(X^3)+I(X^4),data=d)
cv.glm(d,mod1)$delta[1]
```

```
## [1] 7.288162
```

```r
cv.glm(d,mod2)$delta[1]
```

```
## [1] 0.9374236
```

```r
cv.glm(d,mod3)$delta[1]
```

```
## [1] 0.9566218
```

```r
cv.glm(d,mod4)$delta[1]
```

```
## [1] 0.9539049
```

```r
#mod2 had the lowest cost validation error meaning it was the best model
```

Exercise 7:

```r
set.seed(2284)
d=data.frame(X=X,Y=Y)
mod1=glm(Y~X,data=d)
mod2=glm(Y~X+I(X^2),data=d)
mod3=glm(Y~X+I(X^2)+I(X^3),data=d)
mod4=glm(Y~X+I(X^2)+I(X^3)+I(X^4),data=d)
cv.glm(d,mod1)$delta[1]
```

```
## [1] 7.288162
```

```r
cv.glm(d,mod2)$delta[1]
```

```
## [1] 0.9374236
```

```r
cv.glm(d,mod3)$delta[1]
```

```
## [1] 0.9566218
```

```r
cv.glm(d,mod4)$delta[1]
```

```
## [1] 0.9539049
```

```r
set.seed(278)
d=data.frame(X=X,Y=Y)
mod1=glm(Y~X,data=d)
mod2=glm(Y~X+I(X^2),data=d)
mod3=glm(Y~X+I(X^2)+I(X^3),data=d)
mod4=glm(Y~X+I(X^2)+I(X^3)+I(X^4),data=d)
cv.glm(d,mod1)$delta[1]
```

```
## [1] 7.288162
```

```r
cv.glm(d,mod2)$delta[1]
```

```
## [1] 0.9374236
```

```r
cv.glm(d,mod3)$delta[1]
```

```
## [1] 0.9566218
```

```r
cv.glm(d,mod4)$delta[1]
```

```
## [1] 0.9539049
```

```r
set.seed(431)
d=data.frame(X=X,Y=Y)
mod1=glm(Y~X,data=d)
mod2=glm(Y~X+I(X^2),data=d)
mod3=glm(Y~X+I(X^2)+I(X^3),data=d)
mod4=glm(Y~X+I(X^2)+I(X^3)+I(X^4),data=d)
cv.glm(d,mod1)$delta[1]
```

```
## [1] 7.288162
```

```r
cv.glm(d,mod2)$delta[1]
```

```
## [1] 0.9374236
```

```r
cv.glm(d,mod3)$delta[1]
```

```
## [1] 0.9566218
```

```r
cv.glm(d,mod4)$delta[1]
```

```
## [1] 0.9539049
```

```r
#The answers for this model are the same as for the previous model. This shows that this method is stable as it g
ives the same result no matter the seed.
```

Exercise 8:

```r
#Both the graph and the cost validation error show model 2 as the best model. It had the lowest cost validation e
rror. The graph showed a parabola which means that a quadratic formula would likely be the best model to use for
 this data.
```