

```

# set working directory
setwd("C:/Users/jaosi/Desktop/DS-Projects/graduate-project/prostate-cancer")

# load dataset
APPENC05 <- read.csv("./data/processed/APPENC05.txt")
mydata <- APPENC05
summary(mydata)

##      Obs      Y_HighGradeCancer      PSALevel      CancerVol
## Min.   : 1   Min.   :0.0000   Min.   : -2.53370   Min.   : -2.30258
## 1st Qu.:25   1st Qu.:0.0000   1st Qu.: -0.65227   1st Qu.: -0.71613
## Median :49   Median :0.0000   Median : 0.09702   Median : 0.08555
## Mean   :49   Mean   :0.2165   Mean   : 0.00000   Mean   : 0.00000
## 3rd Qu.:73   3rd Qu.:0.0000   3rd Qu.: 0.50654   3rd Qu.: 0.66550
## Max.   :97   Max.   :1.0000   Max.   : 2.70223   Max.   : 2.10683
##      Weight      Age      BenignProstaticHyperplasia
## Min.   : -2.5953   Min.   : -3.0872   Min.   : -0.8406
## 1st Qu.: -0.5519   1st Qu.: -0.5220   1st Qu.: -0.8406
## Median : -0.0663   Median : 0.1531   Median : -0.3929
## Mean    : 0.0000   Mean    : 0.0000   Mean    : 0.0000
## 3rd Qu.: 0.4597   3rd Qu.: 0.5582   3rd Qu.: 0.7375
## Max.    : 4.9712   Max.    : 2.0433   Max.    : 2.5678
## SeminalVesicleInvasion CapsularPenetration
## Min.   :0.0000      Min.   : -0.5966
## 1st Qu.:0.0000      1st Qu.: -0.5966
## Median :0.0000      Median : -0.4772
## Mean    :0.2165      Mean    : 0.0000
## 3rd Qu.:0.0000      3rd Qu.: 0.2681
## Max.    :1.0000      Max.    : 4.2321

View(mydata)
names(mydata)

## [1] "Obs"                  "Y_HighGradeCancer"
## [3] "PSALevel"            "CancerVol"
## [5] "Weight"              "Age"
## [7] "BenignProstaticHyperplasia" "SeminalVesicleInvasion"
## [9] "CapsularPenetration"

# load packages
library(caTools)
library(ROCR)
library(ResourceSelection)

## Warning: package 'ResourceSelection' was built under R version 4.0.3
## ResourceSelection 0.3-5 2019-07-22

library(car)

## Warning: package 'car' was built under R version 4.0.3
## Loading required package: carData
## Warning: package 'carData' was built under R version 4.0.3

# declare SeminalVesicleInvasion a categorical variable (SeminalVesicleInvasion == [0, 1])
mydata$SeminalVesicleInvasion <- factor(mydata$SeminalVesicleInvasion)

```

```

# create training and testing subsets
myseed <- 123
set.seed(myseed)
split <- sample.split(mydata, SplitRatio=0.8)
train <- subset(mydata, split=="TRUE")
test <- subset(mydata, split=="FALSE")

View(train)
View(test)

# write train & test datasets to CSV files
write.csv(train, "./data/processed/train.txt")
write.csv(test, "./data/processed/test.txt")

#####
### Building Helpful Functions ###
#####

freq <- function(data) {
  ### function requires one input parameter: data.
  ### this function will display the table of Y_HighGradeCancer counts (frequency table);
  # i.e. the counts of 0's and 1's in the input data.
  ### the function will then display the proportion of 0 to 1 (I already know via
  # previous analysis that the counts of 0's greatly outweigh the count of 1's).
  ### we can consider this proportion to be a "base accuracy" for model comparison;
  # i.e. if the model just predicted 0's (most frequent classification),
  # for all cases.

  name <- deparse(substitute(data))

  if (name=='train') {
    cat('TRAINING DATA\n')
  }
  else {
    cat('TESTING DATA\n')
  }

  freq_tab <- table(data$Y_HighGradeCancer)
  most_freq_prop <- round(sum(freq_tab[1])/sum(freq_tab), 4)
  less_freq_pop <- round(sum(freq_tab[2])/sum(freq_tab), 4)

  # print out both the table, and calculated base accuracy
  cat('Frequency Table:\n')
  print(freq_tab)
  cat('\nThe proportion of 0 to 1 is:', most_freq_prop, '\n')
  cat('The proportion of 1 to 0 is:', less_freq_pop)
}

accuracy <- function(model, data, val=0.50) {
  ### function requires three input parameters: model, data, and decision value boundry

```

```

# (optional); default 50%.
### this function will first apply the fitted model and create classifications,
# then compare to real values (which we know).
### the confusion matrix and accuracy score will output to the terminal.
### ideally we want the accuracy score to be greater than the base score calculated
# previously (this indicates the logistic model is a better fit).
### decision boundary value may require analysis and adjustments/optimizations afterwards.

name <- deparse(substitute(data))

if (name=='train') {
  cat('TRAINING DATA\n')
}
else {
  cat('TESTING DATA\n')
}

res <- predict(model, data, type="response")
tab <- table(ActualValue=data$Y_HighGradeCancer, PredictedValue=res>=val)
err <- round((1-(sum(diag(tab))/sum(tab)))*100, 2)
acc <- round(sum(diag(tab))/sum(tab)*100, 2)

# print out confusion matrix, and calculated accuracy
cat('Prediction Rule:', val, '\n')
cat('Confusion Matrix:\n', '\n')
print(tab)
cat('\nThe calculated error is:', err, '%')
cat('\nThe calculated accuracy is:', acc, '%')
}

#####
###                               ###
###   Model Fitting               ###
###                               ###
#####

#####
### Second-Order Polynomial Logistic Models ###
#####

# fit full second-order logistic model
logit_poly <- glm(Y_HighGradeCancer ~ poly(PSALevel, 2) + poly(CancerVol, 2) +
  poly(Weight, 2) + poly(Age, 2) + poly(BenignProstaticHyperplasia, 2) +
  SeminalVesicleInvasion + poly(CapsularPenetration, 2),
  data=train, family="binomial")
summary(logit_poly)

##
## Call:
## glm(formula = Y_HighGradeCancer ~ poly(PSALevel, 2) + poly(CancerVol,
## 2) + poly(Weight, 2) + poly(Age, 2) + poly(BenignProstaticHyperplasia,
## 2) + SeminalVesicleInvasion + poly(CapsularPenetration, 2),

```

```

##      family = "binomial", data = train)
##
## Deviance Residuals:
##      Min        1Q      Median        3Q        Max
## -1.56947  -0.29292  -0.11807  -0.02792   2.70547
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -3.1879      1.1911  -2.676  0.00744 **
## poly(PSALevel, 2)1              8.8748      9.4300   0.941  0.34664
## poly(PSALevel, 2)2              9.9300      9.7971   1.014  0.31079
## poly(CancerVol, 2)1             9.0626     12.3092   0.736  0.46158
## poly(CancerVol, 2)2             3.7797      8.9295   0.423  0.67209
## poly(Weight, 2)1              -1.7984      8.8561  -0.203  0.83908
## poly(Weight, 2)2             -22.1802     19.1499  -1.158  0.24676
## poly(Age, 2)1                   3.0558      5.5976   0.546  0.58513
## poly(Age, 2)2                   6.6241      4.6130   1.436  0.15102
## poly(BenignProstaticHyperplasia, 2)1  8.0033      7.2503   1.104  0.26966
## poly(BenignProstaticHyperplasia, 2)2  6.8035      6.3958   1.064  0.28745
## SeminalVesicleInvasion1        -0.9176      1.1728  -0.782  0.43397
## poly(CapsularPenetration, 2)1       2.4574      3.9676   0.619  0.53568
## poly(CapsularPenetration, 2)2      -7.9544      4.4302  -1.795  0.07258 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 72.613  on 75  degrees of freedom
## Residual deviance: 33.998  on 62  degrees of freedom
## AIC: 61.998
##
## Number of Fisher Scoring iterations: 8

step(logit_poly, direction="backward")

## Start:  AIC=62
## Y_HighGradeCancer ~ poly(PSALevel, 2) + poly(CancerVol, 2) +
##      poly(Weight, 2) + poly(Age, 2) + poly(BenignProstaticHyperplasia,
##      2) + SeminalVesicleInvasion + poly(CapsularPenetration, 2)
##
##                                Df Deviance    AIC
## - poly(CancerVol, 2)           2   35.756 59.756
## - poly(BenignProstaticHyperplasia, 2) 2   35.913 59.913
## - SeminalVesicleInvasion         1   34.644 60.644
## - poly(Weight, 2)               2   36.698 60.698
## <none>                          33.998 61.998
## - poly(CapsularPenetration, 2)    2   38.078 62.078
## - poly(Age, 2)                  2   38.511 62.511
## - poly(PSALevel, 2)              2   40.072 64.072
##
## Step:  AIC=59.76
## Y_HighGradeCancer ~ poly(PSALevel, 2) + poly(Weight, 2) + poly(Age,
##      2) + poly(BenignProstaticHyperplasia, 2) + SeminalVesicleInvasion +
##      poly(CapsularPenetration, 2)

```

```

##
##
##      Df Deviance    AIC
## - poly(BenignProstaticHyperplasia, 2)  2   37.400 57.400
## - poly(Weight, 2)                      2   38.191 58.191
## - SeminalVesicleInvasion                1   36.552 58.552
## <none>                                35.756 59.756
## - poly(Age, 2)                        2   39.906 59.906
## - poly(CapsularPenetration, 2)         2   42.124 62.124
## - poly(PSAlevel, 2)                    2   47.961 67.961
##
## Step: AIC=57.4
## Y_HighGradeCancer ~ poly(PSAlevel, 2) + poly(Weight, 2) + poly(Age,
##      2) + SeminalVesicleInvasion + poly(CapsularPenetration, 2)
##
##      Df Deviance    AIC
## - poly(Weight, 2)                      2   38.338 54.338
## - SeminalVesicleInvasion                1   38.128 56.128
## <none>                                37.400 57.400
## - poly(Age, 2)                        2   43.065 59.065
## - poly(CapsularPenetration, 2)         2   43.734 59.734
## - poly(PSAlevel, 2)                    2   48.695 64.695
##
## Step: AIC=54.34
## Y_HighGradeCancer ~ poly(PSAlevel, 2) + poly(Age, 2) + SeminalVesicleInvasion +
##      poly(CapsularPenetration, 2)
##
##      Df Deviance    AIC
## - SeminalVesicleInvasion                1   38.900 52.900
## <none>                                38.338 54.338
## - poly(Age, 2)                        2   44.099 56.099
## - poly(CapsularPenetration, 2)         2   46.605 58.605
## - poly(PSAlevel, 2)                    2   51.230 63.230
##
## Step: AIC=52.9
## Y_HighGradeCancer ~ poly(PSAlevel, 2) + poly(Age, 2) + poly(CapsularPenetration,
##      2)
##
##      Df Deviance    AIC
## <none>                                38.900 52.900
## - poly(Age, 2)                        2   44.200 54.200
## - poly(CapsularPenetration, 2)         2   46.739 56.739
## - poly(PSAlevel, 2)                    2   51.888 61.888
##
## Call: glm(formula = Y_HighGradeCancer ~ poly(PSAlevel, 2) + poly(Age,
##      2) + poly(CapsularPenetration, 2), family = "binomial", data = train)
##
## Coefficients:
##      (Intercept)                poly(PSAlevel, 2)1
##      -2.604                      12.807
##      poly(PSAlevel, 2)2                poly(Age, 2)1
##      7.005                      2.768
##      poly(Age, 2)2 poly(CapsularPenetration, 2)1
##      6.363                      4.741

```

```

## poly(CapsularPenetration, 2)2
## -8.417
##
## Degrees of Freedom: 75 Total (i.e. Null); 69 Residual
## Null Deviance: 72.61
## Residual Deviance: 38.9 AIC: 52.9

# use second-order reduced model setup for quick analysis of adding/removing predictors
logit_poly_red <- glm(Y_HighGradeCancer ~
  poly(PSALevel, 2)
  + poly(CancerVol, 2)
  # + poly(Weight, 2)
  # + poly(Age, 2)
  # + poly(BenignProstaticHyperplasia, 2)
  # + poly(SeminalVesicleInvasion, 2)
  # + poly(CapsularPenetration, 2)
  , data=train, family="binomial")
summary(logit_poly_red)

##
## Call:
## glm(formula = Y_HighGradeCancer ~ poly(PSALevel, 2) + poly(CancerVol,
## 2), family = "binomial", data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.65643  -0.46122  -0.20861  -0.01794   2.41869
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.9802     1.2771  -2.334  0.0196 *
## poly(PSALevel, 2)1    9.0035     8.1120   1.110  0.2670
## poly(PSALevel, 2)2    0.6259     7.4678   0.084  0.9332
## poly(CancerVol, 2)1  17.9731    14.8783   1.208  0.2270
## poly(CancerVol, 2)2  -3.3028    10.0734  -0.328  0.7430
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 72.613  on 75  degrees of freedom
## Residual deviance: 44.511  on 71  degrees of freedom
## AIC: 54.511
##
## Number of Fisher Scoring iterations: 8

step(logit_poly_red, direction="backward")

## Start:  AIC=54.51
## Y_HighGradeCancer ~ poly(PSALevel, 2) + poly(CancerVol, 2)
##
##              Df Deviance    AIC
## - poly(PSALevel, 2)    2  48.120 54.120
## <none>                  44.511 54.511
## - poly(CancerVol, 2)    2  50.767 56.767

```

```
##
## Step: AIC=54.12
## Y_HighGradeCancer ~ poly(CancerVol, 2)
##
##              Df Deviance    AIC
## <none>              48.120 54.120
## - poly(CancerVol, 2)  2   72.613 74.613
##
## Call: glm(formula = Y_HighGradeCancer ~ poly(CancerVol, 2), family = "binomial",
##          data = train)
##
## Coefficients:
##      (Intercept)  poly(CancerVol, 2)1  poly(CancerVol, 2)2
##          -2.6364             20.0802             -0.4743
##
## Degrees of Freedom: 75 Total (i.e. Null);  73 Residual
## Null Deviance:      72.61
## Residual Deviance: 48.12  AIC: 54.12

#####
### First-Order Logistic Models ###
#####

# fit full first-order logistic model
logit_full <- glm(Y_HighGradeCancer ~ PSALevel + CancerVol + Weight +
                  Age + BenignProstaticHyperplasia +
                  SeminalVesicleInvasion + CapsularPenetration, data=train,
                  family="binomial")
summary(logit_full)

##
## Call:
## glm(formula = Y_HighGradeCancer ~ PSALevel + CancerVol + Weight +
##      Age + BenignProstaticHyperplasia + SeminalVesicleInvasion +
##      CapsularPenetration, family = "binomial", data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66861  -0.39211  -0.18721  -0.02467   2.17445
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -2.84305    0.75774  -3.752 0.000175 ***
## PSALevel       1.28155    0.74609   1.718 0.085856 .
## CancerVol      1.40464    0.90103   1.559 0.119014
## Weight        -0.17618    0.75535  -0.233 0.815567
## Age            0.56784    0.43547   1.304 0.192245
## BenignProstaticHyperplasia 0.07823    0.54237   0.144 0.885320
## SeminalVesicleInvasion1 -0.38818    1.04075  -0.373 0.709164
## CapsularPenetration    0.25330    0.44939   0.564 0.572988
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
## Null deviance: 72.613 on 75 degrees of freedom
## Residual deviance: 42.303 on 68 degrees of freedom
## AIC: 58.303
##
## Number of Fisher Scoring iterations: 7

step(logit_full, direction="backward")

## Start: AIC=58.3
## Y_HighGradeCancer ~ PSALevel + CancerVol + Weight + Age + BenignProstaticHyperplasia +
## SeminalVesicleInvasion + CapsularPenetration
##
##
## Df Deviance AIC
## - BenignProstaticHyperplasia 1 42.324 56.324
## - Weight 1 42.358 56.358
## - SeminalVesicleInvasion 1 42.444 56.444
## - CapsularPenetration 1 42.627 56.627
## - Age 1 43.954 57.954
## <none> 42.303 58.303
## - CancerVol 1 45.225 59.225
## - PSALevel 1 45.903 59.903
##
## Step: AIC=56.32
## Y_HighGradeCancer ~ PSALevel + CancerVol + Weight + Age + SeminalVesicleInvasion +
## CapsularPenetration
##
##
## Df Deviance AIC
## - Weight 1 42.358 54.358
## - SeminalVesicleInvasion 1 42.499 54.499
## - CapsularPenetration 1 42.695 54.695
## - Age 1 44.044 56.044
## <none> 42.324 56.324
## - CancerVol 1 45.389 57.389
## - PSALevel 1 46.031 58.031
##
## Step: AIC=54.36
## Y_HighGradeCancer ~ PSALevel + CancerVol + Age + SeminalVesicleInvasion +
## CapsularPenetration
##
##
## Df Deviance AIC
## - SeminalVesicleInvasion 1 42.522 52.522
## - CapsularPenetration 1 42.755 52.755
## - Age 1 44.151 54.151
## <none> 42.358 54.358
## - CancerVol 1 45.390 55.390
## - PSALevel 1 46.059 56.059
##
## Step: AIC=52.52
## Y_HighGradeCancer ~ PSALevel + CancerVol + Age + CapsularPenetration
##
##
## Df Deviance AIC
## - CapsularPenetration 1 42.780 50.780
## - Age 1 44.168 52.168
```



```

## <none>                42.522 52.522
## - CancerVol           1    45.558 53.558
## - PSALevel            1    46.285 54.285
##
## Step: AIC=50.78
## Y_HighGradeCancer ~ PSALevel + CancerVol + Age
##
##           Df Deviance    AIC
## - Age       1    44.628 50.628
## <none>      42.780 50.780
## - PSALevel  1    46.445 52.445
## - CancerVol 1    48.777 54.777
##
## Step: AIC=50.63
## Y_HighGradeCancer ~ PSALevel + CancerVol
##
##           Df Deviance    AIC
## <none>      44.628 50.628
## - PSALevel  1    48.123 52.123
## - CancerVol 1    50.767 54.767
##
## Call: glm(formula = Y_HighGradeCancer ~ PSALevel + CancerVol, family = "binomial",
##           data = train)
##
## Coefficients:
## (Intercept)    PSALevel    CancerVol
##      -2.687         1.058         1.550
##
## Degrees of Freedom: 75 Total (i.e. Null); 73 Residual
## Null Deviance:      72.61
## Residual Deviance: 44.63 AIC: 50.63

# use reduced model setup for quick analysis of adding/removing predictors
logit_red <- glm(Y_HighGradeCancer ~
  PSALevel
  + CancerVol
  # + Weight
  # + Age
  # + BenignProstaticHyperplasia
  # + SeminalVesicleInvasion
  # + CapsularPenetration
  , data=train, family="binomial")
summary(logit_red)

##
## Call:
## glm(formula = Y_HighGradeCancer ~ PSALevel + CancerVol, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.73560  -0.43637  -0.23378  -0.03521   2.42555
##
## Coefficients:

```

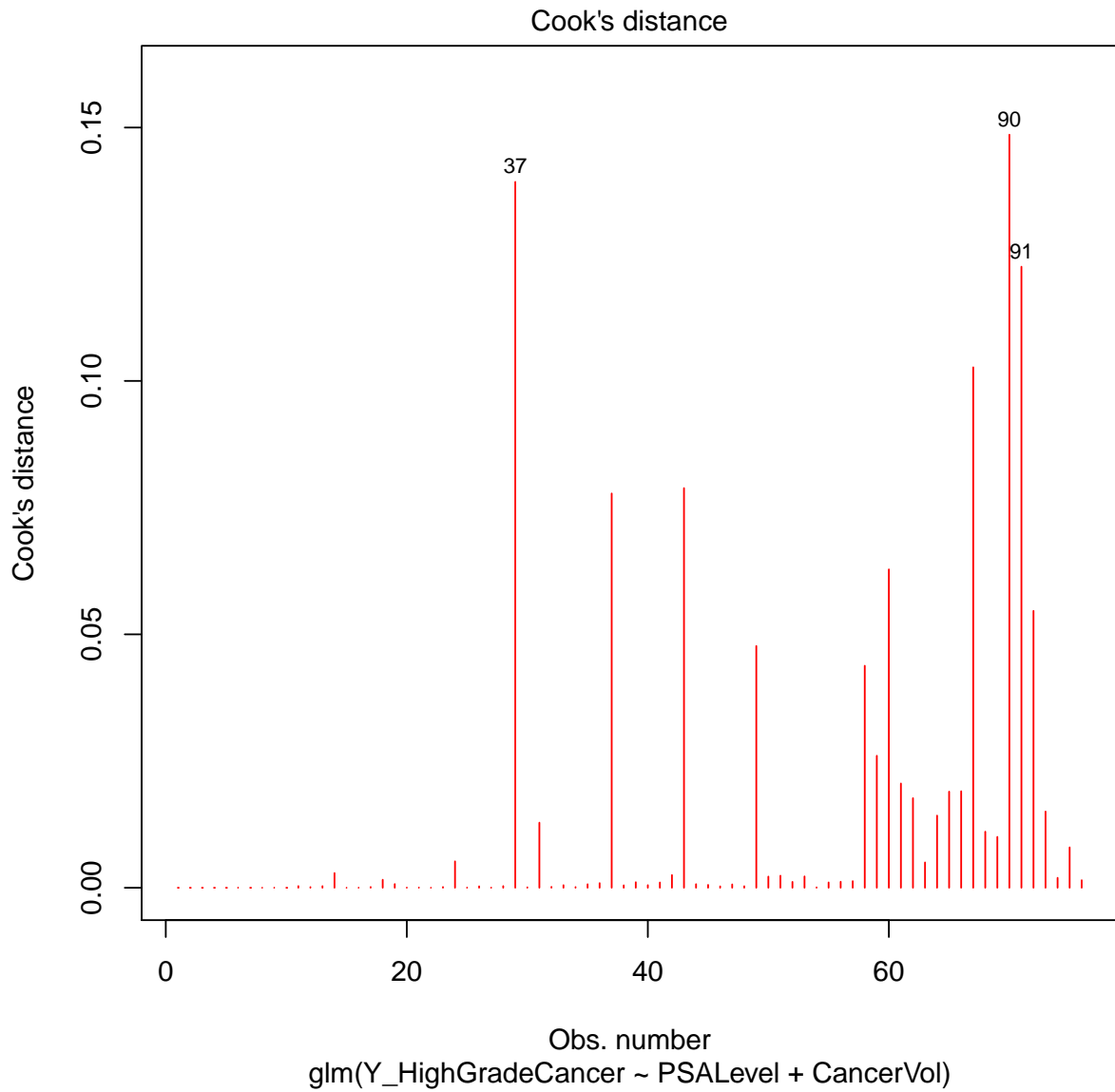
```

##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.6867      0.6186  -4.343 1.41e-05 ***
## PSALevel      1.0577      0.6198   1.707  0.0879 .
## CancerVol     1.5502      0.6859   2.260  0.0238 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 72.613  on 75  degrees of freedom
## Residual deviance: 44.628  on 73  degrees of freedom
## AIC: 50.628
##
## Number of Fisher Scoring iterations: 6

#####
###                                     ###
###   Model Checking and Validation   ###
###                                     ###
#####

#####
### Cook's Distance Diagnostics for Influential Observations ###
#####
plot(logit_red, pch=18, col="red", which=c(4))

```



```
myCDs <- sort(round(cooks.distance(logit_red), 5), decreasing=TRUE)
```

```
myCDs
```

```
##      90      37      91      85      55      47      76      92      63      74
## 0.14859 0.13928 0.12254 0.10270 0.07886 0.07780 0.06282 0.05464 0.04771 0.04380
##      75      78      84      83      79      93      82      39      87      88
## 0.02604 0.02058 0.01905 0.01897 0.01768 0.01504 0.01424 0.01284 0.01107 0.01004
##      96      30      81      18      54      65      67      64      94      22
## 0.00797 0.00521 0.00500 0.00290 0.00251 0.00238 0.00224 0.00221 0.00196 0.00158
##      97      73      72      66      49      70      52      46      24      56
## 0.00149 0.00131 0.00119 0.00116 0.00110 0.00106 0.00105 0.00092 0.00073 0.00071
##      45      60      57      42      51      48      13      36      16      61
## 0.00069 0.00066 0.00057 0.00052 0.00052 0.00048 0.00034 0.00033 0.00032 0.00031
##      33      58      40      21      29      15      43      38      69      27
```

```
## 0.00029 0.00028 0.00018 0.00015 0.00015 0.00014 0.00014 0.00009 0.00008 0.00005
##      25      31      34      7      20      10      11      28      1      2
## 0.00004 0.00004 0.00003 0.00002 0.00002 0.00001 0.00001 0.00001 0.00000 0.00000
##      3      4      6      9      12      19
## 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000

# drop rows for model building (influential observations)
# this step will be visited within Coook's Distance analysis
train_trim <- subset(train, Obs != 90
                     # & Obs != 37
                     # & Obs != 91
)

# view the trimmed data
View(train_trim)

# write train_trim dataset to csv file
write.csv(train_trim, "./data/processed/train_trim.txt")

# re-fit the logistic model
logit_red_trim <- glm(Y_HighGradeCancer ~
                     PSALevel
                     + CancerVol
                     # + Weight
                     # + Age
                     # + BenignProstaticHyperplasia
                     # + SeminalVesicleInvasion
                     # + CapsularPenetration
                     , data=train_trim, family="binomial")
summary(logit_red_trim)

##
## Call:
## glm(formula = Y_HighGradeCancer ~ PSALevel + CancerVol, family = "binomial",
##      data = train_trim)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71156  -0.43371  -0.20855  -0.03378   2.46797
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.9030     0.6907  -4.203 2.63e-05 ***
## PSALevel      0.7495     0.6120   1.225  0.2207
## CancerVol     1.9077     0.7711   2.474  0.0134 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 69.170  on 74  degrees of freedom
## Residual deviance: 41.576  on 72  degrees of freedom
## AIC: 47.576
##
```

```

## Number of Fisher Scoring iterations: 6

# RESULT: the removal of these rows did not improve the model.
# continue forward with original logit_red model

#####
### Hosmer-Lemeshow Goodness of Fit Test ###
#####

gof <- hoslem.test(logit_red$y, fitted(logit_red), g=5) # choosing 5 groups
cbind(gof$expected, gof$observed)

##              yhat0      yhat1 y0 y1
## [0.000206,0.00973] 15.941605 0.05839507 16 0
## (0.00973,0.0521] 14.592690 0.40730980 15 0
## (0.0521,0.103] 13.898096 1.10190404 14 1
## (0.103,0.333] 11.889917 3.11008275 11 4
## (0.333,0.951] 5.677692 9.32230835 6 9

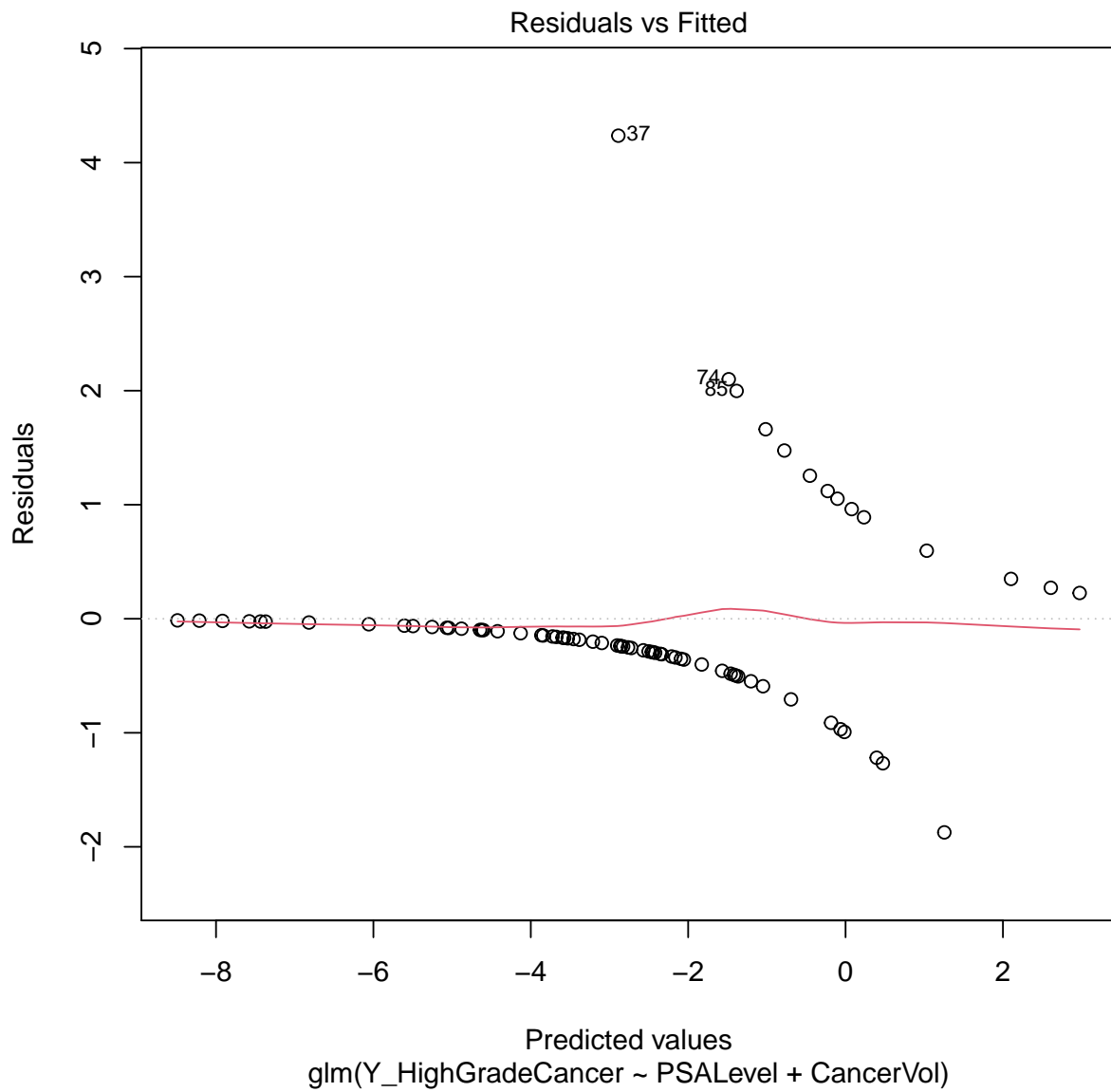
gof

##
## Hosmer and Lemeshow goodness of fit (GOF) test
##
## data: logit_red$y, fitted(logit_red)
## X-squared = 0.83815, df = 3, p-value = 0.8403

#####
### vizualizations ###
#####

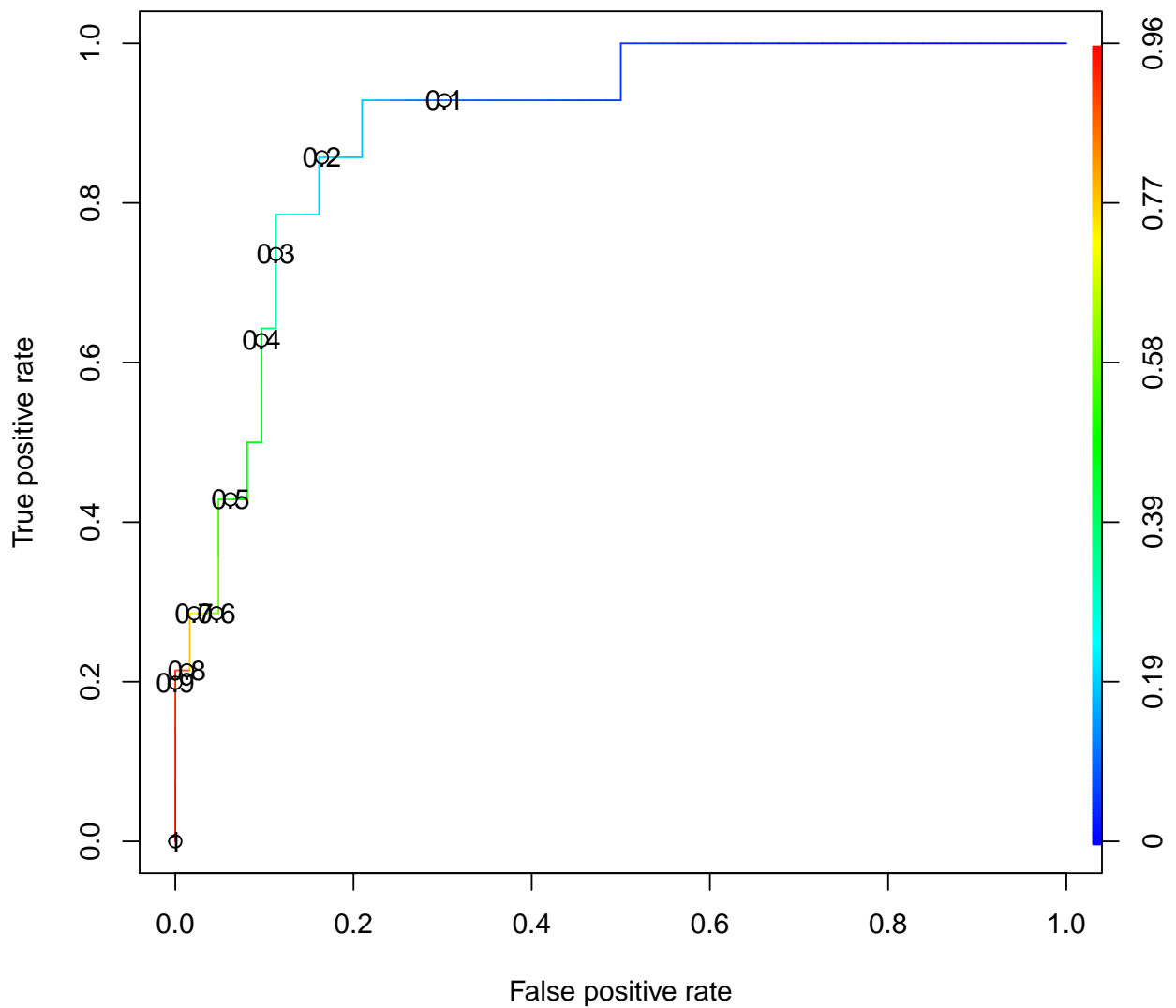
# Residuals vs. Fitted
# Normal Q-Q
# scale-location (Predicted Values vs. sqrt[Std. Pearson Residuals])
# Residuals vs. Leverage
# residualPlot(logit_red, type="pearson")
plot(logit_red, which=c(1))

```



```
### build and plot ROC curve ###
pred <- predict(logit_red, train, type="response")
ROCRPred <- prediction(pred, train$Y_HighGradeCancer)
ROCRPref <- performance(ROCRPred, "tpr", "fpr")
plot(ROCRPref, colorize=TRUE, print.cutoffs.at=seq(0.1, by=0.1),
     main="Reciever Operating Characteristic Curve")
```

## Reciever Operating Characteristic Curve



```
#####
### Accuracy Model Comparisons ###
#####

### invoke functions ###
freq(train)

## TRAINING DATA
## Frequency Table:
##
##  0  1
## 62 14
##
## The proportion of 0 to 1 is: 0.8158
```

```

## The proportion of 1 to 0 is: 0.1842

accuracy(logit_red, train, 0.184) # starting point prediction rule

## TRAINING DATA
## Prediction Rule: 0.184
## Confusion Matrix:
##
##           PredictedValue
## ActualValue FALSE TRUE
##           0    49    13
##           1     1    13
##
## The calculated error is: 18.42 %
## The calculated accuracy is: 81.58 %

accuracy(logit_red, train, 0.20) # final prediction rule

## TRAINING DATA
## Prediction Rule: 0.2
## Confusion Matrix:
##
##           PredictedValue
## ActualValue FALSE TRUE
##           0    52    10
##           1     2    12
##
## The calculated error is: 15.79 %
## The calculated accuracy is: 84.21 %

#####
###                               ###
###   Final Model   ###
###                               ###
#####

# no changes have been made from the reduced model
logit_final <- logit_red
summary(logit_final)

##
## Call:
## glm(formula = Y_HighGradeCancer ~ PSALevel + CancerVol, family = "binomial",
##      data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.73560  -0.43637  -0.23378  -0.03521   2.42555
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.6867     0.6186  -4.343 1.41e-05 ***
## PSALevel       1.0577     0.6198   1.707  0.0879 .
## CancerVol      1.5502     0.6859   2.260  0.0238 *
## ---

```



```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 72.613  on 75  degrees of freedom
## Residual deviance: 44.628  on 73  degrees of freedom
## AIC: 50.628
##
## Number of Fisher Scoring iterations: 6

#####
###                                     ###
###   Model Validation: Test Data   ###
###                                     ###
#####

#####
### Accuracy Model Comparisons ###
#####

### invoke functions ###
freq(test)

## TESTING DATA
## Frequency Table:
##
##  0  1
## 14  7
##
## The proportion of 0 to 1 is: 0.6667
## The proportion of 1 to 0 is: 0.3333

accuracy(logit_final, test, 0.20)

## TESTING DATA
## Prediction Rule: 0.2
## Confusion Matrix:
##
##      PredictedValue
## ActualValue FALSE TRUE
##      0      13      1
##      1       2      5
##
## The calculated error is: 14.29 %
## The calculated accuracy is: 85.71 %

```