```
> data<-read.csv("~/Downloads/BSAP.csv")
> drop=c("Name")
> data = data[,!names(data) %in% drop]
> names(data)
 [1] "overcatch"                                   "overcatch_abort"
 [3] "Number.of.Catch.And.Return.Null.AP"          "Number.of.Destructive.Wrapping.AP"
 [5] "Number.of.Catch.and.Do.Nothing.AP"           "Number.of.Catch.Generic.AP"
 [7] "Number.of.Dummy.Handler.AP"                  "Number.of.Ignoring.Interrupted.Exception.AP"
 [9] "Number.of.Incomplete.Implementation.AP"      "Number.of.Log.and.Return.Null.AP"
[11] "Number.of.Log.and.Throw.AP"                  "Number.of.Mutliline.log.AP"
[13] "Number.of.Nested.Try.AP"                     "Number.of.Relying.on.Get.Cause.AP"
[15] "Number.of.Throw.Within.Finally.AP"           "Number.of.Throws.Generic.AP"
[17] "Number.of.Throws.Kitchen.Sink.AP"            "Code_Ownership_count"
[19] "lines_added"                                 "lines_deleted"
[21] "post_release_defects"                        "total_change"
[23] "pre_release_defects"                         "AvgCyclomatic"
[25] "AvgCyclomaticModified"                       "AvgCyclomaticStrict"
[27] "AvgEssential"                                "AvgLine"
[29] "AvgLineBlank"                                "AvgLineCode"
[31] "AvgLineComment"                              "CountDeclClass"
[33] "CountDeclClassMethod"                        "CountDeclClassVariable"
[35] "CountDeclExecutableUnit"                     "CountDeclFunction"
[37] "CountDeclInstanceMethod"                     "CountDeclInstanceVariable"
[39] "CountDeclMethod"                             "CountDeclMethodDefault"
[41] "CountDeclMethodPrivate"                      "CountDeclMethodProtected"
[43] "CountDeclMethodPublic"                       "CountLine"
[45] "CountLineBlank"                              "CountLineCode"
[47] "CountLineCodeDecl"                           "CountLineCodeExe"
[49] "CountLineComment"                            "CountStmt"
[51] "CountStmtDecl"                               "CountStmtExe"
[53] "MaxCyclomatic"                               "MaxCyclomaticModified"
[55] "MaxCyclomaticStrict"                         "MaxEssential"
[57] "MaxNesting"                                  "RatioCommentToCode"
[59] "SumCyclomatic"                               "SumCyclomaticModified"
[61] "SumCyclomaticStrict"                         "SumEssential"
> drop=c("post_release_defects")
> independant=data[,!(names(data) %in% drop)]
> correlations <- cor(independant, method="spearman")
Warning message:
In cor(independant, method = "spearman") : the standard deviation is zero
> highCorr <- findCorrelation(correlations, cutoff = .75)
Error in if (x[i, j] > cutoff) { : missing value where TRUE/FALSE needed
> highCorr
 [1] 78 82 76 80 91 89 90 81 84 83 74 85 33 29  3 87 77 92 32 75 34 35 65 36  2 44 30 31 60 58 69 66 86
[34] 56 54  1 55 67 48 22 43 46 47 42  6 50 45  5 24
>
> low_cor_names=names(independant[, -highCorr])
> low_cor_data= independant[(names(independant) %in% low_cor_names)]
> dataforredun=low_cor_data
> redun_obj = redun (~. ,data = dataforredun ,nk =0)
Warning messages:
1: In redun(~., data = dataforredun, nk = 0) :
  Number.of.Incomplete.Implementation.AP is constant
2: In redun(~., data = dataforredun, nk = 0) :
  Number.of.Mutliline.log.AP is constant
3: In redun(~., data = dataforredun, nk = 0) :
  Number.of.Throw.Within.Finally.AP is constant
>
> after_redun= dataforredun[,!(names(dataforredun) %in% redun_obj $Out)]
> form=as.formula(paste("post_release_defects>0~",paste(names(after_redun),collapse="+")))
> model=glm(formula=form, data=log10(data+1), family = binomial(link = "logit"))
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
>
> summary(model)
```

```
Call:
glm(formula = form, family = binomial(link = "logit"), data = log10(data +
    1))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8445  -0.1672  -0.1083  -0.0663   3.2624

Coefficients: (3 not defined because of singularities)
                                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                                  -3.8977     0.6662  -5.850 4.90e-09 ***
Number.of.Destructive.Wrapping.AP            -0.3430     1.1654  -0.294  0.76849
Number.of.Dummy.Handler.AP                  -41.4240  1863.9266  -0.022  0.98227
Number.of.Ignoring.Interrupted.Exception.AP   0.4768     1.4155   0.337  0.73623
Number.of.Incomplete.Implementation.AP            NA         NA      NA       NA
Number.of.Log.and.Return.Null.AP            -45.4869  7971.1151  -0.006  0.99545
Number.of.Log.and.Throw.AP                  -46.1832  2933.7313  -0.016  0.98744
Number.of.Mutliline.log.AP                        NA         NA      NA       NA
Number.of.Nested.Try.AP                      -0.6206     0.8293  -0.748  0.45429
Number.of.Relying.on.Get.Cause.AP            -0.2674     1.0851  -0.246  0.80532
Number.of.Throw.Within.Finally.AP                 NA         NA      NA       NA
Number.of.Throws.Generic.AP                   1.4368     0.3001   4.788 1.69e-06 ***
Number.of.Throws.Kitchen.Sink.AP             -0.2445     0.5219  -0.468  0.63945
Code_Ownership_count                          1.4302     0.3347   4.273 1.93e-05 ***
lines_added                                  -0.2614     0.4519  -0.579  0.56290
total_change                                  1.3870     0.4383   3.165  0.00155 **
AvgCyclomatic                                -0.6628     2.2530  -0.294  0.76863
AvgCyclomaticStrict                          -1.4970     2.1279  -0.703  0.48175
AvgEssential                                 -0.8202     1.9210  -0.427  0.66939
AvgLine                                       0.2370     0.6843   0.346  0.72908
AvgLineBlank                                  1.2131     0.7776   1.560  0.11877
CountDeclInstanceVariable                     0.5326     0.2639   2.018  0.04361 *
CountDeclMethod                              -0.5117     0.4064  -1.259  0.20807
CountDeclMethodDefault                       -0.5087     0.2741  -1.856  0.06345 .
CountDeclMethodPrivate                        0.1026     0.3329   0.308  0.75795
CountDeclMethodProtected                     -0.1005     0.2961  -0.339  0.73429
MaxCyclomatic                                 1.3937     1.8355   0.759  0.44769
MaxCyclomaticModified                        -1.1661     1.8568  -0.628  0.52999
RatioCommentToCode                           -4.7369     1.1866  -3.992 6.55e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1295.94  on 5207  degrees of freedom
Residual deviance:  862.23  on 5182  degrees of freedom
AIC: 914.23

Number of Fisher Scoring iterations: 15

> newform= post_release_defects>0~ Number.of.Throws.Generic.AP + Code_Ownership_count + total_change +
CountDeclInstanceVariable + RatioCommentToCode
> newmodel=glm(formula=newform, data=log10(data+1), family = binomial(link = "logit"))
> summary(newmodel)

Call:
glm(formula = newform, family = binomial(link = "logit"), data = log10(data +
    1))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1718  -0.1731  -0.1157  -0.0690   3.1847

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                  -4.7477     0.3394 -13.988  < 2e-16 ***
```

```
Number.of.Throws.Generic.AP   1.0700     0.2465   4.342 1.41e-05 ***
Code_Ownership_count          1.0985     0.2936   3.742 0.000183 ***
total_change                  1.0738     0.1022  10.511  < 2e-16 ***
CountDeclInstanceVariable     0.3022     0.2126   1.422 0.155161    
RatioCommentToCode           -4.3600     1.1124  -3.920 8.87e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1295.94  on 5207  degrees of freedom
Residual deviance:  882.19  on 5202  degrees of freedom
AIC: 894.19

Number of Fisher Scoring iterations: 8

> newform= post_release_defects>0~ Number.of.Throws.Generic.AP + Code_Ownership_count + total_change  + 
RatioCommentToCode
> newmodel=glm(formula=newform, data=log10(data+1), family = binomial(link = "logit"))
> summary(newmodel)

Call:
glm(formula = newform, family = binomial(link = "logit"), data = log10(data + 
    1))

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.1225  -0.1727  -0.1168  -0.0695   3.2283  

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)    
(Intercept)                  -4.5920     0.3204 -14.331  < 2e-16 ***
Number.of.Throws.Generic.AP   1.1220     0.2440   4.597 4.28e-06 ***
Code_Ownership_count          1.2031     0.2835   4.244 2.20e-05 ***
total_change                  1.0898     0.1018  10.702  < 2e-16 ***
RatioCommentToCode           -4.6733     1.0937  -4.273 1.93e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1295.94  on 5207  degrees of freedom
Residual deviance:  884.23  on 5203  degrees of freedom
AIC: 894.23

Number of Fisher Scoring iterations: 8

> 
> 1-884.23/1295.94
[1] 0.3176922
> anova(newmodel)
Analysis of Deviance Table

Model: binomial, link: logit

Response: post_release_defects > 0

Terms added sequentially (first to last)


                            Df Deviance Resid. Df Resid. Dev
NULL                                         5207    1295.94
Number.of.Throws.Generic.AP  1  228.795      5206    1067.14
Code_Ownership_count         1   46.423      5205    1020.72
total_change                 1  110.768      5204     909.95
RatioCommentToCode           1   25.724      5203     884.23
```

```
> testdata=data.frame(Code_Ownership_count =log10(mean(data$Code_Ownership_count)+1),
+ total_change =log10(mean(data$total_change)+1),
+ RatioCommentToCode =log10(mean(data$RatioCommentToCode)+1), Number.of.Throws.Generic.AP
=log10(mean(data$Number.of.Throws.Generic.AP)+1))
> predict(newmodel,testdata, type="response")
         1
0.01463999
> testdata=data.frame(Code_Ownership_count =log10(mean(data$Code_Ownership_count)*2+1),
+ total_change =log10(mean(data$total_change)+1),
+ RatioCommentToCode =log10(mean(data$RatioCommentToCode)+1), Number.of.Throws.Generic.AP
=log10(mean(data$Number.of.Throws.Generic.AP)+1))
> predict(newmodel,testdata, type="response")
         1
0.01971525
> testdata=data.frame(Code_Ownership_count =log10(mean(data$Code_Ownership_count)+1),
+ total_change =log10(mean(data$total_change)*2+1),
+ RatioCommentToCode =log10(mean(data$RatioCommentToCode)+1), Number.of.Throws.Generic.AP
=log10(mean(data$Number.of.Throws.Generic.AP)+1))
> predict(newmodel,testdata, type="response")
         1
0.01982926
> testdata=data.frame(Code_Ownership_count =log10(mean(data$Code_Ownership_count)+1),
+ total_change =log10(mean(data$total_change)+1),
+ RatioCommentToCode =log10(mean(data$RatioCommentToCode)*2+1), Number.of.Throws.Generic.AP
=log10(mean(data$Number.of.Throws.Generic.AP)+1))
> predict(newmodel,testdata, type="response")
         1
0.00558429
> testdata=data.frame(Code_Ownership_count =log10(mean(data$Code_Ownership_count)+1),
+ total_change =log10(mean(data$total_change)+1),
+ RatioCommentToCode =log10(mean(data$RatioCommentToCode)+1), Number.of.Throws.Generic.AP
=log10(mean(data$Number.of.Throws.Generic.AP)*2+1))
> predict(newmodel,testdata, type="response")
         1
0.01769347
>
```