rms (Regression Modeling Strategies) R package Introduction

Pei-Shan Yen

9/30/2020

Purpose: explore and compare the rms package with common modeling function in R

Demonstration version: R (4.0.2)

Author: Pei-Shan Yen, Yi-Fan Chen (Biostatistics Core, Center for Clinical and Translational Science, University of Illinois at Chicago).

Package Introduction

The rms package in R software, originally named 'Design' package, provides a collection of pragmatic functions to construct and evaluate regression models. This package accompanies the book "Regression Modeling Strategies" by Frank Harrell.

This rms package exploration will introduce 1) the function datadist() for summary statistics, 2) the function lrm() for the construction of binary and ordinal logistic regression models, 3) the function ols() for the construction of linear models, and 4) the function xxx() for the construction of cox regression for survival analysis. While there are other functions for performing other regression models, such as quantile regression, they will not be included.

The R document of the rms package. https://www.rdocumentation.org/packages/rms/versions/6.0-1

1. Summary Statistics

1.1 the function datadist() in the rms package

The function datadist() in the rms package is to determine the distribution summaries for the predictor variables in regression models. To demonstrate the use of function datadist(), this exploration will use the resect dataset from Riffenburgh (2006). The dataset includes 134 patients who have undergone resection of tumors in the trachea. The dataset contains 6 variables, defined as follows:

- id = a patient ID,
- age= the patient's age at surgery,
- prior = prior tracheal surgery (1 = yes, 0 = no),
- resection = extent of the resection (in cm),
- intubated = whether intubation was required at the end of surgery (1 = yes, 0 = no), and
- died = the patient's death status (1 = dead, 0 = alive).

```
resect  = read.csv("G:/My Drive/UIC RA CCTS/20200608 Side Project_R rms
package/data/resect.csv")[-1] # Exclude the first variable (Patient ID)
dim(resect) # 134 patients and 5 variables

## [1] 134   5

head(resect, n= 3)

##   age prior resection intubated died
## 1  34     1       2.5         0    0
## 2  57     0       5.0         0    0
## 3  60     1       4.0         1    1
```

The resect data is used to demonstrate the use of the function datadist(). This output specifies the following:

- Low/High effect: The first/third Quantile
- Adjust to: Median
- Low/High: Minimum/Maximum
- Low/High Prediction: The 10th smallest/largest predicted probability
- Values: The level of categorical variables.

```
# install.packages("rms")
library(rms)
# function datadist() in the rms package
data_resect = datadist(resect)
data_resect

##                   age prior resection intubated died
## Low:effect      36.00     0       2.0         0    0
## Adjust to       51.00     0       2.5         0    0
## High:effect     61.00     1       4.0         1    1
## Low:prediction  20.95     0       1.0         0    0
## High:prediction 69.35     1       5.0         1    1
## Low              8.00     0       1.0         0    0
## High            80.00     1       6.0         1    1
##
## Values:
##
## prior : 0 1
## intubated : 0 1
## died : 0 1

# function summary() in the base package
summary(resect)

##       age            prior          resection       intubated
##  Min.   : 8.00   Min.   :0.0000   Min.   :1.000   Min.   :0.0000
##  1st Qu.:36.00   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:0.0000
##  Median :51.00   Median :0.0000   Median :2.500   Median :0.0000
##  Mean   :47.84   Mean   :0.2537   Mean   :2.963   Mean   :0.1418
##  3rd Qu.:61.00   3rd Qu.:0.7500   3rd Qu.:4.000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :1.0000   Max.   :6.000   Max.   :1.0000
```

```
##       died
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1269
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

2. Logistic Regression Model

2.1 the function lrm() in the rms package

Within a logistic regression model, the binary outcome variable Y takes on the value 1 or 0. In the rms package, the function lrm() is used to construct a logistic regression model. This output specifies the following:

- Obs: The total number of observations used to fit the model. Observations are subdivided into groups 0 and 1. The value 0 indicates the outcome "alive" and the value 1 indicates the outcome "died".
- maxmax |deriv| : the maximum absolute value of the derivative at the point where the maximum likelihood function was estimated.
- Model likelihood ratio test: the result of the model compared with the null model
- Discrimination: R^2, g, gr, gp, and Brier.
- Rank Discrimination: C statistic (area under the ROC curve) and Somers' D (D_xy), gamma, and tau-a. To decide the model accuracy, C statistics is are the most commonly used. If the value of the C statistics falls into a) 0.6-0.7, b) 0.7-0.8, c) 0.8-0.9, and d)0.9-1.0, this indicates the model does a a) poor, b) fair, c) good, and d) excellent job at discrimination, respectively.
- A table of coefficients, standard errors, and Wald Z statistics with their p values.

2.2 the demonstration of logistic regression

2.2.1 Using function lrm() to construct a logistic regression for all of the predictors

Using resect data to demonstrate the use of the function lrm(), a multiple logistic regression model was constructed to predict the outcome variable died. The predictors include age, prior, resection, and intubated. Among the four predictors, only the predictors resection and intubated are statistically significant.

```
# function lrm() in the rms package
options(datadist="data_resect") # to store information with fit without accessing the
original dataset

LR_fun_lrm_rms01 = lrm(died ~ age + prior + resection  + intubated,
                       data=resect,
                       x=TRUE, y=TRUE) # x=TRUE, y=TRUE allows use of resid(),
which.influence below
LR_fun_lrm_rms01
```

```
## Logistic Regression Model
##
##  lrm(formula = died ~ age + prior + resection + intubated, data = resect,
##      x = TRUE, y = TRUE)
##
##                         Model Likelihood    Discrimination    Rank Discrim.
##                               Ratio Test           Indexes          Indexes
##  Obs           134    LR chi2      34.58    R2       0.427    C       0.862
##   0            117    d.f.             4    g        1.534    Dxy     0.723
##   1             17    Pr(> chi2) <0.0001    gr       4.637    gamma   0.726
##  max |deriv| 2e-08                          gp       0.164    tau-a   0.161
```

```
##                                            Brier    0.070
##
##            Coef    S.E.   Wald Z Pr(>|Z|)
##  Intercept -5.1529 1.4695 -3.51  0.0005
##  age        0.0012 0.0206  0.06  0.9547
##  prior      0.8147 0.7048  1.16  0.2477
##  resection  0.6122 0.2828  2.16  0.0304
##  intubated  2.8108 0.6584  4.27  <0.0001
##
```

2.2.2 Using function fastbw() to perform model selection

The function fastbw() in the rms package aims to perform backward elimination on predictors. The output produces the deletion statistics for variables, one at a time and in descending order of insignificance. The output also shows the parameter estimates for the final model after deleting variables.

Now, the complete model with 4 predictors is used to demonstrate the use of the function fastbw(). With a cutoff of 0.20, the predictors age and prior are removed from the complete model. The final model only includes the predictors resection and intubated.

```
# Model Selection
# rule: Stopping rule. Defaults to "aic" for Akaike's information criterion. Use rule="p"
to use P-values
# sls: Significance level for staying in a model if rule="p". Default is .05.
fastbw(LR_fun_lrm_rms01, rule="p", sls=0.20)
```

```
##
##  Deleted Chi-Sq d.f. P      Residual d.f. P      AIC
##  age     0.00   1    0.9547 0.00     1    0.9547 -2.00
##  prior   1.33   1    0.2482 1.34     2    0.5126 -2.66
##
## Approximate Estimates after Deleting Factors
##
##              Coef   S.E. Wald Z         P
## Intercept -4.5392 1.0606 -4.280 0.0000187
## resection  0.5355 0.2722  1.967 0.0491694
## intubated  2.8038 0.6546  4.284 0.0000184
##
## Factors in Final Model
##
## [1] resection intubated
```

In the final model, the R^2 of this model is 0.413, and the C statistic is 0.867. For the predictor resection, the parameter estimation is 0.5475 with P-value of 0.0418; and for the predictor intubated, the parameter estimation is 2.8640 with P-Value <0.001.

```
LR_fun_lrm_rms = lrm(died ~ resection  + intubated,
                     data=resect,
                     x=TRUE, y=TRUE)
LR_fun_lrm_rms
```

```
## Logistic Regression Model
##
##  lrm(formula = died ~ resection + intubated, data = resect, x = TRUE,
##      y = TRUE)
##
##                        Model Likelihood    Discrimination    Rank Discrim.
```

```
##                                 Ratio Test            Indexes          Indexes
##  Obs           134    LR chi2      33.27    R2          0.413    C         0.867
##   0            117    d.f.             2    g           1.397    Dxy       0.734
##   1             17    Pr(> chi2) <0.0001    gr          4.043    gamma     0.757
##  max |deriv| 5e-10                          gp          0.160    tau-a     0.164
##                                             Brier       0.073
##
##             Coef    S.E.   Wald Z Pr(>|Z|)
##  Intercept -4.6370 1.0430 -4.45  <0.0001
##  resection  0.5475 0.2689  2.04  0.0418
##  intubated  2.8640 0.6479  4.42  <0.0001
##
```

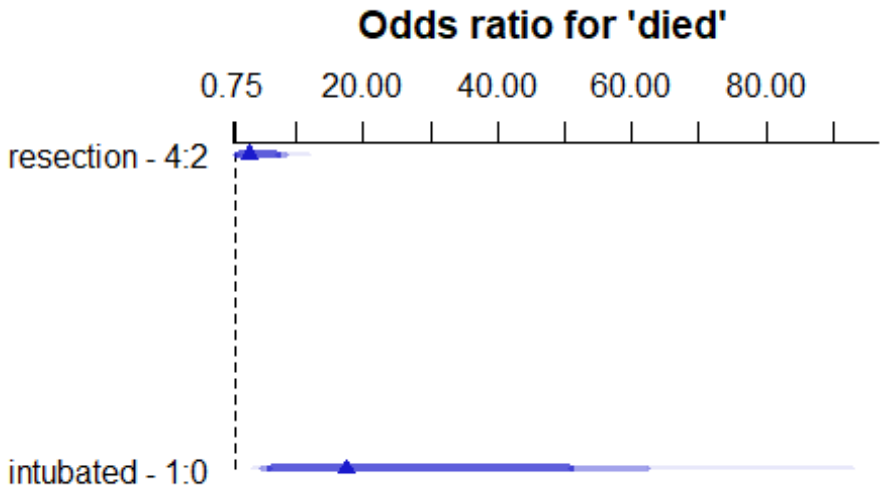
2.2.3 Using function summary() and plot() to demonstrate the odds ration of the predictors

The function summary() and plot() in the base package for lrm() subject produce a more detailed summary of information about the model. The summary result for the function lrm() reveals the odds ratio and its 95% confidence interval for the continuous predictors resection and intubated. The plot for the function lrm() can visualize the odds ratio for the predictors.

```
summary(LR_fun_lrm_rms)
```

```
##              Effects              Response : died
##
##  Factor      Low High Diff. Effect  S.E.    Lower 0.95 Upper 0.95
##  resection   2   4    2      1.0949 0.53783 0.04082     2.1491
##   Odds Ratio 2   4    2      2.9890      NA 1.04170     8.5769
##  intubated   0   1    1      2.8640 0.64790 1.59410     4.1338
##   Odds Ratio 0   1    1     17.5310      NA 4.92390    62.4160
```

```
plot(summary(LR_fun_lrm_rms), main="Odds ratio for 'died'")
```



2.2.4 Using function anova() to perform the Lack of Fit F-test

The function anova() in the stats package is used to evaluate the Lack of Fit F-Test. The final model is compared to the null model (intercept model). The final model is significantly better than the null model (intercept model).

```
anova(LR_fun_lrm_rms) # compare to the null model
```

```
##                 Wald Statistics          Response: died
##
##  Factor     Chi-Square d.f. P
##  resection   4.14      1    0.0418
##  intubated  19.54      1    <.0001
##  TOTAL      25.47      2    <.0001
```

2.2.5 Using function which.influence() and show.influence() to identify the influential points

The function which.influence() in the rms package indicates the influential points in the regression model. We use the cutoff of dfbetas 0.3 to indicate important influential points. In this dataset, patients 84 and 94 are influential points.

```
inf_0.2 = which.influence(fit = LR_fun_lrm_rms, cutoff=0.2)
inf_0.2
```

```
## $Intercept
## [1]  53  84  94 128
##
## $resection
## [1]  55  84  94 128
##
## $intubated
##  [1]  29  42  53  55  73  84  88  94 103 109 128
```

```
inf_0.3 = which.influence(fit = LR_fun_lrm_rms, cutoff=0.3)
inf_0.3
```

```
## $Intercept
## [1] 84 94
##
## $resection
## [1] 84 94
```

```
show.influence(object = inf_0.3, dframe = data.frame(resect))
```

```
##    Count resection
## 84     2        *2
## 94     2        *6
```

2.2.6 Using function Prediction() to perform prediction for new dataset

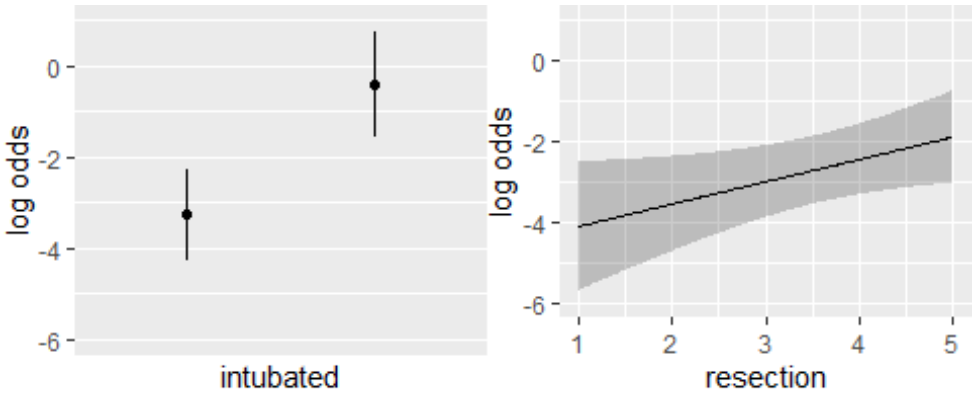
The function Predict() in the rms package shows the prediction result. The ggplot shows the effect of the predictors resection and intubated.

```
head(Predict(LR_fun_lrm_rms))
```

```
##             resection intubated      yhat     lower     upper .predictor.
## resection.1  1.000000         0 -4.089544 -5.667312 -2.511777   resection
## resection.2  1.020101         0 -4.078540 -5.647322 -2.509758   resection
```

```
## resection.3  1.040201         0 -4.067535 -5.627353 -2.507718   resection
## resection.4  1.060302         0 -4.056531 -5.607404 -2.505658   resection
## resection.5  1.080402         0 -4.045526 -5.587476 -2.503577   resection
## resection.6  1.100503         0 -4.034522 -5.567569 -2.501475   resection
##
## Response variable (y): log odds
##
## Limits are 0.95 confidence limits
```

```
library(ggplot2)
ggplot(Predict(LR_fun_lrm_rms))
```

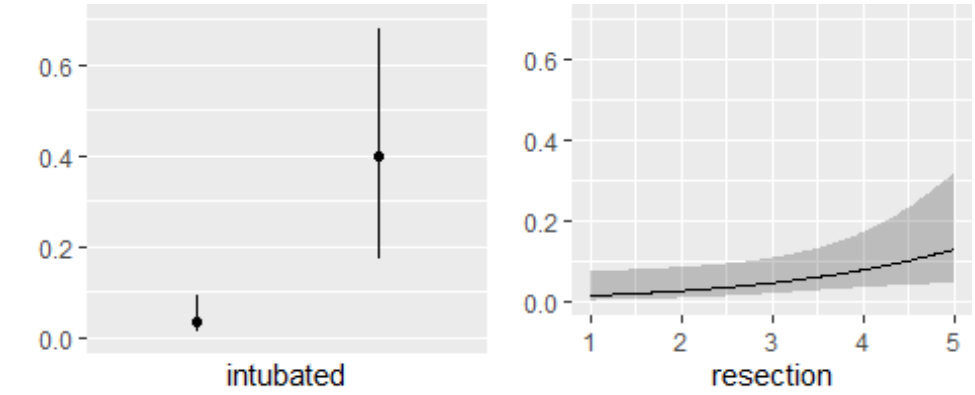


The estimated probability of death can be shown from the function Predict(), the rms package, using the function plogis(), the stats package.

```
head(Predict(LR_fun_lrm_rms, fun = plogis))
```

```
##             resection intubated       yhat       lower      upper .predictor.
## resection.1  1.000000         0 0.01647103 0.003445237 0.07503669   resection
## resection.2  1.020101         0 0.01665025 0.003514554 0.07517696   resection
## resection.3  1.040201         0 0.01683139 0.003585189 0.07531889   resection
## resection.4  1.060302         0 0.01701446 0.003657163 0.07546249   resection
## resection.5  1.080402         0 0.01719949 0.003730499 0.07560781   resection
## resection.6  1.100503         0 0.01738650 0.003805220 0.07575487   resection
##
## Response variable (y):
##
## Limits are 0.95 confidence limits
```

```
ggplot(Predict(LR_fun_lrm_rms, fun = plogis))
```



2.2.7 Using function nomogram() to plot the nomogram

The function nomogram() in the rms package is used to draw the nomogram for the regression fit with a reference line produced from scoring points (default range 0–100). In this nomogram, each predictor is scaled according to the size of its effect on a common scale of 0–100 "points."

A representative observation is shown by the marked points, corresponding to a person of tumor extent of the resection 4.78 cm, was required intubation at the end of surgery. Adding the points associated with each variable value gives the result shown on the scale of total points. For this observation, the result is 72 + 100 = 1, for which the scale of log odds at the bottom gives a predicted logit of 0.84, or a predicted probability of death of 1/(1 + exp(−0.84)) = 0.70.

```
nomogram(fit = LR_fun_lrm_rms, fun=plogis, fun.at=c(0.05, seq(0.1, 0.9, by = 0.1), 0.95),
funlabel="Pr(died)")
```

```
## Points per unit of linear predictor: 34.91661
## Linear predictor units per point   : 0.02863966
##
##
##  resection Points
##  1.0        0
##  1.5       10
##  2.0       19
##  2.5       29
##  3.0       38
##  3.5       48
##  4.0       57
##  4.5       67
##  5.0       76
##  5.5       86
##  6.0       96
##
##
##  intubated Points
##  0           0
##  1         100
##
##
##  Total Points Pr(died)
##            40     0.05
##            66     0.10
##            94     0.20
##           113     0.30
##           129     0.40
##           143     0.50
##           157     0.60
##           172     0.70
##           191     0.80
```

```
# plot(nomogram(fit = LR_fun_lrm_rms, fun=plogis, fun.at=c(0.05, seq(0.1, 0.9, by = 0.1),
0.95),  funlabel="Pr(died)"))
```

2.2.8 Using function validate() to Validate the discrimation index

The function validate() in the rms package to perform resampling validation of a model, with or without backwards step-wise variable selection. The table below includes the results of the model validation using 100 bootstrap replications.

The area under the ROC curve, C statistic, is 0.5 + (D_{xy}/2) = 0.8670 and R^2 = 0.3780.

```
set.seed(20201001)
validate(LR_fun_lrm_rms, B = 100)
```

```
##           index.orig training    test optimism index.corrected   n
## Dxy           0.7340   0.7398  0.7306   0.0092          0.7249 100
## R2            0.4128   0.4415  0.4067   0.0348          0.3780 100
## Intercept     0.0000   0.0000 -0.0150   0.0150         -0.0150 100
## Slope         1.0000   1.0000  0.9510   0.0490          0.9510 100
## Emax          0.0000   0.0000  0.0136   0.0136          0.0136 100
## D             0.2408   0.2620  0.2367   0.0253          0.2156 100
## U            -0.0149  -0.0149  0.0041  -0.0190          0.0041 100
## Q             0.2558   0.2769  0.2326   0.0443          0.2115 100
## B             0.0727   0.0680  0.0756  -0.0076          0.0803 100
## g             1.3970   1.4725  1.3473   0.1253          1.2717 100
## gp            0.1597   0.1606  0.1569   0.0037          0.1560 100
```

2.3 compare function lrm() in the rms package to the function glm() in the stats package

The table below compares the function lrm() in the rms package with the function glm() in the stats package.

Using the lrm() in the rms package to construct a logistic regression provide more detailed information, including the discrimination index, the visualization, model selection, and the validation tool.

The following content demonstrate the use of the function glm() in the stats package.

```
# function glm() in the stats package
LR_fun_glm = glm(died ~ resection  + intubated, data=resect,
family="binomial"(link="logit"))
LR_fun_glm

##
## Call:  glm(formula = died ~ resection + intubated, family = binomial(link = "logit"),
##     data = resect)
##
## Coefficients:
## (Intercept)    resection    intubated
##     -4.6370       0.5475       2.8640
##
## Degrees of Freedom: 133 Total (i.e. Null);  131 Residual
## Null Deviance:       101.9
## Residual Deviance: 68.67     AIC: 74.67

summary(LR_fun_glm)

##
## Call:
## glm(formula = died ~ resection + intubated, family = binomial(link = "logit"),
##     data = resect)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8499  -0.3570  -0.2734  -0.2087   2.6723
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.6370     1.0430  -4.446 8.76e-06 ***
## resection     0.5475     0.2689   2.036   0.0418 *
## intubated     2.8640     0.6479   4.420 9.85e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 101.943  on 133  degrees of freedom
## Residual deviance:  68.669  on 131  degrees of freedom
## AIC: 74.669
##
## Number of Fisher Scoring iterations: 6

# The coefficient of resection has a point estimate of 0.5475 with 95% confidence
interval of (0.0307, 1.1019) after adjusting the predictor intubated
round(confint(LR_fun_glm, level = 0.95),4)
```

```
##                2.5 %  97.5 %
## (Intercept) -6.9456 -2.7946
## resection    0.0307  1.1019
## intubated    1.6288  4.1969
```

```
# To understand the impact of changing a predictor on the odds of the outcome. Estimate
the odds ratio for death associated with a 1 cm increase in resection size is 1.7289,
with a 95% CI of (1.0312, 3.0098) adjusting the predictor intubated.
round(exp(coef(LR_fun_glm)),4)
```

```
## (Intercept)   resection   intubated
##      0.0097      1.7289     17.5309
```

```
round(exp(confint(LR_fun_glm)),4)
```

```
##              2.5 %  97.5 %
## (Intercept) 0.0010  0.0611
## resection   1.0312  3.0098
## intubated   5.0980 66.4771
```

```
# The function anova() in the stats package is used to evaluate the Lack of Fit F-Test.
The final model is compared to the null model (intercept model). The final model is
significantly better than the null model (intercept model).


anova(LR_fun_glm) # compare to the null model
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: died
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                        133    101.943
## resection  1   12.450       132     89.493
## intubated  1   20.823       131     68.669
```

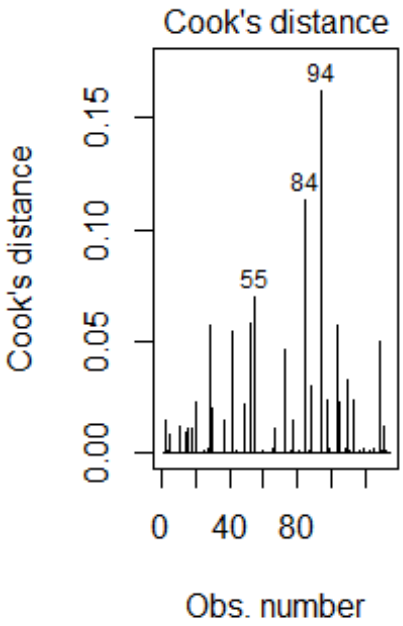
```
pchisq(q = anova(LR_fun_glm)[3,2], df = 133-131, lower.tail = FALSE)
```

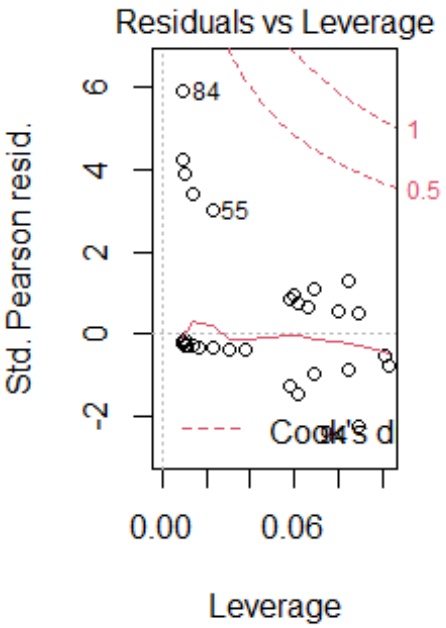
```
## [1] 3.007699e-05
```

```
# predict the outcome
#predict(LR_fun_glm, resect, type="response")[1:5]

#library(tibble)
#predict(LR_fun_glm, newdata = data_frame(resection = c(4,5)))

# Residual plots
# In this case, the highly influential points 84 and 94 fall outside of the Cook's
distance (0.10) contours.
par(mfrow=c(1,2))
plot(LR_fun_glm, which=c(4:5))
```


Cook's distance
Cook's distance
0.00
0.05
0.10
0.15
94
84
55
0
40
80
Obs. number


Residuals vs Leverage
Std. Pearson resid.
-2
0
2
4
6
84
55
1
0.5
Cook's d
0.00
0.06
Leverage