Logistic Regression using R

Dr. James. Ready

April 20, 2020

Logistic regression is a method for fitting a regression curve when the dependent variable is categorical. Logistic regression is similar to discriminant analysis in that a user is examining the differences between two or more groups based on several variables simultaneously. The difference is in its approach to application. Under discriminant analysis, a user is examining potential classification, while under logistic regression, a user is examining the development of a predictive model. Research has shown than logistic regression, which employs maximum likelihood estimation (MLE), outperforms classical discriminant analysis; however, the differences are small (Halperin, Blackwelder, & Verter, 1971; Press & Wilson, 1978).

Logistic regression is a subset of linear modeling called Generalized Linear Modeling (GLM). Thus to perform logistic regression in R, you would use the *glm()* function. While this document explains and illustrates the use of logistic regression in marketing research, the approach described can be applied to any area of research.

Preparatory Steps

Before beginning, three packages need to be loaded into the R environment -

- tidyverse (Wickham, 2019). {tidyverse} is a set of packages that work in harmony because they share common data representations and an Application Program Interface (API) design. For this instruction, only one package in the tidyverse is needed, {readxl}; however, the entire tidyverse can be used in future applications, so load them all at this time.
- rcompanion (Mangiafico, 2020). {rcompanion} is a set of functions developed to support the book *An R Companion for the Handbook of Biological Statistics* (McDonald, 2015), but apply to this illustration.
- car (Fox, 2020). {car} contains a set of functions developed to support the book *An R Companion to Applied Regression* (Fox & Weisberg, 2019), but apply to this illustration.

The following code will check to see if these packages are installed on your computer. If not, the packages will be downloaded from the Comprehensive R Archive Network (CRAN) cloud-based mirror site (https://cloud.r-project.org/) and installed. Once downloaded and installed, the package functions will be loaded into your current R environment for access.

```
if(!require(tidyverse)){
    install.packages("tidyverse")
}
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  3.0.0     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
if(!require(rcompanion)){
    install.packages("rcompanion")
}
```

```
## Loading required package: rcompanion
```

```
if(!require(car)){
    install.packages("car")
}
```

```
## Loading required package: car

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some
```

```
library(tidyverse)
library(rcompanion)
library(car)
```

Logistic Regression

To understand how to perform logistic regression using R, use the Malhotra (2019) Chapter 18.6 data set. First, let us start with a hypothetical research question -

> RQ: Which characteristics of consumer behavior predict loyalty?

Using Null Hypothesis Significance Testing (NHST), a null/alternative hypothesis pairing could be -

- H_0: No characteristics of consumer behavior explain loyalty
- H_A: Some characteristics explain consumer loyalty.

To begin the analysis, read the associated Microsoft Excel file containing the data using the *read_xl* function from the {readxl} package. Once loaded into the R Environment, display some records to confirm the import. The function *file.choose()* allows a user to navigate to where the file was stored when downloaded. The item *sheet = 3* selects the third sheet from the Microsoft Excel file, which contains the data for the illustration. The function *head()* displays the first six records for each variable.

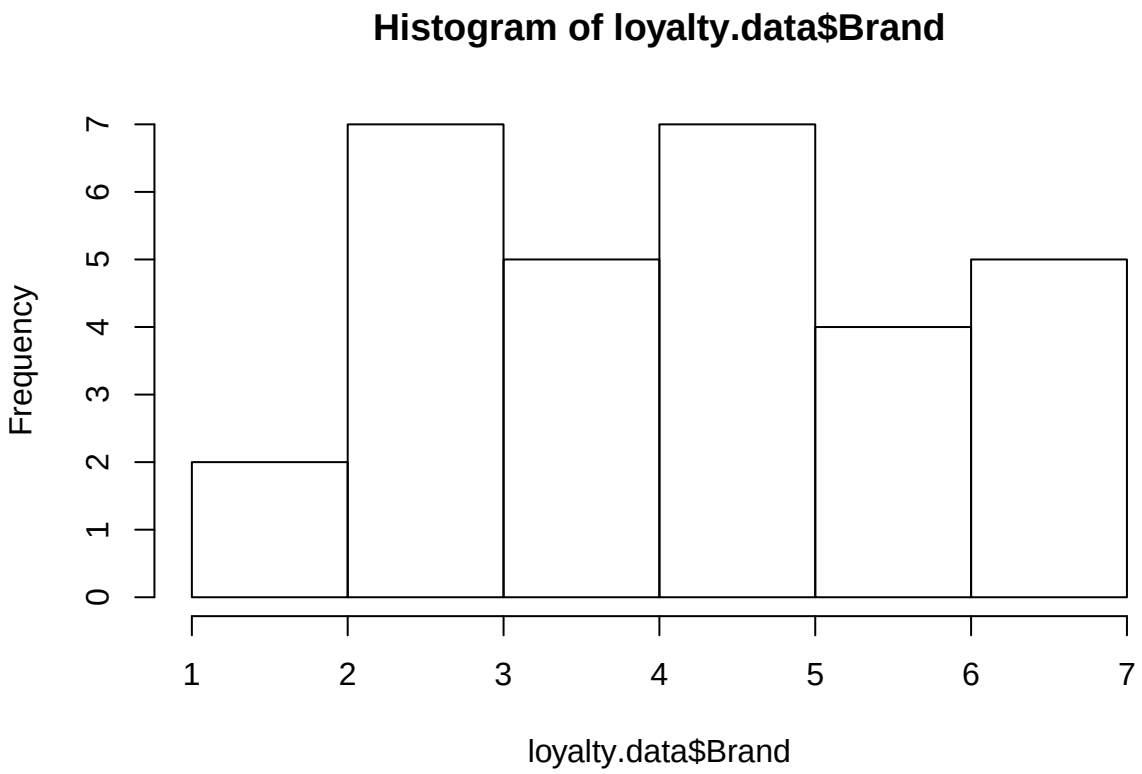
```
# import the training data from Malhotra (2019), Table 18.6 (Information on
# Loyalty) to the data frame "loyalty.data." The data for the illustration can
# be found on the third sheet of the Excel file
loyalty.data <- readxl::read_xls(file.choose(), sheet = 3)
# list the first six records along with the column headings
head(loyalty.data)
```

```
## # A tibble: 6 x 5
##      no Loyalty Brand Product Shopping
##   <dbl>   <dbl> <dbl>   <dbl>    <dbl>
## 1     1       1     4       3        5
## 2     2       1     6       4        4
## 3     3       1     5       2        4
## 4     4       1     7       5        5
## 5     5       1     6       3        4
## 6     6       1     3       4        5
```

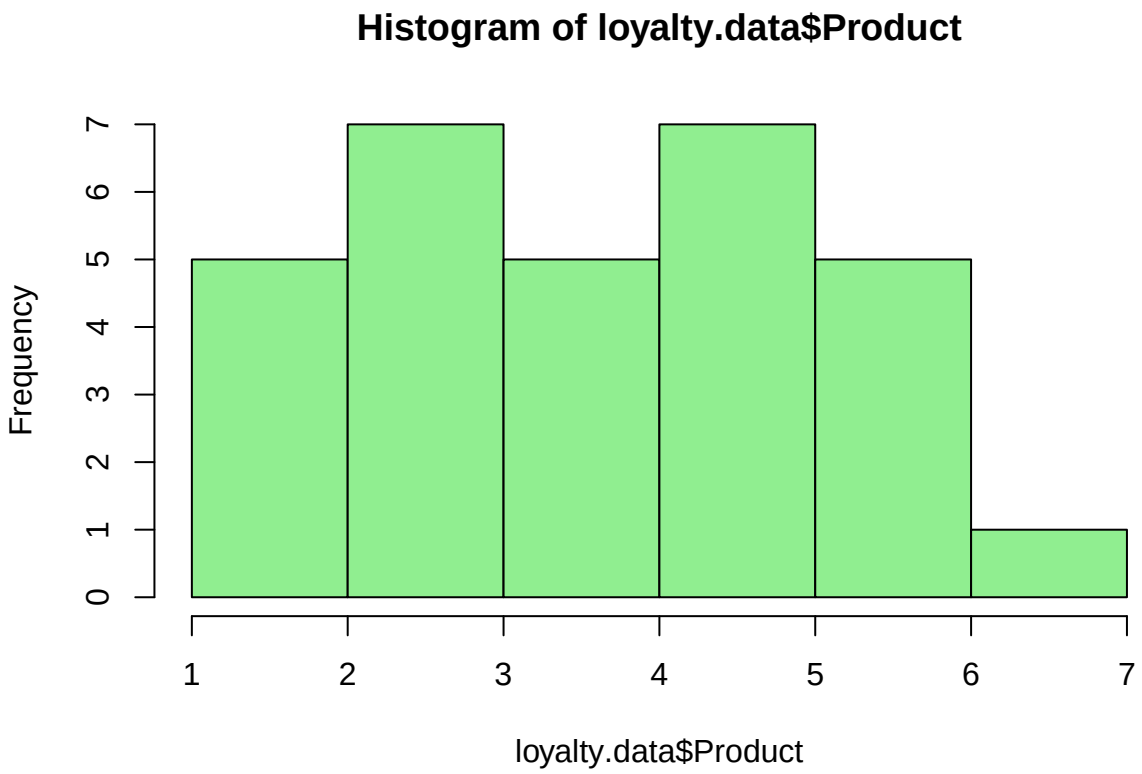
In our example, the dependent variable is Loyalty. Loyalty has two attributes: 1 = Visited, and 2 = Did not visit. The independent variables are Brand, Product, and Shopping (Malhotra, 2019), which are represented by consumer responses on a 7-point Likert scale.

Next, perform an exploratory data analysis of each independent variable using a histogram via the *hist()* function.

```
# the built-in command 'hist' creates a histogram based on the referenced
# variable. the first example displays the Brand variable with default attributes
hist(loyalty.data$Brand)
```




```
# the second example displays the Product variable, and colors the histogram a
# light green
hist(loyalty.data$Product, col = "light green")
```




```r
# the third example displays the Shopping variable, colors the bars in yellow,
# changes the default title of the chart, and labels the x-axis as "Score."
hist(loyalty.data$Shopping,
     col = "yellow",
     main = "Histogram of Shopping",
     xlab = "Score")
```

Construct the Logistic Regression Model

To construct a model, the user must specify the dependent and independent variables. In R, the format is -

$$\text{DV} \sim \text{IV1} + \text{IV2} + \ldots \text{ IV}i$$

Constructing a model requires an understanding of critical elements -

-The item on the left side of "~" is the dependent variable, and the items on the right side of the "~" are the independent variables. A user would place a "+" between the variables, in this case, we're examining individual effect of the independent variables on the dependent variable, and linear models are considered additive. -The "data =" attribute tells the *GLM* function the location of the variables. -Since logistic regression assumes the distribution of the dependent variable follows the binomial distribution function, the *family = binomial* attribute informs the *GLM* function of that assumption.

Once the model is created, the *summary()* function displays information about the model. The *confint()* provides the 95% confidence intervals of the constant (or intercept) and independent variables.

```
# Create the model and store the results to the logR1.model variable. By storing
# the model to a variable, it can be called from memory as long as the R
# environment is active.
logR1.model <- stats::glm(Loyalty ~ Brand + Product + Shopping,
                  data = loyalty.data,
                  family = "binomial"
                  )
summary(logR1.model)
```

```
##
## Call:
## stats::glm(formula = Loyalty ~ Brand + Product + Shopping, family = "binomial",
##     data = loyalty.data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.50745  -0.54518   0.04585   0.58398   1.67736
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -8.6419     3.3456  -2.583  0.00979 **
## Brand         1.2739     0.4789   2.660  0.00782 **
## Product       0.1862     0.3218   0.579  0.56292
## Shopping      0.5900     0.4912   1.201  0.22974
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 41.589  on 29  degrees of freedom
## Residual deviance: 23.471  on 26  degrees of freedom
## AIC: 31.471
##
## Number of Fisher Scoring iterations: 5
```

```
confint(logR1.model)
```

```
## Waiting for profiling to be done...
```

```
##                    2.5 %     97.5 %
## (Intercept) -16.9146286 -3.2190827
## Brand         0.4835753  2.4314888
## Product      -0.4285137  0.8863323
## Shopping     -0.3342646  1.6754398
```

The system-generated output from the *glm* function reports four pieces of information for the model -

- Call, which allows the user to confirm the model was constructed correctly
- Deviance residuals, which is information about the minimum, median, maximum, and first and third quartiles
- Coefficients, which displays the coefficients, standard errors, Z-values, and the statistical significance for each variable
- Null and Residual deviance, which compares the intercept-only (Null) model to the model with all variables included (Residual)
- AIC, which stands for the Akaike Information Criteria, is a maximum likelihood estimation statistic. For further reading on AIC, see Cavanaugh (2011).
- Fisher Scoring, which is the number iterations needed to fit the model.

Based on the output, the formula to predict the binary outcome would be displayed -

$$-8.6419 + (Brand * 1.2739) + (Product * 0.1862) + (Shopping * 0.5900)$$

The individual independent variable coefficients are interpreted as log odds. For example, Brand increases the log odds of predicting Loyalty by 1.2739 (95% CI [0.4836, 2.4315]).

Examining Model Variance

Once the model has been constructed, a user would perform an Analysis of Variance (ANOVA). Normally the built-in *aov* function would suffice -

```
aov(logR1.model)
```

```
## Call:
##    aov(formula = logR1.model)
##
## Terms:
##                     Brand  Product Shopping Residuals
## Sum of Squares  3.316062 0.006786 0.182013  3.995138
## Deg. of Freedom        1        1        1        26
##
## Residual standard error: 0.3919938
## Estimated effects may be unbalanced
```

However, note the statement in the output that the estimated effects may be unbalanced. This situation is a common occurrence. No worries - there is a function to address this phenomenon. The *Anova* function from the {car} package provides the user with three types of variance analyses -

- Balanced number of observations in each group (Type I Sum of Squares), which is the same function as the *aov()* function
- Unbalanced number of observations in each group (Type II Sum of Squares)
- Interaction between variables (Type III Sum of Squares)

For logistic regression with unbalanced effects and no user *a priori* interaction specified, select the Type II test (either "II" or "2") with the **Wald** test statistic -

```
car::Anova(logR1.model, type = "II", test = "Wald")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Loyalty
##          Df  Chisq Pr(>Chisq)
## Brand     1 7.0754   0.007815 **
## Product   1 0.3347   0.562925
## Shopping  1 1.4424   0.229744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output from this test can be interpreted as Brand being statistically significant in the prediction of loyalty, while Product and Shopping are not, based on the deviance of model residuals. This could be reported in APA format as -

An Analysis of Variance (ANOVA) was performed to examine the deviance between the null model and the model with variables. Using the Wald test statistic, only the variable Brand was considered statistically significant,

$$\chi^2(1) = 7.075, p = .008$$

.

Calculating the Model Fit

Due to how a logistic model is constructed, there is no R^2 calculation to assess model fit. However, through the years, attempts have been made to create a *pseudo* R^2. Three types of approaches are usually considered: McFadden (1974), Cox and Snell (1989), and Nagelkerke (1991). The latter's approach was an extension of the Craig and Uhler (1970) model. For a detailed review of each of these approaches and others, see Smith and McKenna (2013).

A user can obtain the three statistics through the *nagelkerke* function from the {rcompanion} package -

```
rcompanion::nagelkerke(logR1.model)
```

```
## $Models
##
## Model: "stats::glm, Loyalty ~ Brand + Product + Shopping, binomial, loyalty.data"
## Null:  "stats::glm, Loyalty ~ 1, binomial, loyalty.data"
##
## $Pseudo.R.squared.for.model.vs.null
##                              Pseudo.R.squared
## McFadden                            0.435632
## Cox and Snell (ML)                  0.453332
## Nagelkerke (Cragg and Uhler)        0.604443
##
## $Likelihood.ratio.test
##  Df.diff LogLik.diff  Chisq    p.value
##       -3     -9.0587 18.117 0.00041599
##
## $Number.of.observations
##
## Model: 30
## Null:  30
##
## $Messages
## [1] "Note: For models fit with REML, these statistics are based on refitting with ML"
##
## $Warnings
## [1] "None"
```

The three pseudo-R^2 values range from .453 to .604, which is a large effect (Cohen, 1992). Another statistic, which is similar to the *pseudo* R^2 is the Likelihood Ratio (LR). The LR test was significant,

$$\chi^2(30) = 18.117, p < .001$$

, which can be interpreted as the three predictors contribute significantly to the model in comparison to the intercept only.

Diagnostic Plots for GLMs

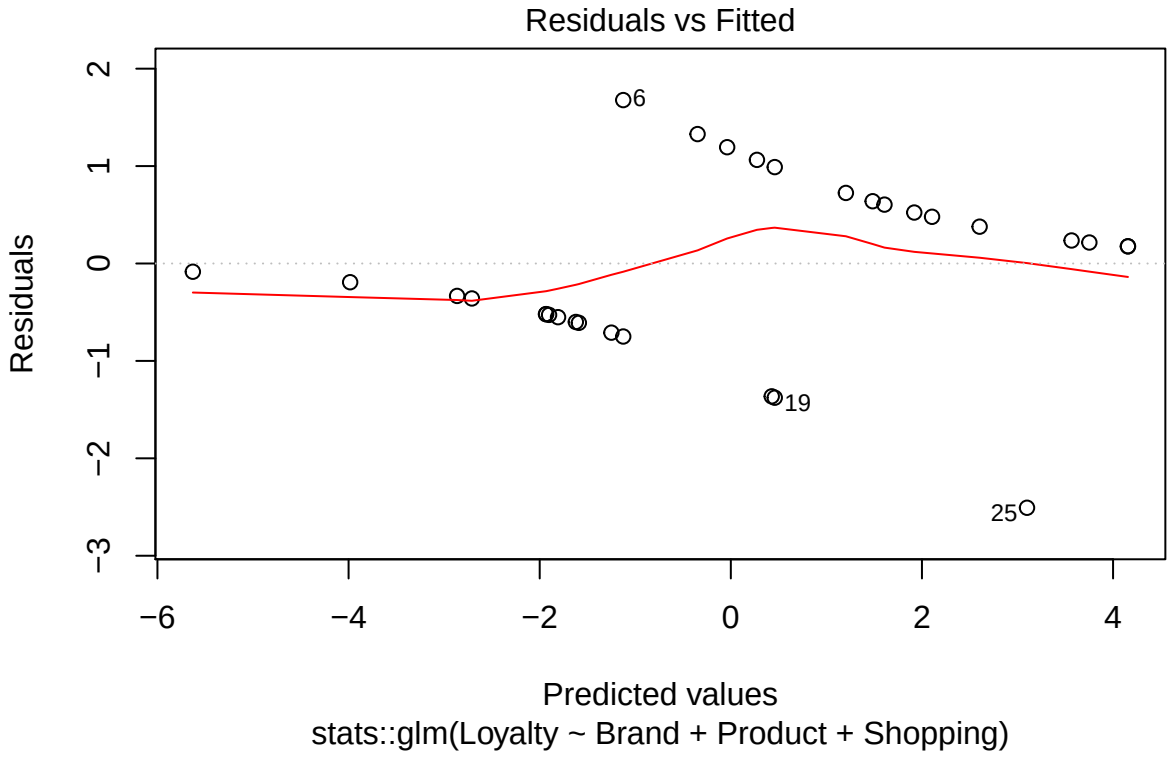
Using the *plot* command, a user can view six common graphs associated with modeling GLMs -

- Residuals vs Fitted (which = 1)
- Q-Q Plot (which = 2)
- Scale-Location (which = 3)
- Cook's Distance (which = 4)
- Residuals vs. Leverage (which = 5)
- Cook's Distance vs. Leverage (which = 6)

Residuals vs. Fitted The first plot listed compares the logistic regression residual errors vs the fitted values. The dotted line displays the y = 0 value. Points above the dotted line are positive residual values, while points below the dotted line are negative residual values. The red line is a smoothed curve to provide the user with a sense of the pattern of residual movement.

Plotting residuals is useful for identifying problems; however, as stated by Fox and Weinberg (2018), "less useful in determining the exact nature of the problem" (p. 388). By selecting "which = 1", a user can obtain the plot of the residuals compared to fitted values -

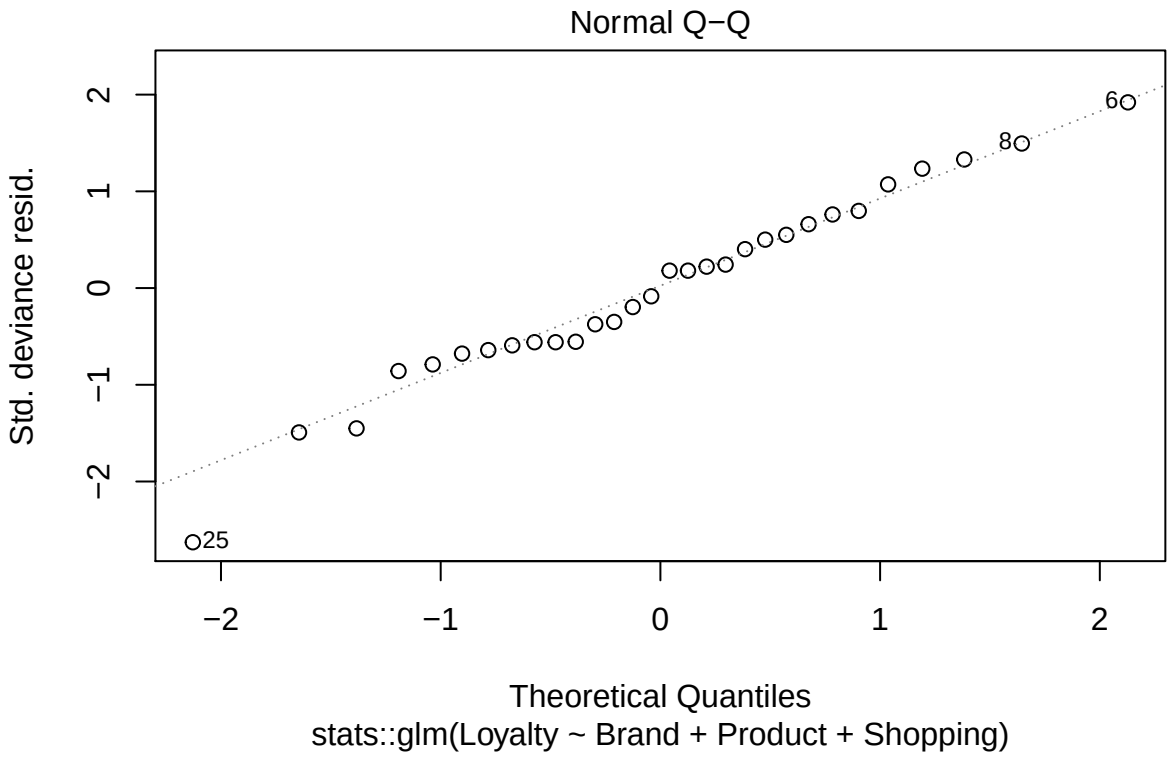
```
plot(logR1.model, which = 1)
```



In this example, one can see how a series of greater positive and negative residuals influence the curve, specifically Records 6, 19, and 25.

Normal Q-Q Plot In this situation, the purpose of a Q-Q plot is to display the model's residuals across a theoretical normal distribution. Points on or near the line are considered normally distributed. By selecting "which = 2", a user can obtain the Normal Q-Q plot -

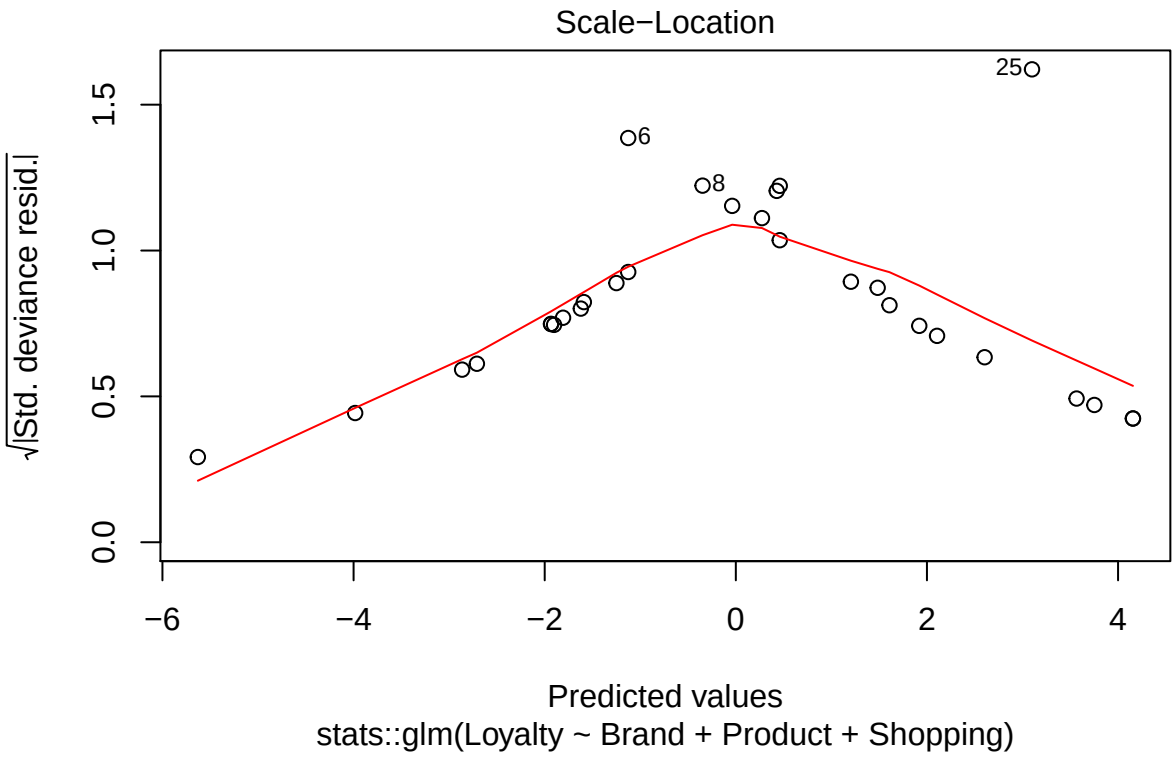
```
plot(logR1.model, which = 2)
```



Note Record 25 and its distance from the line; however, it could be within a 95% confidence interval. For an understanding of how to create a Q-Q plot with confidence intervals, see Kassambara (2020) or Ready (2019).

Scale-Location Plot The scale-location plot indicates the spread of points across predicted value ranges. All forms of regression assume the variance should be equal across the predictor range. Thus, the red line should be nearly horizontal. By selecting "which = 3", a user can obtain the Scale-Location plot -

```
plot(logR1.model, which = 3)
```



In our case, a peak is formed. This peak could be attributed to the leverage of three items (Records 6, 8, and 25), or perhaps some of the independent variables should be transformed.

Cook's Distance Cook (1977, 1979) explored detecting influential observations in regression. Influential observations, or outliers, can negatively influence models and has been the focus of much research. From this work, Cook formulated a distance statistic (Cook's D) based on the least-squares estimate. This distance applies to not only linear but logistic regression.

By selecting "which = 4", a user can obtain not only the Cook's distance for each record, but a plot of the Cook's distance -

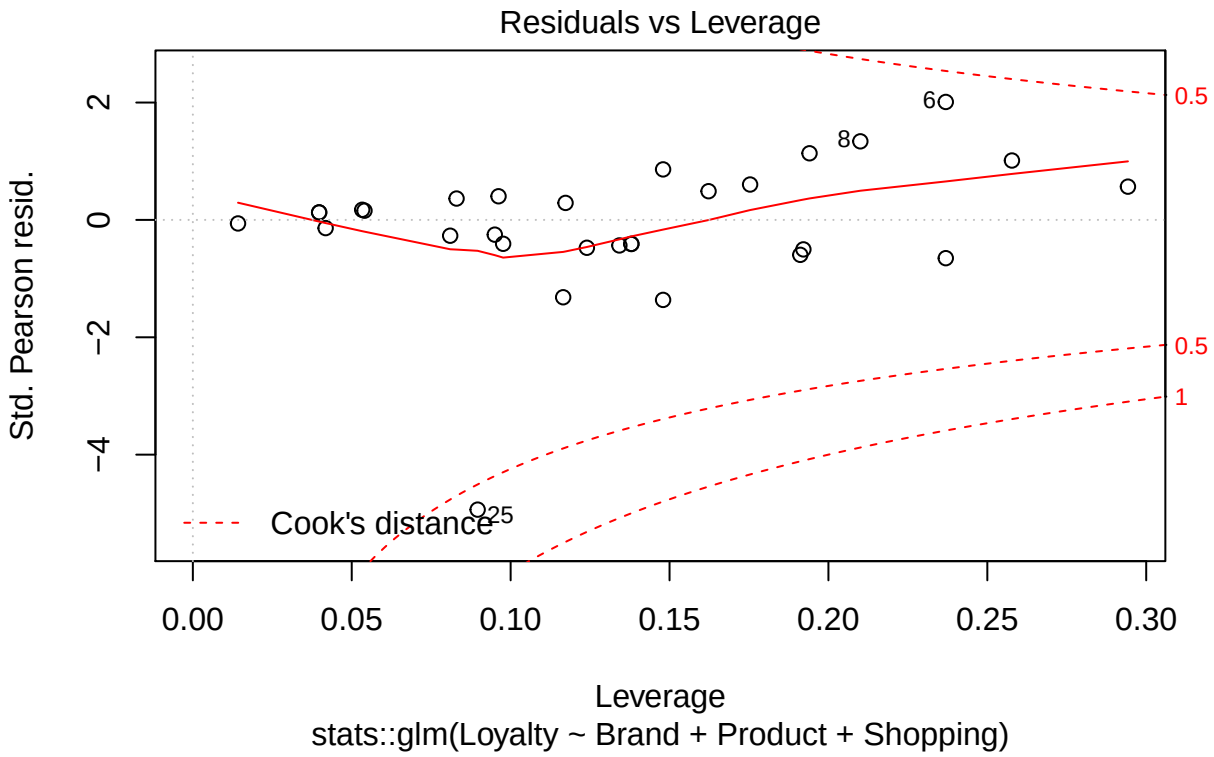
```
plot(logR1.model, which = 4)
```

While the Cook's distance for Records 6 and 8 is large, the distance for Record 25 is larger. Move to the Residual vs. Leverage section for an additional discussion.

Residual vs. Leverage The next plot incorporates the analysis of residual, leverage, influence, and Cook's Distance. Leverage is defined as the difference between the predictor variable and the mean predictor variable. Influence is defined as the contribution of each data point to the determination of the least-squares estimate of the parameter vector (Cook, 1977).

By selecting "which = 5", a user can obtain a plot of the Cook's distance -

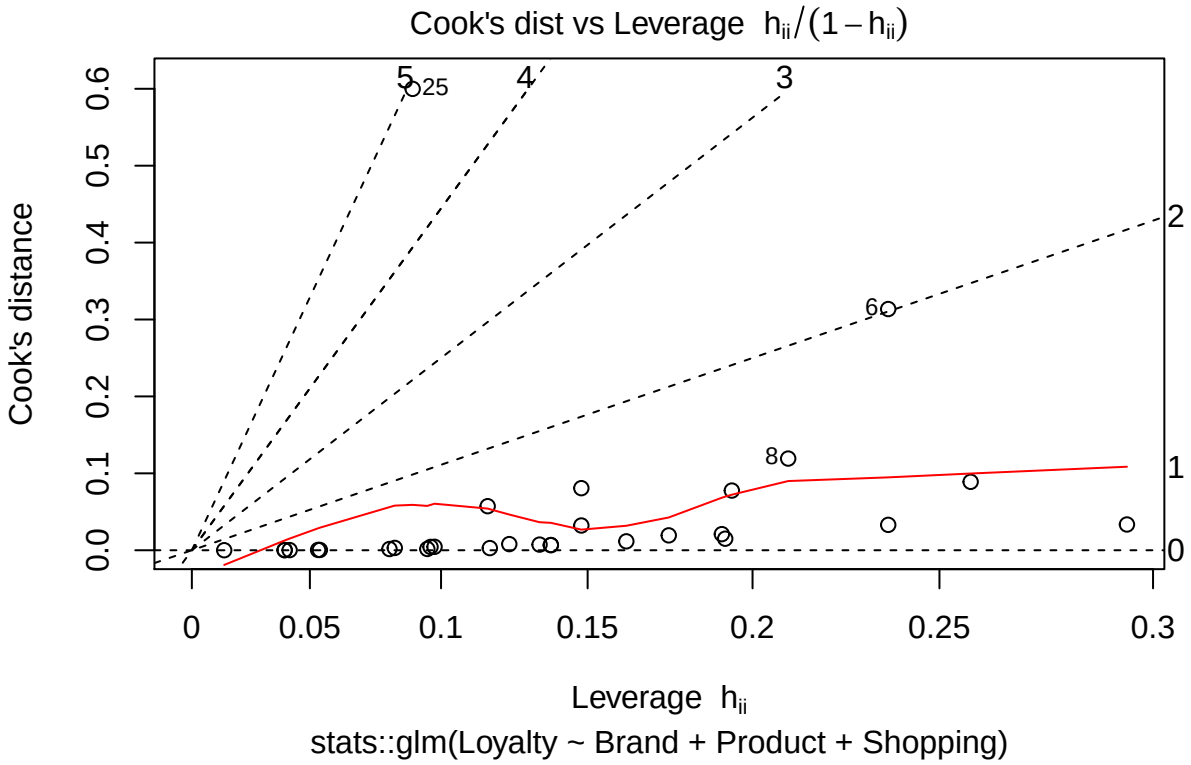
```
plot(logR1.model, which = 5)
```



As reported before, Record 25 is outside Cook's distance. A researcher would use this information to consider (a) removing record 25 and rerunning the model, (b) transform the independent variable(s) to eliminate the outlier, or (c) retain the abnormal record and explore its existence.

Cook's Distance vs. Leverage The final diagnostic plot displays Cook's *D' compared to variable leverage. A rule of thumb is if Cook's D > 1, it exerts too much leverage. By selecting "which $= 6$", a user can obtain the Cook's distance vs. Leverage plot.

```
plot(logR1.model, which = 6)
```



Note that both Records 6 and 25 appear to exert excessive leverage. A common approach would be to remove Record 25 from the model, and reconstruct the model.

References:

Cavanaugh, J. E. (2011). Akaike information criterion. In N. J. Salkind (Ed.), *Encyclopedia of measurement and statistics* (pp. 16-17). SAGE Publications.

Cook, R. D. (1977). Detection of influential observation in linear regression. *Technometrics. 19*(1), 15-18. https://doi.org/10.1080/00401706.1977.10489493

Cox, D. R., & Snell, E. J. (1989). *The analysis of binary data* (2nd Ed.). Erlbaum.

Cragg, S. G., & Uhler, R. (1970). The demand for automobiles. *Canadian Journal of Economics, 3*(3), 386-406. https://doi.org/10.2307/133656

Fox, J. (2020, March 10). *car: Companion to applied regression.* https://cran.r-project.org/web/packages/car/car.pdf

Fox, J., & Weisberg, S. (2019). *An R companion to applied regression* (3rd Ed.). SAGE Publications.

Halperin, M., Blackwelder, W. C., & Verter, J. I. (1971). Estimation of the multivariate logistic risk function: a comparison of the discriminant function and maximum likelihood approaches. *Journal of Chronic Diseases, 24*(2-3), 125-158. https://doi.org/10.1016/0021-9681(71)90106-8

Kassambara, A. (2020, February 13). ggpubr: 'ggplot2' based publication ready plots. https://cran.r-project.org/web/packages/ggpubr/index.html

Malhotra, N. K. (2019). *Marketing research: An applied orientation* (7th ed.). Pearson.

Mangiafico, S. (2020, February 9). *rcompanion: Functions to support extension education program evaluation.* https://cran.r-project.org/web/packages/rcompanion/rcompanion.pdf

McDonald, J. H. (2015). *Handbook of biological statistics* (3rd ed.). Sparky House Publishing.

McFadden, D. (1974). Conditional logit analysis of qualitative choice behavior. In P. Zarembka (Ed.), *Frontiers of econometrics* (pp. 104-142). Academic Press.

Nagelkerke, N. (1991). A note on a general definition of the coefficient of determination. *Biometrika, 78*, 691-692.

Press, S. J., & Wilson, S. (1978). Choosing between logistic regression and discriminant analysis. *Journal of the American Statistical Association, 73*(364), 699-705. https://doi.org/10.1080/01621459.1978.10480080

Ready, J. (2019, December 23). *Testing for Normality - Performing Tests with R and R Commander*. https://online.columbiasouthern.edu/webapps/discussionboard/do/message?action=list_messages&course_id=_29920_1&nav=discussion_board&conf_id=_32807_1&forum_id=_759918_1&message_id=_17623761_1

Smith, T. J., & McKenna, C. M. (2013). A comparison of logistic regression pseudo R^2 indices. *Multiple Linear Regression Viewpoints, 39*(2), 17-26. https://www.glmj.org/archives/articles/Smith_v39n2.pdf

Wickham, H. (2019, November 21). *tidyverse: Easily install and load the 'Tidyverse'*. https://cran.r-project.org/web/packages/tidyverse/tidyverse.pdf