Logistic Regression Tutorial

In this tutorial, we will learn how to fit a (binary) logistic regression model in R. Logistic regression is used when the response variable is binary. We model the log odds of “success” as a linear combination of coefficients and predictors:

$$\log(\frac{\pi}{1-\pi}) = \beta_0 + \beta_1 x_1 + \cdots \beta_k x_k.$$

We have a dataset, `students.txt`, that contains information on about 250 college students at a large public university and their study and party habits. The variables are:

- `Gender`: gender of student
- `Smoke`: whether the student smokes
- `Marijuan`: whether the student uses marijuana
- `DrivDrnk`: whether the student has driven while drunk
- `GPA`: student’s GPA
- `PartyNum`: number of times the student parties in a month
- `DaysBeer`: number of days the student drinks at least 2 beers in a month
- `StudyHrs`: number of hours the students studies in a week

Suppose we want to relate the likelihood of a student driving while drunk with the other variables. Notice that the response variable, `DrivDrnk` is a binary variable with yes or no as levels. When the response variable is binary and not quantitative, we have to use logistic regression instead of linear regression.

Let us read the data in:

```r
library(tidyverse)
Data<-read.table("students.txt", header=T, sep="")
```

We are going to perform some basic data wrangling for our dataframe:

- Remove the first column, as it is just an index.
- Keep observations that have no missing values in any variable. There are about a dozen observations with missing values in at least one variable.
- Apply `factor()` to categorical variables. As a reminder, this should be done to categorical variables if you want to change the reference class.

```r
##first column is index, remove it
Data<-Data[,-1]
##some NAs in data. Remove them
Data<-Data[complete.cases(Data),]

##convert categorical to factors. needed for contrasts
Data$Gender<-factor(Data$Gender)
Data$Smoke<-factor(Data$Smoke)
Data$Marijuan<-factor(Data$Marijuan)
Data$DrivDrnk<-factor(Data$DrivDrnk)
```

We are going to split the dataset into equal sets: one a training set, and another a test set. Recall that the training set is used to build the model, and the test set is used to assess how the model performs on new observations. We use the `set.seed()` function so that we can replicate the same split each time this block of code is run. An integer needs to be supplied to the function.

```r
##set seed so results (split) are reproducible
set.seed(6021)

##evenly split data into train and test sets
sample.data<-sample.int(nrow(Data), floor(.50*nrow(Data)), replace = F)
train<-Data[sample.data, ]
test<-Data[-sample.data, ]
```

1 Visualizations with Logistic Regression

Given that the response variable, `DrivDrnk`, is categorical, we use slightly different visualizations than with linear regression.

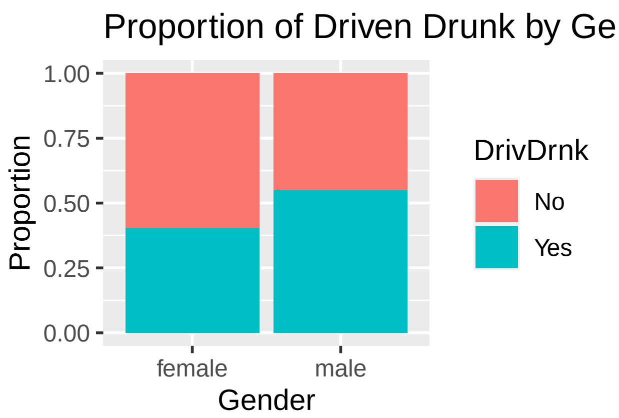
1.1 Barcharts

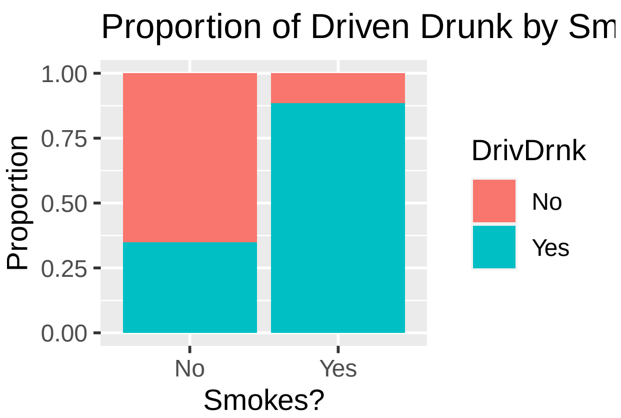
Barcharts are useful to visualize how categorical predictors may be related to the categorical response variable. Since we have three categorical predictors, `Gender`, `Smoke`, and `Marijuan`, we will be creating three barcharts:

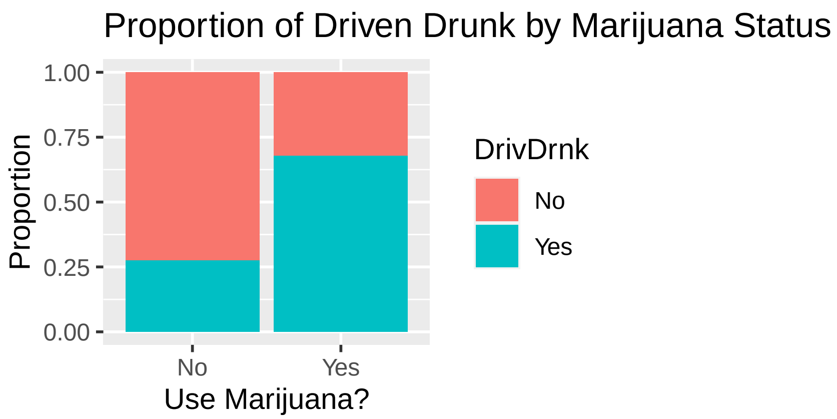
```r
chart1<-ggplot2::ggplot(train, aes(x=Gender, fill=DrivDrnk))+
  geom_bar(position = "fill")+
  labs(x="Gender", y="Proportion",
       title="Proportion of Driven Drunk by Gender")

chart2<-ggplot2::ggplot(train, aes(x=Smoke, fill=DrivDrnk))+
  geom_bar(position = "fill")+
  labs(x="Smokes?", y="Proportion",
       title="Proportion of Driven Drunk by Smoking Status")

chart3<-ggplot2::ggplot(train, aes(x=Marijuan, fill=DrivDrnk))+
  geom_bar(position = "fill")+
  labs(x="Use Marijuana?", y="Proportion",
       title="Proportion of Driven Drunk by Marijuana Status")
```

Instead of displaying these barcharts individually, we can display them simultaneously, using the `grid.arrange()` function from the `gridExtra` package. We will display these 3 barcharts in a 2 by 2 matrix:

```r
##put barcharts in a matrix
library(gridExtra)
gridExtra::grid.arrange(chart1, chart2, chart3, ncol = 2, nrow = 2)
```







- We can see that a slightly higher proportion of male students have driven drunk, compared to female students.
- There is a much higher proportion of smokers who have driven drunk, compared to non smokers.
- Similarly, a higher proportion of students who use marijuana have driven drunk, compared to non users.
- Each of these categorical predictors have some relationship with whether the student has driven drunk.

1.2 Two way tables

We can create two way tables to summarize the relationship between each categorical predictor and whether students have driven drunk:

```
##two way tables of counts
table(train$Gender, train$DrivDrnk)
```

```
##
##          No Yes
##   female 40  27
##   male   23  28
```

```
table(train$Smoke, train$DrivDrnk)
```

```
##
##       No Yes
##   No   60  32
##   Yes   3  23
```

```
table(train$Marijuan, train$DrivDrnk)
```

```
##
```

```
##       No Yes
##   No  45  17
##   Yes 18  38
```

Or we can display the tables via proportions instead of counts:

```
##two way tables using proportions
prop.table(table(train$Gender, train$DrivDrnk),1)
```

```
## 
##                  No       Yes
##   female 0.5970149 0.4029851
##   male   0.4509804 0.5490196
```

```
prop.table(table(train$Smoke, train$DrivDrnk),1)
```

```
## 
##               No       Yes
##   No  0.6521739 0.3478261
##   Yes 0.1153846 0.8846154
```

```
prop.table(table(train$Marijuan, train$DrivDrnk),1)
```

```
## 
##               No       Yes
##   No  0.7258065 0.2741935
##   Yes 0.3214286 0.6785714
```

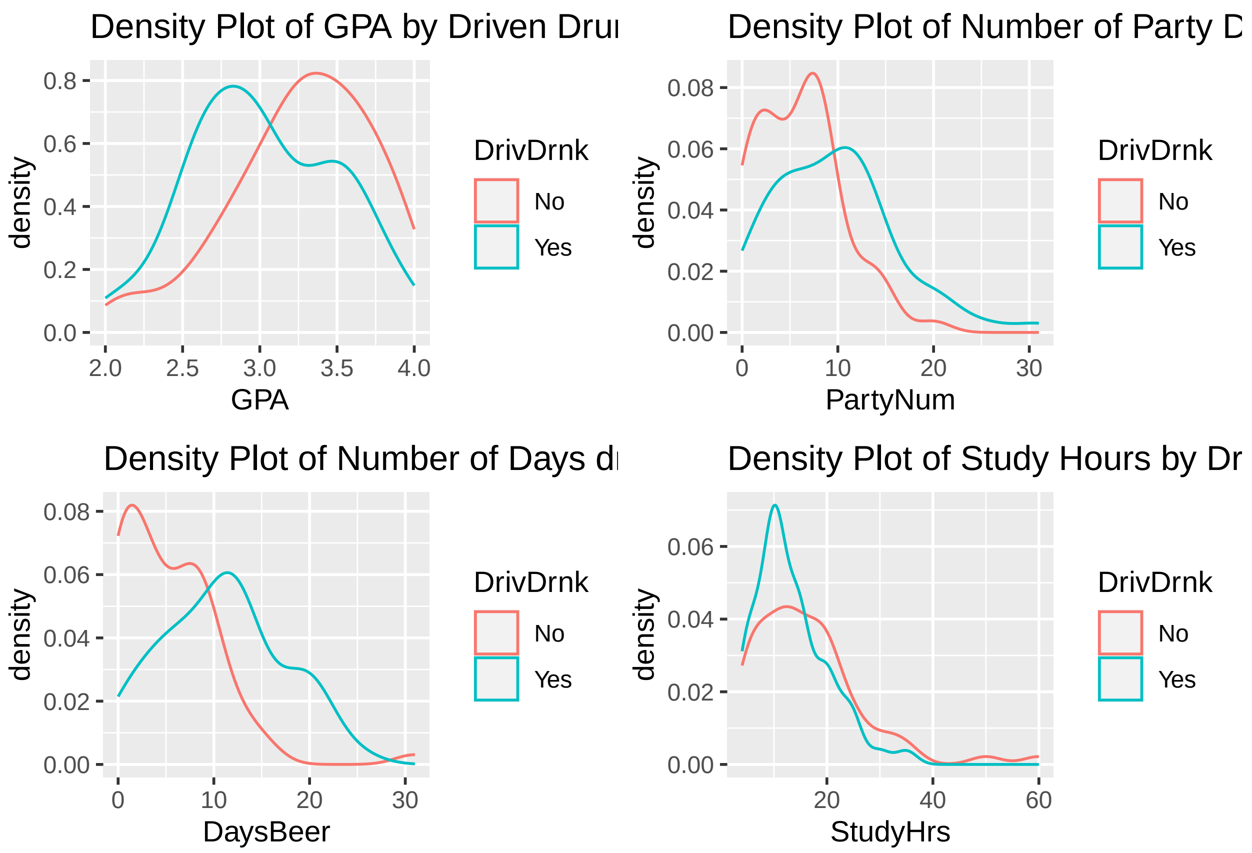
- About 55% of male students have driven drunk, compared to about 40% of female students.
- About 88% of smokers have driven drunk, compared to 35% of non smokers.
- About 68% of marijuana users have driven drunk, compared to 27% of non users.

The earlier barcharts are the visualizations of these tables.

1.3 Quantitative predictors

To see how the quantitative predictors, `GPA`, `PartyNum`, `DaysBeer`, and `StudyHrs` may differ between those who have driven drunk and those who have not, we can compare the distributions of these quantitative variables between the two groups:

```
dp1<-ggplot2::ggplot(train,aes(x=GPA, color=DrivDrnk))+
  geom_density()+
  labs(title="Density Plot of GPA by Driven Drunk")

dp2<-ggplot2::ggplot(train,aes(x=PartyNum, color=DrivDrnk))+
  geom_density()+
  labs(title="Density Plot of Number of Party Days by Driven Drunk")

dp3<-ggplot2::ggplot(train,aes(x=DaysBeer, color=DrivDrnk))+
  geom_density()+
  labs(title="Density Plot of Number of Days drank Beer by Driven Drunk")

dp4<-ggplot2::ggplot(train,aes(x=StudyHrs, color=DrivDrnk))+
  geom_density()+
  labs(title="Density Plot of Study Hours by Driven Drunk")

gridExtra::grid.arrange(dp1, dp2, dp3, dp4, ncol = 2, nrow = 2)
```



- Among those who have driven drunk, we see a higher proportion of students lower GPAs (below 3.0). For those who have not driven drunk, a higher proportion of students have higher GPAs (above 3.0).
- A higher proportion of students who have not driven drunk party less than 10 days a month, compared to students who have driven drunk.
- Similarly, a higher proportion of students who have not driven drunk spend less than 10 days drinking beer, compared to students who have driven drunk.
- A much high proportion of students who have driven drunk spend less than 15 hours studying a week, compared to students who have not drive drunk.
- Each of these quantitative predictors may be related to whether students have driven drunk. It looks like lower GPAs, partying more, drinking more, and studying less is associated with increased likelihood of having driven drunk.

1.4 Correlations between quantitative predictors

We can quickly check the correlations between the quantitative predictors:

```
##correlations between quantitative predictors
round(cor(train[,5:8]),3)
```

```
##              GPA PartyNum DaysBeer StudyHrs
## GPA        1.000   -0.259   -0.361    0.207
## PartyNum  -0.259    1.000    0.773   -0.014
## DaysBeer  -0.361    0.773    1.000   -0.186
## StudyHrs   0.207   -0.014   -0.186    1.000
```

Notice that `DaysBeer` and `PartyNum` are highly correlated. Probably not surprising since drinking is probably done at parties, so a student who parties more is likely to drink more.

2 Fit Logistic Regression

Based on the visualizations, we suspect that all the predictors may influence the response variable. We use the `glm()` function to fit logistic regression:

```
##fit logistic regression
result<-glm(DrivDrnk~., family=binomial, data=train)
```

Notice we specify the argument `family = "binomial"`. This has to be specified for a logistic regression. If this is not specified, a linear regression is fitted instead. The function `glm()` uses maximum likelihood estimation whereas `lm()` uses ordinary least squares. We can look at the Wald tests and deviance of our model using `summary()`:

```
summary(result)
```

```
##
## Call:
## glm(formula = DrivDrnk ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4529  -0.7418  -0.4578   0.6672   2.0707
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.93400    1.94169  -0.996 0.319231
## Gendermale   0.20055    0.47903   0.419 0.675462
## SmokeYes     2.36384    0.71312   3.315 0.000917 ***
## MarijuanYes  0.94538    0.49745   1.900 0.057374 .
## GPA          0.08094    0.55353   0.146 0.883751
## PartyNum    -0.02539    0.07337  -0.346 0.729282
## DaysBeer     0.13664    0.06921   1.974 0.048346 *
## StudyHrs    -0.02221    0.03065  -0.724 0.468764
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 163.04  on 117  degrees of freedom
## Residual deviance: 114.52  on 110  degrees of freedom
## AIC: 130.52
##
## Number of Fisher Scoring iterations: 5
```

Notice a few predictors have highly insignificant coefficients: `Gender`, `GPA`, `PartyNum`, and `StudyHrs`. `PartyNum` being insignificant is not surprising since we noted it is highly correlated with `DaysBeer`. We can also assess variance inflation factors (VIFs) in logistic regression like before:

```
library(faraway)
faraway::vif(result)
```

```
##  Gendermale    SmokeYes MarijuanYes         GPA    PartyNum    DaysBeer
##    6.644806   10.308653    7.281197    8.420435   20.957260   23.110089
##    StudyHrs
##    9.156448
```

Not surprisingly, we have evidence of multicollinearity.

2.1 Likelihood ratio tests

Let us see if we can drop `Gender`, `GPA`, `PartyNum`, and `StudyHrs` from the model, via a likelihood ratio test (LRT). The ΔG^2 test statistic is found by finding the difference in the deviances of the two models. The deviance can be found by extracting the component `deviance` from an object created by `glm()`:

```
reduced<-glm(DrivDrnk~Smoke+Marijuan+DaysBeer, family=binomial, data=train)

##test to compare reduced and full model
##test stat
TS<-reduced$deviance-result$deviance
TS
```

```
## [1] 1.007939
```

```
##pvalue
1-pchisq(TS,4)
```

```
## [1] 0.9085899
```

```
##critical value
qchisq(1-0.05,4)
```

```
## [1] 9.487729
```

- The null hypothesis supports dropping the 4 predictors, and the alternative hypothesis supports not dropping the 4 predictors.
- The ΔG^2 test statistic is 1.008, and is compared with a χ^2_4 distribution, i.e. a chi-squared distribution with 4 degrees of freedom. The degrees of freedom is equal to the number of terms we are dropping.
- The p-value is 0.9086, and the critical value us 9.4877.
- So we fail to reject the null hypothesis. Data do not support using the full model with all the predictors, so we drop `Gender`, `GPA`, `PartyNum`, and `StudyHrs` to use the reduced model.

Let us take a look at the estimated coefficients for this model:

```
summary(reduced)
```

```
##
## Call:
## glm(formula = DrivDrnk ~ Smoke + Marijuan + DaysBeer, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3922  -0.7793  -0.4980   0.7128   2.0730
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.02459    0.43398  -4.665 3.08e-06 ***
## SmokeYes     2.39376    0.68727   3.483 0.000496 ***
## MarijuanYes  0.95839    0.47955   1.999 0.045659 *
## DaysBeer     0.12480    0.04353   2.867 0.004143 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 163.04  on 117  degrees of freedom
## Residual deviance: 115.53  on 114  degrees of freedom
## AIC: 123.53
##
## Number of Fisher Scoring iterations: 5
```

So our logistic regression equation is

$$\log\left(\frac{\hat{\pi}}{1-\hat{\pi}}\right) = -2.0246 + 2.3938I_1 + 0.9584I_2 + 0.1248DaysBeer,$$

where $I_1 = 1$ if the student smokes, and $I_2 = 1$ if the student uses marijuana.

- Given that these coefficients are positive, smoking, using marijuana, and drinking on more days are associated with higher likelihood of having driven drunk.
- The odds of driving drunk for smokers is $\exp(2.3938) = 10.955$ times the odds for non smokers, when controlling for marijuana use and days drinking.
- The odds of driving drunk for marijuana users is $\exp(0.9584) = 2.6075$ times the odds for non users, when controlling for smoking and days drinking.
- The odds of driving drunk is multiplied by a factor of $\exp(0.1248) = 1.1329$ for each additional day of drinking, when controlling for smoking and marijuana use.

2.2 Predicted log odds and probabilities

We can use the `predict()` function to calculate predicted log odds for our test data, using the reduced model:

```
##predicted log odds for test data
logodds<-predict(reduced,newdata=test)
head(logodds)
```

```
##         2         3         5         6         7         9
## -2.024593 -1.525404  3.199523  1.951552 -2.024593  1.429744
```

To find probabilities instead, supply `type="response"` within the `predict()` function:

```
##predicted probabilities for test data
preds<-predict(reduced,newdata=test, type="response")
head(preds)
```

```
##         2         3         5         6         7         9
## 0.1166449 0.1786671 0.9608163 0.8756158 0.1166449 0.8068613
```

So for observation index 2, the student's predicted log odds of driving drunk is -2.024593, with corresponding predicted probability of 0.1166449.