Logistic Regression Tutorial

We use **logistic regression** to predict the probability of a *categorical* dependent variable (with 2 values, usually 0 and 1), with some other *continuous* independent variable(s). That is, the logistic regression model predicts P(Y=1) as a function of X.

Look at the data

To get started, let’s load in a dataset (info about the variables can be found here (http://stanford.edu/class/psych252/data/)):

```
d0 = read.csv("http://www.stanford.edu/class/psych252/data/hw2data.csv")
summary(d0)
```

```
##       Type         Pasthapp      Responsible   Futurehapp
##  Min.   :1.00   Min.   : 0.00   Min.   : 0   Min.   : 0.00
##  1st Qu.:1.00   1st Qu.: 2.00   1st Qu.: 5   1st Qu.: 2.00
##  Median :2.00   Median : 4.00   Median :14   Median : 5.00
##  Mean   :1.94   Mean   : 4.78   Mean   :11   Mean   : 4.25
##  3rd Qu.:3.00   3rd Qu.: 7.00   3rd Qu.:16   3rd Qu.: 5.00
##  Max.   :3.00   Max.   :15.00   Max.   :20   Max.   :15.00
##       FTP          complain
##  Min.   : 3.0   Min.   :0.000
##  1st Qu.:10.0   1st Qu.:0.000
##  Median :13.0   Median :0.000
##  Mean   :12.4   Mean   :0.476
##  3rd Qu.:15.0   3rd Qu.:1.000
##  Max.   :19.0   Max.   :1.000
```

```
# factor d0$Type (i.e., memory groups), since it is categorical
d0$Type = factor(d0$Type)
head(d0)
```

```
##   Type Pasthapp Responsible Futurehapp FTP complain
## 1    2        2          10          5  12        1
## 2    3       10           5          5  18        1
## 3    1        2          18          3   9        0
## 4    3        5          15          0  15        1
## 5    1        7          17          7  17        1
## 6    1        0          20          5  15        0
```

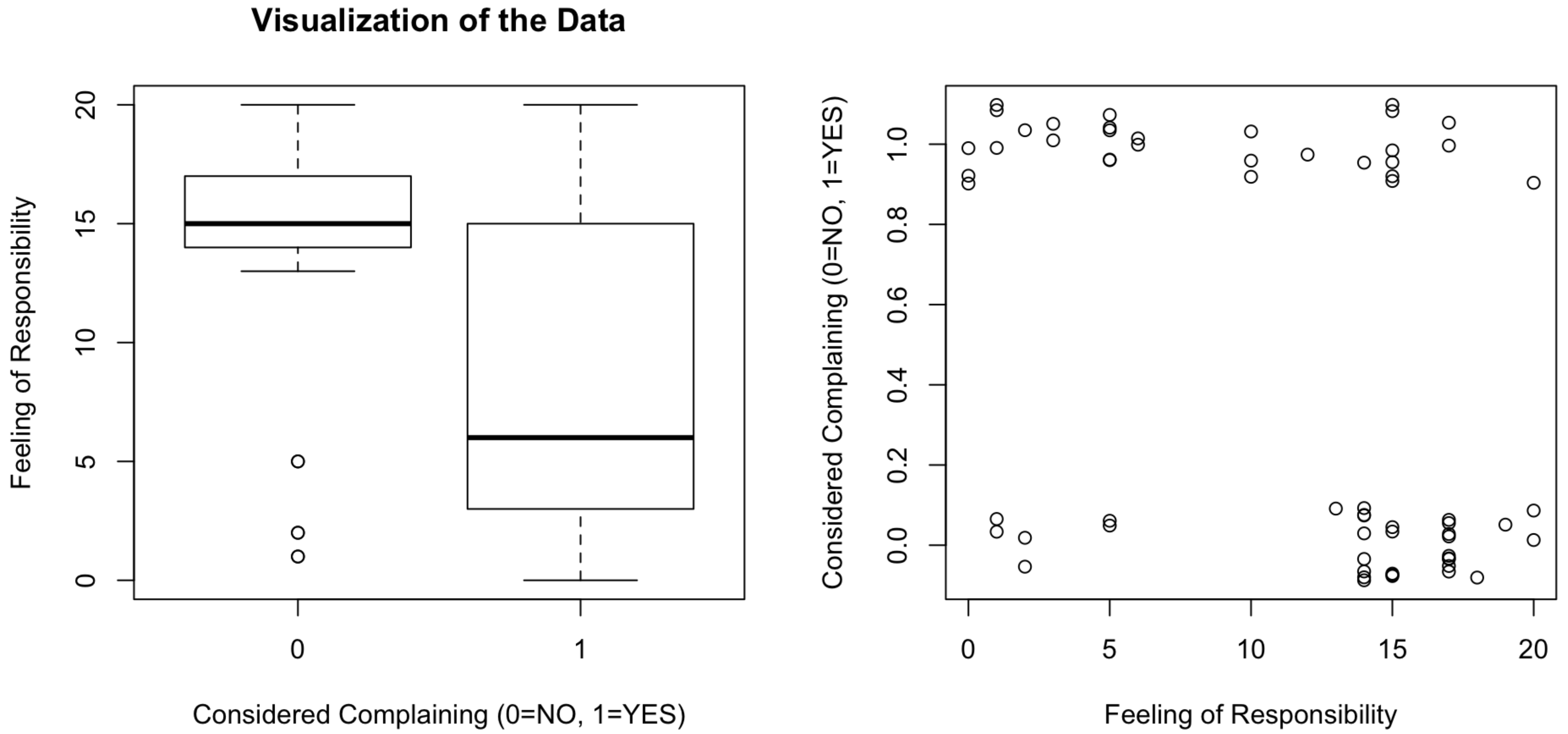
We can see that the variable `d0$complain` takes on values of 1 or 0. Here, 1 and 0 code for YES/NO responses for whether or not a participant considers complaining (1=YES, 2=NO).

Now, we might be interested in whether or not a participant's **self-reported feelings of responsibility** of missing a plane or a train (`d0$Responsible`) influences whether or not they **considered complaining** (`d0$complain`).

First, let's take a look at the data in a few different ways:

```
par(mfrow=c(1,2))
plot(factor(d0$complain), d0$Responsible, xlab=("Considered Complaining (0=NO, 1=YES)"), ylab="Feeling of Responsibility")
title(main="Visualization of the Data")

plot(jitter(d0$complain, factor=0.5)~d0$Responsible, ylab="Considered Complaining (0=NO, 1=YES)", xlab="Feeling of Responsibility")
```



Based on these plots, it looks like participants who felt more responsible about missing a plane or a train considered complaining less than participants who felt less responsible for missing a plane/train. Let’s test this hypothesis formally.

Hypothesis Testing

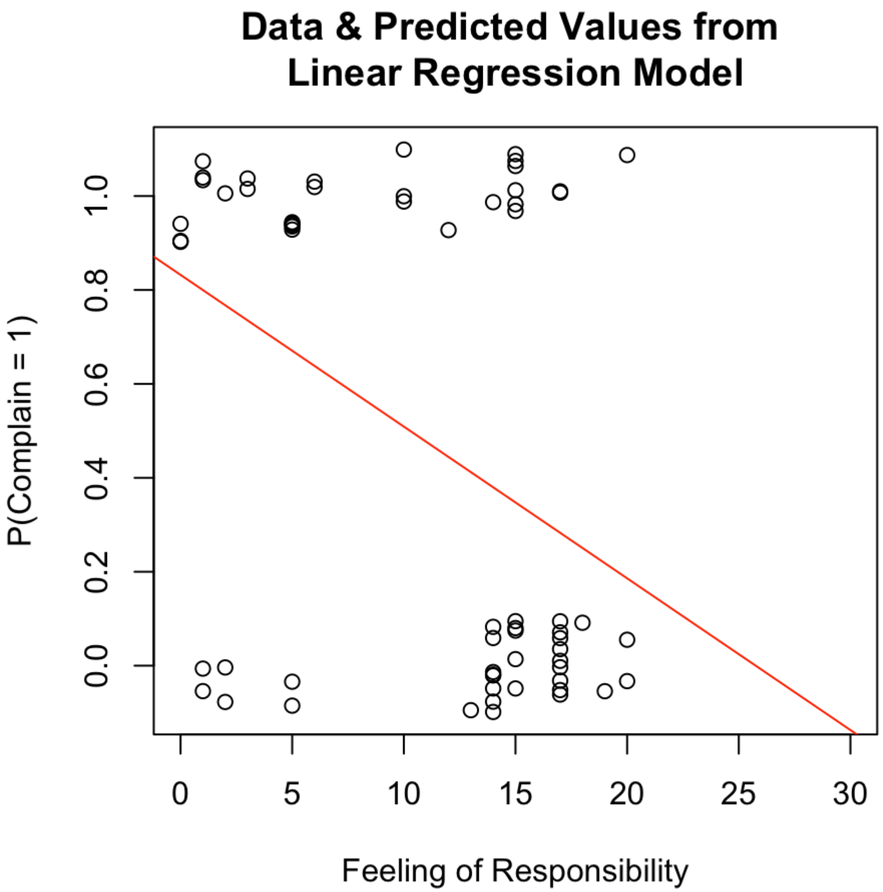
Here we will examine how a participant’s feeling of *responsibility* influences their tendency to *complain*. Since `complain` is a binary coded variable (i.e., with values of 0 or 1), `complain` is categorical rather than continuous. Thus, we are faced with a **classification** problem, rather than a **linear regression** problem. That is, given a person with some value of `Responsible`, we want to predict whether or not that person is likely to complain; in other words, we want to *classify* that person as “complainer”" (`complain` = 1), or a non-complainer (`complain` = 0). **Logistic regression** can be used to solve a classification problem like this.

What if we were to model the data with linear regression?

Here, we’ve plotted the data, and the linear regression line-of-best-fit:

```
plot(jitter(d0$complain, factor=0.5)~d0$Responsible, ylab="P(Complain = 1)", xlab="Feeling of Responsibility", xlim=c(0,30))

abline(lm(d0$complain~d0$Responsible), col='red')
title(main="Data & Predicted Values from \nLinear Regression Model")
```



A few things to note:

1. The line-of-best-fit tells us the **predicted probability** of complaining (i.e., P(`complain` = 1)) for each level of `Responsible`. For example, someone with a responsible level of "10" would have a 50% probability of complaining; someone with a responsible level of "5" would have a 70% probability of complaining.
2. For greater levels of `Responsible`, the model from linear regression predicts that the probability of complaining would be *less than* 0. This is impossible!

Sigmoid (logistic) function

Since a straight line won't fit our data well, we instead will use an S-shaped, **sigmoid** function. This function ensures that our output values will fall between 0 and 1, for any value of `x`. Further, please note that this function outputs the *probability* that `y` = 1.

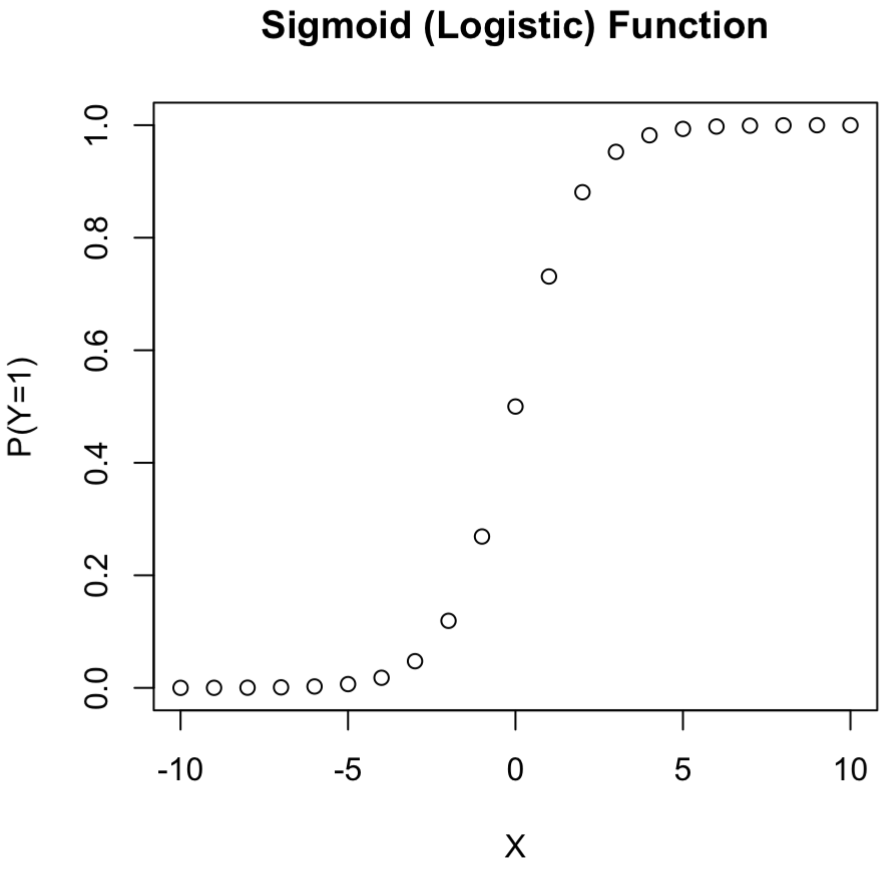
The sigmoid (or "logistic") function is given by the equation:

$$P(y = 1) = \frac{e^{(b+mx)}}{1 + e^{(b+mx)}}$$

That is, the probability of a person complaining (`complain` = 1) is a function of `x`, *e* (the base of the natural logarithm, ≈ 2.718), and the coefficients from the generalized linear model (`b` = intercept, `m` = coefficient for `Responsible`).

Here's an example of the sigmoid function plotted:

```
x <- c(-10:10)
b = 0 # intercept
m = 1 # slope
y <- exp((b + m*x)) / (1 + exp((b + m*x)))
plot(x, y, xlab="X", ylab="P(Y=1)")
title(main="Sigmoid (Logistic) Function")
```



In general, changing the **intercept** (b) shifts the sigmoid along the x-axis; positive intercepts result in a sigmoid to the *right* of x=0, and negative intercepts result in a sigmoid to the *left* of x=0. Changing the **slope** (m) changes both the *direction* and the *steepness* of the function. That is, positive slopes result in an “s”-shaped curve, and negative slopes result in a “z”-shaped curve. In addition, larger absolute values of slope result in a steeper function, and smaller absolute values result in a more gradual slope.

If you want to explore how varying the slope and intercept change the shape of the sigmoid function, try changing the coefficients in the .Rmd file, or check out the app here (http://spark.rstudio.com/supsych/logistic_regression/)).

Before, we mentioned that the sigmoid (logistic) function outputs the *probability* that $Y = 1$. This is is because the *logistic* regression is essentially *linear* regression on the **logit transform** of our original Y. Solving our sigmoid function above for `b + mx` $= \hat{y}$, we find that $\hat{y}$ is the **logit**, sometimes referred to as **log odds.**

$$logit(p) = \hat{y} = b + mx = log(\frac{p}{1-p}) = log(\frac{P(Y=1)}{P(Y=0)})$$

Running a general linear model w/ glm()

To estimate the intercept and slope coefficients, we can run a **generalized linear model** using the R function `glm()`.

```
rs1 = glm(complain ~ Responsible, data = d0, family = binomial, na.action = na.omit)
summary(rs1)
```

```
## 
## Call:
## glm(formula = complain ~ Responsible, family = binomial, data = d0, 
##     na.action = na.omit)
## 
## Deviance Residuals: 
##    Min      1Q  Median      3Q     Max  
## -1.775  -0.915  -0.673   0.898   1.786  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)   
## (Intercept)   1.4866     0.5917    2.51    0.012 * 
## Responsible  -0.1428     0.0462   -3.09    0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 87.194  on 62  degrees of freedom
## Residual deviance: 76.013  on 61  degrees of freedom
## AIC: 80.01
## 
## Number of Fisher Scoring iterations: 4
```

```
# show coeffecients
rs1$coefficients
```

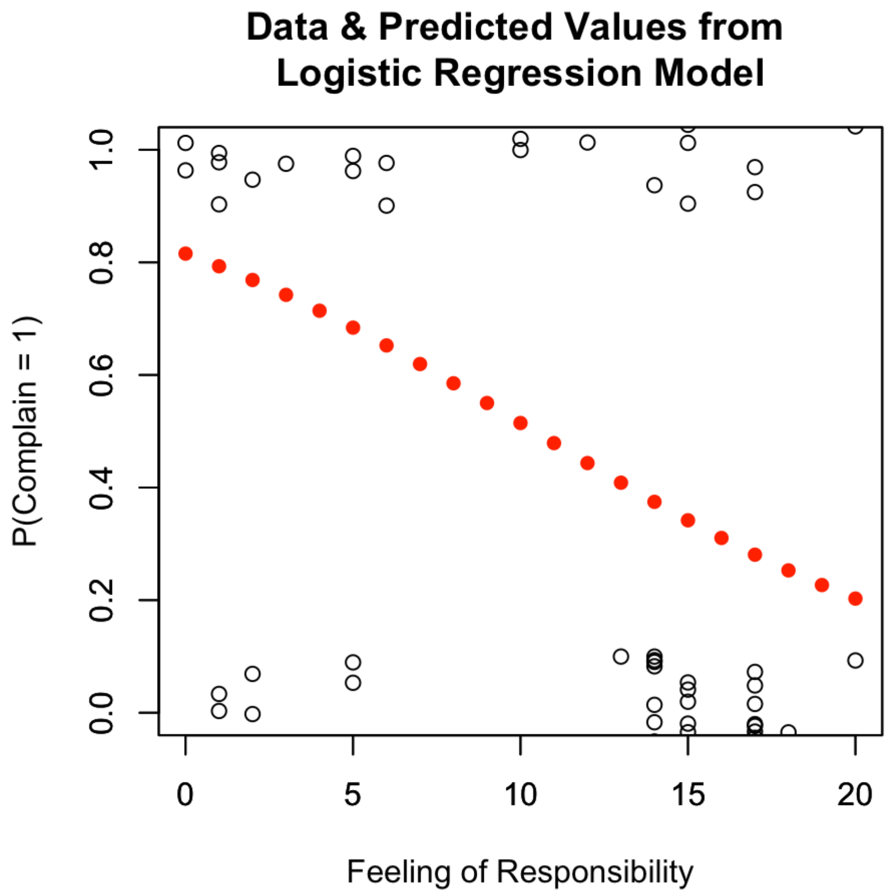
```
## (Intercept) Responsible 
##      1.4866     -0.1428
```

Now, let’s take the coefficients from the model, and put them into our sigmoid function to **generate predicted values**:

```
# plot the data
plot(jitter(d0$complain, factor=0.5)~d0$Responsible, ylab="P(Complain = 1)", xlab="Feeling of Responsibility", ylim=c(0,1), xlim=c(0,20))

# plot the predicted values using the sigmoid function
x <- c(0:20)
b = rs1$coefficients[1] # intercept
m = rs1$coefficients[2] # slope
y <- exp((b + m*x)) / (1 + exp((b + m*x)))

par(new=TRUE) # don't erase what's already on the plot!
plot(x, y, xlab="", ylab="", pch = 16, ylim=c(0,1), xlim=c(0,20), col='red')
title(main="Data & Predicted Values from \nLogistic Regression Model")
```



Since the slope (m = -0.1428; i.e., the coefficient for `Responsible`) is negative, the function is flipped from the S-shaped sigmoid shown above. Further, since the slope is small, the drop-off in the function is more gradual.

Interpreting the output from glm()

Before we noted that the logistic/sigmoid function outputs the **probability** that Y = 1 for a given value of x. In this example, P(`complain` = 1) – the probability that someone considered complaining – if they had a self-reported feeling of responsibility = 5 would be equal to:

$$P(Y = 1) = \frac{e^{(1.4866 + -0.1428 * 5)}}{1 + e^{(1.4866 + -0.1428 * 5)}}$$

Which would equal:

```
x = 5
b = rs1$coefficients[1] # intercept
m = rs1$coefficients[2] # slope
y <- exp((b + m*x)) / (1 + exp((b + m*x)))

as.numeric(y)
```

```
## [1] 0.6841
```

Thus, based on our logistic regression model, the probability that a person with a self-reported responsibility = 5 would have a 68.4087% probability of complaining.

Log odds

We can also extract the **log odds** from our `glm()` output. In our example, log odds essentially captures the ratio of (being a "complainer"):(not being a "complainer"). In logistic regression, the dependent variable ($\hat{y}$, or $b + mx$) is referred to as the **logit**, which is the natural log of the odds.

As noted above, by rearranging the logistic function, we get the equation for the logit:

$$\hat{y} = b + mx = log(\frac{p}{1-p})$$

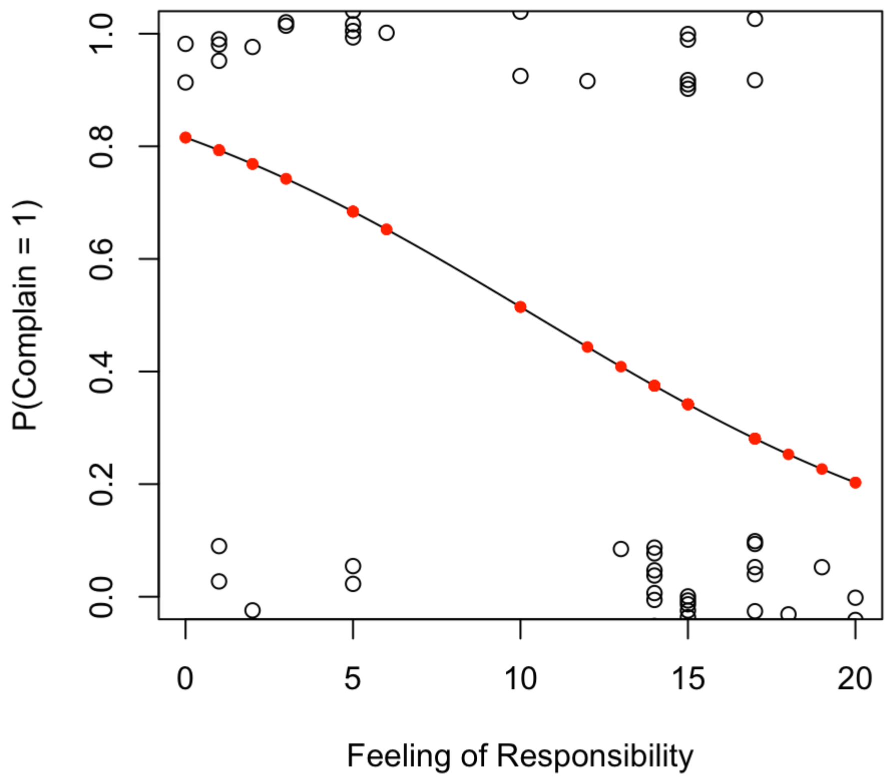
Since the predicted variable here is log odds, the coefficient of "Responsible" can be interpreted as **"for every one unit increase in self-reported responsibility, the odds of complaining change by $e^{(-0.1428)}$ = 0.8669 times."** Note that when the *log odds* are negative, the *odds* will be < 1. This means that here we would expect to see a 13.3062% *decrease* in the odds of complaining.

Some more ways to vizualize logistic regression data/results

```
# plot the data
plot(jitter(d0$complain, factor=0.5)~d0$Responsible, ylab="P(Complain = 1)", xlab="Feeling of Responsibility", ylim=c(0,1), xlim=c(0,20))

# draw a curve based on prediction from logistic regression model
Responsible = d0$Responsible
curve(predict(rs1, data.frame(Responsible = x), type = "resp"), add = TRUE)

points(d0$Responsible,fitted(rs1),pch=20, col='red')
```



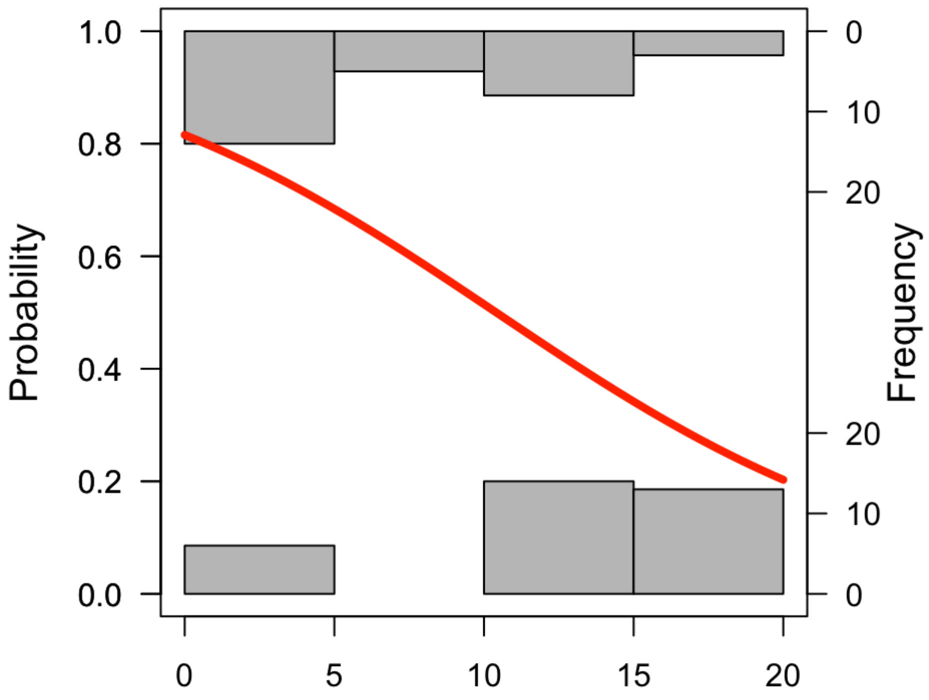
```
install.packages("popbio", repos="http://ftp.ussg.iu.edu/CRAN/"); library(popbio)
```

```
## Installing package into '/Users/thomas/Library/R/3.1/library'
## (as 'lib' is unspecified)
```

```
##
## The downloaded binary packages are in
##  /var/folders/63/fff36nnd7ll815rc_1kgy2dc0000gp/T//RtmpgT31k3/downloaded_packages
```

```
## Loading required package: quadprog
```

```
logi.hist.plot(d0$Responsible,d0$complain,boxp=FALSE,type="hist",col="gray")
```


Probability
Frequency