Logistic Regression Using Zoo Animals

Mashhood Syed

November 9, 2014

Dataset location: http://archive.ics.uci.edu/ml/datasets/Zoo

First we do some standard housekeeping:

```
setwd("~/Desktop/PS4")
zooFile = "zoo.data.txt"
zoo = read.csv(zooFile, header=FALSE)
colnames(zoo) = c("Animal Name", "Hair", "Feathers", "Eggs", "Milk", "Airborne", "Aquatic", "Predator",
#install.packages("dplyr", repos="http://cran.rstudio.com/")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#install.packages("arm", repos="http://cran.rstudio.com/")
library(arm)
```

```
## Loading required package: MASS
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
##
## Loading required package: Matrix
## Loading required package: lme4
## Loading required package: Rcpp
##
## arm (Version 1.7-07, built: 2014-8-27)
##
## Working directory is /Users/mashhoodsyed/Desktop/PS4
```

```
#install.packages("ggplot2", repos="http://cran.rstudio.com/")
library(ggplot2)
attach(zoo)
```

1. Choose 4 Continuous/Discrete Variables as Predictors: I am going to choose Legs, Tail, Venomous, and Toothed to predict whether the
   animal is a PREDATOR or not. I expect the coefficients to be…

Legs:negative Tail:positive Venomous:positive Toothed:positive

1. All 4 predictors in single logistic regression

```
zoo.all4 = glm(formula = Predator ~ Legs + Tail + Venomous + Toothed, family = binomial(link = "logit"))
display(zoo.all4, digits = 4)
```

```
## glm(formula = Predator ~ Legs + Tail + Venomous + Toothed, family = binomial(link = "logit"))
##             coef.est coef.se
## (Intercept)  0.2059   0.6377
## Legs        -0.0956   0.1098
## Tail        -0.1544   0.5232
## Venomous     1.0410   0.8771
## Toothed      0.5491   0.4384
## ---
##   n = 101, k = 5
##   residual deviance = 134.7, null deviance = 138.8 (difference = 4.1)
```

Below is a logistic regression using just one variable to start 2. Legs

```
zoo.1 = glm(formula = Predator ~ Legs, family = binomial(link = "logit"))
display(zoo.1, digits = 4)
```

```
## glm(formula = Predator ~ Legs, family = binomial(link = "logit"))
##             coef.est coef.se
## (Intercept)  0.5044   0.3516
## Legs        -0.0998   0.1000
## ---
##   n = 101, k = 2
##   residual deviance = 137.8, null deviance = 138.8 (difference = 1.0)
```

Lets look at the other variables individually.. 2. Tail:

```
zoo.2 = glm(formula = Predator ~ Tail, family = binomial(link = "logit"))
display(zoo.2, digits = 4)
```

```
## glm(formula = Predator ~ Tail, family = binomial(link = "logit"))
##             coef.est coef.se
## (Intercept) 0.1542   0.3934
## Tail        0.0870   0.4570
## ---
##   n = 101, k = 2
##   residual deviance = 138.8, null deviance = 138.8 (difference = 0.0)
```

2. Venomous (this is a no brainer..or is it?) Why isnt it 100% ?

```
zoo.3 = glm(formula = Predator ~ Venomous, family = binomial(link = "logit"))
display(zoo.3, digits = 4)
```

```
## glm(formula = Predator ~ Venomous, family = binomial(link = "logit"))
##             coef.est coef.se
## (Intercept) 0.1508   0.2080 
## Venomous    0.9478   0.8426 
## ---
##   n = 101, k = 2
##   residual deviance = 137.4, null deviance = 138.8 (difference = 1.4)
```

2. Toothed

```
zoo.4 = glm(formula = Predator ~ Toothed, family = binomial(link = "logit"))
display(zoo.4, digits = 4)
```

```
## glm(formula = Predator ~ Toothed, family = binomial(link = "logit"))
##             coef.est coef.se
## (Intercept) -0.1001   0.3166
## Toothed      0.5329   0.4110
## ---
##   n = 101, k = 2
##   residual deviance = 137.1, null deviance = 138.8 (difference = 1.7)
```

Calculating the logistic regression using "mean evaluation" and "divide by 4": Mean Evaluation, Divide by 4, and 95% confidence Interval:

Using Legs as our predictor, we get:

```
Intercept_Legs = .5044
Coeff_Est_of_Legs = -.0998
Mean_value_of_Legs = 2.84
Std_error_of_Legs = .1000
conf__bottom_Legs = (Coeff_Est_of_Legs - 2*Std_error_of_Legs) / 4
conf__top_Legs = (Coeff_Est_of_Legs + 2*Std_error_of_Legs) / 4
mean_eval_of_Legs = Intercept_Legs - Coeff_Est_of_Legs*Mean_value_of_Legs
Div_4_Legs = Coeff_Est_of_Legs / 4
conf__bottom_Legs
```

```
## [1] -0.07495
```

```
conf__top_Legs
```

```
## [1] 0.02505
```

```
mean_eval_of_Legs
```

```
## [1] 0.7878
```

```
Div_4_Legs
```

```
## [1] -0.02495
```

Using Tail as our predictor, we get:

```
Intercept_Tail = .15
Coeff_Est_of_Tail = .09
Mean_value_of_Tail = mean(Tail)
Std_error_of_Tail = .46
conf__bottom_Tail = (Coeff_Est_of_Tail - 2*Std_error_of_Tail) / 4
conf__top_Tail = (Coeff_Est_of_Tail + 2*Std_error_of_Tail) / 4
mean_eval_of_Tail = Intercept_Tail - Coeff_Est_of_Tail*Mean_value_of_Tail
Div_4_Tail = Coeff_Est_of_Tail / 4
conf__bottom_Tail
```

```
## [1] -0.2075
```

```
conf__top_Tail
```

```
## [1] 0.2525
```

```
mean_eval_of_Tail
```

```
## [1] 0.08317
```

```
Div_4_Tail
```

```
## [1] 0.0225
```

Using Venomous as our predictor, we get:

```
Intercept_Venomous = .15
Coeff_Est_of_Venomous = .95
Mean_value_of_Venomous = mean(Venomous)
Std_error_of_Venomous = .84
conf__bottom_Venomous = (Coeff_Est_of_Venomous - 2*Std_error_of_Venomous) / 4
conf__top_Venomous = (Coeff_Est_of_Venomous + 2*Std_error_of_Venomous) / 4
mean_eval_of_Venomous = Intercept_Venomous - Coeff_Est_of_Venomous*Mean_value_of_Venomous
Div_4_Venomous = Coeff_Est_of_Venomous / 4
conf__bottom_Venomous
```

```
## [1] -0.1825
```

```
conf__top_Venomous
```

```
## [1] 0.6575
```

```
mean_eval_of_Venomous
```

```
## [1] 0.07475
```

```
Div_4_Venomous
```

```
## [1] 0.2375
```

Using Toothed as our predictor, we get:

```
Intercept_Toothed = -.10
Coeff_Est_of_Toothed = .53
Mean_value_of_Toothed = mean(Toothed)
Std_error_of_Toothed = .41
conf__bottom_Toothed = (Coeff_Est_of_Toothed - 2*Std_error_of_Toothed) / 4
conf__top_Toothed = (Coeff_Est_of_Toothed + 2*Std_error_of_Toothed) / 4
mean_eval_of_Toothed = Intercept_Toothed - Coeff_Est_of_Toothed*Mean_value_of_Toothed
Div_4_Toothed = Coeff_Est_of_Toothed / 4
conf__bottom_Toothed
```

```
## [1] -0.0725
```

```
conf__top_Toothed
```

```
## [1] 0.3375
```

```
mean_eval_of_Toothed
```

```
## [1] -0.4201
```

```
Div_4_Toothed
```

```
## [1] 0.1325
```

2. Now lets add more variables to try and improve our model

In terms of the difference in the residual and null deviance, we have a differenc of 3.3. We can see that we have added two predictors and that our decrease is > 2. We have a better model, as insignificant as it may be by 1.3. In regards to the mean values of Venomous and Toothed, there is a 55.7% chance
that our Animal is a Predator

```
zoo.5 = glm(formula = Predator ~ + Venomous + Toothed, family = binomial(link = "logit"))
log_odds5 = cbind(1, mean(Venomous), mean(Toothed)) %*% zoo.5$coef
invlogit(log_odds5)
```

```
##        [,1]
## [1,] 0.5571
```

```r
display(zoo.5, digits = 4)
```

```
## glm(formula = Predator ~ +Venomous + Toothed, family = binomial(link = "logit"))
##             coef.est coef.se
## (Intercept) -0.1992   0.3286
## Venomous     1.0312   0.8524
## Toothed      0.5746   0.4163
## ---
##   n = 101, k = 3
##   residual deviance = 135.5, null deviance = 138.8 (difference = 3.3)
```

2. Now lets make the coefficients comparable by multiplying each coefficient by its predictors SD.

We can see that a one SD difference in Venomous translates to a 6.99 point increase in the probability of the animal being a Predator. In addition, we get a 7.06 point increase if the animal is Toothed.

```r
((coef(zoo.5)["Venomous"] * sd(Venomous))/4) * 100
```

```
## Venomous
##    6.997
```

```r
((coef(zoo.5)["Toothed"] * sd(Toothed))/4) * 100
```

```
## Toothed
##    7.06
```

3. Lets add in a interaction predictor to see if that improves our model

```r
zoo.6 = glm(Predator ~ + Venomous + Toothed + Toothed:Venomous, family = binomial(link = "logit"))
display(zoo.6)
```

```
## glm(formula = Predator ~ +Venomous + Toothed + Toothed:Venomous,
##     family = binomial(link = "logit"))
##                  coef.est coef.se
## (Intercept)        -0.11     0.33
## Venomous            0.11     1.05
## Toothed             0.43     0.43
## Venomous:Toothed   16.14  1199.77
## ---
##   n = 101, k = 4
##   residual deviance = 132.9, null deviance = 138.8 (difference = 5.9)
```

3. In order to interpret our results, we need to use the mean of our variables Venomous and Toothed. Our results tell us that the base probability of an animal being a Predator given an average Venomous value and a average Toothed value is 71.6%

```r
log_odds.6 = cbind(1, mean(Venomous), mean(Toothed), mean(Venomous)*mean(Toothed)) %*% coef(zoo.6)
invlogit(log_odds.6)
```

```
##         [,1]
## [1,] 0.7169
```

4. Lets look at some other predictors to see if we can improve our model once again

Lets try these three new variables: Hair, Backbone, Breathers

I expect the coefficients to be…

Hair: positive Backbone: positive Breathers: negative

```
zoo.1a = glm(Predator ~ Hair, family = binomial(link = "logit"))
display(zoo.1a, digits = 4)
```

```
## glm(formula = Predator ~ Hair, family = binomial(link = "logit"))
##             coef.est coef.se
## (Intercept)  0.4925   0.2706
## Hair        -0.6322   0.4083
## ---
##   n = 101, k = 2
##   residual deviance = 136.4, null deviance = 138.8 (difference = 2.4)
```

```
zoo.2a = glm(Predator ~ Backbone, family = binomial(link = "logit"))
display(zoo.2a, digits = 4)
```

```
## glm(formula = Predator ~ Backbone, family = binomial(link = "logit"))
##             coef.est coef.se
## (Intercept) 0.0000   0.4714
## Backbone    0.2666   0.5208
## ---
##   n = 101, k = 2
##   residual deviance = 138.6, null deviance = 138.8 (difference = 0.3)
```

```
zoo.3a = glm(Predator ~ Breathers, family = binomial(link = "logit"))
display(zoo.3a, digits = 4)
```

```
## glm(formula = Predator ~ Breathers, family = binomial(link = "logit"))
##             coef.est coef.se
## (Intercept)  1.4469   0.5557
## Breathers   -1.4969   0.5990
## ---
##   n = 101, k = 2
##   residual deviance = 131.3, null deviance = 138.8 (difference = 7.5)
```

4. Using Hair as our predictor, we get:

```
Intercept_Hair = .4925
Coeff_Est_of_Hair = -.6322
Mean_value_of_Hair = mean(Hair)
Std_error_of_Hair = .4083
conf__bottom_Hair = (Coeff_Est_of_Hair - 2*Std_error_of_Hair) / 4
conf__top_Hair = (Coeff_Est_of_Hair + 2*Std_error_of_Hair) / 4
mean_eval_of_Hair = Intercept_Hair - Coeff_Est_of_Hair*Mean_value_of_Hair
Div_4_Hair = Coeff_Est_of_Hair / 4
conf__bottom_Hair
```

```
## [1] -0.3622
```

```
conf__top_Hair
```

```
## [1] 0.0461
```

```
mean_eval_of_Hair
```

```
## [1] 0.7617
```

```
Div_4_Hair
```

```
## [1] -0.158
```

4. Using Backbone as our predictor, we get:

```
Intercept_Backbone = .0000
Coeff_Est_of_Backbone = .2666
Mean_value_of_Backbone = mean(Backbone)
Std_error_of_Backbone = .5208
conf__bottom_Backbone = (Coeff_Est_of_Backbone - 2*Std_error_of_Backbone) / 4
conf__top_Backbone = (Coeff_Est_of_Backbone + 2*Std_error_of_Backbone) / 4
mean_eval_of_Backbone = Intercept_Backbone - Coeff_Est_of_Backbone*Mean_value_of_Backbone
Div_4_Backbone = Coeff_Est_of_Backbone / 4
conf__bottom_Backbone
```

```
## [1] -0.1938
```

```
conf__top_Backbone
```

```
## [1] 0.3271
```

```
mean_eval_of_Backbone
```

```
## [1] -0.2191
```

```
Div_4_Backbone
```

```
## [1] 0.06665
```

4. Using Breathers as our predictor, we get:

```
Intercept_Breathers = 1.4469
Coeff_Est_of_Breathers = -1.4969
Mean_value_of_Breathers = mean(Breathers)
Std_error_of_Breathers = .5990
conf__bottom_Breathers = (Coeff_Est_of_Breathers - 2*Std_error_of_Breathers) / 4
conf__top_Breathers = (Coeff_Est_of_Breathers + 2*Std_error_of_Breathers) / 4
mean_eval_of_Breathers = Intercept_Breathers - Coeff_Est_of_Breathers*Mean_value_of_Breathers
Div_4_Breathers = Coeff_Est_of_Breathers / 4
conf__bottom_Breathers
```

```
## [1] -0.6737
```

```
conf__top_Breathers
```

```
## [1] -0.07472
```

```
mean_eval_of_Breathers
```

```
## [1] 2.633
```

```
Div_4_Breathers
```

```
## [1] -0.3742
```

4. Now lets add more variables to try and improve our model

In terms of the difference in the residual and null deviance, we have a differenc of 8.3. We can see that we have added two predictors and that our decrease is > 2. We have a better model, 6.3 points higher than our last model. In regards to the mean values of Breathers and Backbone, there is a 56.59% chance
that our Animal is a Predator (only slightly better then our first model).

```
##       [,1]
## [1,] 0.566
```

```
## glm(formula = Predator ~ +Breathers + Backbone, family = binomial(link = "logit"))
##             coef.est coef.se
## (Intercept)  1.0349   0.6524
## Breathers   -1.6600   0.6283
## Backbone     0.6636   0.5812
## ---
##   n = 101, k = 3
##   residual deviance = 130.0, null deviance = 138.8 (difference = 8.8)
```

4. Now lets make the coefficients comparable by multiplying each coefficient by its predictors SD.

We can see that one SD difference in Backbone translates to a 6.38 point increase in the probability of the animal being a Predator. In addition, we get a -16.93 point decrease if the animal has Breathers.

```
## Backbone
##    6.381
```

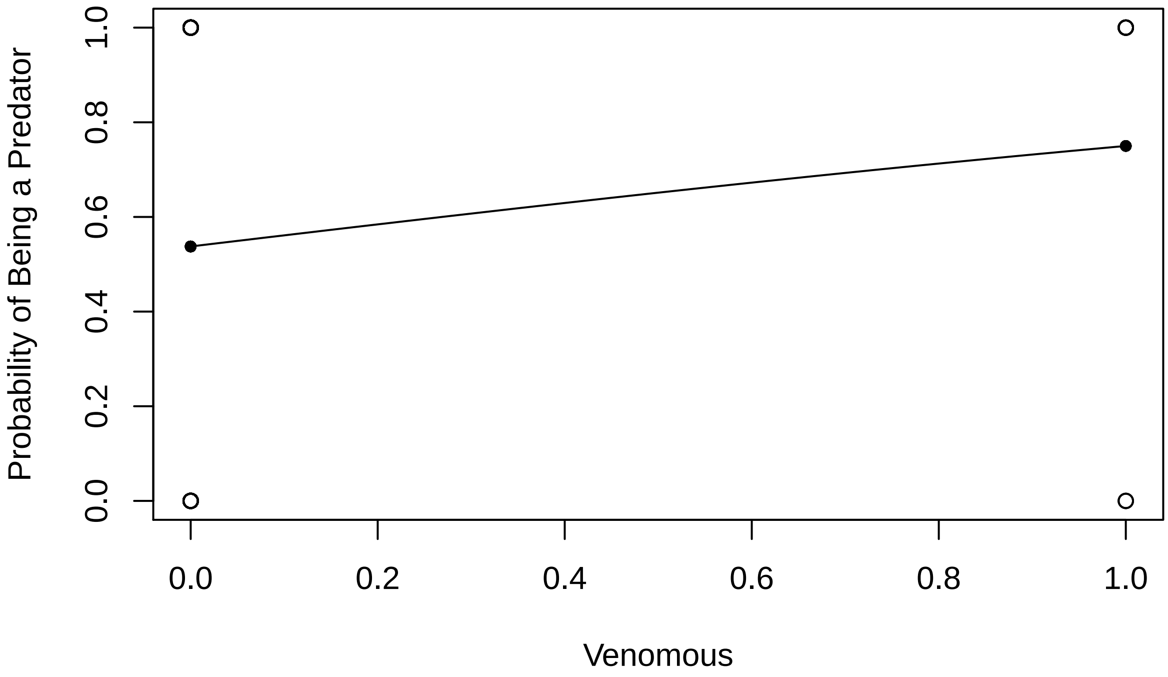
```
## Breathers
##    -16.93
```

5. Summary

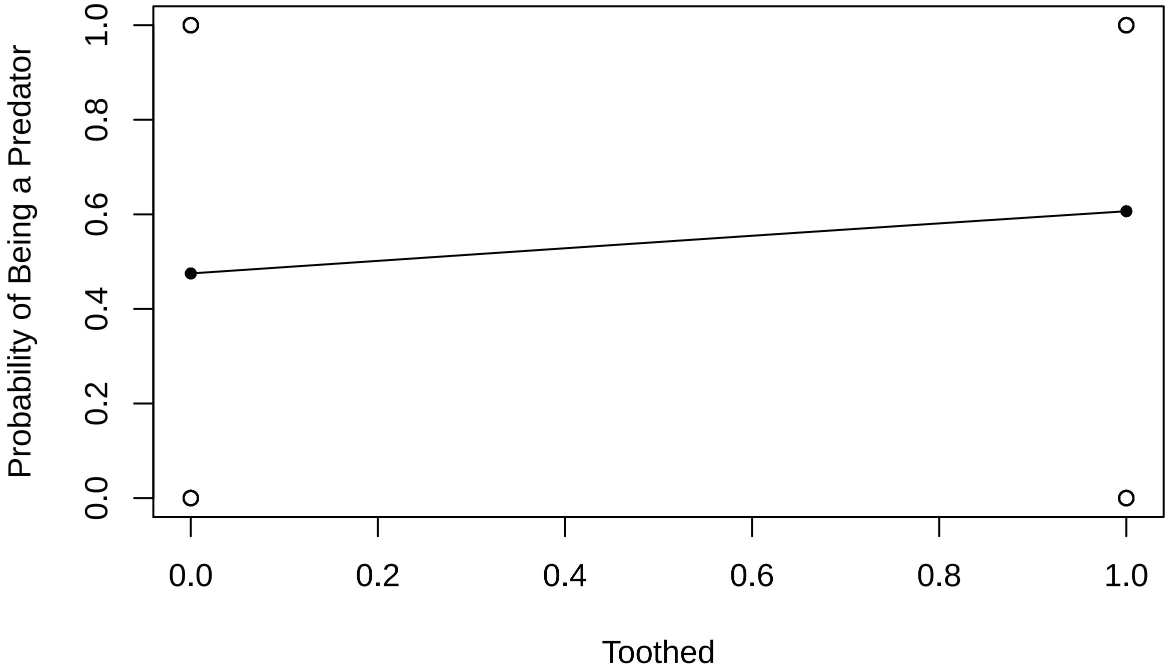
Based on the data we were given, we performed analysis to see which attributes provided us the greatest probability of arriving at whether a given animal at the zoo was a Predator or not. The data set had only categorical values that were discrete. For example, we knew whether or not a given animal had a particular attribute or not (ex: hair, feathers, toothed, backbone). These were all essentially yes/no indicators. We were able to combine different attributes in the data to see which combination would yield us a better probability. By doing this, we are able to "hone" in on what one variable (or combination thereof) really decides whether or not our animal in question is a Predator. While he had a fair number of attributes to work with, they were all the same type of attribute (binary). If we had a combination of both binary and numeric values, we could have gained even more insight into the model and possibly improved our prediction. An example of some numeric values that could have been helpful are: Speed (how fast can the animal run) and sleep (how much does the animal sleep).

Below are a few graphs that plot a few of the variables discussed in this Problem Set.

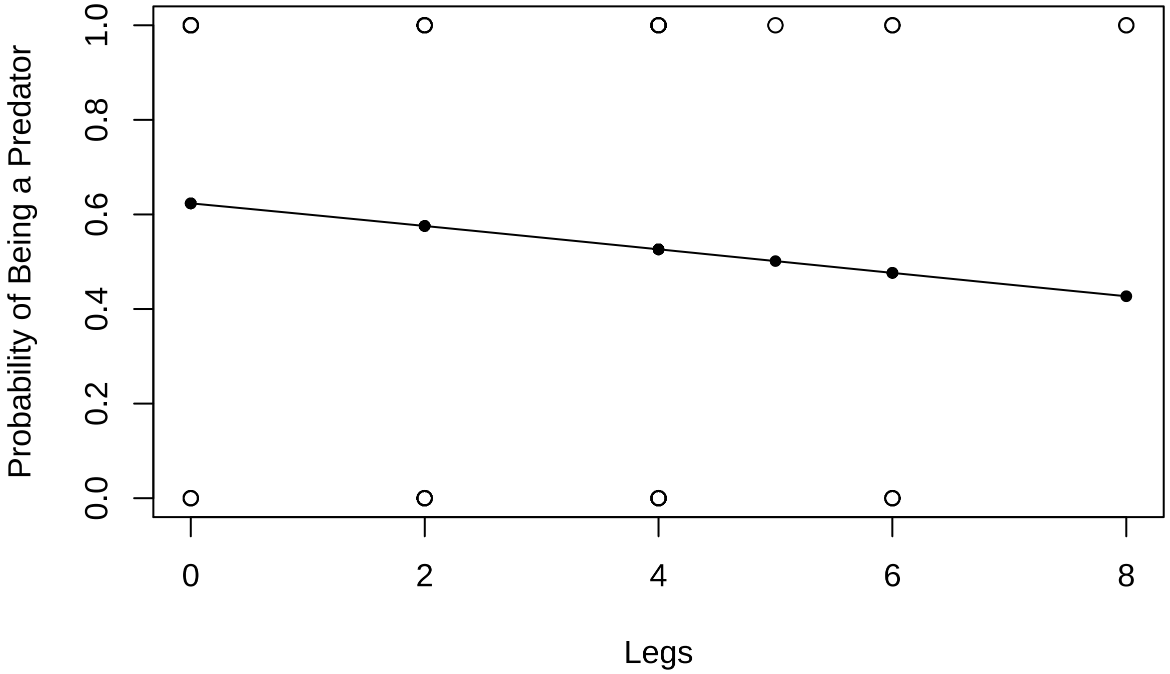
```
plot(Venomous, Predator,xlab="Venomous",ylab="Probability of Being a Predator") # plot with body size o
g=glm(Predator~Venomous,family=binomial) # run a logistic regression model (in this case, generalized l

curve(predict(g,data.frame(Venomous=x),type="resp"),add=TRUE) # draws a curve based on prediction from

points(Venomous,fitted(g),pch=20) # optional: you could skip this draws an invisible set of points of b
```



```
plot(Toothed, Predator,xlab="Toothed",ylab="Probability of Being a Predator") # plot with body size on
g=glm(Predator~Toothed,family=binomial) # run a logistic regression model (in this case, generalized li

curve(predict(g,data.frame(Toothed=x),type="resp"),add=TRUE) # draws a curve based on prediction from l

points(Toothed,fitted(g),pch=20) # optional: you could skip this draws an invisible set of points of bo
```



```
plot(Legs, Predator,xlab="Legs",ylab="Probability of Being a Predator") # plot with body size on x-axis
g=glm(Predator~Legs,family=binomial) # run a logistic regression model (in this case, generalized linea

curve(predict(g,data.frame(Legs=x),type="resp"),add=TRUE) # draws a curve based on prediction from logi

points(Legs,fitted(g),pch=20) # optional: you could skip this draws an invisible set of points of body
```


Probability of Being a Predator
Legs