HM_ClassNotes_23January2018

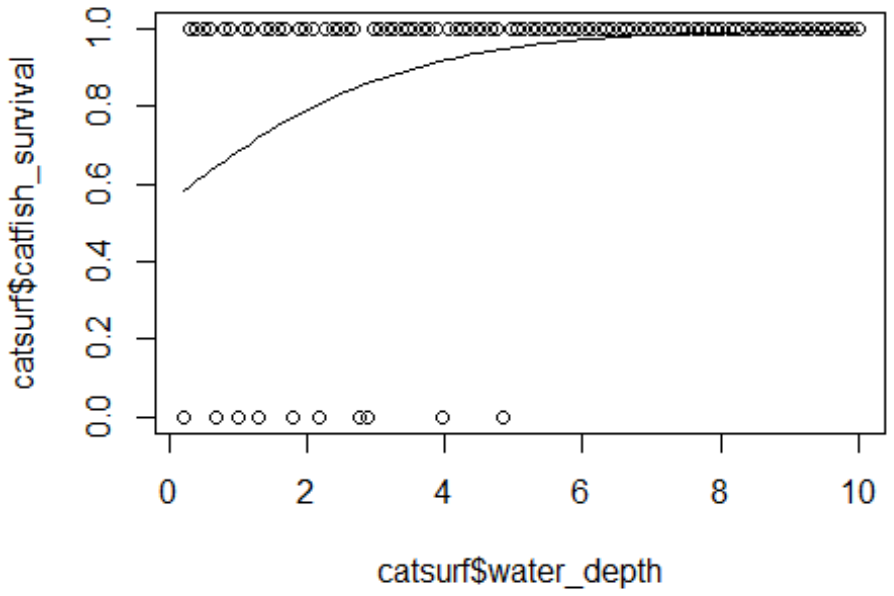
C. Merriman

January 23, 2018

REVIEW:

All models have stochastic part and deterministic part. QUESTION: We ended last week with catfish survival data

```
catsurf<-read.csv("catfish_survival.csv")
plot(catsurf$catfish_survival~catsurf$water_depth)
#glm = GENERLALIZED linear model
#parameterized using maximum liklihood estimate (MLE)
#find the value of p that results in MLE
catmod1<-glm(catfish_survival~water_depth, data=catsurf, family=binomial)
coef(catmod1) #calling only the coefficients...these corroborate with rest of
the class

## (Intercept) water_depth
##   0.2177667   0.5563107

plot(catsurf$catfish_survival~catsurf$water_depth)
curve(plogis(0.2177667+0.553107*x), add=T)
```



The above is how you would interpret parameters if it was a linear model.

BUT as a consequence of logit-link, the parameters are on the real number scale. To interpret, we need to transform to the probability scale. This makes for an importat change because it accounts for the 0-1 scale that binary response variables may need.

QUESTION: How can we interpret these parameters? Intercept= 0.2177667 <- Baseline survival when water depth is near zero. Put on a probability scale using plogis(intercept) Baseline survival for catfish: logistic(0.21777) Slope: trickier versus a normal linear regression because not linear. At beginning: not a lot of effect IN middle: big effect At end: not a lot of effect Trick: Divide by Four RUle: Value of the derivative at the steepest part of the line< this means a fair amount of calculus, which essentiallly breaks down into the slope parameter divided by four. ((be$^{0)/((1+e}$0)^2))=(b/4) Soooo....effect of water depth is ~ 0.56/4=0.14. By increasing water depth by 1 unit we see a 14% increase in catfish survival.

ANOTHER WAY TO DO THIS: evaluate how probability is different between values of x. If the intercept is 0.5, and slope is 0.1, we can calculate difference between two values of x:

```
x1<-2
x2<-0
print("Difference between a value of x=2 and x=0:")

## [1] "Difference between a value of x=2 and x=0:"

plogis(0.5+0.1*x1)-plogis(0.5+0.1*x2)

## [1] 0.04572844
```

This is another way to talk about effect size. See example on hunting/logging on animals...Roopsind et al 2017.

HYPOTHESIS TESTS: LOGISTIC REGRESSION

Look at confidence intervals for catfish

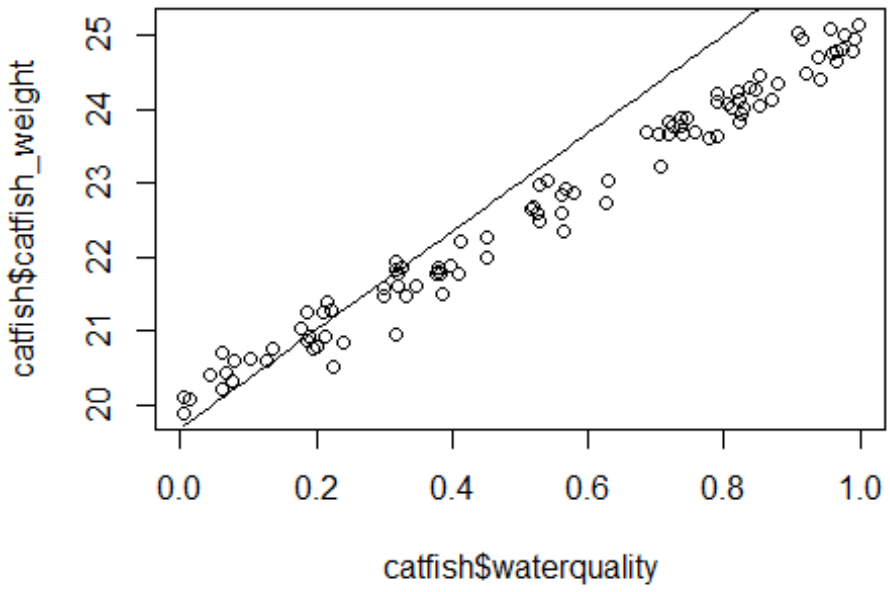
```
confint(catmod1) #corroborates with class

## Waiting for profiling to be done...

##                   2.5 %   97.5 %
## (Intercept) -0.9432583 1.403034
## water_depth  0.2320441 1.001251
```

But is this really that handy? We want o interpret this on the REAL Number scale. EFFECT SIZE: Put on probability scale HYP. TEST: Put on Real number scale

IMPROVING PARAMETER INTERPRETATION

Using Water Depth and Fish Weight

```
catfish<-read.csv("catdat16.csv")
catmod2<-lm(catfish$catfish_weight~catfish$waterquality)
catmod2

##
## Call:
## lm(formula = catfish$catfish_weight ~ catfish$waterquality)
##
## Coefficients:
##          (Intercept)  catfish$waterquality
##               20.000                 4.987

plot(catfish$catfish_weight~catfish$waterquality)
curve(19.687+6.648*x, add=T)
```



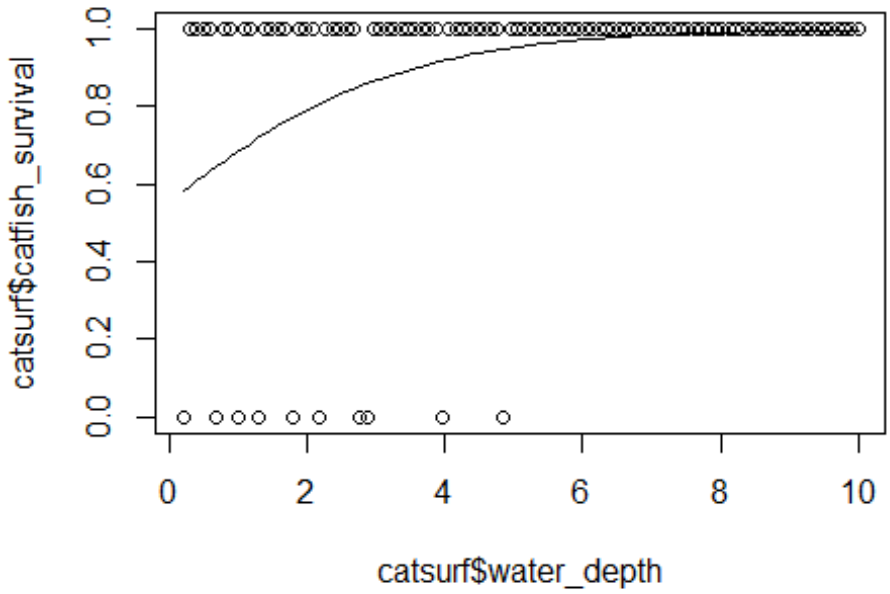
BUT this graph tells us that you'll have a 20 unit fish in no water, which makes no biological sense.

How do you fix this? SOLUTION 1: zero-intercept regression: set intercept to zero and fit two parameters: slope and sigma. But this doesn't fit the model as well! USE FOR THIS TOOL: assess model bias. Look at how different the slope is from that the models would say if it was a perfect fit.

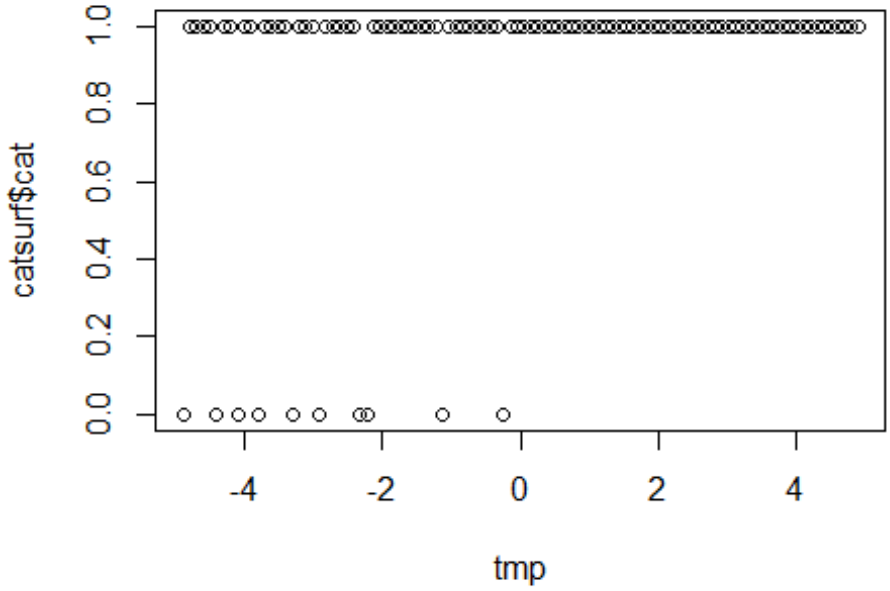
SOLUTION 2: Center around the mean 1. Create a new predictor variable: xc= x- x_hat 2. New interpretation of intercept: baseline value is for mean of x

```
#use catfish survival data = "catsurf"
plot(catsurf$catfish_survival~catsurf$water_depth)
#old glm:
```

```r
catmod1<-glm(catfish_survival~water_depth, data=catsurf, family=binomial)
coef(catmod1) #calling only the coefficients...these corroborate with rest of the class

## (Intercept) water_depth
##   0.2177667   0.5563107

plot(catsurf$catfish_survival~catsurf$water_depth)
curve(plogis(0.2177667+0.553107*x), add=T) #use values from catmod1
```




```r
#create new means: THANK YOU JAKE!
x<-catsurf$water_depth
x_hat<-mean(catsurf$water_depth)
MeanCenter<-x-x_hat
tmp <- catsurf$water - (mean(catsurf$water))
plot(catsurf$cat ~ tmp)
catmod2<-glm(catfish_survival~tmp, data=catsurf, family=binomial)

plot(catsurf$cat ~ tmp)
```



Likelihood: P(data|hypothesis)

Remember that nasty equation for the likelihood function? use dbinom

```
dbinom(x=5, size=10,prob=0.8) #confirms out guess that max likelihood is
0.5<-see last week's notes.

## [1] 0.02642412
```

HOW DO WE CALCULATE THE LIKLIHOOD FOR MULTIPLE OBSERVATIONS? Two Experiments: 1. Ten seeds added; rats eat five seeds 2. Fifteen seeds added, rats eat three seeds What's the likelihood that p=0.5? You just multiply! p(both events)=p(one event)*p(another events)

```
exp1<-dbinom(x=5, size=10, prob=0.5)
exp2<-dbinom(x=3, size=15, prob=0.5)
exp1 #0.25

## [1] 0.2460938

exp2 #0.14

## [1] 0.0138855

exp1*exp2 #0.0034

## [1] 0.003417134
```

```
#Dr.C's code:
dbinom(x=5, size=10, prob=0.5)*dbinom(x=3, size=15, prob=0.5) #same answer

## [1] 0.003417134
```

MORE THAN TWO EVENTS: Just multiply all of the elements in the series. --But that leads to really small numbers, so we just take the log of the distribution. This is where "log likelihood" comes in!

Optimization for maximum likelihood: negative log-likelihood is standard currency. To find max. likelihood= minimize the negative log-likelihood!

```
args(dbinom)

## function (x, size, prob, log = FALSE)
## NULL
```

glm: generalized linear model. Parameterized using maximum likelihood estimate (MLE) Optimization algorithms find parameter values that minimize negative log-likihood.

PROPORTIONAL DATA IN BINOMIAL REGRESSIONS

Yi~Binomial(pi,Ni) Examples: Canopy cover and etc using LiDAR data. w/in each single 30m2 Landsat pixels there are 876 LiDAR "pixels" So take the proportion of LiDAR "pixels" in each Landsat pixel and extrapolate?