HW2 Edward Trout

Question 1: Seedling Survival

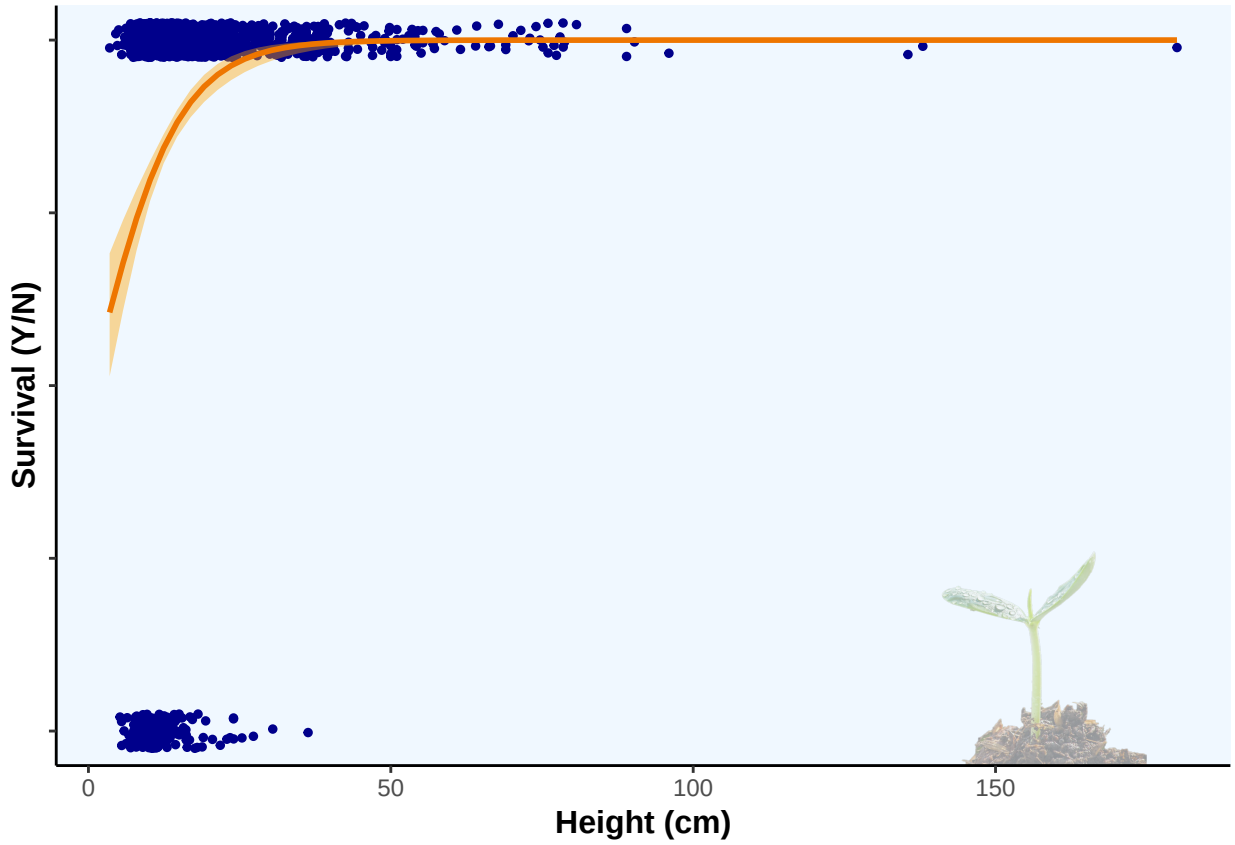
```
#load and inspect data
seeds <- read.csv("SEEDLING_SURVIVAL.csv")

head(seeds, 3)
```

```
##   survival HEIGHT LIGHT
## 1        1   47.0  2.40
## 2        1   70.2 14.83
## 3        1   16.3  9.15
```

A. Effect of Height on Seedling Survival

```
# build model
seed.height <- glm(seeds$survival~seeds$HEIGHT, family = "binomial")
```

a) Plot



b) Point Estimates for Intercept and Slope Parameters

```
coef(seed.height)
```

```
##  (Intercept) seeds$HEIGHT
##  -0.06271111   0.14071141
```

```
#transform into probability unit space
plogis(coef(seed.height)[1])
```

```
## (Intercept)
##   0.4843274
```

```
coef(seed.height)[2]/4
```

```
## seeds$HEIGHT
##   0.03517785
```

The baseline survival probability of a seed is .484, or 48.4%. When the height of a seedling is 0, it has around 48% chance of surviving. When at its height of influence, a seedling gains .035 or 3.5% probability of survival every centimeter it grows in height.

c) Confidence Intervals for Intercept and Slope

```
confint(seed.height)
```

```
##                    2.5 %    97.5 %
## (Intercept)  -0.5791061 0.4268167
## seeds$HEIGHT  0.1038803 0.1815477
```

```
#transform into probability unit space
plogis(confint(seed.height))
```

```
##                   2.5 %    97.5 %
## (Intercept)  0.3591383 0.6051133
## seeds$HEIGHT 0.5259467 0.5452627
```

```
.1038803/4
```

```
## [1] 0.02597007
```

```
.1815477/4
```

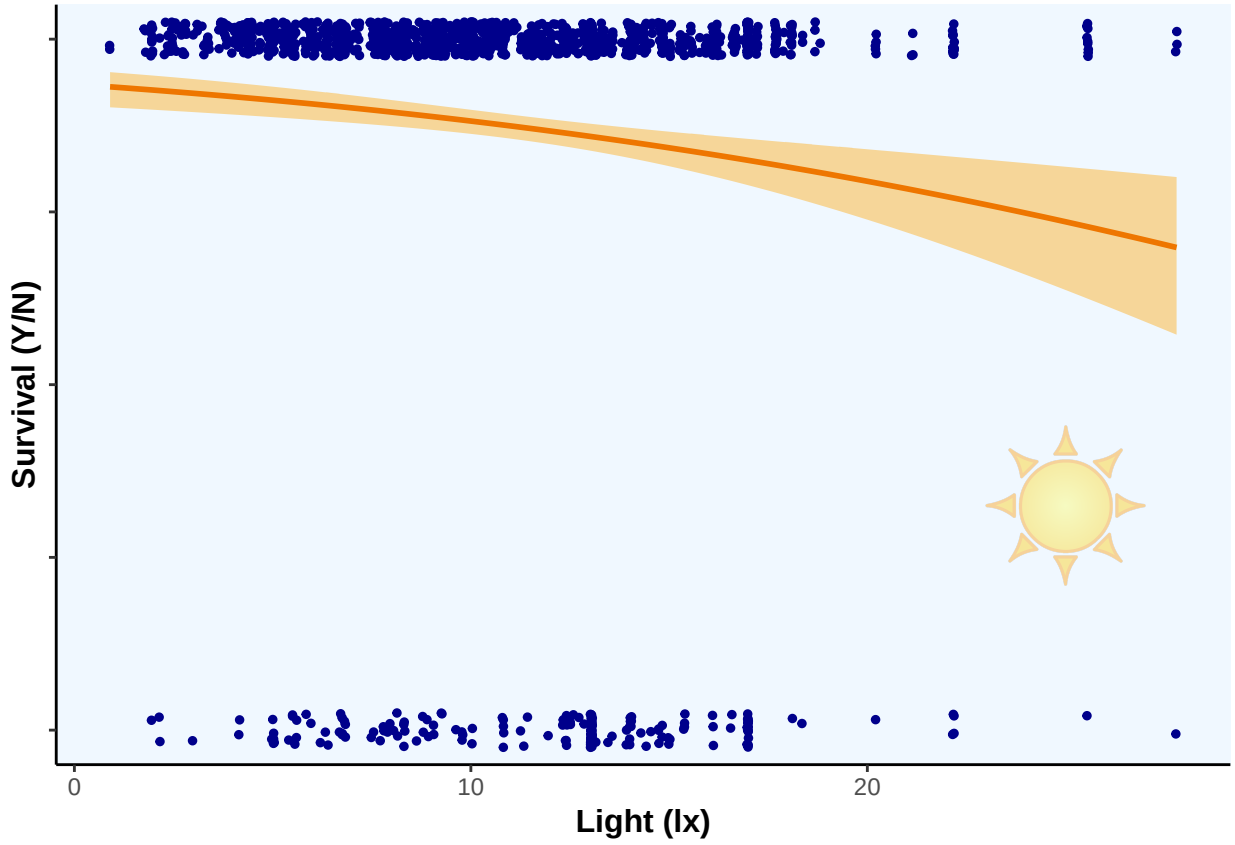
```
## [1] 0.04538693
```

This informs us that the intercept, while significant at 48.4%, has a wide range between 35.9% and 60.5% survivability. This makes sense as there would be high variability in a seedling's survival at height = 0. The slope parameter does not overlap 0 and is positive, with a confidence interval of 2.6%-4.5%, suggesting that there is an effect of height on seedling survival, and survival probability increases with height gradually. Observing the data and fitted line on the plot, we can see that once a seedling reaches 50cm, it is guaranteed survival.

B. Effect of Light on Seedling Survival

```
# build model
seed.light <- glm(seeds$survival~seeds$LIGHT, family = "binomial")
```

a) Plot



b) Point Estimates for Intercept and Slope Parameters

```
coef(seed.light)
```

```
## (Intercept) seeds$LIGHT
##  2.66194692 -0.06552684
```

```
# transform into probability space
plogis(coef(seed.light)[1])
```

```
## (Intercept)
##   0.9347435
```

```
coef(seed.light)[2]/4
```

```
## seeds$LIGHT
## -0.01638171
```

According to the model there is a 93.4% probability of survival at the baseline light level, perhaps 0 lux. The effect of light on survivability is a negative and a slight one, at maximum a 1.6% decrease in survival probability per unit of light exposure.

c) Confidence Intervals for Intercept and Slope

```
confint(seed.light)
```

```
##                    2.5 %     97.5 %
## (Intercept)  2.25136434  3.0876309
## seeds$LIGHT -0.09841747 -0.0325795
```

```
# transform into probability space

plogis(confint(seed.light))
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.9047682 0.9563796
## seeds$LIGHT 0.4754155 0.4918558
```

```
-.09841747/4
```

```
## [1] -0.02460437
```

```
-.0325795/4
```

```
## [1] -0.008144875
```

The confidence interval for the slope does not overlap zero, suggesting a true significance. The bounds of -.8% to -3.2% are quite broad for the maximum value of -1.6%, suggesting a wide variability in effect size.

Is Height or Light a Stronger Predictor of Seedling Survival?

Seedling height is a stronger predictor than light, the slope is higher and the confidence interval is tighter suggesting less variance.

Question 2: Recruitment of Seeds

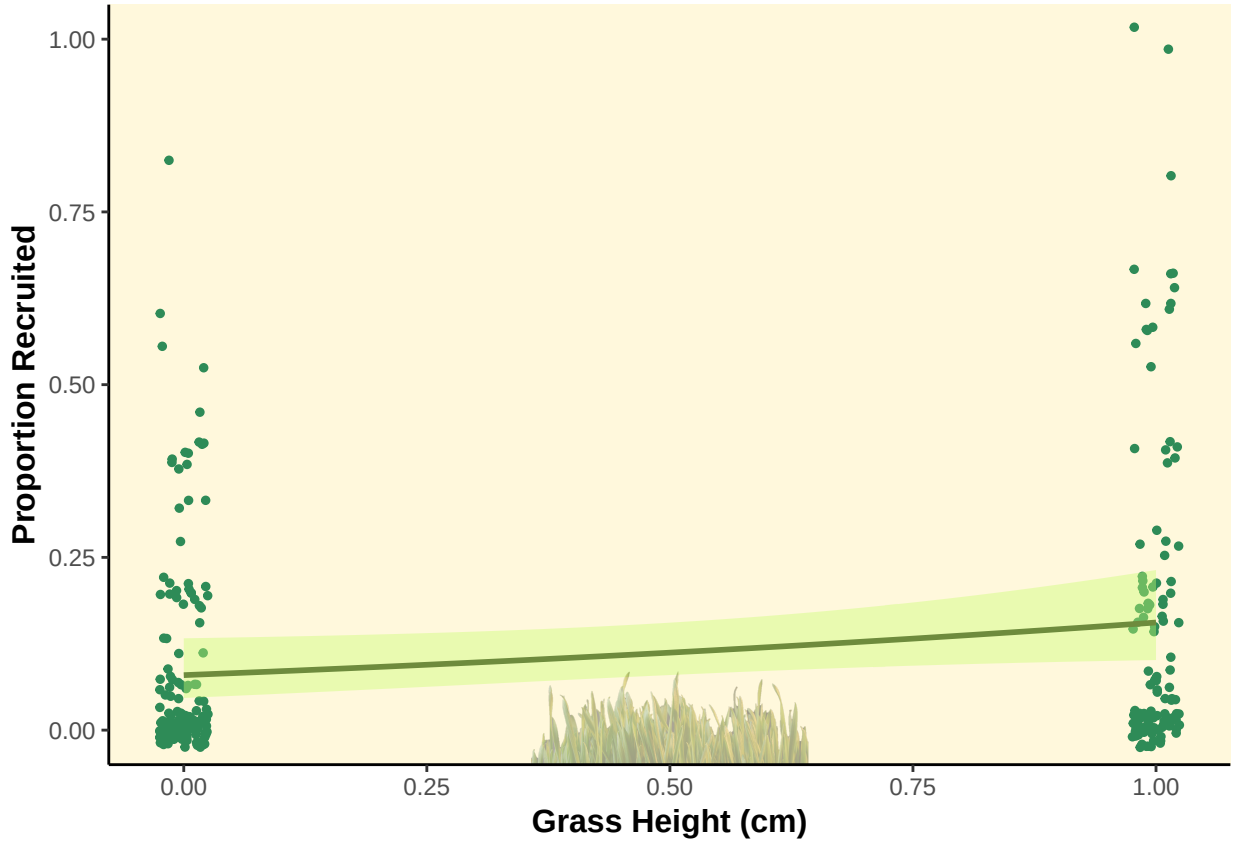
```
# read and inspect data
seeds2 <- read.csv("seeds.csv")

head(seeds2,4)
```

```
##   Site   Pile   DBH seedlings seeds recruits grass light
## 1   m1  m1.15 21.59         0    15        2     1  9.35
## 2   m1  m1.45  0.00         0    45        2     0 17.00
## 3   m1   m1.5 46.99         0     5        1     0  6.68
## 4  m10 m10.15  0.00         0    15        0     0  6.72
```

A. Select a Predictor Variable and Test Effect on Seedling Recruitment

```
# build model- chosen predictor: Grass
response <- cbind(seeds2$recruits, seeds2$seeds - seeds2$recruits)

seed.grass <- glm(response~seeds2$grass, family = "binomial")
```

a) Plot



It's clear here in the plot that the predictor variable was measured at 0 and 1, perhaps 0 and 1 cm. While computationally ok, what if we tried to imagine a more gradated measurement of grass height in these data? (note that this is diverges from the homework a bit here but I thought it would be a little fun).

```
#create uniform distribution of theoretical grass heights (between 0 and 1)
gradate <- runif(length(seeds2$grass), 0, 1)

#add this new vector to the data frame
seeds2$grad <- gradate

#create theoretical seed recruitment responses for each theoretical gradated grass height
#  predictors based on the parameters from the original model.
seeds2$grad.resp <- rbinom(length(seeds2$grad),
                           prob = plogis(-2.5593279 + .7280516 * seeds2$grad),
                           size = seeds2$seeds)

#make the new theoretical response a proportion, as with the original response, and bind
#  into columns as before.
seeds2$grad.prop <- seeds2$grad.resp/seeds2$seeds
response2 <- cbind(seeds2$grad.resp, seeds2$seeds)

#create new model based on theoretical gradation.
grad.grass <- glm(response2~seeds2$grad, family = "binomial")

#examine coefficients of our new gradated glm compared to the original glm
```

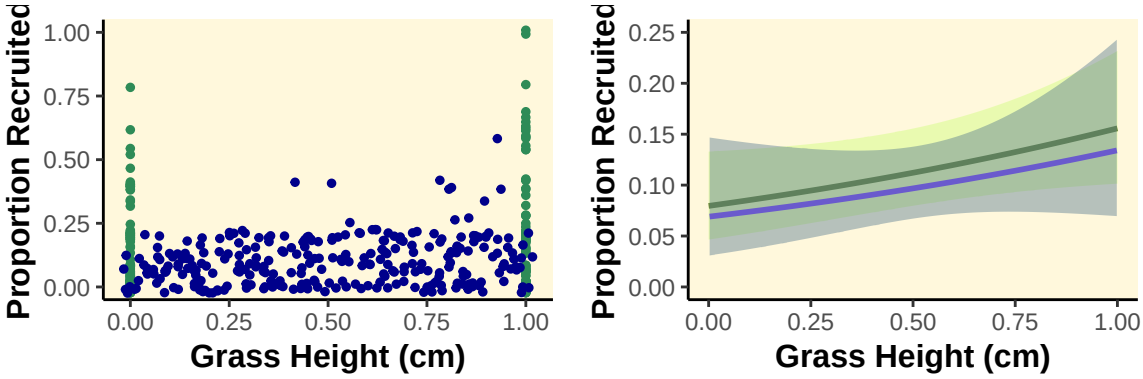
```
coef(grad.grass)
```

```
## (Intercept) seeds2$grad
##  -2.5617672   0.5724578
```

```
coef(seed.grass)
```

```
##  (Intercept) seeds2$grass
##   -2.5593279    0.7280516
```

So the theoretical gradated model checks out more or less, and if we were to visualize the two data sets, first just the data and the theoretical data, and then comparing the two models...



...the generated data are a good approximation, and the model slope and effect sizes are well matched.

b) Coefficients

```
coef(seed.grass)
```

```
##  (Intercept) seeds2$grass
##   -2.5593279    0.7280516
```

```
#transform into probability space
plogis(coef(seed.grass)[1])
```

```
## (Intercept)
##  0.07180233
```

```
coef(seed.grass)[2]/4
```

```
## seeds2$grass
##    0.1820129
```

c) Confidence Intervals

```
confint(seed.grass)
```

```
##                   2.5 %     97.5 %
## (Intercept)  -2.691223 -2.4322244
## seeds2$grass  0.558850  0.8988073
```

```
# transform into probability space
plogis(confint(seed.grass))
```

```
##                    2.5 %    97.5 %
```

```
## (Intercept)  0.06349324 0.0807482
## seeds2$grass 0.63618642 0.7107043
```

```
.55885/4
```

```
## [1] 0.1397125
```

```
.8988073/4
```

```
## [1] 0.2247018
```

B. Do your results support the hypothesis that your selected predictor variable has a significant effect on seedling germination?

There is good evidence that grass height has a significant positive effect on the probability of germination.The baseline value of the probability of seedling germination is 7.1% when there is no grass, and there is a strong positive effect of grass height on seedling germination, for every cm of growth there is 18.2% increase in germination or recruitment probability.The confidence interval for the slope parameter does not overlap with 0, and suggests a wide positive significant effect of grass height on germination by a margin of 14.0% - 22.5% increase in probability of survival per cm of grass height.The 95% CI for the baseline germination probability of seeds at 0 grass height is 6.3% - 8.1%.

Question 3: Mosquito Model Management

```
# read in and inspect data
el.mosquo <- read.csv("mosquito_data.csv")
head(el.mosquo,4)
```

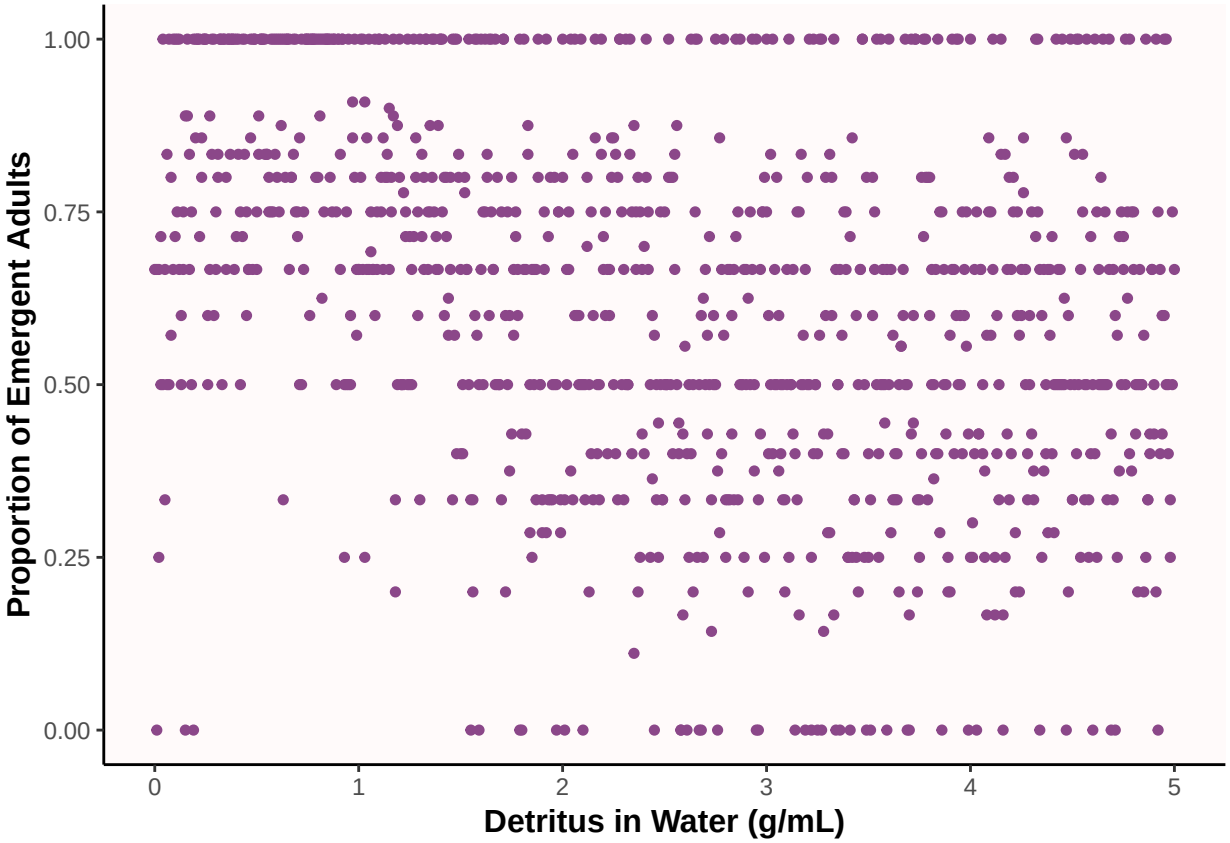
```
##   Emergent_adults Egg_Count Detritus
## 1               2         3     0.00
## 2               0         2     0.01
## 3               2         3     0.01
## 4               4         6     0.02
```

```
# build glm model
response3 <- cbind(el.mosquo$Emergent_adults, el.mosquo$Egg_Count - el.mosquo$Emergent_adults)
mos.mod <- glm(response3 ~ el.mosquo$Detritus, family = "binomial")
coef(mos.mod)
```

```
##        (Intercept) el.mosquo$Detritus
##          1.3240425         -0.3216083
```

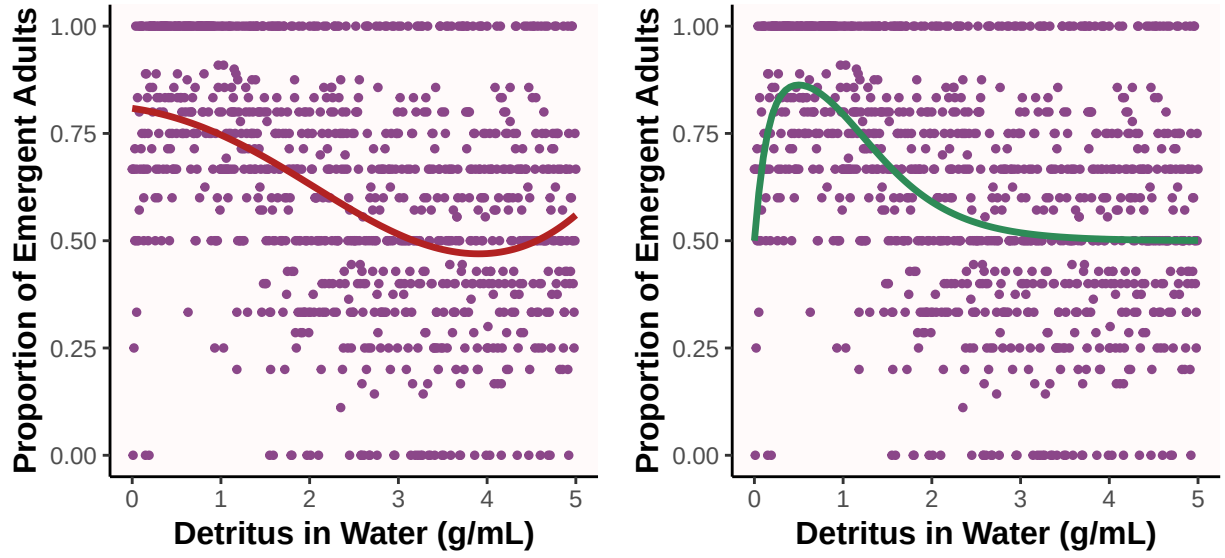
A. Plot Data

```
# create proportionate vector for visualizing properly
el.mosquo$prop <- el.mosquo$Emergent_adults/el.mosquo$Egg_Count
```



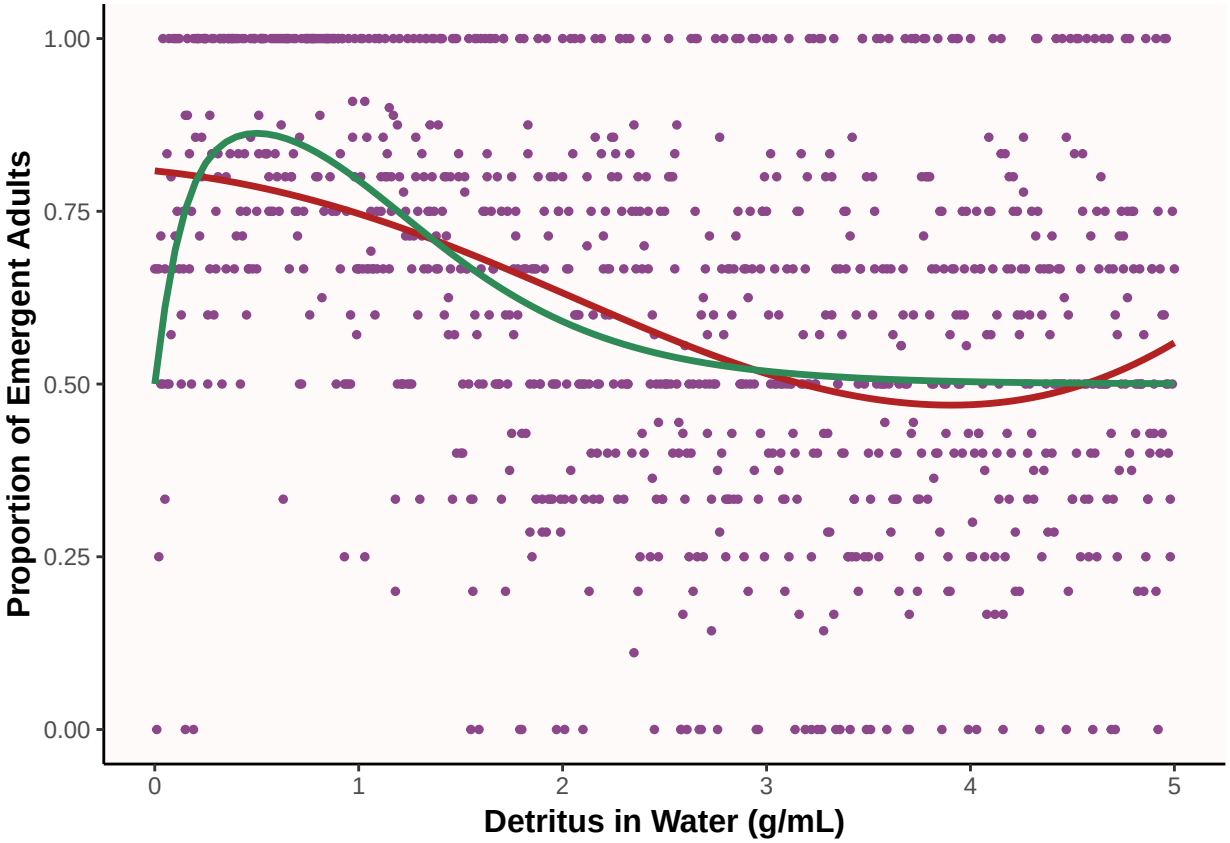
The data are quite noisy across most detritus concentrations. There appears to be a rough consolidation of high (.5-.75) proportion emergence at low detritus and a low (.25-.5) proportion emergence at high detritus.

B. Add Curves, Polynomial: $1.44 - 0.19x - 0.21x^2 + 0.04x^3$ and Ricker: $10xe^{-2x}$.



C. Biological Implications of each model.

Here is the added curves polynomial (left) and Ricker's (right). The polynomial implies a decline in mosquito emergence as detritus content in the water increases.The Ricker formula implies a very small window of acceptable low detritus content that there is a high probability of emergence within. The important difference between the two models is that the Ricker does not allow a proportion of emergence below .5, suggesting that if the assumed baseline emergence probability was 50%, detritus cannot have a negative effect on mosquito emergence below natural naive values. The polynomial suggests that there are two optima in detritus content, low detritus = high emergence, and high detritus = low emergence. The problem with the polynomial is the fact that due to the model parameters the model suggests that after a certain high detritus content emergence actually increases.



D. Likelihood of Each Model

```
# The x value here is number of successes in EACH trial- Emergent_adults
x <- el.mosquo$Emergent_adults

# the size here is the number of observations in EACH trial- Egg_Count
N <- el.mosquo$Egg_Count

# The probability is determined by the formula (poly or Rickers) with the predictor
#   variable as the dependent variable - Detritus. The formula has to be in
#   the logistic (plogis)
p1 <- plogis(1.44 - .19*el.mosquo$Detritus - .21*el.mosquo$Detritus^2 + .04*el.mosquo$Detritus^3)
p2 <- plogis(10*el.mosquo$Detritus*exp(-2*el.mosquo$Detritus))
```

There are 1000 trials, so to find total Likelihood we will have to multiply every independent event. If we take

the log of the product (log = TRUE), we can instead sum the events (-sum).

```
# For Ricker Equation
-sum(dbinom(x,
       size = N,
       prob = p2,
       log = TRUE))
```

```
## [1] 1385.847
```

```
# For Polynomial
-sum(dbinom(x,
            size = N,
            prob = p1,
            log = TRUE))
```

```
## [1] 1415.63
```

E. The data has a higher likelihood for which model?

The output of -sum(dbinom()) is the negative log-likelihood, and the minumum negative log likelihood is the maximum likelihood. So the polynomial output (NLL) (1415.63) is actually less likely than the Rickers (1385.85). The Ricker equation has the minimum log likelihood and therefore the maximum likelihood. This must be because the Ricker models the cluster of low-detritus/high emergence data better than the polynomial, and becuase it fits the high number of .500 emergence proportions.

Question 4: Power Play

1. Decide on Values for the true slope, intercept, and standard deviation (for the Normal Values).

System: Gecko (*Mediodactylus kotschyi*) Autotomy (tail loss) models with viper (a predator) abundance as the predictor. Autotomy expressed as a proportion.

Source: Itescu, Y. et al., 2017. *Intraspecific Competition, not Predation, Drives Lizard Tail Loss on Islands*, J. Animal Ecology 86.1, 66-74.

```
# take parameters from system
intercept <- .521
slope <- 1.167
sd <- .183 * sqrt(10)   # calculated from given standard error
```

2. Create a Predictor Variable, using either runif or seq functions.

```
vipers <- seq(from = 1, to = 50, by = 1)
```

3. Create a Vector of Sample Size.

```
sample_size <- c(3:80)  # 41 was the highest sample size of groups, assuming double
                        #    as total possible maximum.
```

4. Create an empty Vector to fill in with Results from the Power Analysis. Note that the number of elements in the vector needs to be equal in length of the sample size vector.

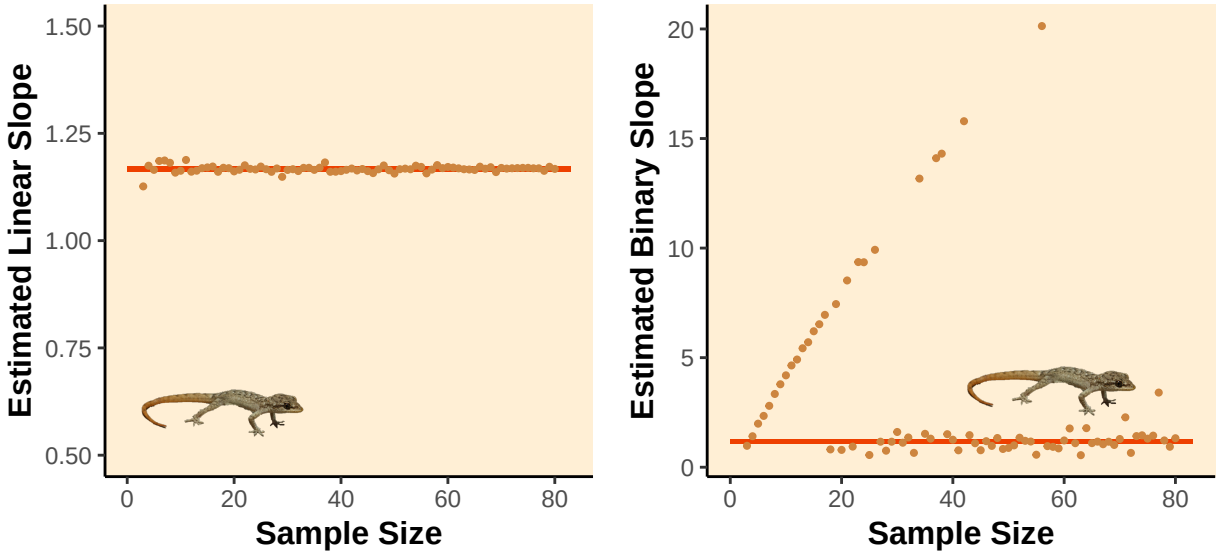
```
#power vector for both linear (1) and binary (2) data
power_vector1 <- rep(NA, times = length(sample_size))
power_vector2 <- rep(NA, times = length(sample_size))
```

5. Write a for loop that fills in the empty vector with results from simulated analyses.

```
  # linear regeression
for(i in 1:length(sample_size)) {
  tail.loss <- rnorm(sample_size[i],
                     mean = intercept + slope*seq(from = 1,
                                                  to = 50,
                                                  length = sample_size[i]),
                     sd = sd)
  tomy_mod1 <- glm(tail.loss ~ seq(from = 1, to = 50, length = sample_size[i]))
  power_vector1[i] <- coef(tomy_mod1)[2]
}
  # binary data
for(j in 1:length(sample_size)) {
  tail.loss2 <- rbinom(sample_size[j],
                     prob= plogis(intercept + slope*seq(from = 1,
                                                        to = 50,
                                                        length = sample_size[j])),
                     size= 40)  # The average number of geckos in a group is 40.
  response <- cbind(tail.loss2, 40 - tail.loss2)
  tomy_mod2 <- glm(response ~ seq(from = 1,
                                  to = 50,
                                  length = sample_size[j]),
                   family= "binomial")
  power_vector2[j] <- coef(tomy_mod2)[2]
}
```

6. Plot Results, with the filled in vector on the y axis and the sample size on the x axis.

```
#make data.frames of each power vector
geckos_power1 <- data.frame(sample_size, power_vector1)
colnames(geckos_power1) <- c("Sample_Size", "Estimated_Slope")

geckos_power2 <- data.frame(sample_size, power_vector2)
colnames(geckos_power2) <- c("Sample_Size", "Estimated_Slope")
```



A. How many samples do you need to accurately estimate the slope parameter in a binomial vs. linear regression? Use MSE to calculate the accuracy and precision of your estimate vs. the real value.

The Linear Data attenuates to the true parameter almost immediately, there is an original sinusoid that is still very close to the true slope, and even that disappears after one phase at around sample size = 10. The Binary Data is much more stochastic. There is a rising string of error between sample size = 5 and sample size = 49. General attenuation begins at sample size = 20, however it would appear that sample size = 50 is a good place to start here. Please note the different scaling of the y-axis on the plots above, the linear regression power plot is zoomed to better see the slight variations in the data.

```
# Calculate MSE for linear (1) and binary (2) data.
mean((slope - geckos_power1$Estimated_Slope)^2)
```

```
## [1] 6.51851e-05
```

```
mean((slope - geckos_power2$Estimated_Slope)^2)
```

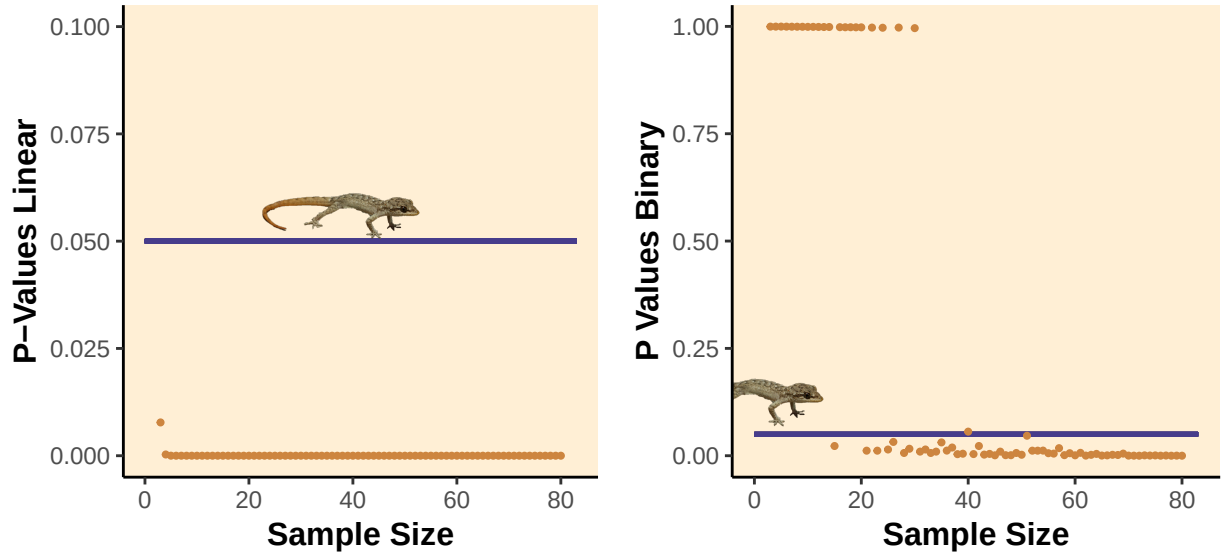
```
## [1] 19.88344
```

The MSE for the linear data is 6E-5, very small overall. The MSE for binary data is 16.6, very high. These values support the eyeball findings.

B. How many samples do you need to ensure a p_value<0.05 for binomial vs. linear regression?

```
# create p_value empty vectors
p_values1 <- rep(NA, times = length(sample_size))
p_values2 <- rep(NA, times = length(sample_size))

# run for loops again with new output fill

# linear regression
for(i in 1:length(sample_size)) {
  tail.loss <- rnorm(sample_size[i],
                     mean = intercept + slope*seq(from = 1,
                                                  to = 50,
                                                  length = sample_size[i]),
```

```
                      sd = sd)
  tomy_mod1 <- glm(tail.loss ~ seq(from = 1, to = 50, length = sample_size[i]))
  p_values1[i] <- summary(tomy_mod1)$coefficients[2,4]
}

# binary regression
for(j in 1:length(sample_size)) {
  tail.loss2 <- rbinom(sample_size[j],
                       prob= plogis(intercept + slope*seq(from = 1,
                                                          to = 50,
                                                          length = sample_size[j])),
                       size= 40)  # The average number of geckos in a group is 40.
  response <- cbind(tail.loss2, 40 - tail.loss2)
  tomy_mod2 <- glm(response ~ seq(from = 1,
                                  to = 50,
                                  length = sample_size[j]),
                   family= "binomial")
  p_values2[j] <- summary(tomy_mod2)$coefficients[2,4]
}

# data.frames for plotting
geckos_pvalue1 <- data.frame(sample_size, p_values1)
colnames(geckos_pvalue1) <- c("Sample_Size", "pvalues")

geckos_pvalue2 <- data.frame(sample_size, p_values2)
colnames(geckos_pvalue2) <- c("Sample_Size", "pvalues")
```



For the Linear Regression, the minimum sample size = 3 is all that's required to fall well below a p-value of .05. For the Binomial Regression, the first sample size that allows a p-value under .05 is sample size = 20, however the safe sample size is again after sample size = 50, as with the slope estimates.

C. In general, why is statistical power generally higher for continuous data than discrete response variables?

In continuous data, variance can be evaluated and accounted for, whereas in binary or discrete data, the attribute of variance is lost.It is also harder to determine effect size and power when the units are 0 and 1 than when a continuous string.