

Lab 6

Dr. Alejandro Molina-Moctezuma (based on Dr. Rushing's lab)

Count data and glmm

Up to this point, we have been working on linear models, in which the response variable has been a continuous, normally distributed variable. For today's lab we will focus on Generalized Linear Model (called so, because it generalizes the linear regression using a link function, see the slides for week 8!)

I will be borrowing from Dr. Clark Rushing <https://warnell.uga.edu/directory/people/clark-rushing> for this lab, taken from his **FANR 6750** course.

Dataset

Let's take a look at a data example to demonstrate this concept. The data are modified from [Ver Hoef and Boveng 2007](#) and include 400 observed counts of harbor seals from aerial surveys conducted in coastal Alaska.

Download the harbordata.csv from the files tab in Canvas, and load it into an object called harbordata.

```
harbordata = read.csv("harbordata.csv")
#library(FANR6750)
#data("harbordata")
```

After you have downloaded it, let's explore it

```
head(harbordata)
```

	Survey	Number	substrate	Reltolow	SITE
1	1	250	rock	0.0	Abbess I.
2	2	144	rock	3.2	Abbess I.
3	3	0	rock	4.9	Alava Bay

4	4	42	rock	3.6 Alava Bay
5	5	10	rock	0.0 Alava Bay
6	6	0	rock	1.6 Alava Bay

And finally... let's visualize it.

We will use the ggplot package. If you have never used it, let's download it:

You only need to install a package once in your computer.

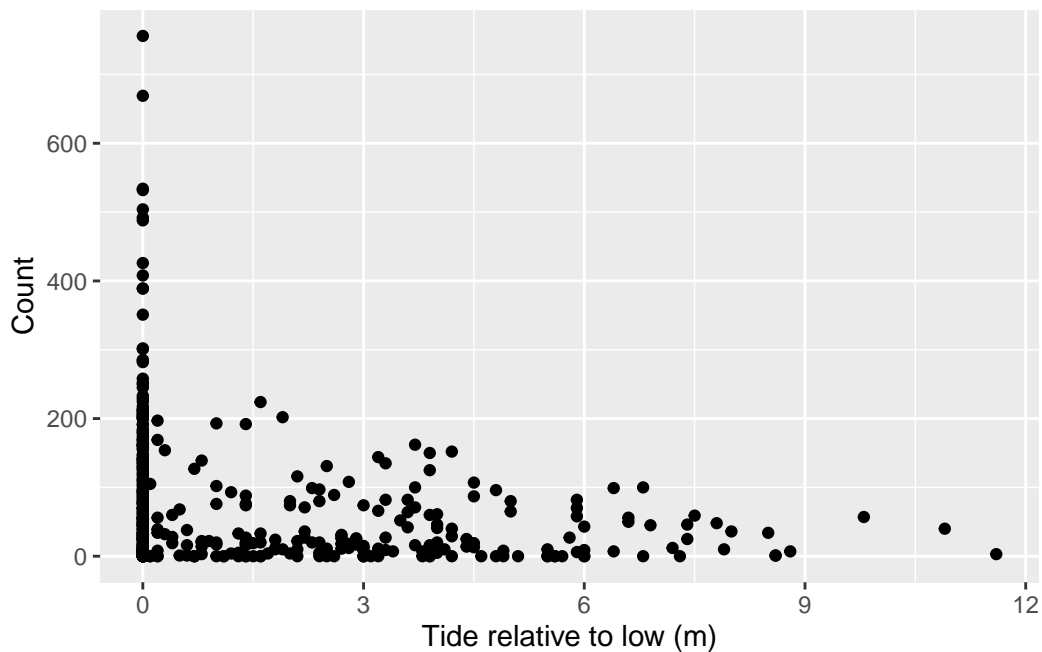
Then, let's load the package

```
library(ggplot2)
```

Warning: package 'ggplot2' was built under R version 4.3.3

And finally, let's plot the data:

```
ggplot() +  
  geom_point(data = harbordata, aes(x = Reltolow, y = Number)) +  
  scale_y_continuous("Count") +  
  scale_x_continuous("Tide relative to low (m)")
```



Remember, this data includes aerial observations of seals

Looking at the plot, it seems like there are a ton of observations in which 0 individuals were seen, and some in which up to 12 were seen. This might be a bit overdispersed for a Poisson distribution, but we'll worry about that later!

Running a simple linear model

First off, let's look at what would happen if we ran a simple linear model. Hopefully you know how to do this by now.

Run a linear model using the "lm" function, in which Number is the response variable, and Reltolow is the explanatory variable using the harbordata dataset.

```
mod1 <- lm(Number ~ Reltolow, data = harbordata)
summary(mod1)
```

Call:

```
lm(formula = Number ~ Reltolow, data = harbordata)
```

Residuals:

Min	1Q	Median	3Q	Max
-92.39	-63.27	-29.01	38.68	663.61

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	92.390	6.052	15.265	< 2e-16 ***
Reltolow	-11.524	2.172	-5.306	1.86e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 99.42 on 398 degrees of freedom

Multiple R-squared: 0.06607, Adjusted R-squared: 0.06372

F-statistic: 28.16 on 1 and 398 DF, p-value: 1.862e-07

While this model is significant, we know there's some stuff wrong with it (or, at least, you should know it!)

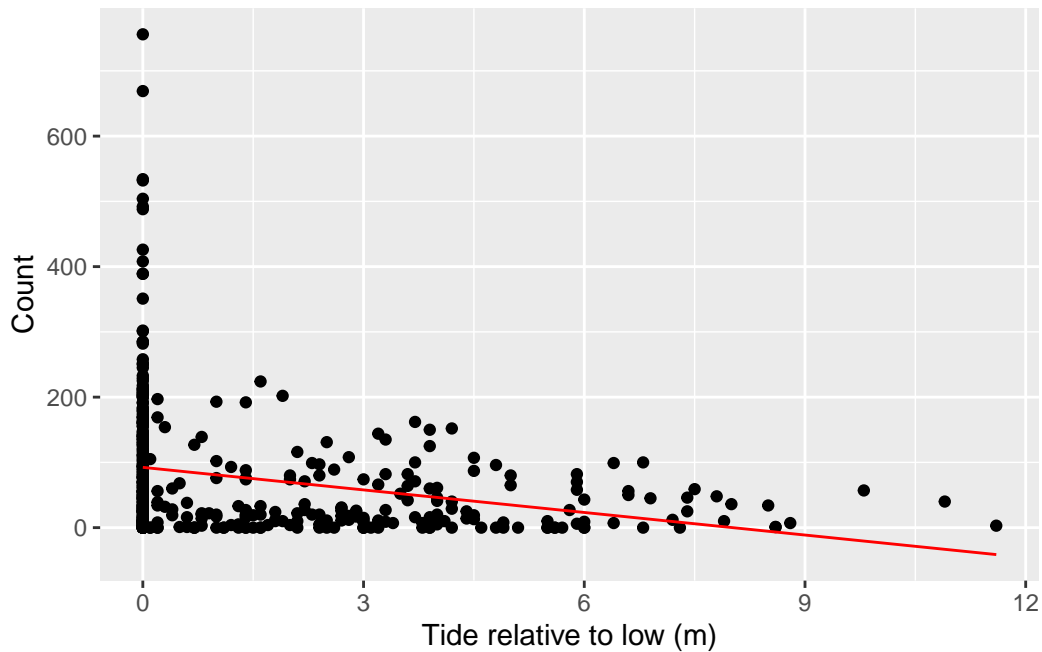
Let's add this model prediction to the plot:

```

model_pred <- predict(mod1)

ggplot() +
  geom_point(data = harbordata, aes(x = Reltolow, y = Number)) +
  geom_line(aes(harbordata$Reltolow, model_pred), color= 'red') +
  scale_y_continuous("Count") +
  scale_x_continuous("Tide relative to low (m)")

```



Uggh, this is not good.

But remember! We can use a link function to estimate a different response. In this case:

$$\log(\lambda_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}$$

Now, that doesn't have a random component. In this case, the random component is given by:

$$y_i \sim \text{Poisson}(\lambda_i)$$

One last problem... that random component uses λ_i , not $\log(\lambda_i)$. Well, to obtain λ_i we simply use:

$$\lambda_i = e^{\beta_0} + e^{\beta_1 x_{i1}} + e^{\beta_2 x_{i2}} + \dots + e^{\beta_z x_{zp}}$$

Fortunately, we don't have to worry about anything other than knowing which link function to use.

Let's run the same model, but using a GLM:

```
fm2 <- glm(Number ~ Reltolow, family = poisson(link = "log"), data = harbordata)
summary(fm2)
```

Call:

```
glm(formula = Number ~ Reltolow, family = poisson(link = "log"),
    data = harbordata)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	4.563811	0.006535	698.37	<2e-16 ***
Reltolow	-0.229939	0.003864	-59.51	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 42725 on 399 degrees of freedom
 Residual deviance: 37943 on 398 degrees of freedom
 AIC: 39829

Number of Fisher Scoring iterations: 5

New things in this function: family, and link!

This model is looking at the effect of flow, but maybe substrate type is also affecting the number of observed seals. Let's run it:

```
fm3 <- glm(Number ~ substrate + Reltolow, family = poisson(link = "log"),
            data = harbordata)
summary(fm3)
```

Call:

```
glm(formula = Number ~ substrate + Reltolow, family = poisson(link = "log"),
    data = harbordata)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	5.422800	0.013433	403.69	<2e-16 ***
substraterock	-1.006149	0.015286	-65.82	<2e-16 ***
substratesand	-1.284207	0.035859	-35.81	<2e-16 ***
Reltolow	-0.191000	0.003858	-49.51	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 42725 on 399 degrees of freedom
 Residual deviance: 34227 on 396 degrees of freedom
 AIC: 36116

Number of Fisher Scoring iterations: 6

The GLM summary gives us an AIC value. Based on this value, which of the two models is better?



Q1. 2 pts. Which of the two models do you think explains the observed data better? Why? save the model in a new object called "smodel"



A1. The second model fm3 is better because the AIC which compares the model to the null model is lower than that of the fm2.

```
smodel<-fm3
```

Let's use predict to plot the better model. We will only plot it for one type of substrate

```
predData <- data.frame(Reltolow = seq(min(harbordata$Reltolow),
                                     max(harbordata$Reltolow), length = 10000),
                      substrate= 'ice')

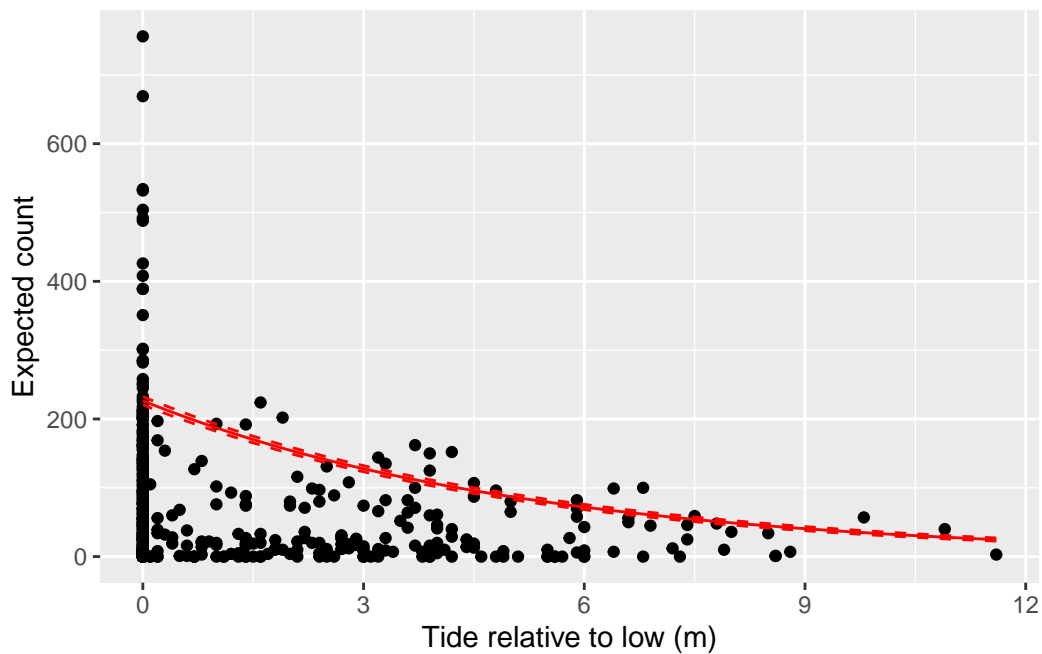
pred.link <- predict(smodel, newdata = predData, se.fit = TRUE)
predData$lambda <- exp(pred.link$fit) # exp is the inverse-link function
predData$lower <- exp(pred.link$fit - 1.96 * pred.link$se.fit)
```

```

predData$upper <- exp(pred.link$fit + 1.96 * pred.link$se.fit)

ggplot() +
  geom_point(data = harbordata, aes(x = Reltolow, y = Number)) +
  geom_path(data = predData, aes(x = Reltolow, y = lambda), color= 'red') +
  geom_ribbon(data = predData, aes(x = Reltolow, ymin = lower, ymax = upper),
            fill = NA, color = "red", linetype= 'dashed') +
  scale_x_continuous("Tide relative to low (m)") +
  scale_y_continuous("Expected count")

```



Do you think this looks OK? Look at the (tiny) confident intervals compared to the great spread of the data!

Remember that I mentioned that the data looked a bit overdispersed? In Poisson, the mean and the variance are equal. That's why as λ increases, so does the spread of the data.



Q2. 4 pts. Estimate the mean and the variance of the "number" variable in the dataframe. Do you think there is overdispersion?



Q2. The mean is 74.07 and the variance is 10557.64. Because the variance is so large, I think there is overdispersion.

```
mean(harbordata$Number)
```

```
[1] 74.07
```

```
var(harbordata$Number)
```

```
[1] 10557.64
```

A long time ago, I talked about the use of negative binomial instead of Poisson when there is overdispersion. It allows us a lot more flexibility!

To run a glm using the negative binomial, we need a new package. Download the “MASS” package.

And now, let’s run the code!

We will

1. Run the NB model
2. Look at the summary
3. Obtain the predictors to plot
4. Plot!
5. Add the Poisson model to the plot

```
library(MASS)
nb1 <- glm.nb(Number ~ substrate + Reltolow, data = harbordata)
summary(nb1)
```

Call:

```
glm.nb(formula = Number ~ substrate + Reltolow, data = harbordata,
       init.theta = 0.4661173743, link = log)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	5.4190	0.2933	18.477	< 2e-16 ***
substraterock	-1.0372	0.3077	-3.371	0.000749 ***
substratesand	-1.3827	0.4500	-3.072	0.002123 **
Reltolow	-0.1608	0.0327	-4.917	8.78e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(0.4661) family taken to be 1)

Null deviance: 532.29 on 399 degrees of freedom
Residual deviance: 484.04 on 396 degrees of freedom
AIC: 4009.5

Number of Fisher Scoring iterations: 1

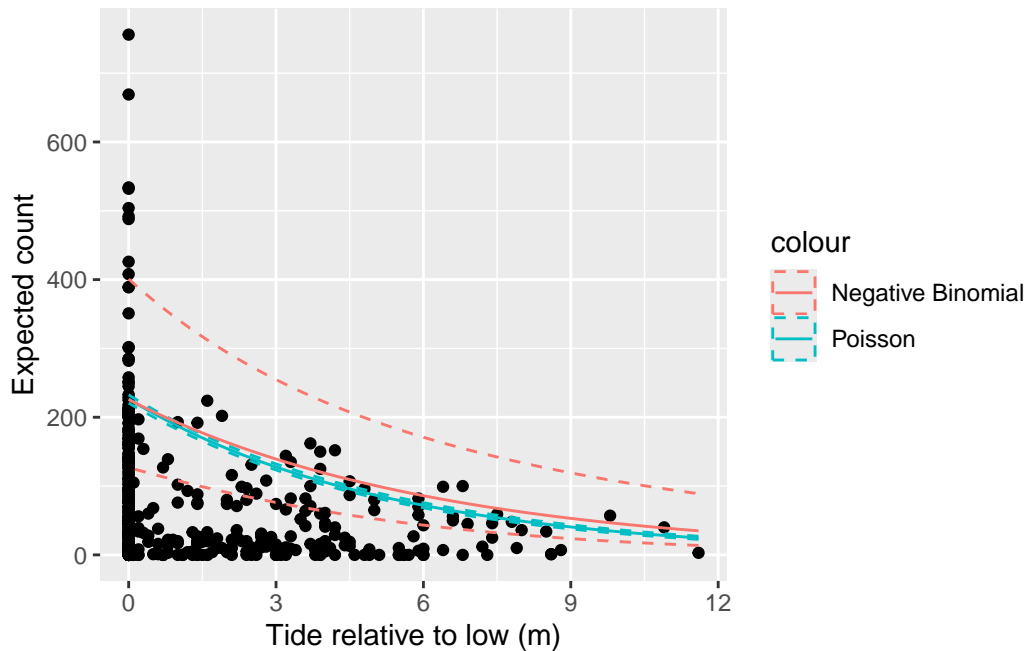
Theta: 0.4661
Std. Err.: 0.0319

2 x log-likelihood: -3999.4500

```
predDatanb <- data.frame(Reltolow = seq(min(harbordata$Reltolow),
                                         max(harbordata$Reltolow), length = 10000),
                        substrate= 'ice')

pred.linknb <- predict(nb1, newdata = predDatanb, se.fit = TRUE)
predDatanb$lambda <- exp(pred.linknb$fit) # exp is the inverse-link function
predDatanb$lower <- exp(pred.linknb$fit - 1.96 * pred.linknb$se.fit)
predDatanb$upper <- exp(pred.linknb$fit + 1.96 * pred.linknb$se.fit)

ggplot() +
  geom_point(data = harbordata, aes(x = Reltolow, y = Number)) +
  geom_path(data = predData, aes(x = Reltolow, y = lambda, color= 'Poisson')) +
  geom_ribbon(data = predData, aes(x = Reltolow, ymin = lower,
                                  ymax = upper, color= 'Poisson'),
            fill = NA, linetype= 'dashed') +
  geom_path(data = predDatanb, aes(x = Reltolow, y = lambda,
                                   color= 'Negative Binomial')) +
  geom_ribbon(data = predDatanb, aes(x = Reltolow, ymin = lower, ymax = upper,
                                   color= 'Negative Binomial'), fill = NA,
            linetype= 'dashed') +
  scale_x_continuous("Tide relative to low (m)") +
  scale_y_continuous("Expected count")
```



Even though the CI is larger, this better represents the real world observations!

Download the `alligatordata.csv` file.



Q3. 12 pts. Estimate clutch size as a function of length, habitat, and length + habitat using both methods (Poisson and NB)

```
alligator=read.csv("alligatordata.csv")
head(alligator)
```

	Nest	Eggs	Length	Habitat
1	1	62	13.8	Brackish marsh
2	2	22	8.1	Brackish marsh
3	3	41	10.7	Intermediate marsh
4	4	37	11.1	Brackish marsh
5	5	23	7.0	Intermediate marsh
6	6	29	7.3	Swamp

```
alligator$Habitat = as.factor(alligator$Habitat)
summary(alligator)
```

Nest	Eggs	Length	Habitat
------	------	--------	---------

Min. : 1.00	Min. :20.00	Min. : 6.00	Brackish marsh :150
1st Qu.: 88.25	1st Qu.:31.00	1st Qu.: 8.10	Intermediate marsh: 78
Median :175.50	Median :39.00	Median :10.60	Swamp :122
Mean :175.50	Mean :40.91	Mean :10.53	
3rd Qu.:262.75	3rd Qu.:48.75	3rd Qu.:12.90	
Max. :350.00	Max. :82.00	Max. :14.90	

```
am_pois <- glm(Eggs ~ Length + Habitat + Length:Habitat, family = poisson(link = "log"), data = alligator)
summary(am_pois)
```

Call:

```
glm(formula = Eggs ~ Length + Habitat + Length:Habitat, family = poisson(link = "log"),
    data = alligator)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	2.855675	0.058373	48.921	<2e-16 ***
Length	0.076246	0.005266	14.478	<2e-16 ***
HabitatIntermediate marsh	0.059500	0.097301	0.612	0.541
HabitatSwamp	-0.053845	0.082606	-0.652	0.515
Length:HabitatIntermediate marsh	0.004644	0.008254	0.563	0.574
Length:HabitatSwamp	0.005720	0.007405	0.772	0.440

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 1342.39 on 349 degrees of freedom
 Residual deviance: 608.48 on 344 degrees of freedom
 AIC: 2547.6

Number of Fisher Scoring iterations: 4

```
am_nb <- glm.nb(Eggs ~ Length + Habitat + Length:Habitat, data = alligator)
summary(am_nb)
```

Call:

```
glm.nb(formula = Eggs ~ Length + Habitat + Length:Habitat, data = alligator,
    init.theta = 55.03075957, link = log)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	2.862381	0.075015	38.157	<2e-16 ***
Length	0.075611	0.006887	10.980	<2e-16 ***
HabitatIntermediate marsh	0.053040	0.126509	0.419	0.675
HabitatSwamp	-0.053845	0.105997	-0.508	0.611
Length:HabitatIntermediate marsh	0.005259	0.010920	0.482	0.630
Length:HabitatSwamp	0.005721	0.009696	0.590	0.555

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(55.0308) family taken to be 1)

Null deviance: 766.96 on 349 degrees of freedom
Residual deviance: 349.10 on 344 degrees of freedom
AIC: 2482.3

Number of Fisher Scoring iterations: 1

Theta: 55.03
Std. Err.: 9.79

2 x log-likelihood: -2468.269



Q4. 4 pts. Choose the best model from Q3 and describe it



A4. Since the negative binomial model's AIC is lower than that of the poisson model.



Q5. 6 pts. Use the poisson regression model to plot the relationship between number of eggs and female length, for all three habitat types, on the same graph. The graph should include points color-coded by habitat, a legend and confidence intervals

```

#Set up data frame for predictor data for each habitat type
predData2_brack = data.frame(Length = seq(min(alligator$Length),
                                          max(alligator$Length), length = 10000),
                             Habitat = as.factor('Brackish marsh'),
                             show.legend = TRUE)

predData2_int = data.frame(Length = seq(min(alligator$Length),
                                          max(alligator$Length), length = 10000),
                             Habitat = as.factor('Intermediate marsh'))

predData2_swamp = data.frame(Length = seq(min(alligator$Length),
                                          max(alligator$Length), length = 10000),
                             Habitat = as.factor('Swamp'))

#Implement prediction for Brackish Marsh
pred.link2_brack <- predict(am_pois, newdata = predData2_brack, se.fit = TRUE)
predData2_brack$lambda <- exp(pred.link2_brack$fit) # exp is the inverse-link function
predData2_brack$lower <- exp(pred.link2_brack$fit - 1.96*pred.link2_brack$se.fit)
predData2_brack$upper <- exp(pred.link2_brack$fit + 1.96*pred.link2_brack$se.fit)

#Implement prediction for Intermediate Swamp
pred.link2_int <- predict(am_pois, newdata = predData2_int, se.fit = TRUE)
predData2_int$lambda <- exp(pred.link2_int$fit)
predData2_int$lower <- exp(pred.link2_int$fit - 1.96*pred.link2_int$se.fit)
predData2_int$upper <- exp(pred.link2_int$fit + 1.96*pred.link2_int$se.fit)

#Implement prediction for Swamp
pred.link2_swamp <- predict(am_pois, newdata = predData2_swamp, se.fit = TRUE)
predData2_swamp$lambda <- exp(pred.link2_swamp$fit)
predData2_swamp$lower <- exp(pred.link2_swamp$fit - 1.96*pred.link2_swamp$se.fit)
predData2_swamp$upper <- exp(pred.link2_swamp$fit + 1.96*pred.link2_swamp$se.fit)

#Extract all data points into separate data points for each habitat type
brack_df = data.frame(alligator[which(alligator$Habitat == "Brackish marsh"), ])
int_df = data.frame(alligator[which(alligator$Habitat == "Intermediate marsh"), ])
swamp_df = data.frame(alligator[which(alligator$Habitat == "Swamp"), ])

## Begin plot
ggplot() +
#plot all data points by color
geom_point(data = brack_df, aes(x = Length, y = Eggs, color='Brackish Marsh'),
           shape=1) +

```

```

geom_point(data = int_df, aes(x = Length, y = Eggs, color='Intermediate Marsh'),
           shape=2) +
geom_point(data = swamp_df, aes(x = Length, y = Eggs, color='Swamp'),
           shape=3) +

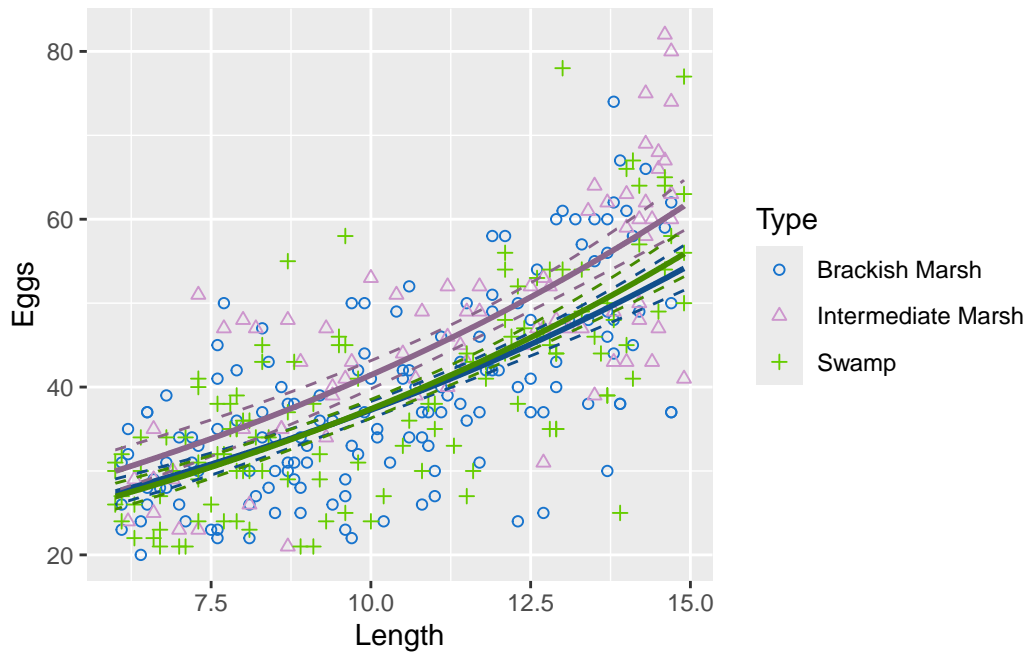
#plot poisson regression model and confidence intervals for Brackish Marsh
geom_path(data = predData2_brack, aes(x = Length, y = lambda),
          linewidth=1, color= 'dodgerblue4') +
geom_ribbon(data = predData2_brack, aes(x = Length,
                                       ymin = lower,
                                       ymax = upper),
           fill = NA, linewidth=0.5, color = "dodgerblue4", linetype= 'dashed') +

#plot poisson regression model and confidence intervals for Intermediate Marsh
geom_path(data = predData2_int, aes(x = Length, y = lambda),
          linewidth = 1, color= 'plum4') +
geom_ribbon(data = predData2_int, aes(x = Length,
                                       ymin = lower,
                                       ymax = upper),
           linewidth=0.5, fill = NA, color = "plum4", linetype= 'dashed') +

#plot poisson regression model and confidence intervals for Swamp
geom_path(data = predData2_swamp, aes(x = Length, y = lambda),
          linewidth = 1, color= 'chartreuse4') +
geom_ribbon(data = predData2_swamp, aes(x = Length,
                                       ymin = lower,
                                       ymax = upper),
           linewidth=0.5, fill = NA, color = "chartreuse4", linetype= 'dashed') +

#Change color of standard data for person preference and visibility
scale_color_manual(name = 'Type',
                   values = c('Brackish Marsh' = 'dodgerblue3',
                              'Intermediate Marsh' = 'plum3',
                              'Swamp' = 'chartreuse3'))

```



EXTRA CREDIT: up to 4 pts. Question 5 might be a bit hard for some of you. If you figure out a good way to plot it, start a new discussion in Canvas and share your plot/code. There are many ways to plot this, so, if you have an alternative way of plotting it, you can upload it even if someone else has done it already. You can also modify previous responses if you have suggestions on how to do it. In order to get the extra credit, you need to post by end of day Friday 29th. This is so we have a chance to help students working on this Question.

Total points: 28