**Section 4.2 – Poisson regression models for count responses**

**Section 4.2.1 – Model for mean: Log link**

Suppose the mean parameter of a Poisson distribution is now dependent on a function of explanatory variables. When there is only one explanatory variable x, a model is

μ = β0 + β1x

Depending on the value of the regression parameters and x, we could obtain a negative value for μ which would not make sense for a count! Instead, we can use

log(μ) = β0 + β1x

which alternatively can be written as

μ = exp(β0 + β1x)

Now, μ is guaranteed to be greater than 0. This is referred to a Poisson regression model.

When needed, we can emphasize that the mean changes as a function of the variable x for the ith observation with

μi = exp(β0 + β1xi)

If there are p explanatory variables, we can write the model as

μ = exp(β0 + β1x1 + βpxp)

or

log(μ) = β0 + β1x1 + βpxp

Generalized linear model

A Poisson regression model is a generalized linear model with the following components:

1. Random: Y has a Poisson distribution
2. Systematic: β0 + β1x1 + βpxp
3. Link: log

**Section 4.2.2 – Parameter estimation and inference**

Maximum likelihood estimation is used again to find the MLEs. Suppose my sample is denoted as (yi, xi1, …, xip) with i = 1, …, n. The likelihood function is



where . For most situations, the likelihood function needs to be maximized using iterative numerical procedures. The glm() function in R completes this maximization where the family argument needs to be given as poisson(link = log).

The covariance matrix for the regression parameter estimators follows from using standard likelihood procedures as outlined in Appendix B. Wald and LR-based inference methods are performed in essentially the same ways as what was used for logistic regression.

Example: Horseshoe crabs and satellites (Horseshoe.R, Horseshoe.csv)

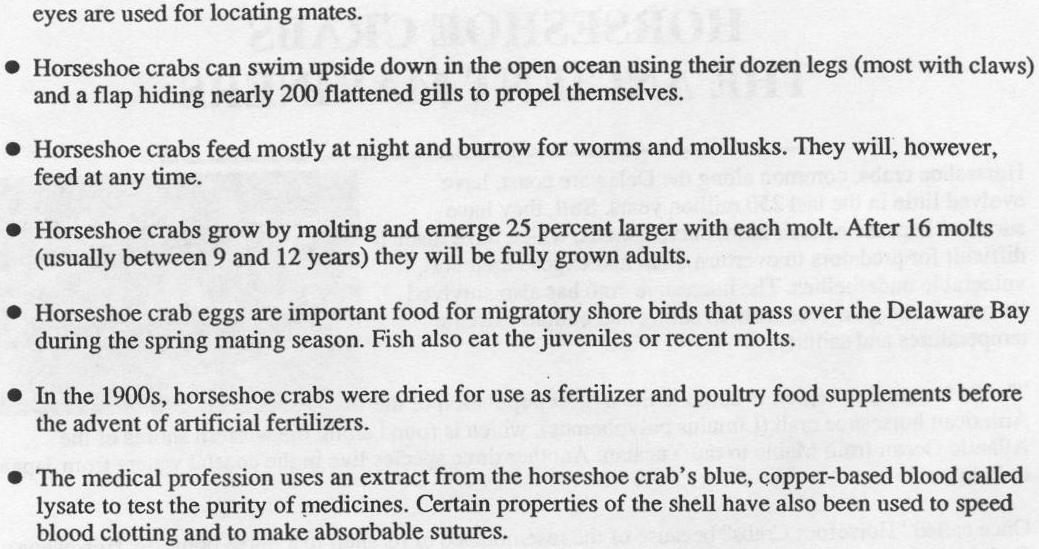
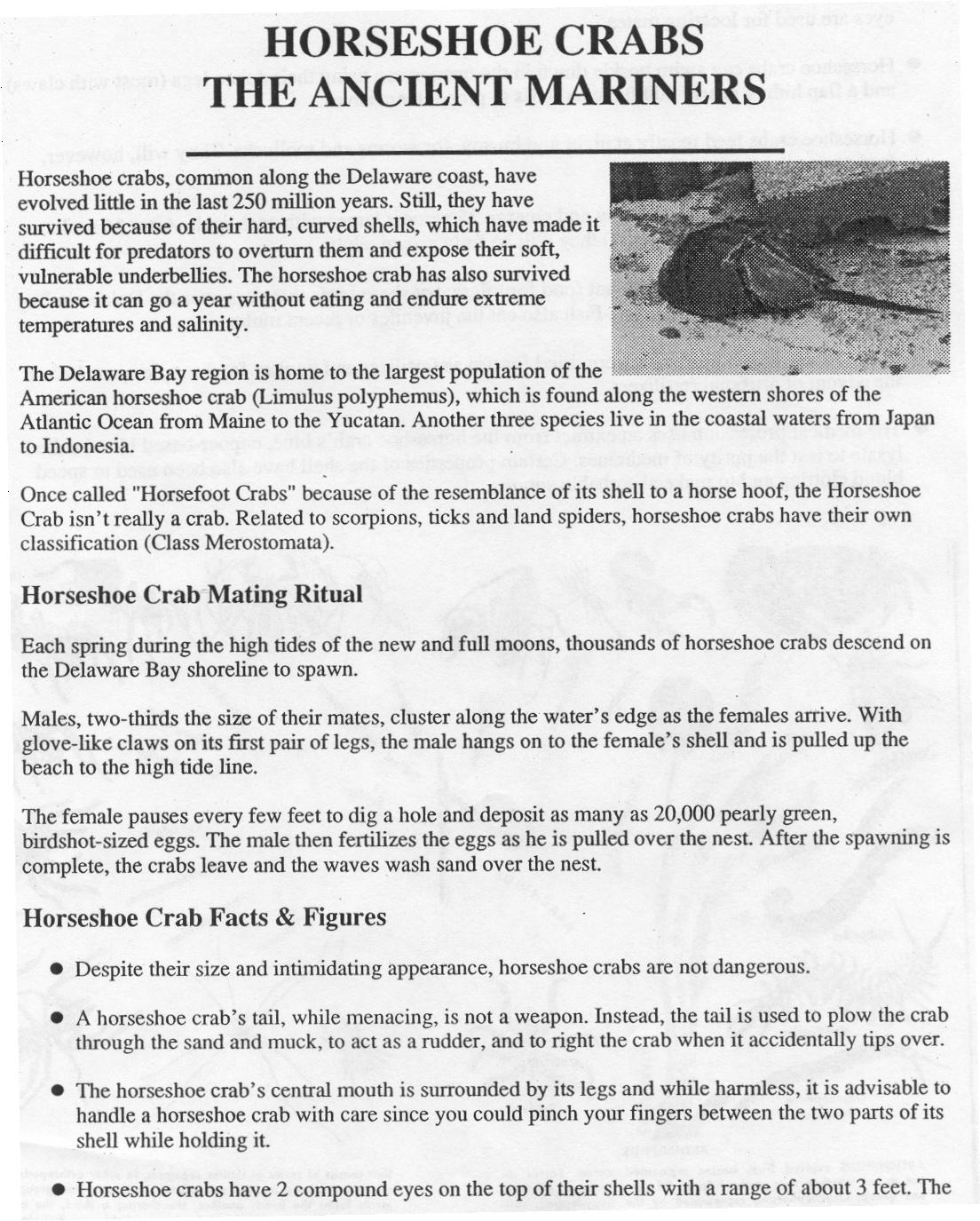
Information about horseshoe crabs:

* The television show Nature –

<https://youtu.be/BuOguPBmqrI?t=1204>

* An NPR story –[www.npr.org/templates/story/story.php?storyId=106489695](http://www.npr.org/templates/story/story.php?storyId=106489695)

Below are some additional details:



The purpose of this example is to determine if the shell width of a female (x) is related to the number of satellites (Y) she has around her. A simple Poisson regression model:

log(μ) = β0 + β1x

where

Y = Number of satellites

x = Shell width of female (measured in cm)

can be used to estimate the mean number of satellites given a shell width.

Below is how I read in the data:

> crab <- read.csv(file = "c:\\data\\horseshoe.csv”)

> head(crab)

satellite width

1 8 28.3

2 0 22.5

3 9 26.0

4 0 24.8

5 4 26.0

6 0 23.8

Below is how I estimate the model:

> mod.fit <- glm(formula = satellite ~ width, data = crab,

family = poisson(link = log))

> summary(mod.fit)

Call:

glm(formula = satellite ~ width, family = poisson(link = log), data = crab)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.8526 -1.9884 -0.4933 1.0970 4.9221

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -3.30476 0.54224 -6.095 1.1e-09 \*\*\*

width 0.16405 0.01997 8.216 < 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: 632.79 on 172 degrees of freedom

Residual deviance: 567.88 on 171 degrees of freedom

AIC: 927.18

Number of Fisher Scoring iterations: 6

The estimated Poisson regression model is



where x represents the number of satellites. The model can be written also as:



Questions:

* What happens to the estimated mean number of satellites as the width increases?
* What type of female crabs do male crabs prefer?

Now that we have the estimated model, many of the basic types of analyses performed in Chapters 2 and 3 can be performed here. The R code used is very similar as well.

1. What does a Wald test for β1 conclude?
2. How can we perform a LRT for an explanatory variable?
3. How can we estimate the expected number of satellites when the shell width is 23?



1. How can we find a Wald confidence interval for μ? Are there any worries about interval limits being outside of the appropriate numerical range? If so, what can be done?
2. How can we find a Profile LR confidence interval for μ?

> K <- matrix(data = c(1, 23), nrow = 1, ncol = 2)

> K

[,1] [,2]

[1,] 1 23

> # Calculate -2log(Lambda)

> linear.combo <- mcprofile(object = mod.fit, CM = K)

> # CI for beta\_0 + beta\_1 \* x

> ci.logmu.profile <- confint(object = linear.combo, level

= 0.95)

> ci.logmu.profile

mcprofile - Confidence Intervals

level: 0.95

adjustment: single-step

Estimate lower upper

C1 0.468 0.284 0.647

> ci.logmu.profile$confint

lower upper

C1 0.2841545 0.6471545

> exp(ci.logmu.profile)

mcprofile - Confidence Intervals

level: 0.95

adjustment: single-step

Estimate lower upper

C1 1.6 1.33 1.91

The 95% interval is 1.33 < μ < 1.91, which is quite similar to the Wald interval.

1. How could you estimate the covariance matrix and print it in R?
2. How could you include some type of transformation of an explanatory variable(s) in the model?

When there is only one explanatory variable in the model, we can easily examine the estimated model through a plot:

> plot(x = crab$width, y = crab$satellite, xlab = "Width (cm)", ylab = "Number of satellites", main = "Horseshoe crab data set \n with Poisson regression model fit", panel.first = grid())

> curve(expr = exp(mod.fit$coefficients[1] + mod.fit$coefficients[2]\*x), col = "red", add = TRUE, lty = "solid")

> # Can also use this to plot the model:

> # curve(expr = predict(object = mod.fit, newdata = data.frame(width = x), type ="response"), col = "red", add = TRUE, lty = 1)

> # Function to find confidence interval

> ci.mu <- function(newdata, mod.fit.obj, alpha) {

lin.pred.hat <- predict(object = mod.fit.obj, newdata = newdata, type = "link", se = TRUE)

lower <- exp(lin.pred.hat$fit - qnorm(1 - alpha/2) \*

lin.pred.hat$se)

upper <- exp(lin.pred.hat$fit + qnorm(1 - alpha/2) \*

lin.pred.hat$se)

list(lower = lower, upper = upper)

}

> # Test

> ci.mu(newdata = data.frame(width = 23), mod.fit.obj = mod.fit, alpha = 0.05)

$lower

1

1.332135

$upper

1

1.915114

> # Add confidence interval bands

> curve(expr = ci.mu(newdata = data.frame(width = x), mod.fit.obj = mod.fit, alpha = 0.05)$lower, col = "blue", add = TRUE, lty = "dotdash")

> curve(expr = ci.mu(newdata = data.frame(width = x), mod.fit.obj = mod.fit, alpha = 0.05)$upper, col = "blue", add = TRUE, lty = "dotdash")

> legend(x = 21, y = 14, legend = c("Poisson regression model", "95% individual C.I."), bty = "n", lty = c("solid", "dotdash"), col = c("red", "blue"))



There is somewhat of an upward trend as width increases.

You may be alarmed by the number of plotting points far from the estimated model. However, remember that the model is trying to estimate the “average” number of satellites given the width. The plot below examines this more closely where I have added the average number of satellites for a width group.

> # Put the data into groups

> min(crab$width)

[1] 21

> max(crab$width)

[1] 33.5

> crab$groups <- cut(x = crab$width, c(20, seq(from = 23.25, to = 29.25, by = 1), 34))

> head(crab)

satellite width groups

1 8 28.3 (28.2,29.2]

2 0 22.5 (20,23.2]

3 9 26.0 (25.2,26.2]

4 0 24.8 (24.2,25.2]

5 4 26.0 (25.2,26.2]

6 0 23.8 (23.2,24.2]

> tail(crab)

satellite width groups

168 2 26.2 (25.2,26.2]

169 3 26.1 (25.2,26.2]

170 4 29.0 (28.2,29.2]

171 0 28.0 (27.2,28.2]

172 0 27.0 (26.2,27.2]

173 0 24.5 (24.2,25.2]

> #Find the average number of satellites per group and plot

> ybar <- aggregate(x = satellite ~ groups, data =

crab, FUN = mean)

> xbar <- aggregate(x = width ~ groups, data = crab,

FUN = mean)

> count <- aggregate(x = satellite ~ groups, data = crab,

FUN = length)

groups satellite mean.width count

1 (20,23.2] 1.000000 22.69286 14

2 (23.2,24.2] 1.428571 23.84286 14

3 (24.2,25.2] 2.392857 24.77500 28

4 (25.2,26.2] 2.692308 25.83846 39

5 (26.2,27.2] 2.863636 26.79091 22

6 (27.2,28.2] 3.875000 27.73750 24

7 (28.2,29.2] 3.944444 28.66667 18

8 (29.2,34] 5.142857 30.40714 14

> points(x = xbar$width, y = ybar$satellite, pch = 17, col

= "darkgreen", cex = 2)

> legend(x = 21, y = 14, legend = c("Poisson regression

model", "95% individual C.I.", "Sample mean"), bty =

"n", lty = c("solid", "dotdash", NA), col = c("red",

"blue", "darkgreen"), pch = c(NA, NA, 17))



Notice how the red line goes through the middle of the green triangles (group means).

Comments:

* Does the model fit the data well? This is a difficult question to answer solely based on this plot. At the very least, it looks like the model is doing what it is supposed to do.
* What if we used different groupings? My program gives details on how to use a more general method (based on quantiles of the data) to break the data up into groups. Below is the corresponding plot:



This plot does not look as good as the previous plot. There is more variability of the green triangles around the red line. Overall, this helps to illustrate that different groupings can produce somewhat different results, and this method should not be used alone to judge if a model is fitting well.

* One can construct a similar type of plot to provide an ad-hoc assessment of how well a logistic regression model fits the data. This is especially useful in situations when the explanatory variable is continuous.