ST623 Midterm

Nick Sun

Question 1

Part a.

The binomial pmf can be factored into

$$f(y) = exp\left[yln\left(\frac{p}{1-p}\right) + nln(1-p) + ln\left(\binom{n}{y}\right)\right]$$

This can be rewritten into:

$$f(y) = exp\left[yln\left(\frac{p}{1-p}\right) + nln(1-p) - nln\left(\frac{1}{2}\right) + ln\left(\binom{n}{y}\left(\frac{1}{2}\right)^n\right)\right]$$

This gives us the following:

- $\theta = ln\left(\frac{p}{1-p}\right)$
- $b(\theta) = nln(1-p) - nln\left(\frac{1}{2}\right)$
- $f_0(y) = ln\left(\binom{n}{y}\left(\frac{1}{2}\right)^n\right)$ which is the Binomial(n, $\frac{1}{2}$) pmf

Part b.

The Poisson pmf can be factored into

$$f(y) = exp\left[yln(\lambda) - \lambda - ln(y!)\right]$$

This can be rewritten into:

$$f(y) = exp\left[yln(\lambda) - (\lambda - 1) + ln((y!)^{-1}e^{-1})\right]$$

This gives us the following:

- $\theta = ln(\lambda)$
- $b(\theta) = (\lambda - 1)$
- $f_0(y) = e^{-1}y^{-1}$ which is the Poisson(1) pmf

Part c.

The Exponential pdf can be factored into

$$f(y) = exp\left[-\lambda y + ln(\lambda)\right]$$

This can be rewritten into:

$$f(y) = exp\left[-\lambda y + ln(\lambda) - ln(e^{-y}) + ln(e^{-y})\right]$$

This gives us the following:

- $\theta = -\lambda$
- $b(\theta) = ln(\lambda) - ln(e^{-1})$
- $f_0(y) = e^{-y}$ which is the Exponential(1) pmf

Part d.

The Geometric pmf can be factored into

$$f(y) = exp\left[(y-1)ln(1-p) + ln(p)\right]$$

This can be rewritten into:

$$\begin{aligned} f(y) &= exp\left[(y-1)ln(1-p) + ln(p) - ln\left(\frac{1}{2}\right)^y + ln\left(\frac{1}{2}\right)^y\right] \\ &= exp\left[yln(1-p) - ln((1-p)p) - yln\left(\frac{1}{2}\right) + ln\left(\frac{1}{2}\right)^y\right] \end{aligned}$$

This gives us the following:

- $\theta = ln(1-p)$
- $b(\theta) = ln((1-p)p) - yln\left(\frac{1}{2}\right)$
- $f_0(y) = \left(\frac{1}{2}\right)^y$ which is the geometric(1/2) pmf

Part e.

If we fit the model

$$y_i \sim exp\left[\theta_i y_i - b(\theta_i) + ln(f_0(y_i))\right]$$

this allows us to specify the mean response $\mu = b'(\theta)$. If we have $\theta_i = x_i^T\beta$, then consider $\mu_i = b'(x_i^T\beta)$. Assuming the mean function $b'(\theta_i)$ is an invertible function, then we have $x_i^T\beta = b^{-1'}(\mu)$. If β increases then the function $b^{-1'}(\mu)$ increases. If β decreases then the function $b^{-1'}(\mu)$ decreases. Therefore we can interpret β as having a direct impact on the inverse of the mean function of y.

The mean function in the case of OLS is just the identity, so that allows us to interpret β as being composed of individual coefficients β_i where each β_i is the change in mean response if the associated variable x_i is changed by one unit and all other variables $x_j, i \neq j$ remain the same.

Question 2

Here we assume the following model:

$$y_{ij} = ln(r_{ij}) = ln(w_i) - ln(w_j) = \alpha_i - \alpha_j + \epsilon_{ij}$$

Part a.

Estimating α_i and $\alpha_i + c$ are equivalent in this model because the model definition effectively remains unchanged:

$$y_{ij} = (\alpha_i + c) - (\alpha_j + c) + \epsilon_{ij} = \alpha_i - \alpha_j + \epsilon_{ij}$$

This however tells us that we cannot estimate α_i directly since if we are able to offset the parameter estimates by a constant and have the model unchanged, then we can have no reliable estimate for the α_is themselves.

However, we should still be able to estimate the weights since the ratio of the weights is captured in this model. Consider that

$$\begin{aligned} y_{ij} = ln(w_i) - ln(w_j) &= \alpha_i - \alpha_j + \epsilon_{ij} \\ ln\left(\frac{w_i}{w_j}\right) &= \alpha_i - \alpha_j + \epsilon_{ij} \\ w_i &= exp(\alpha_i - \alpha_j) w_j \end{aligned}$$

If we use the following as our parameter vector, we can estimate the weights:

$$\beta = \begin{pmatrix} \alpha_1 - \alpha_8 \\ \alpha_2 - \alpha_8 \\ \alpha_3 - \alpha_8 \\ \alpha_4 - \alpha_8 \\ \alpha_5 - \alpha_8 \\ \alpha_6 - \alpha_8 \\ \alpha_7 - \alpha_8 \end{pmatrix}$$

This model is not overparameterized, and we can estimate all of the weights using the relationship we illustrated above:

$w_1 = exp(\alpha_1 - \alpha_8) w_8 w_2 = exp(\alpha_2 - \alpha_8) w_8 w_3 = exp(\alpha_3 - \alpha_8) w_8 w_4 = exp(\alpha_4 - \alpha_8) w_8 w_5 = exp(\alpha_5 - \alpha_8) w_8 w_6 = exp(\alpha_6 - \alpha_8) w_8 u$

where

$$\begin{aligned} w_8 &= 1 - \sum_{i=1}^{7} w_i \\ w_8 &= 1 - w_8 \sum_{i=1}^{7} (\alpha_i - \alpha_8) \\ w_8 + w_8 \sum_{i=1}^{7} (\alpha_i - \alpha_8) &= 1 \\ w_8 &= \frac{1}{1 + \sum_{i=1}^{7} (\alpha_i - \alpha_8)} \end{aligned}$$

Part b.

To fit the model, we first have to define the appropriate model matrix.

```
observed_pairwise_log_ratios <- c(5, 3, 7, 6, 6, 1/3, 1/4,
                                  1/3, 5, 3, 3, 1/5, 1/7,
                                  6, 3, 4, 6, 1/5,
                                  1/3, 1/4, 1/7, 1/8,
                                  1/2, 1/5, 1/6,
                                  1/5, 1/6,
                                  1/2)

model_matrix <- rbind(
  rbind(cbind(rep(1,6),-1*diag(6)),
        c(1,0,0,0,0,0,0)),
  rbind(cbind(rep(0,5), rep(1,5), -1*diag(5)),
        c(0,1,0,0,0,0,0)),
  rbind(cbind(rep(0,4), rep(0,4), rep(0,4), -1*diag(4)),
        c(0,0,1,0,0,0,0)),
  rbind(cbind(rep(0,3), rep(0,3), rep(0,3), rep(0,3), -1*diag(3)),
        c(0,0,0,1,0,0,0)),
  rbind(cbind(rep(0,2), rep(0,2), rep(0,2), rep(0,2), rep(0,2), -1*diag(2)),
        c(0,0,0,0,1,0,0)),
  rbind(cbind(rep(0,1), rep(0,1), rep(0,1), rep(0,1), rep(0,1), rep(0,1), -1*diag(1)),
        c(0,0,0,0,0,1,0)),
  c(0,0,0,0,0,0,1)
)
# model_matrix

model0 <- lm(observed_pairwise_log_ratios ~ model_matrix - 1)
summary(model0)
```

```
##
## Call:
## lm(formula = observed_pairwise_log_ratios ~ model_matrix - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0326 -0.3058  0.6867  1.5152  5.5085
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## model_matrix1  2.8744577  0.9624006   2.987  0.00703 **
## model_matrix2  0.2610649  0.9624006   0.271  0.78884
## model_matrix3  0.0007297  1.3687895   0.001  0.99958
## model_matrix4 -3.6848694  1.1631523  -3.168  0.00463 **
## model_matrix5 -1.8062288  1.0282273  -1.757  0.09355 .
## model_matrix6 -1.7413018  0.9311849  -1.870  0.07549 .
## model_matrix7 -0.4915240  0.8571465  -0.573  0.57244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.187 on 21 degrees of freedom
## Multiple R-squared:  0.6608, Adjusted R-squared:  0.5478
```

```
## F-statistic: 5.845 on 7 and 21 DF,  p-value: 0.0007337
```

Part c.

Let's plug our estimated values into the formulas we derived above!

```
estimates <- summary(model0)$coef[,1]
w8 <- 1/(1+sum(exp(estimates)))

weights = exp(estimates)*w8
weight_df <- data.frame(weights = c(weights, w8))
rownames(weight_df) <- (paste0("w", 1:8))
pander(weight_df,
       caption = "Estimated Weights")
```

Table 1: Estimated Weights

	weights
w1	0.8056
w2	0.05904
w3	0.04551
w4	0.001141
w5	0.00747
w6	0.007971
w7	0.02782
w8	0.04547

Part d.

In the case where all 8 criterion are equally important, then we have a scenario where each $w_i = \frac{1}{8}$. From here, we can assume that all of the α_i will also be equal. If this is the case, then $y_{ij} = \alpha_i - \alpha_j + \epsilon_{ij} = \epsilon_{ij}$

In other words, we want to see if at least one of our $(\alpha_i - \alpha_j) \neq 0$. This can be done using the omnibus F-statistic provided in the `summary()` function.

```
## value
##  TRUE
```

Our F-statistic is greater than the critical F-statistic in the omnibus F-test so we have significant evidence against the hypothesis that there are equal weights on all 8 variables.

Part e.

While the QQ plot seems to be roughly in line with what we expected, the residual plot should give us some concern. There appears to be a distinct pattern in the residuals, indicating that our model is probably not a great fit for the data. We might be better off using a different model, perhaps even a nonlinear model.

This is further confirmed by also looking at the studentized residuals which pinpoint 5 log(ratio) points that have unusually high residual values.

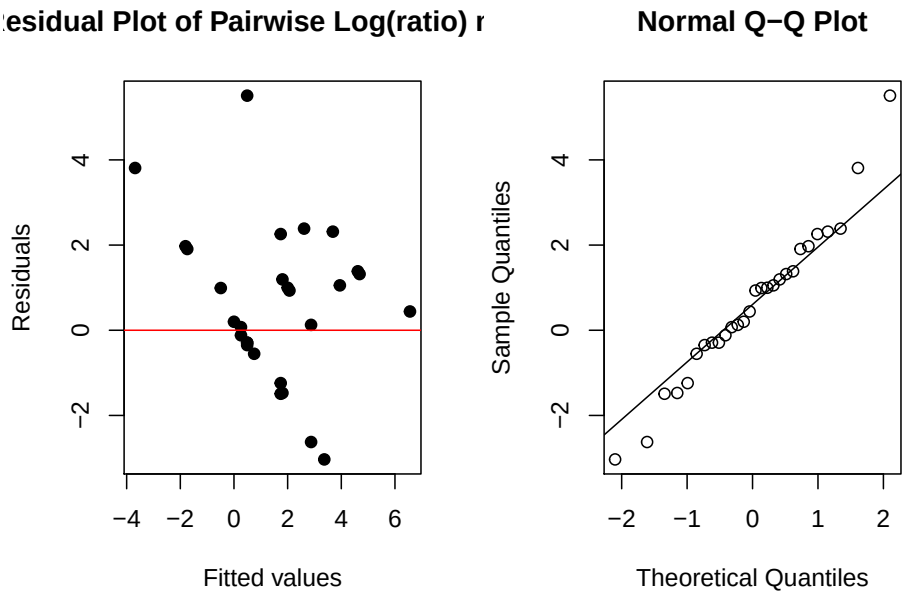


Figure 1: The residuals are not great.

```
res.studentized= model0$resid/sqrt( 1-influence(model0)$hat)
res.studentized[res.studentized > qt(.95, 28-1)]
```

```
##        1       14       16       17       22       25       27
## 2.755817 2.733674 2.496190 5.987326 4.498641 2.235260 2.108583
```

Question 3

The first part of this question is inputting the dataset. We should have a data.frame that looks something like:

toes_removed	returned	size	captured	not_captured
1	0.75	$<$ 15 mm	3	1
2	0.6842	$<$ 15 mm	26	12
3	0.6557	$<$ 15 mm	80	42
4	0.6044	$<$ 15 mm	110	72
5	0.544	$<$ 15 mm	68	57
6	0.5263	$<$ 15 mm	20	18

Part a.

Fitting a logistic regression to check the effect on size of the return rate is straightforward.

```
##
## Call:
## glm(formula = cbind(captured, not_captured) ~ as.factor(size) +
##     toes_removed, family = binomial(link = "logit"), data = dtf)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.72846  -0.38050  -0.01561   0.06506   1.54708
##
## Coefficients:
```

More specifically, a frog that is larger than 15mm has on average lower odds of returning than a frog that is smaller than 15mm.

Part b.

Return rate **does decrease** with the number of toes removed.

We can tell this via the estimated coefficients for the `toes removed` variable. Notice that the estimate is negative, so as the number of toes removed increases, the odds of returning is multiplied by a value that is less than 1, specifically $\approx e^{-.24} = .7866$. Therefore, the return odds decrease with the number of toes removed.

Part c.

For this question, we want to fit a model with a constant multiplicative effect m such that for every additional toe removed, return rate are multiplied by *(1+m)*.

We know for a log link function, we have the following setup:

$$\begin{aligned} ln(y) &= \beta_0 + \beta_1 X \\ y &= exp(\beta_0 + \beta_1 X) \end{aligned}$$

Therefore, if we increase X by 1, we have

$$\begin{aligned} y &= exp(\beta_0 + \beta_1 (X+1)) \\ y &= exp(\beta_0 + \beta_1 X) exp(\beta_1) \end{aligned}$$

If we set this multiplicative factor $e^{\beta_1} = (1+m)$, we get that the constant factor $m = e^{\beta_1} - 1$

From here, I think there are two things you could possibly do:

- Poisson regression
- Binomial regression with a log-link function

The Poisson regression model uses a log link function, so if we assume that the rate of frogs returning is Poisson distributed, we can model this relationship. We have to make sure to use an offset however, since the rate of frog return is influenced by how many frogs were initially released. The issue with this approach is that in our data there were occasionally no frogs released in a toe removal category and taking the log of 0 in the offset leads to $-\infty$.

For binomial regression with a log-link function, we only have to respecify the link function in the `glm()` function.

Part d.

Here we fit the Poisson regression model.

```r
dtf$total <- dtf$captured + dtf$not_captured

pois_model <- glm(captured ~ toes_removed + as.factor(size) + offset(log(total)),
                  data = dtf[dtf$total != 0,],
                  family = poisson(link = "log"),
                  control = list(maxit = 50))

summary(pois_model)
```

```
##
## Call:
## glm(formula = captured ~ toes_removed + as.factor(size) + offset(log(total)),
##     family = poisson(link = "log"), data = dtf[dtf$total != 0,
##         ], control = list(maxit = 50))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.60957  -0.37876  -0.15027   0.08789   1.39276
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -0.08062    0.18903  -0.427   0.6697
## toes_removed              -0.10903    0.04653  -2.343   0.0191 *
## as.factor(size)>= 15 mm -0.73024    0.14322  -5.099 3.42e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 45.9758  on 13  degrees of freedom
## Residual deviance:  3.5175  on 11  degrees of freedom
## AIC: 66.552
##
## Number of Fisher Scoring iterations: 4
```

The unfortunate part of using this method is that we have to filter our data to remove the cells where there were no frogs released. This is not good, especially if we are interested in comparing different modelling approaches using the same data. Still it might be interesting to see what the results are of this model.

For every toe removed, the expected multiplicative effect on the rate of return is `e round(exp(summary(pois_model)$coe[2]`
4). Similar to the logistic model, we see that for every additional toe removed the average rate of return decreases.

Now let's fit a binomial regression modle with a log link function.

```r
log_model <- glm(cbind(captured, not_captured) ~ as.factor(size) + toes_removed,
                 data = dtf,
                 family = binomial(link = "log"))
summary(log_model)
```

```
##
## Call:
## glm(formula = cbind(captured, not_captured) ~ as.factor(size) +
```

```
##     toes_removed, family = binomial(link = "log"), data = dtf)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.81320  -0.36622  -0.16519   0.01792   1.93735
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -0.13343    0.11505  -1.160  0.24615
## as.factor(size)>= 15 mm -0.72708    0.11646  -6.243 4.28e-10 ***
## toes_removed              -0.09621    0.02965  -3.245  0.00117 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 84.3276  on 13  degrees of freedom
## Residual deviance:  6.0471  on 11  degrees of freedom
## AIC: 60.554
##
## Number of Fisher Scoring iterations: 5
```

Here we calculate that the multiplicative effect of having an addition toe removed on the return rate is 0.9083.

Part e.

To compare the models, we will be interested in comparing their relative AIC values since that is the only metric that is available for this cross-model comparison.

Logit link	Log link (binomial)	Poisson
58.88	60.55	66.55

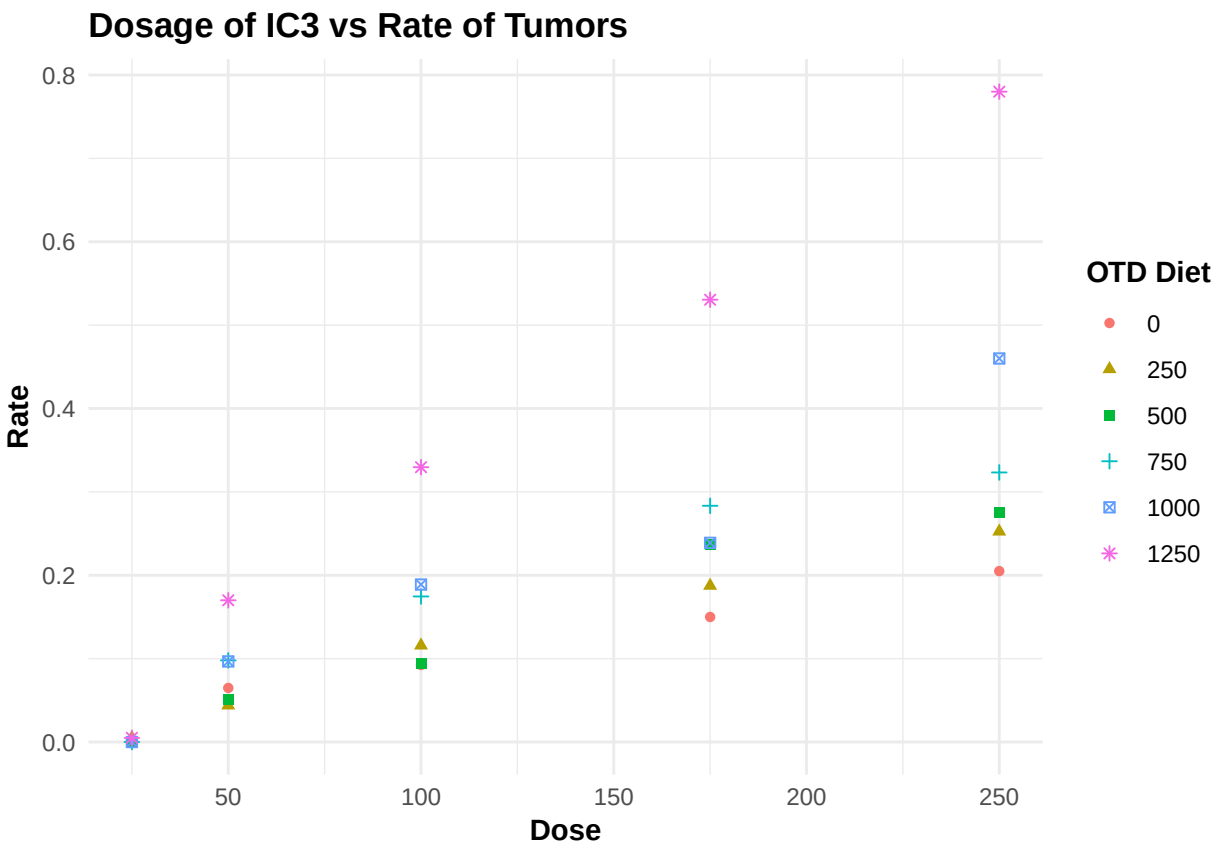
The original model with a logit link function has the lowest AIC, and by that criterion it is the "best" model.

Question 4

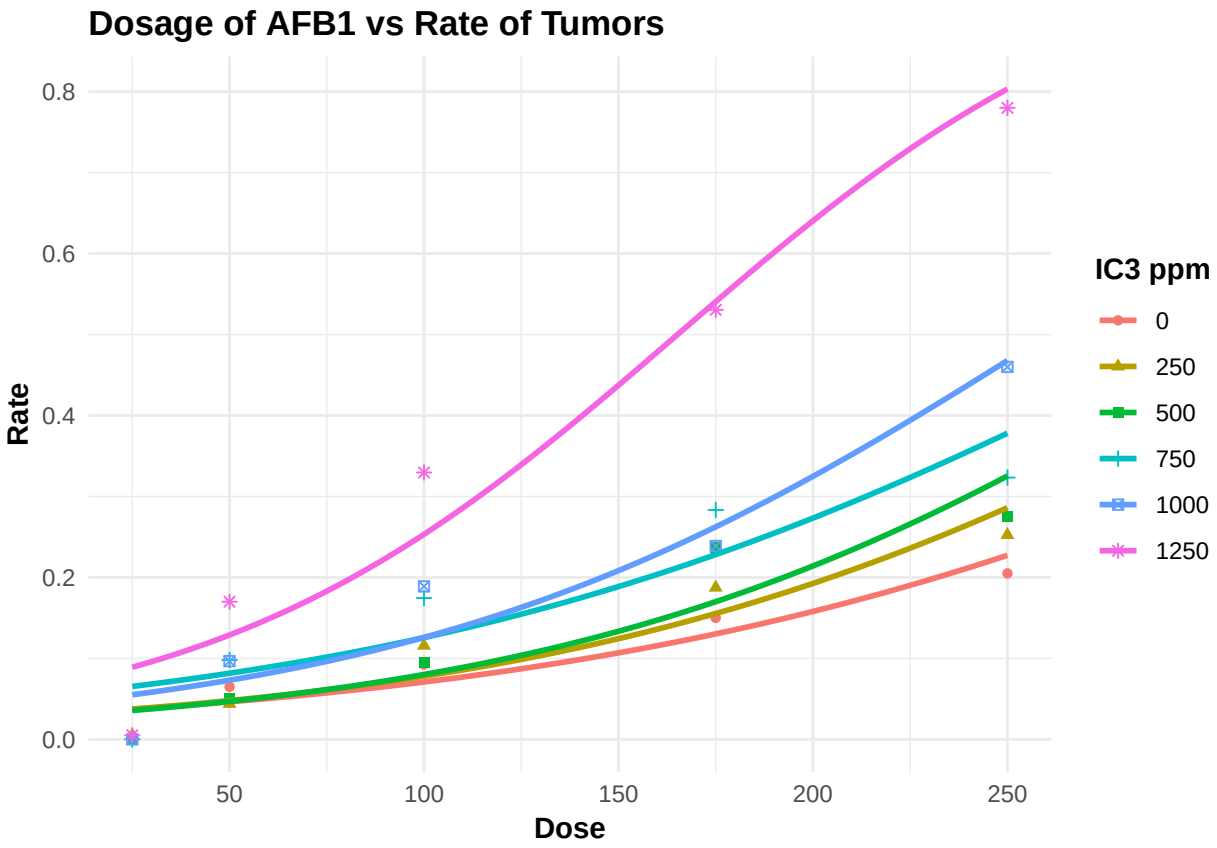
We are analyzing the effects of the metabolite I3C and tumors on the rainbow trout.

Part a.

Here we are going to visualize the tumor incidence rate as a function of AFB1 dosage.


Dosage of IC3 vs Rate of Tumors
Rate
0.0
0.2
0.4
0.6
0.8
50
100
150
200
250
Dose
OTD Diet
0
250
500
750
1000
1250

Part b.



If we consider this graph, it appears that across all OTD diets, the higher the dosage of AFB1, the higher the rate of liver tumors. The relationship within each OTD diet group appears pretty linear and the lines do not appear to be parallel - they each have a different slope.

Part c.

First, we fit a model where there is an interaction term for each of the six IC3 dosage groups. Note that the dosage used in the model is actually the log(dosage).

```
nonparallel_model <- glm(cbind(Presence, Absence) ~ log(Dose)*OTD,
                data = fish,
                family = binomial(link = "logit"))
pander(summary(nonparallel_model))
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-7.105	0.6811	-10.43	1.775e-25
log(Dose)	1.045	0.1338	7.805	5.948e-15
OTD250	-1.097	0.9753	-1.125	0.2606
OTD500	-1.582	0.9763	-1.621	0.1051
OTD750	-0.04393	0.9139	-0.04807	0.9617
OTD1000	-1.164	0.9185	-1.268	0.2049
OTD1250	-2.311	0.8718	-2.651	0.008033

	Estimate	Std. Error	z value	Pr(>|z|)
log(Dose):OTD250	0.2547	0.1908	1.335	0.182
log(Dose):OTD500	0.3757	0.1908	1.969	0.04895
log(Dose):OTD750	0.141	0.1802	0.7825	0.4339
log(Dose):OTD1000	0.4082	0.1814	2.251	0.0244
log(Dose):OTD1250	0.8611	0.1745	4.934	8.058e-07

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	1568.05 on 29 degrees of freedom
Residual deviance:	85.04 on 18 degrees of freedom

Then we fit a parallel lines model where there is only a single slope parameter for dosage. In the model formula, this is just not specifying an interaction term between OTD and log(dosage).

```
parallel_model <- glm(cbind(Presence, Absence) ~ log(Dose)+OTD,
                data = fish,
                family = binomial(link = "logit"))
pander(summary(parallel_model))
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-9.15	0.2795	-32.73	5.256e-235
log(Dose)	1.445	0.05194	27.82	2.674e-170
OTD250	0.2023	0.1099	1.84	0.06576
OTD500	0.3368	0.1078	3.125	0.001775
OTD750	0.6924	0.1107	6.254	4.003e-10
OTD1000	0.9196	0.1104	8.327	8.284e-17
OTD1250	1.94	0.1065	18.22	3.813e-74

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	1568.1 on 29 degrees of freedom
Residual deviance:	117.3 on 23 degrees of freedom

To asses which model is better, we can use a Chi-squared test.

Table 8: Analysis of Deviance Table

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
23	117.3	NA	NA	NA
18	85.04	5	32.28	5.231e-06

Our low p-value indicates that we can reject the null hypothesis that the simpler model (aka the parallel lines model) sufficiently describes the data. We should instead opt to use the nonparallel lines model.

Part d.

Let's fit the model with some alternative link functions.

The probit, complementary-loglog, and loglog link functions can be specified in the `glm()` function.

Table 9: AIC with different link functions

logit	probit	cloglog	loglog
245.8	233.2	249.5	227.5

The model with the lowest AIC should be the model with the best fit. Here we observe that the log-log link function creates the model with the best fit, so we should opt to use that model.

Part e.

Here we want to estimate the relative potencies and 95% confidence intervals for the ratio of AFB1 doses required to get the same tumor incidence between different levels of I3C.

That means for each pair of I3C treatments *i,j*, we want to calculate the ratio $\frac{x_i}{x_j}$ of AFB1 that will ensure that

$$\alpha_i + \beta_i \log(x_i) = \alpha_j + \beta_j \log(x_j)$$

We are assuming here that our model of choice has separate intercepts α_i and separate slopes β_i. Furthermore, we are also assuming a logit link function.

In office hours, we discussed using something like the following formula:

$$\begin{aligned}
\alpha_i + \beta_i \log(x_i) &= \alpha_j + \beta_j \log(x_j) \\
\alpha_i + \beta_i \log(x_i) &= \alpha_j + \beta_i \log(x_j) + (\beta_j - \beta_i) \log(x_j) \\
\alpha_i - \alpha_j - (\beta_j - \beta_i) \log(x_j) &= \beta_i \log\left(\frac{x_j}{x_i}\right) \\
\log\left(\frac{x_j}{x_i}\right) &= \frac{\alpha_i - \alpha_j - (\beta_j - \beta_i) \log(x_j)}{\beta_i}
\end{aligned}$$

where we can eventually figure out an estimate for $\frac{x_j}{x_i}$ by plugging in the respective estimated values for $\alpha_i, \alpha_j, \beta_i, \beta_j$. However, I don't really understand this formula and it's pretty late so I'm just going to take this fat L, Arpita.