Monday, March 3

Proportions as Response Variables

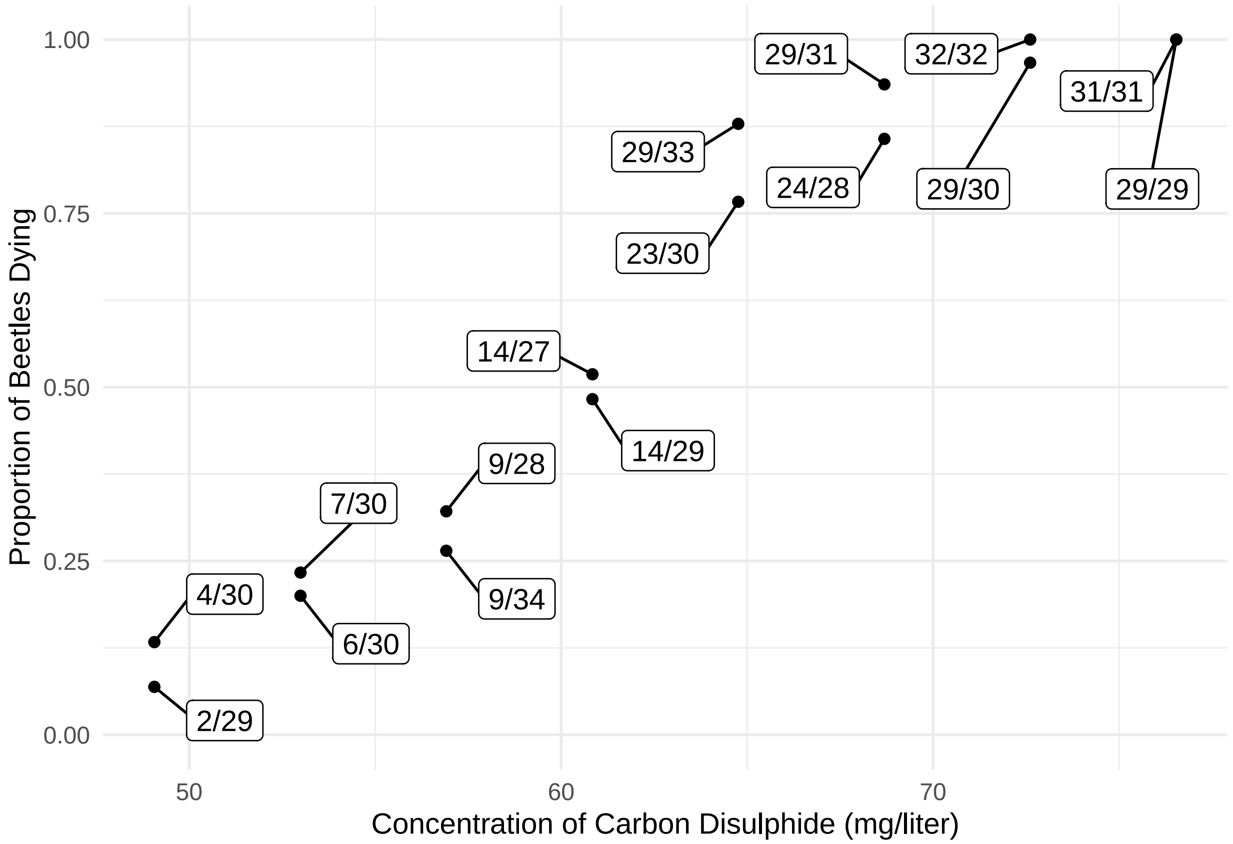
Consider the following data from an experiment that exposed batches of beetles to carbon disulphide.

```
library(trtools)
library(ggplot2)
library(ggrepel) # used for geom_label_repel (see below)

bliss$proportion <- paste(bliss$dead, "/", bliss$exposed, sep = "")
bliss
```

```
   concentration dead exposed proportion
1           49.1    2      29       2/29
2           49.1    4      30       4/30
3           53.0    7      30       7/30
4           53.0    6      30       6/30
5           56.9    9      28       9/28
6           56.9    9      34       9/34
7           60.8   14      27      14/27
8           60.8   14      29      14/29
9           64.8   23      30      23/30
10          64.8   29      33      29/33
11          68.7   29      31      29/31
12          68.7   24      28      24/28
13          72.6   29      30      29/30
14          72.6   32      32      32/32
15          76.5   29      29      29/29
16          76.5   31      31      31/31
```

```
p <- ggplot(bliss, aes(x = concentration, y = dead/exposed)) +
  geom_point() + ylim(0, 1) + theme_minimal() +
  geom_label_repel(aes(label = proportion), box.padding = 0.75) +
  labs(x = "Concentration of Carbon Disulphide (mg/liter)",
    y = "Proportion of Beetles Dying")
plot(p)
```



The interest here is in modeling the *proportion* of dead beetles as a response variable.

A proportion Y_i can be defined as $Y_i = C_i/m_i$ where C_i is a count and m_i is a total so that $C_i = 0, 1, \dots, m_i$ and $Y_i = 0, 1/m_i, 2/m_i, \dots, 1$. Note that proportions are not quite the same as rates. Proportions are bounded between zero and one, but rates are only bounded below by zero.

1. Proportions may require nonlinear models because $0 \le E(Y_i) \le 1$.
2. Proportions tend to exhibit heteroscedasticty with variance depending on $E(Y_i)$ and m_i. The variance of Y_i tends to be smaller as $E(Y_i)$ gets closer to zero or one, and is inversely proportional to m_i.
3. Non-normal discrete distribution.

The Binomial Distribution

Assume m independent “trials” with a probability of a “success” on each trial of p (and thus the probability of a “failure” is $1 - p$). The number of successes then has a *binomial distribution* such that

$$P(C = c) = \binom{m}{c} p^c (1-p)^{m-c}$$

where

$$\binom{m}{c} = \frac{m!}{c!(m-c)!}.$$

The possible values of C are $0, 1, \dots, m$. Note that $\binom{m}{c}$ is the number of outcomes where we can have a count of c out of m, and $p^c(1-p)^{m-c}$ is the probability of each of these outcomes.

Example: Suppose that the probability of observing a seed germinate under certain conditions is 0.2, and we observe four seeds. Let C be the number of seeds that germinate. Then $m = 4$ and $p = 0.2$. The probability that, say, $C = 3$ is then

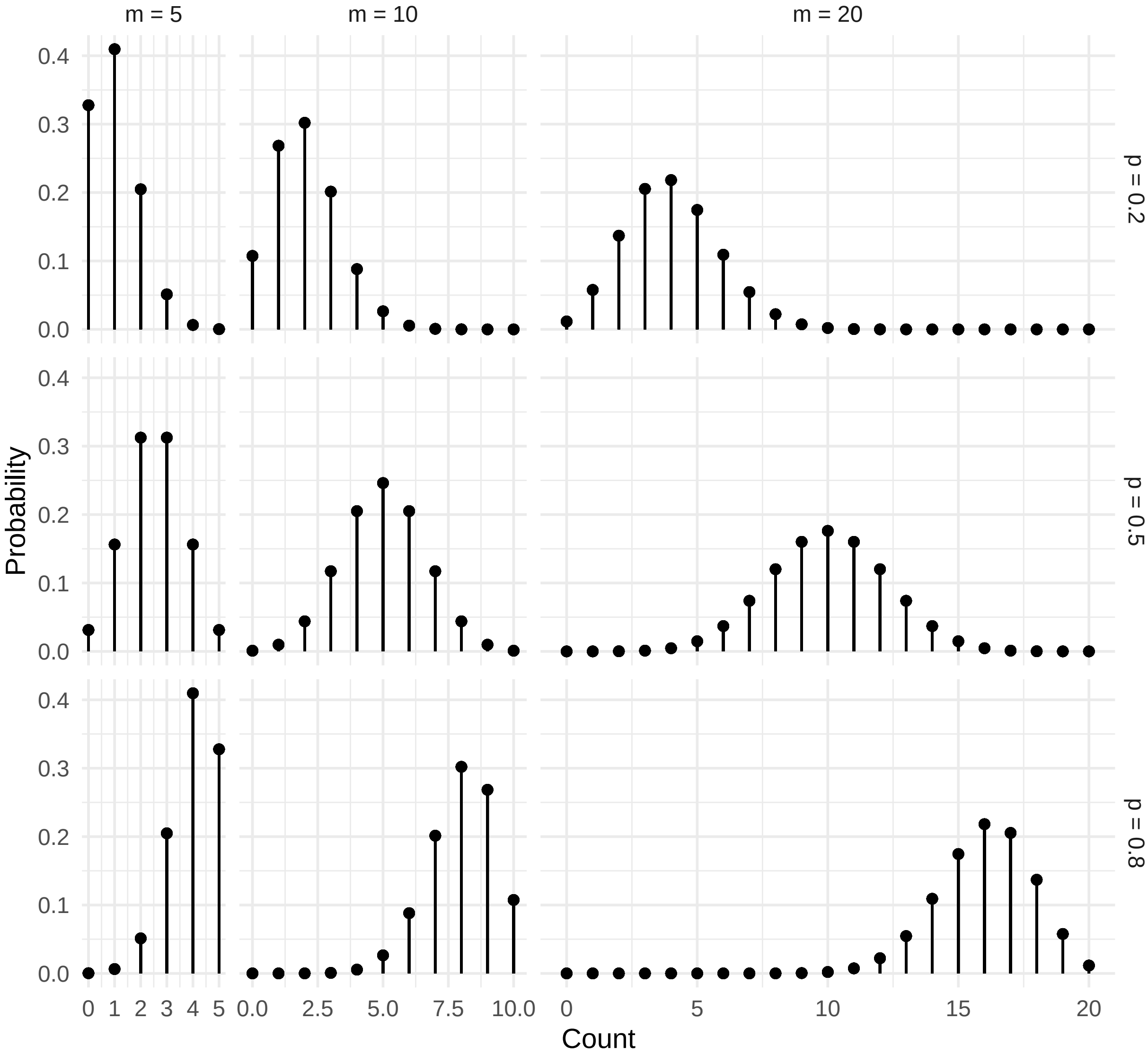
$$P(C=3) = \underbrace{\frac{4!}{3!(4-3)!}}_{4}\underbrace{0.2^3(1-0.2)^{4-3}}_{0.0064} = 0.0246.$$

There are four outcomes that give three successes, and each of these outcomes has a probability of 0.0064.

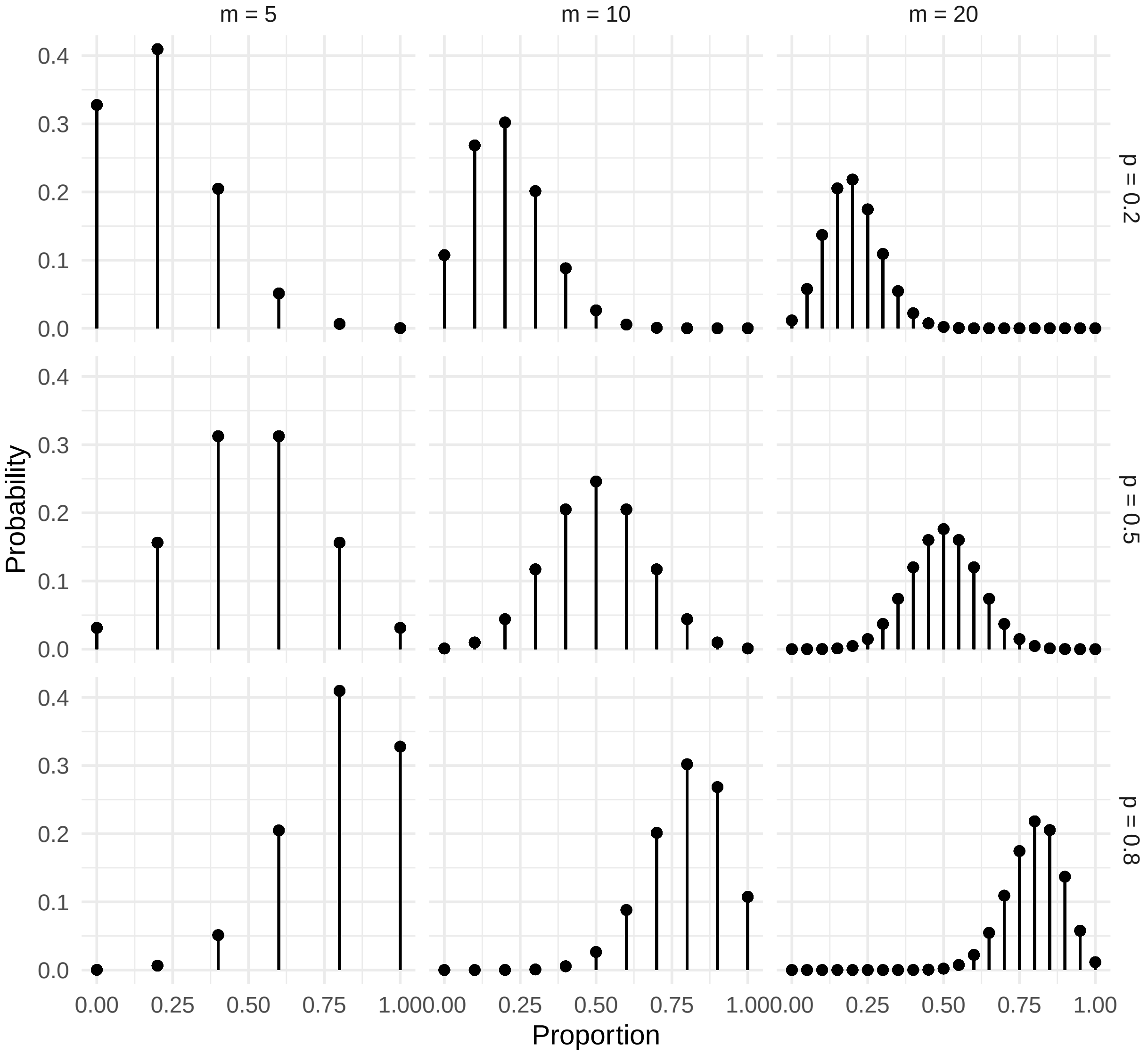
Outcome	Probability
SSSF	$0.2 \times 0.2 \times 0.2 \times 0.8$
SSFS	$0.2 \times 0.2 \times 0.2 \times 0.8$
SFSS	$0.2 \times 0.2 \times 0.2 \times 0.8$
FSSS	$0.2 \times 0.2 \times 0.2 \times 0.8$

The proportion is obtained as $Y = C/m$.

The figures below show several binomial distributions for different values of m and p.



The figures below show the distributions of the proportion C/m.



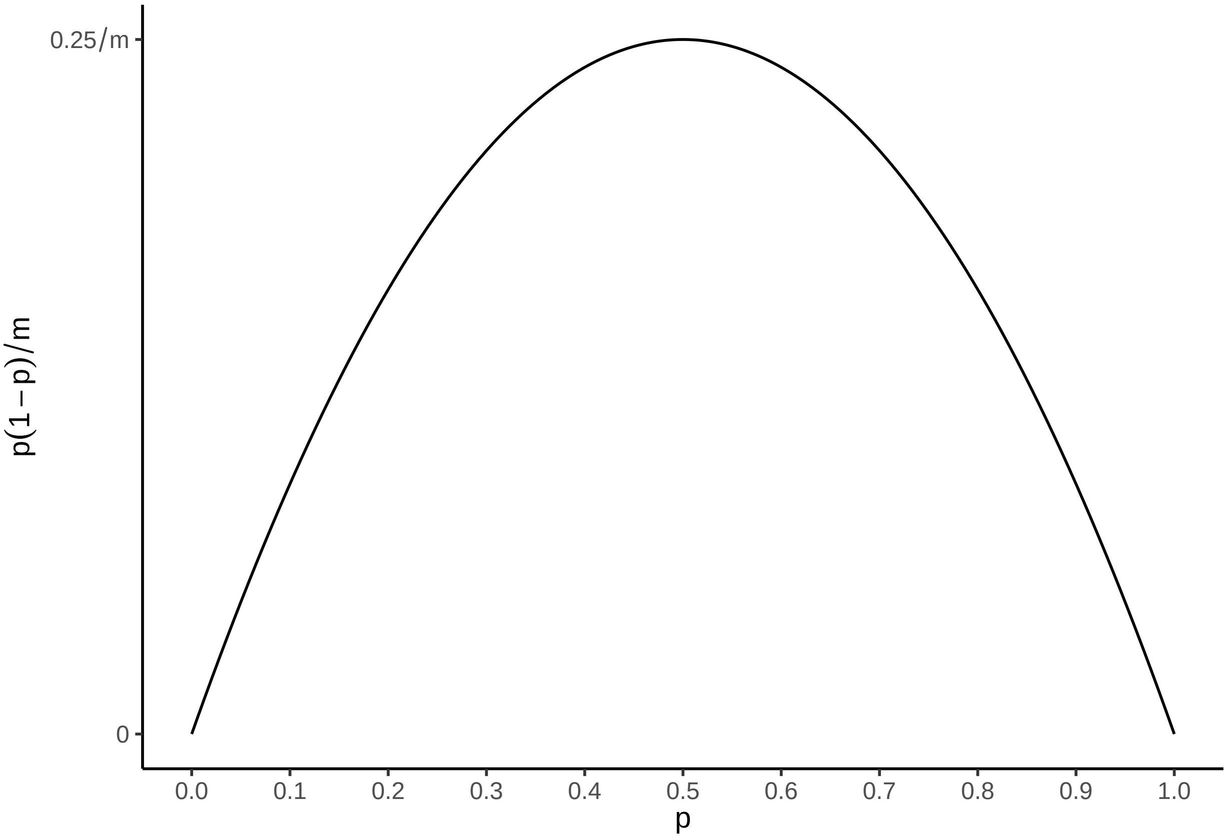
It can be shown that

$$E(C) = mp \quad \text{and} \quad \text{Var}(C) = mp(1-p).$$

Then for the *proportion* $Y = C/m$ it follows that

$$E(Y) = p \quad \text{and} \quad \text{Var}(Y) = p(1-p)/m.$$

This is because $E(Y) = E(C/m) = E(C)/m = mp/m = p$ and $\text{Var}(Y) = \text{Var}(C/m) = \text{Var}(C)/m^2 = mp(1-p)/m^2 = p(1-p)/m$. Note that the variance is at its maximum when $p = 0.5$ and gets smaller as p moves away from 0.5 toward $p = 0$ or $p = 1$.



An important special case of the binomial distribution is the *Bernoulli distribution* where $m = 1$ so that $C = 0, 1$ and $Y = 0, 1$.

Binomial Generalized Linear Models

Assume that each $C_1, C_2, \ldots, C_n$ has a binomial distribution with parameters $p_1, p_2, \ldots, p_n$ and $m_1, m_2, \ldots, m_n$, respectively, but $m_1, m_2, \ldots, m_n$ are observed/known). A binomial GLM will then specify the expected value of $Y_i = C_i/m_i$ as

$$g[E(Y_i)] = \eta_i \quad \text{or} \quad E(Y_i) = g^{-1}(\eta_i),$$

where $\eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}$.

Recall that $E(Y_i) = p_i$ so we are effectively specifying a model for the *probability of a success*. The variance of Y_i is then

$$\text{Var}(Y_i) = E(Y_i)[1 - E(Y_i)]/m_i = p_i(1 - p_i)/m_i,$$

so that $0 \le \text{Var}(Y_i) \le 0.25m_i$. Like rates, it is preferable to *not* model proportions as response variables without accounting for the denominator m_i since it affects the variance.

Logistic Regression

Logistic regression is a binomial generalized linear model that uses a "logit" link function such that

$$g[E(Y_i)] = \log\left[\frac{E(Y_i)}{1 - E(Y_i)}\right] = \log\left(\frac{p_i}{1 - p_i}\right),$$

and therefore

$$E(Y_i) = \frac{e^{\eta_i}}{1 + e^{\eta_i}} \quad \text{or} \quad p_i = \frac{e^{\eta_i}}{1 + e^{\eta_i}},$$

where again $\eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}$. Note that this guarantees that $0 < E(Y_i) < 1$.

Example: Consider again the `bliss` data. The `glm` function can be used to estimate the logistic regression model where

$$E(Y_i) = \frac{e^{\eta_i}}{1 + e^{\eta_i}},$$

where $\eta_i = \beta_0 + \beta_1 x_i$ and x_i is the concentration for the i-th observation (i.e., the i-th batch of beetles).

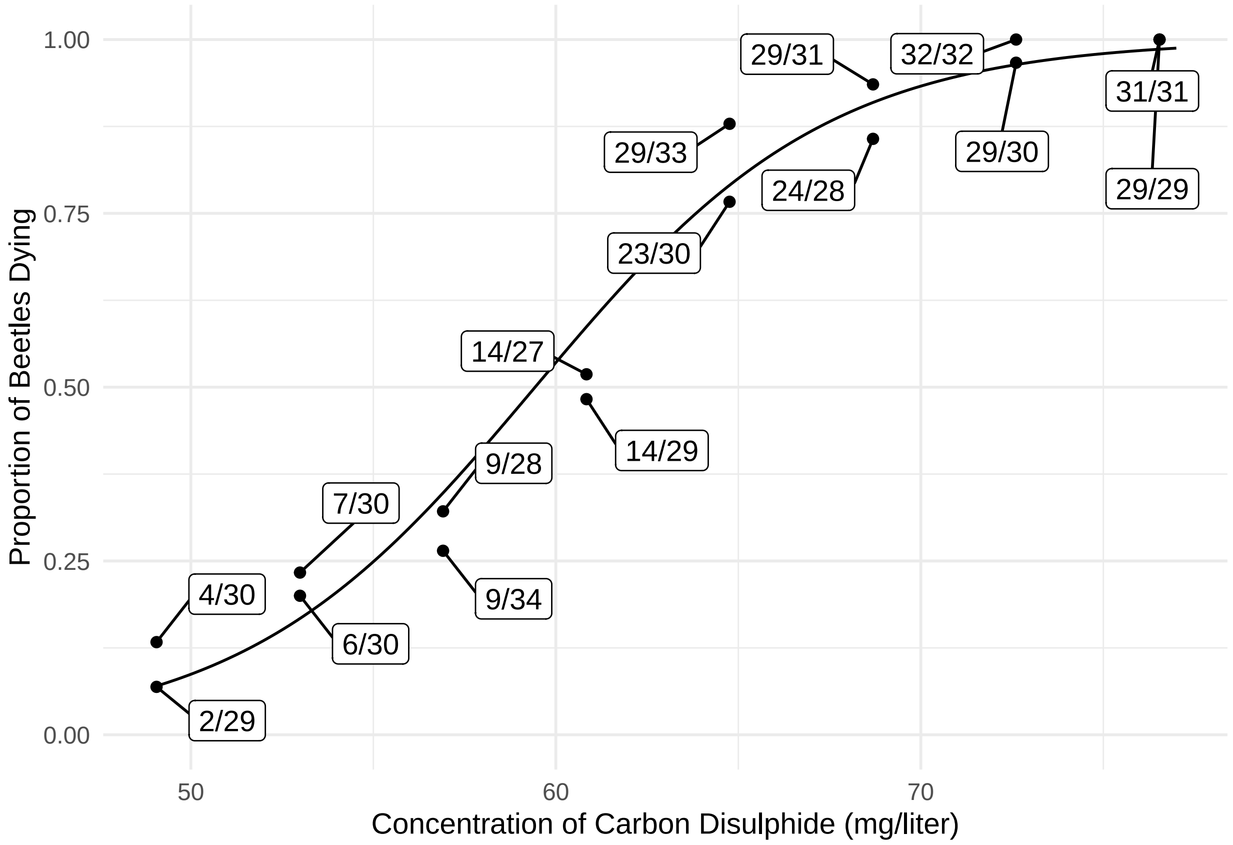
```
m <- glm(cbind(dead, exposed - dead) ~  concentration,
  family = binomial(link = logit), data = bliss)
cbind(summary(m)$coefficients, confint(m))
```

```
              Estimate Std. Error z value Pr(>|z|)   2.5 %  97.5 %
(Intercept)    -14.808     1.2898   -11.5 1.63e-30 -17.478 -12.409
concentration    0.249     0.0214    11.7 2.25e-31   0.209   0.294
```

Here the two variables in `cbind` are *the number of times the event occurred* (i.e., C_i) and *the number of times the event did not occur* (i.e., $m_i - C_i$). If the variables had been `dead` and `alive`, representing the number of dead and alive beetles, respectively, then we'd write `cbind(dead, alive)`. Also for `family = binomial` the logit link function is the default so you can use `family = binomial` for logistic regression.

```
d <- data.frame(concentration = seq(49, 77, length = 1000))
d$yhat <- predict(m, newdata = d, type = "response")

p <- ggplot(bliss, aes(x = concentration, y = dead/exposed)) +
  geom_point() + ylim(0, 1) + theme_minimal() +
  geom_line(aes(y = yhat), data = d) +
  geom_label_repel(aes(label = proportion), box.padding = 0.75) +
  labs(x = "Concentration of Carbon Disulphide (mg/liter)",
    y = "Proportion of Beetles Dying")
plot(p)
```



Predicted probabilities, with confidence intervals, can also be obtained using `contrast` or `glmint`. Note that the function $e^x/(1+e^x)$ is known to R as `plogis`.

```
trtools::contrast(m, list(concentration = c(50,60,70)),
  cnames = c("50 mg/liter","60 mg/liter","70 mg/liter"), tf = plogis)
```

```
            estimate  lower upper
50 mg/liter   0.0871 0.0551 0.135
60 mg/liter   0.5354 0.4712 0.598
70 mg/liter   0.9330 0.8949 0.958
```

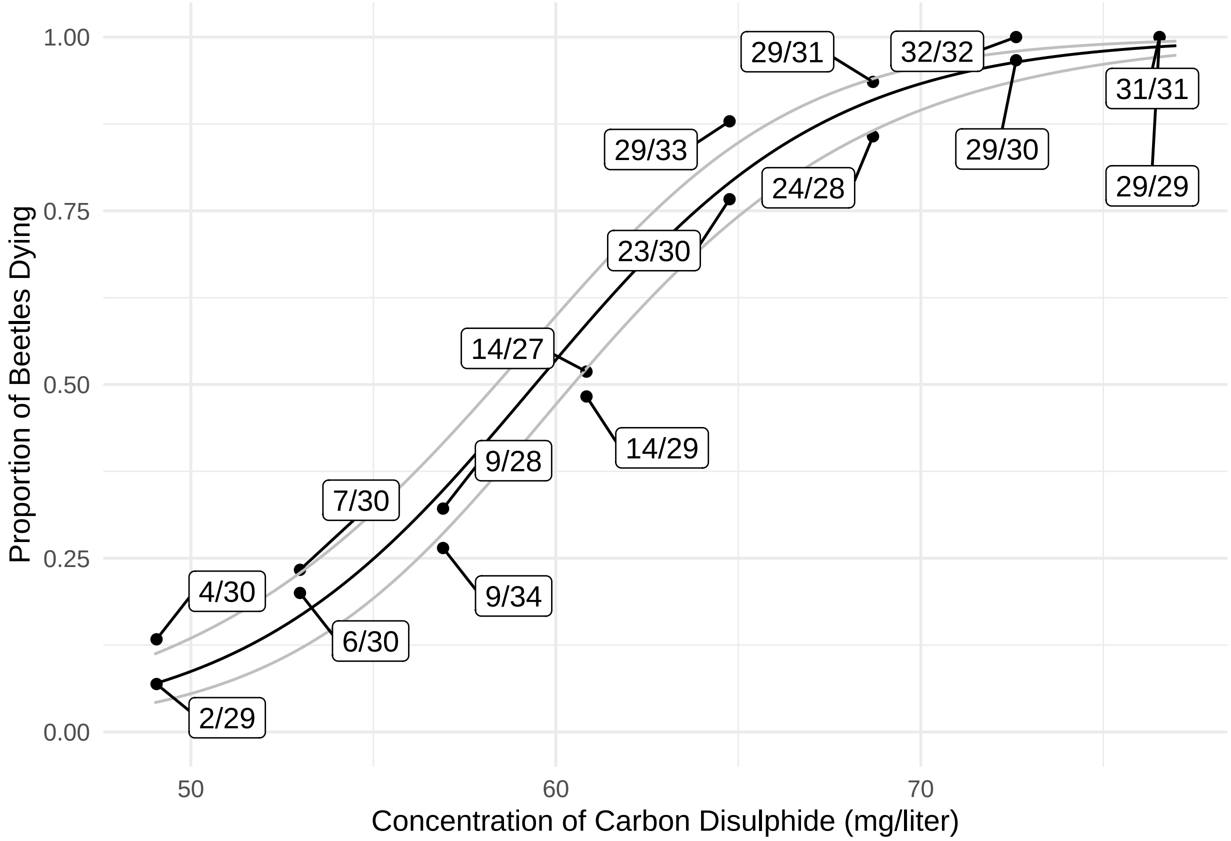
```
trtools::glmint(m, newdata = data.frame(concentration = c(50,60,70)))
```

```
     fit    low   upp
1 0.0871 0.0551 0.135
2 0.5354 0.4712 0.598
3 0.9330 0.8949 0.958
```

```
d <- data.frame(concentration = seq(49, 77, length = 1000))
d <- cbind(d, trtools::glmint(m, newdata = d))
head(d)
```

```
  concentration    fit    low   upp
1          49.0 0.0692 0.0420 0.112
2          49.0 0.0696 0.0423 0.113
3          49.1 0.0701 0.0427 0.113
4          49.1 0.0706 0.0430 0.114
5          49.1 0.0710 0.0433 0.114
6          49.1 0.0715 0.0437 0.115
```

```r
p <- ggplot(bliss, aes(x = concentration, y = dead/exposed)) +
  geom_point() + ylim(0, 1) + theme_minimal() +
  geom_line(aes(y = fit), data = d) +
  geom_line(aes(y = low), data = d, color = grey(0.75)) +
  geom_line(aes(y = upp), data = d, color = grey(0.75)) +
  geom_label_repel(aes(label = proportion), box.padding = 0.75) +
  labs(x = "Concentration of Carbon Disulphide (mg/liter)",
    y = "Proportion of Beetles Dying")
plot(p)
```



Parameter and Contrast Interpretation: Odds Ratios

A logistic regression model can be written as

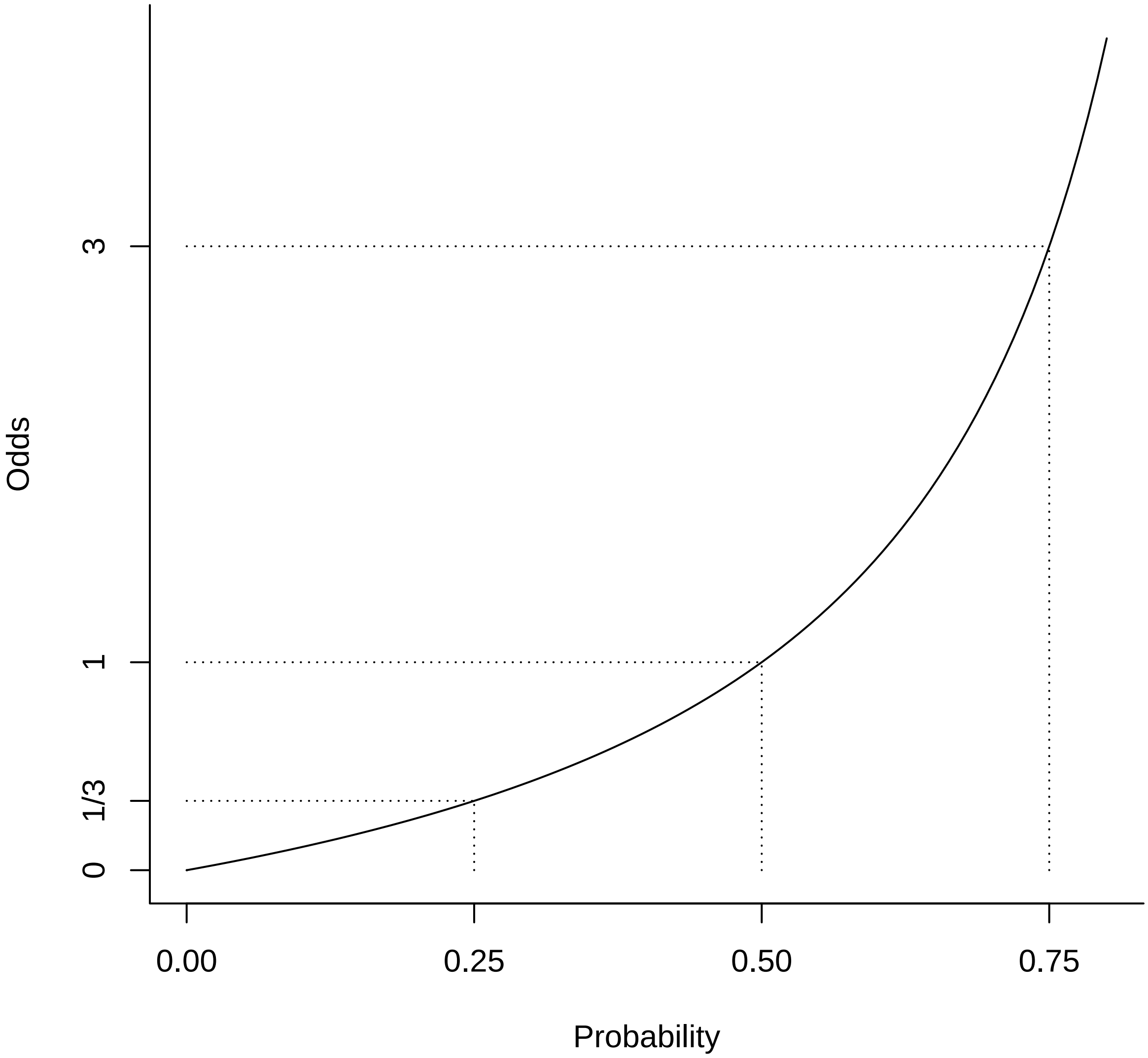
$$\frac{p_i}{1 - p_i} = \exp(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik})$$

where $p_i/(1-p_i)$ is the *odds* of the event. The *odds* is simply the ratio of the probability of the event occurring (p_i) to the probability of the event not occurring $(1 - p_i)$.

Odds are sometimes stated in “fractional form” as two numbers separated by a colon or other character (e.g., an odds of 1.5 might be written as “3:2” or “three to two”). Note that in its fractional form the odds $a : b$ implies a probability of $a/(a + b)$.

It is important to note that probabilities and odds are related but not equal.

	Odds	
Probability	Numeric	Fractional
0.01	0.01	1:99
0.1	0.11	1:9
0.25	0.33	1:3
1/3	0.50	1:2
0.4	0.67	2:3
0.5	1.00	1:1
0.6	1.50	3:2
2/3	2.00	2:1
0.75	3.00	3:1
0.9	9.00	9:1
0.99	99.00	99:1



Let O_i be the odds for the i-th observation. Then $O_i = p_i/(1 - p_i)$ and $p_i = O_i/(1 + O_i)$. Note that $0 \leq p_i \leq 1$ but $0 \leq O_i \leq \infty$.

We can write a logistic regression model in terms of the *odds* of an event as

$$O_i = \exp(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}),$$

or

$$O_i = e^{\beta_0} e^{\beta_1 x_{i1}} e^{\beta_2 x_{i2}} \cdots e^{\beta_k x_{ik}}.$$

Here we can use `contrast` to make inferences about the *odds* of death.

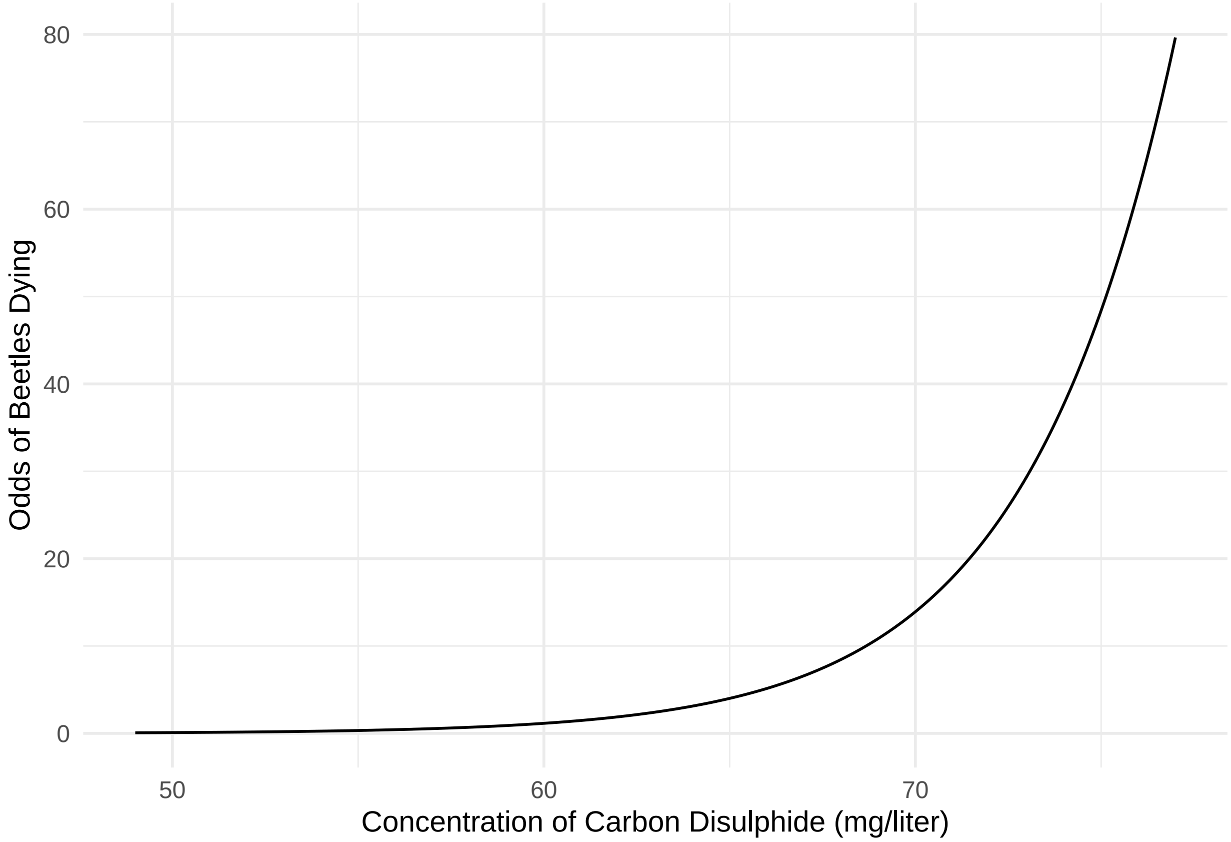
```
trtools::contrast(m, list(concentration = c(50,60,70)),
  cnames = c("50 mg/liter","60 mg/liter","70 mg/liter"), tf = exp)
```

```
            estimate  lower  upper
50 mg/liter   0.0954 0.0583  0.156
60 mg/liter   1.1523 0.8911  1.490
70 mg/liter  13.9222 8.5143 22.765
```

We can even plot the estimated odds of death.

```
d <- data.frame(concentration = seq(49, 77, length = 1000))
d$yhat <- predict(m, newdata = d, type = "response")
d$odds <- d$yhat / (1 - d$yhat)

p <- ggplot(d, aes(x = concentration, y = odds)) +
  geom_line() + theme_minimal() +
  labs(x = "Concentration of Carbon Disulphide (mg/liter)",
    y = "Odds of Beetles Dying")
plot(p)
```



The model for the odds is "log-linear" like the model for expected counts in Poisson regression. To interpret the parameters of a logistic regression model, we can use *odds ratios* which are similar to rate ratios in Poisson regression.

Odds Ratio: Quantitative Explanatory Variable

Suppose we have the logistic regression model

$$O_i = \exp(\beta_0 + \beta_1 x) = e^{\beta_0} e^{\beta_1 x},$$

were x_i is a quantitative explanatory variable. Consider the odds at x and $x + 1$ for arbitrary x,

$$O_a = e^{\beta_0} e^{\beta_1 (x+1)} \quad \text{and} \quad O_b = e^{\beta_0} e^{\beta_1 x}.$$

Then the *odds ratio* is

$$\frac{O_a}{O_b} = \frac{e^{\beta_0} e^{\beta_1 (x+1)}}{e^{\beta_0} e^{\beta_1 x}} = \frac{e^{\beta_0} e^{\beta_1 x} e^{\beta_1}}{e^{\beta_0} e^{\beta_1 x}} = e^{\beta_1} \Leftrightarrow O_a = O_b e^{\beta_1},$$

so that an increase x by one unit changes the odds by a factor of e^{β_1}. Also, we can compute the percent change in the odds as

$$100\% \times [O_a/O_b - 1],$$

where $O_a/O_b = e^{\beta_1}$ is the odds ratio. Again, the sign tells us if this is a percent increase or decrease in the odds.

Example: Consider again the model for the `bliss` data.

```r
cbind(summary(m)$coefficients, confint(m))
```

```
              Estimate Std. Error z value Pr(>|z|)   2.5 %  97.5 %
(Intercept)    -14.808     1.2898   -11.5 1.63e-30 -17.478 -12.409
concentration    0.249     0.0214    11.7 2.25e-31   0.209   0.294
```

```r
exp(cbind(coef(m), confint(m)))
```

```
                          2.5 %   97.5 %
(Intercept)   3.70e-07 2.57e-08 4.08e-06
concentration 1.28e+00 1.23e+00 1.34e+00
```

```r
trtools::contrast(m, tf = exp,
  a = list(concentration = 2),
  b = list(concentration = 1))
```

```
 estimate lower upper
     1.28  1.23  1.34
```

An odds ratio is then simply the ratio of the odds at two different values of an explanatory variable. We could compute the odds ratio, for example, for an increase of 1, 5, 10, and 20 mg/liter.

```r
trtools::contrast(m, tf = exp,
  a = list(concentration = c(1,5,10,20)),
  b = list(concentration = 0),
  cnames = c("+1 mg/liter", "+5 mg/liter", "+10 mg/liter", "+20 mg/liter"))
```

```
             estimate lower  upper
+1 mg/liter      1.28  1.23   1.34
+5 mg/liter      3.48  2.82   4.29
+10 mg/liter    12.08  7.95  18.37
+20 mg/liter   145.97 63.13 337.54
```

Suppose that we model instead the probability of *survival* rather than death.

```
m <- glm(cbind(exposed - dead, dead) ~ concentration,
  family = binomial, data = bliss)
cbind(summary(m)$coefficients, confint(m))
```

```
              Estimate Std. Error z value Pr(>|z|)  2.5 % 97.5 %
(Intercept)     14.808     1.2898    11.5 1.63e-30 12.409 17.478
concentration   -0.249     0.0214   -11.7 2.25e-31 -0.294 -0.209
```

```
exp(cbind(coef(m), confint(m)))
```

```
                          2.5 %   97.5 %
(Intercept)   2.70e+06 2.45e+05 3.90e+07
concentration 7.79e-01 7.46e-01 8.11e-01
```

```
trtools::contrast(m, tf = exp,
  a = list(concentration = 2),
  b = list(concentration = 1))
```

```
 estimate lower upper
    0.779 0.747 0.813
```

Note the “symmetry” of logistic regression. Whether we model the probability of the event or its complement is just a matter of parameterization.

Odds Ratio: Categorical Explanatory Variable

Suppose we have the model

$$O_i = \exp(\beta_0 + \beta_1 x) = e^{\beta_0} e^{\beta_1 x},$$

were x is an indicator variable so that

$$x = \begin{cases} 1, & \text{if the observation is from group } a, \\ 0, & \text{if the observation is from group } b, \end{cases}$$

so that the model can be written as

$$O_i = \begin{cases} e^{\beta_0} e^{\beta_1}, & \text{if the observation is from group } a, \\ e^{\beta_0}, & \text{if the observation is from group } b. \end{cases}$$

So we can write the odds as

$$O_a = e^{\beta_0} e^{\beta_1} \quad \text{and} \quad O_b = e^{\beta_0}.$$

The *odds ratio* is then

$$\frac{O_a}{O_b} = \frac{e^{\beta_0} e^{\beta_1}}{e^{\beta_0}} = e^{\beta_1} \quad \text{or} \quad \frac{O_b}{O_a} = \frac{e^{\beta_0}}{e^{\beta_0} e^{\beta_1}} = \frac{1}{e^{\beta_1}} = e^{-\beta_1}.$$

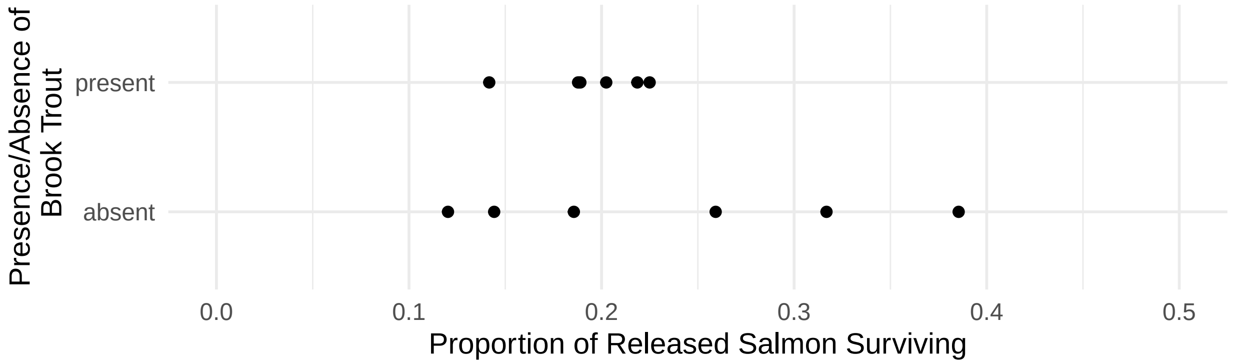
So the odds for group a is e^{β_1} times that for group b, and the odds for group b is $e^{-\beta_1} = 1/e^{\beta_1}$ times that for group a. We can compute how much larger (or smaller) O_a is relative to O_b with

$$100\% \times [O_a/O_b - 1],$$

where $O_a/O_b = e^{\beta_1}$ is the odds ratio. The sign tells us if O_a is a percent larger or smaller than O_b.

Example: Consider the following data from a study that investigated the effect of non-indigenous brook trout on the survival of salmon.

```
library(abd) # for BrookTrout data
p <- ggplot(BrookTrout, aes(x = trout, y = salmon.survived/salmon.released)) +
  geom_point() + ylim(0, 0.5) + coord_flip() + theme_minimal() +
  labs(x = "Presence/Absence of\n Brook Trout",
    y = "Proportion of Released Salmon Surviving")
plot(p)
```



```
m <- glm(cbind(salmon.survived, salmon.released - salmon.survived) ~ trout,
  data = BrookTrout, family = binomial)
cbind(summary(m)$coefficients, confint(m))
```

```
             Estimate Std. Error z value  Pr(>|z|)  2.5 %  97.5 %
(Intercept)     -1.30     0.0367  -35.43 5.00e-275 -1.372 -1.2283
troutpresent    -0.14     0.0519   -2.69  7.12e-03 -0.241 -0.0379
```

```
exp(cbind(coef(m), confint(m)))
```

```
                   2.5 % 97.5 %
(Intercept)  0.273 0.254  0.293
troutpresent 0.870 0.786  0.963
```

```
trtools::contrast(m, a = list(trout = "present"), b = list(trout = "absent"), tf = exp)
```

```
 estimate lower upper
     0.87 0.786 0.963
```

```
trtools::contrast(m, a = list(trout = "absent"), b = list(trout = "present"), tf = exp)
```

```
 estimate lower upper
     1.15  1.04  1.27
```

Recall that estimated probabilities can be computed using `contrast` with `tf = plogis`.

```
trtools::contrast(m, a = list(trout = c("present","absent")),
    tf = plogis, cnames = c("prob @ present","prob @ absent"))
```

```
               estimate lower upper
prob @ present    0.192 0.181 0.203
prob @ absent     0.214 0.202 0.227
```

Similarly the estimated odds can be computed if `tf = exp`.

```
trtools::contrast(m, a = list(trout = c("present","absent")),
    tf = exp, cnames = c("odds @ present","odds @ absent"))
```

```
               estimate lower upper
odds @ present    0.237 0.221 0.255
odds @ absent     0.273 0.254 0.293
```

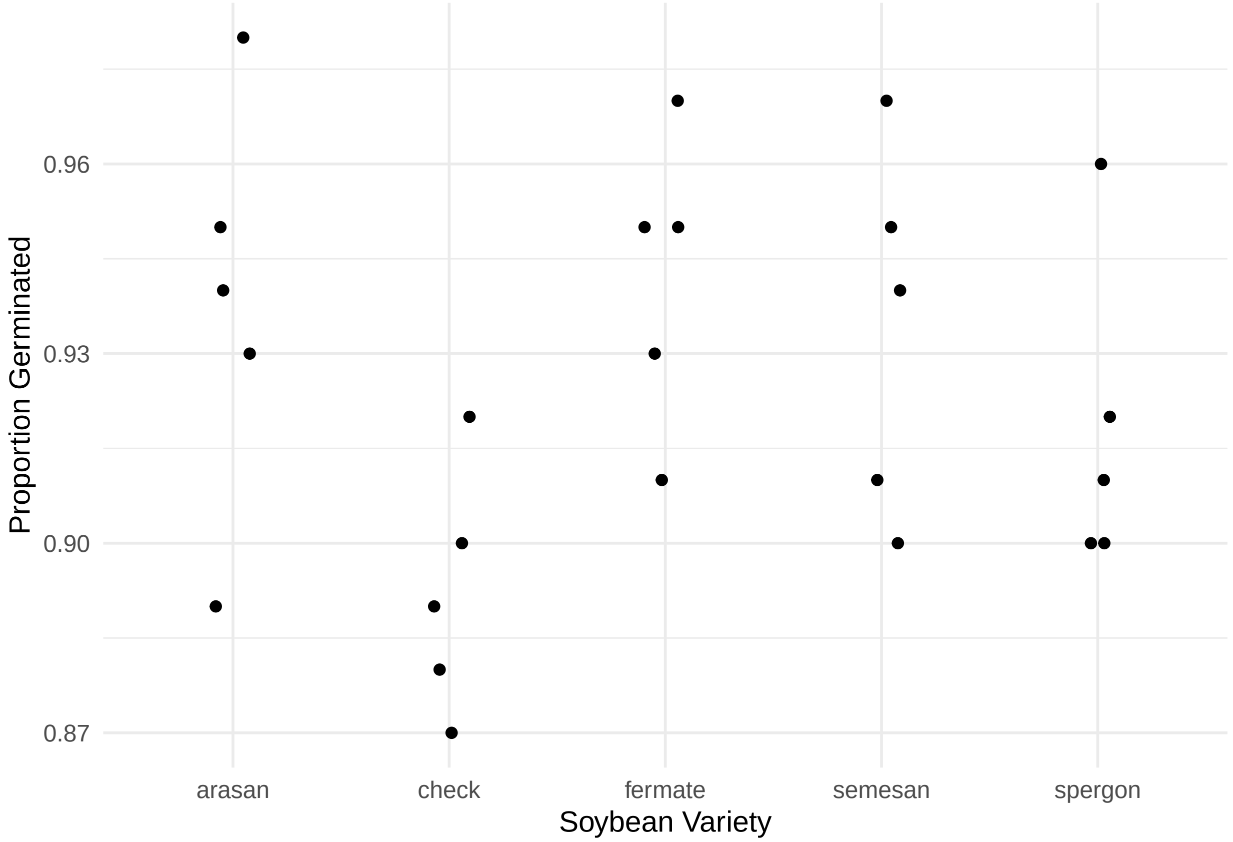
The odds ratios are then simply a ratio of these odds.

Example: Consider the following study of the germination of five varieties of soybean seeds. Note that each observation was the number of seeds that *failed* to germinate out of 100 seeds.

```r
head(faraway::soybean, 10)
```

```
   variety replicate failure
1    check         1       8
2    check         2      10
3    check         3      12
4    check         4      13
5    check         5      11
6   arasan         1       2
7   arasan         2       6
8   arasan         3       7
9   arasan         4      11
10  arasan         5       5
```

```r
p <- ggplot(faraway::soybean, aes(x = variety, y = (100-failure)/100)) +
  geom_jitter(height = 0, width = 0.1) + theme_minimal() +
  labs(x = "Soybean Variety", y = "Proportion Germinated")
plot(p)
```



```r
m <- glm(cbind(100 - failure, failure) ~ variety, family = binomial, data = faraway::soybean)
exp(cbind(coef(m), confint(m)))
```

```
                      2.5 % 97.5 %
(Intercept)    15.129 10.711 22.213
varietycheck    0.546  0.341  0.859
varietyfermate  1.074  0.636  1.817
varietysemesan  0.935  0.562  1.554
varietyspergon  0.740  0.453  1.197
```

```
# compute odds ratio of germination for arasan, fermate, semesan, and spergon versus check
trtools::contrast(m, tf = exp,
  a = list(variety = c("arasan","fermate","semesan","spergon")),
  b = list(variety = "check"),
  cnames = c("arasan/check","fermate/check","semesan/check","spergon/check"))
```

```
              estimate lower upper
arasan/check      1.83 1.156  2.90
fermate/check     1.97 1.230  3.14
semesan/check     1.71 1.090  2.69
spergon/check     1.36 0.885  2.08
```

Aggregated Versus Binary Responses

Suppose the observations in the `bliss` data were for individual beetles.

```
blissbin <- bliss |> mutate(alive = exposed - dead) |>
  dplyr::select(concentration, dead, alive) |>
  pivot_longer(cols = c(dead,alive), names_to = "state", values_to = "count") |>
  uncount(count)
head(blissbin)
```

```
# A tibble: 6 x 2
  concentration state
          <dbl> <chr>
1          49.1 dead
2          49.1 dead
3          49.1 alive
4          49.1 alive
5          49.1 alive
6          49.1 alive
```

We can specify the response variable as follows.

```
m <- glm(state == "dead" ~ concentration, family = binomial, data = blissbin)
summary(m)$coefficients
```

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -14.808     1.2897   -11.5 1.63e-30
concentration    0.249     0.0214    11.7 2.24e-31
```

But do not use this the method above if using **emmeans**. Or if the response variable is binary we can specify the model as follows.

```
blissbin <- blissbin |> mutate(y = ifelse(state == "dead", 1, 0))
m <- glm(y ~ concentration, family = binomial, data = blissbin)
summary(m)$coefficients
```

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -14.808     1.2897   -11.5 1.63e-30
concentration    0.249     0.0214    11.7 2.24e-31
```

```
m <- glm(cbind(y, 1-y) ~ concentration, family = binomial, data = blissbin)
summary(m)$coefficients
```

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -14.808     1.2897   -11.5 1.63e-30
concentration    0.249     0.0214    11.7 2.24e-31
```

Note that our parameter estimates and other inferences are the same as what we obtained with the aggregated data.

```
head(bliss)
```

```
  concentration dead exposed proportion
1          49.1    2      29       2/29
2          49.1    4      30       4/30
3          53.0    7      30       7/30
4          53.0    6      30       6/30
5          56.9    9      28       9/28
6          56.9    9      34       9/34
```

```
m <- glm(cbind(dead, exposed - dead) ~ concentration,
  family = binomial, data = bliss)
```

It is usually not necessary to transform aggregate data into binary data, but it is sometimes useful to transform binary data into aggregate data. Here is how that can be done. Note that any explanatory variables (separated by commas) are listed in `group_by` and the response variable is listed in `count`.

```
blissagg <- blissbin |> group_by(concentration) |> count(state) |>
  pivot_wider(names_from = state, values_from = n, values_fill = 0)
blissagg
```

```
# A tibble: 8 x 3
# Groups:   concentration [8]
  concentration alive  dead
          <dbl> <int> <int>
1          49.1    53     6
2          53.0    47    13
3          56.9    44    18
4          60.8    28    28
5          64.8    11    52
6          68.7     6    53
7          72.6     1    61
8          76.5     0    60
```

```
m <- glm(cbind(dead, alive) ~ concentration, family = binomial, data = blissagg)
summary(m)$coefficients
```

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -14.808     1.2898   -11.5 1.63e-30
concentration    0.249     0.0214    11.7 2.25e-31
```