Friday, Mar 25

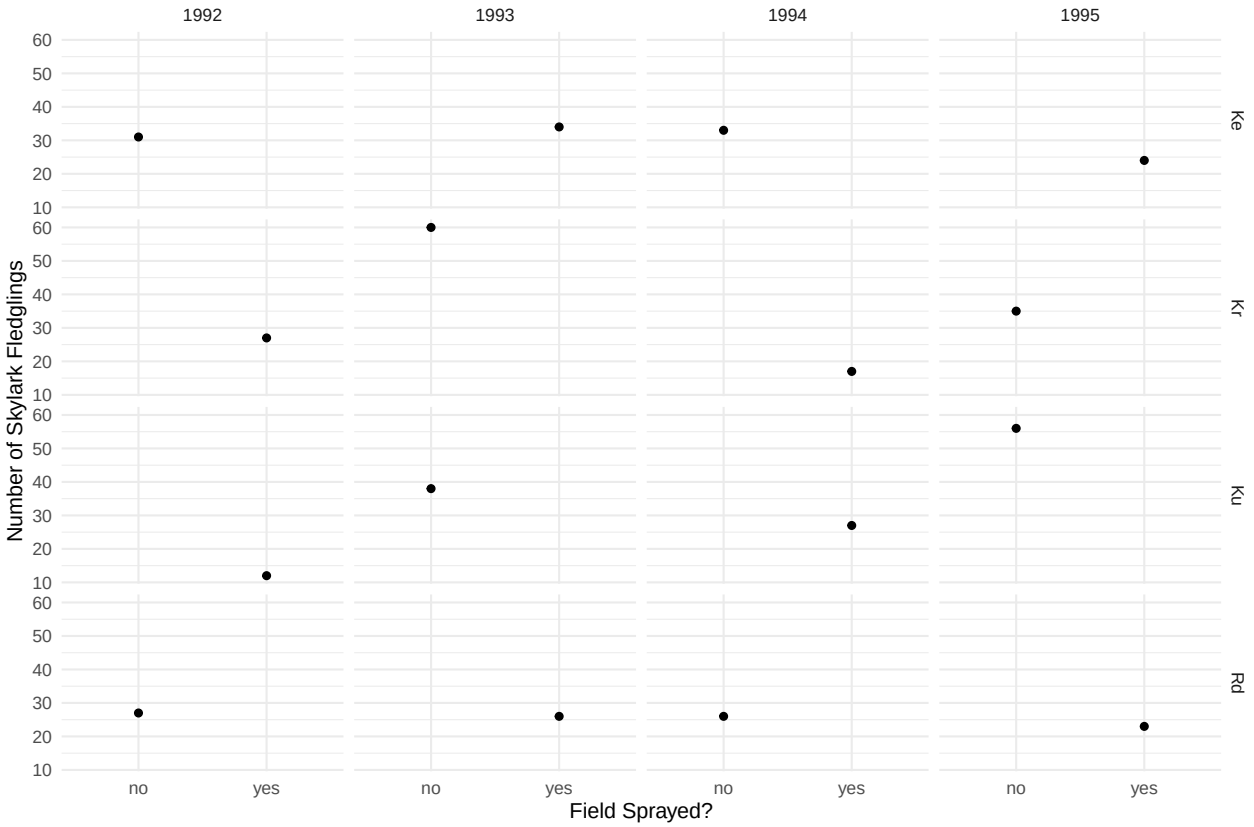
Impact of Pesticides on Skylark Reproductivity

During the four summers from 1992 to 1995 researchers from the National Environmental Research Institute in the Ministry of Environment and Energy in Denmark conducted a study to examine how pesticide use impacts skylark reproduction in barley fields.[1] The study used a fractional factorial design in which each year two of four fields were sprayed with pesticides while the other two fields were not.[2] Which fields were sprayed was alternated so that a field was sprayed every other year. The number of fledgling skylarks produced in each field each year was recorded. The data are in the `skylark` data frame from the **trtools** package. The data are plotted below.

```
library(trtools)
library(ggplot2)
p <- ggplot(skylark, aes(x = spray, y = count)) +
   geom_point() + facet_grid(field ~ year) +
   labs(shape = "Field", x = "Field Sprayed?",
    y = "Number of Skylark Fledglings") + theme_minimal()
plot(p)
```

[1] Odderskær, P., Prang, A., Eknegaard, N., & Andersen, P. N. (1997). Skylark reproduction in pesticide treated fields (Comparative studies of Alauda arvensis breeding performance in sprayed and unsprayed barley fields). *Bekæmpelsesmiddelforskning fra Miljøstyrelsennr*, 32, National Environmental Research Institute, Ministry of the Environment and Energy, Denmark: Danish Environmental Protection Agency.

[2] A fractional factorial design is a design in which observations are made at only a subset of the possible combinations of levels of two or more factors. Such designs are quite economical but can preclude the estimation of interactions. This does not mean that such interactions are not present, but rather that if they are they are confounded with the main effects. For this particular design it is only possible to fully estimate a model with "main effects" for each of the three factors. Ideally factional factorial designs are used when interactions are negligible.



The plot clearly shows the incomplete nature of the fractional factorial design. In any given year, a field either was or was not sprayed. The objective is to investigate the effect of spraying on the number of skylarks while controlling for the effects of year and field.

1. Estimate a Poisson regression model for the number of skylark fledglings as your response variable that will reproduce the following results.

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

```
             Estimate Std. Error  z value   Pr(>|z|)    2.5 %   97.5 %
(Intercept)  3.430943    0.13262 25.86999 1.450e-147  3.16352  3.68367
sprayyes    -0.456126    0.09385 -4.86011  1.173e-06 -0.64141 -0.27324
fieldKr      0.049089    0.12672  0.38738  6.985e-01 -0.19929  0.29806
fieldKu      0.004964    0.12800  0.03879  9.691e-01 -0.24611  0.25625
fieldRd     -0.179048    0.13417 -1.33452  1.820e-01 -0.44342  0.08326
year1993     0.462623    0.13064  3.54108  3.985e-04  0.20868  0.72149
year1994     0.060018    0.14149  0.42420  6.714e-01 -0.21735  0.33816
year1995     0.327281    0.13411  2.44041  1.467e-02  0.06596  0.59240
```

Note that here `m` is a model object created using the `glm` function.

Solution: The results can be replicated as follows. Note that the output above indicates that only the "main effects" of spray, field, and year were specified. We can see that there are indicator variables for spray, field, and year, but no interaction terms.

```
m <- glm(count ~ spray + field + year, family = poisson, data = skylark)
cbind(summary(m)$coefficients, confint(m))
```

```
             Estimate Std. Error  z value   Pr(>|z|)    2.5 %   97.5 %
(Intercept)  3.430943    0.13262 25.86999 1.450e-147  3.16352  3.68367
```

```
sprayyes    -0.456126    0.09385 -4.86011  1.173e-06 -0.64141 -0.27324
fieldKr      0.049089    0.12672  0.38738  6.985e-01 -0.19929  0.29806
fieldKu      0.004964    0.12800  0.03879  9.691e-01 -0.24611  0.25625
fieldRd     -0.179048    0.13417 -1.33452  1.820e-01 -0.44342  0.08326
year1993     0.462623    0.13064  3.54108  3.985e-04  0.20868  0.72149
year1994     0.060018    0.14149  0.42420  6.714e-01 -0.21735  0.33816
year1995     0.327281    0.13411  2.44041  1.467e-02  0.06596  0.59240
```

2. According to the model, the expected number of skylark fledglings when a field *is* sprayed is about 0.63 times that when a field is *not* sprayed. In other words, we estimate when the fields are sprayed the expected number of skylark fledglings by about 37 percent lower. Note that these effects are qualified as "when controlling for field and year" or "for any given field or year" since the model conditions the distribution of counts on these variables as well. Confirm this result by applying the exponential function to the parameter estimates.

   Solution: The quantity we want to estimate is e^{β_1} which is the "rate ratio" for the effect of spraying. This gives the "multiplicative effect" of spraying — i.e., the ratio of the expected count when spraying to the expected count when not spraying. You could compute this in R or with a hand calculator by computing `exp(-0.4561258)`. Also we can do this in R as follows.

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

```
                    2.5 %  97.5 %
(Intercept) 30.9058 23.6537 39.7921
sprayyes     0.6337  0.5265  0.7609
fieldKr      1.0503  0.8193  1.3472
fieldKu      1.0050  0.7818  1.2921
fieldRd      0.8361  0.6418  1.0868
year1993     1.5882  1.2321  2.0575
year1994     1.0619  0.8046  1.4024
year1995     1.3872  1.0682  1.8083
```

   Note that I've included the endpoints of the confidence intervals as well. If you just wanted the point estimate you could use `exp(coef(m))`. The function `coef` extracts the parameter estimates from the model object `m`.

3. Replicate the result that was obtained in the previous problem using the `contrast` function and the `tf = exp` option. Note that you will need to specify a field and a year, but since the model does not contain any interactions between spray and those variables the multiplicative effect of spray will not depend on the field or year you specify.

   Solution: Here is how we could make inferences about the spray effect using the `contrast` function.

```
trtools::contrast(m, tf = exp,
    a = list(spray = "yes", field = "Ke", year = "1992"),
    b = list(spray = "no", field = "Ke", year = "1992"))
```

```
 estimate  lower  upper
   0.6337 0.5273 0.7617
```

   Also we can "flip" the rate ratio.

```
trtools::contrast(m, tf = exp,
    a = list(spray = "no", field = "Ke", year = "1992"),
    b = list(spray = "yes", field = "Ke", year = "1992"))
```

```
 estimate lower upper
    1.578 1.313 1.897
```

This shows that if a field is not sprayed the the expected number of fledglings is about 1.58 (58%) higher than when it is sprayed. Again, note that the field and year does not matter since there are no interactions between spray and either of those variables, so the multiplicative effect of spray will be the same regardless of the field and year. You could verify this by specifying a different field and/or year. Note that there are slight differences in the confidence intervals obtained here and in the previous problem. That is because `confint` computes a profile likelihood confidence interval whereas `contrast` computes a Wald confidence interval. Also note that while we were mainly interested in the effect of spray here, we could easily compare different fields or years.

You can also use the **emmeans** package to estimate rate ratios, provided that the explanatory variable in question is categorical. Here is how you can use it to estimate the rate ratios for every combination of year and field (note that they are all the same for this model).

```
library(emmeans)
pairs(emmeans(m, ~ spray | field*year),
  type = "response", infer = TRUE)
```

```
field = Ke, year = 1992:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Kr, year = 1992:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Ku, year = 1992:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Rd, year = 1992:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Ke, year = 1993:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Kr, year = 1993:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Ku, year = 1993:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Rd, year = 1993:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Ke, year = 1994:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Kr, year = 1994:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
```

```
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Ku, year = 1994:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Rd, year = 1994:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Ke, year = 1995:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Kr, year = 1995:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Ku, year = 1995:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

field = Rd, year = 1995:
 contrast ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.58 0.148 Inf      1.31       1.9    1   4.860  <.0001

Confidence level used: 0.95
Intervals are back-transformed from the log scale
Tests are performed on the log scale
```

4. Use `contrast` to estimate the expected number of fledglings with and without spraying for any given field or year.

   Solution: For example, the expected counts with and without spraying for field `Ke` in 1992 is

```
trtools::contrast(m, a = list(spray = c("yes","no"), field = "Ke", year = "1992"),
    tf = exp, cnames = c("spray","no spray"))
```

```
          estimate lower upper
spray        19.59 15.07 25.45
no spray     30.91 23.83 40.08
```

   Also note that the ratio of these estimated expected counts (spray divided by no spray) is about 0.63 which is what we estimated earlier. If you wanted to estimate a bunch of expected counts it might be easier to use `glmint`. Below I have estimated the estimated expected count for every combination of spray, field, and year. Note that `glmint` does not require you to specify a transformation function. It knows what function to use based on the model.

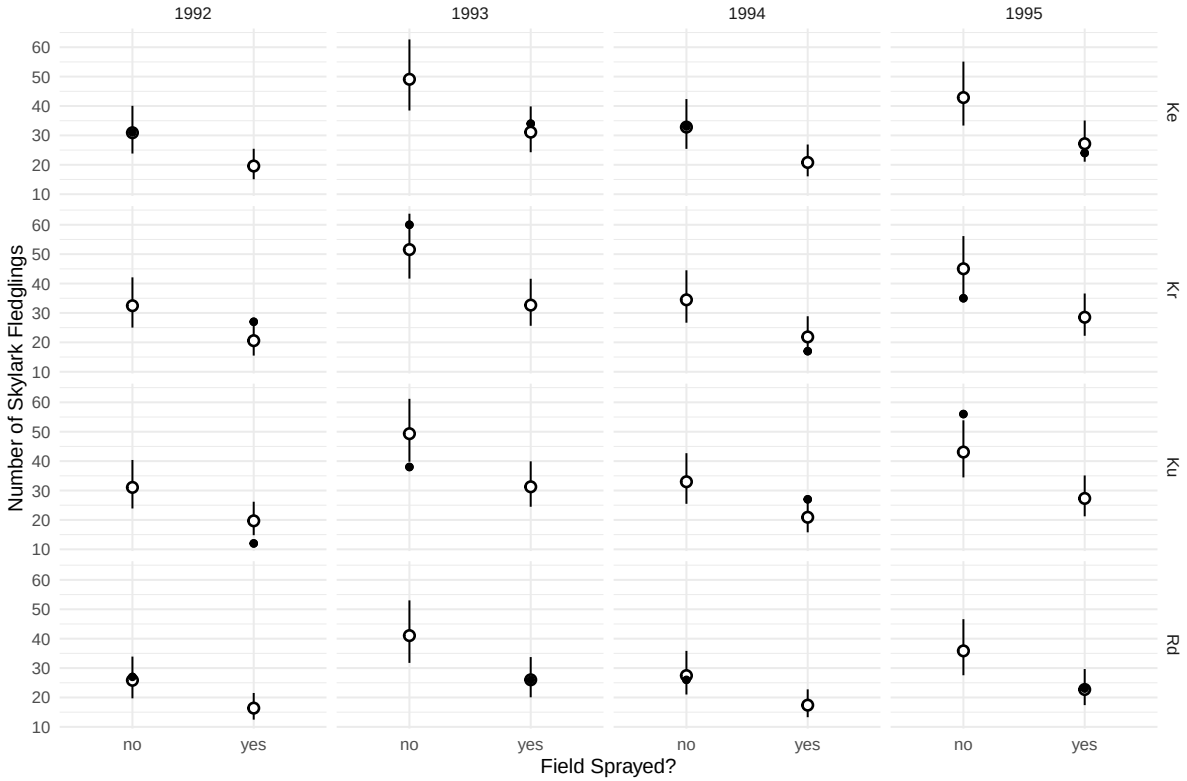
```
d <- expand.grid(spray = c("yes","no"), field = c("Ke","Kr","Ku","Rd"),
    year = c("1992","1993","1994","1995"))
cbind(d, glmint(m, newdata = d))
```

```
   spray field year   fit   low   upp
1    yes    Ke 1992 19.59 15.07 25.45
2     no    Ke 1992 30.91 23.83 40.08
3    yes    Kr 1992 20.57 15.50 27.31
4     no    Kr 1992 32.46 25.02 42.11
```

```
5    yes    Ku 1992 19.68 14.80 26.18
6     no    Ku 1992 31.06 23.88 40.39
7    yes    Rd 1992 16.38 12.46 21.52
8     no    Rd 1992 25.84 19.69 33.90
9    yes    Ke 1993 31.11 24.27 39.87
10    no    Ke 1993 49.09 38.46 62.65
11   yes    Kr 1993 32.67 25.64 41.63
12    no    Kr 1993 51.56 41.68 63.76
13   yes    Ku 1993 31.26 24.47 39.93
14    no    Ku 1993 49.33 39.77 61.19
15   yes    Rd 1993 26.01 20.05 33.74
16    no    Rd 1993 41.04 31.76 53.03
17   yes    Ke 1994 20.80 16.08 26.90
18    no    Ke 1994 32.82 25.42 42.37
19   yes    Kr 1994 21.84 16.52 28.88
20    no    Kr 1994 34.47 26.69 44.52
21   yes    Ku 1994 20.90 15.78 27.69
22    no    Ku 1994 32.98 25.47 42.70
23   yes    Rd 1994 17.39 13.29 22.76
24    no    Rd 1994 27.44 21.00 35.84
25   yes    Ke 1995 27.17 21.05 35.07
26    no    Ke 1995 42.87 33.35 55.11
27   yes    Kr 1995 28.54 22.24 36.62
28    no    Kr 1995 45.03 36.11 56.15
29   yes    Ku 1995 27.30 21.22 35.13
30    no    Ku 1995 43.09 34.46 53.88
31   yes    Rd 1995 22.72 17.39 29.67
32    no    Rd 1995 35.84 27.55 46.63
```

Here `fit` is the estimated expected count, and `low` and `upp` are the endpoints of the confidence interval for the expected count. Note that even though observations were not made at every combination of spray, field, and year, we can still *estimate* the expected counts for each combination. But we can only do this by assuming (implicitly when specifying the model) that there are no interactions among these variables. We could add a layer to the plot above to show the estimates and confidence intervals for the expected counts.

```
d <- cbind(d, glmint(m, newdata = d))
p <- p + geom_pointrange(aes(y = fit, ymin = low, ymax = upp),
  shape = 21, fill = "white", data = d) + geom_point()
plot(p)
```



Note that I "added" a layer after `geom_pointrange` to show points corresponding to the observed counts so that they would be on top of the points showing the estimated expected counts.

Here is how to use the **emmeans** package to estimate the expected counts (rates) for each combination of spray, field, and year.

```
library(emmeans)
emmeans(m, ~ spray*field*year, type = "response")
```

```
 spray field year rate   SE  df asymp.LCL asymp.UCL
 no    Ke    1992 30.9 4.10 Inf      23.8      40.1
 yes   Ke    1992 19.6 2.62 Inf      15.1      25.4
 no    Kr    1992 32.5 4.31 Inf      25.0      42.1
 yes   Kr    1992 20.6 2.97 Inf      15.5      27.3
 no    Ku    1992 31.1 4.16 Inf      23.9      40.4
 yes   Ku    1992 19.7 2.87 Inf      14.8      26.2
 no    Rd    1992 25.8 3.58 Inf      19.7      33.9
 yes   Rd    1992 16.4 2.28 Inf      12.5      21.5
 no    Ke    1993 49.1 6.11 Inf      38.5      62.6
 yes   Ke    1993 31.1 3.94 Inf      24.3      39.9
 no    Kr    1993 51.6 5.59 Inf      41.7      63.8
 yes   Kr    1993 32.7 4.04 Inf      25.6      41.6
 no    Ku    1993 49.3 5.42 Inf      39.8      61.2
 yes   Ku    1993 31.3 3.90 Inf      24.5      39.9
 no    Rd    1993 41.0 5.37 Inf      31.8      53.0
 yes   Rd    1993 26.0 3.45 Inf      20.1      33.7
 no    Ke    1994 32.8 4.28 Inf      25.4      42.4
 yes   Ke    1994 20.8 2.73 Inf      16.1      26.9
 no    Kr    1994 34.5 4.50 Inf      26.7      44.5
 yes   Kr    1994 21.8 3.11 Inf      16.5      28.9
```

```
 no    Ku    1994 33.0 4.34 Inf      25.5      42.7
 yes   Ku    1994 20.9 3.00 Inf      15.8      27.7
 no    Rd    1994 27.4 3.74 Inf      21.0      35.8
 yes   Rd    1994 17.4 2.39 Inf      13.3      22.8
 no    Ke    1995 42.9 5.49 Inf      33.4      55.1
 yes   Ke    1995 27.2 3.54 Inf      21.1      35.1
 no    Kr    1995 45.0 5.07 Inf      36.1      56.1
 yes   Kr    1995 28.5 3.63 Inf      22.2      36.6
 no    Ku    1995 43.1 4.91 Inf      34.5      53.9
 yes   Ku    1995 27.3 3.51 Inf      21.2      35.1
 no    Rd    1995 35.8 4.81 Inf      27.6      46.6
 yes   Rd    1995 22.7 3.09 Inf      17.4      29.7

Confidence level used: 0.95
Intervals are back-transformed from the log scale
```

```
emmeans(m, ~ spray|field*year, type = "response")
```

```
field = Ke, year = 1992:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    30.9 4.10 Inf      23.8      40.1
 yes   19.6 2.62 Inf      15.1      25.4

field = Kr, year = 1992:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    32.5 4.31 Inf      25.0      42.1
 yes   20.6 2.97 Inf      15.5      27.3

field = Ku, year = 1992:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    31.1 4.16 Inf      23.9      40.4
 yes   19.7 2.87 Inf      14.8      26.2

field = Rd, year = 1992:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    25.8 3.58 Inf      19.7      33.9
 yes   16.4 2.28 Inf      12.5      21.5

field = Ke, year = 1993:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    49.1 6.11 Inf      38.5      62.6
 yes   31.1 3.94 Inf      24.3      39.9

field = Kr, year = 1993:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    51.6 5.59 Inf      41.7      63.8
 yes   32.7 4.04 Inf      25.6      41.6

field = Ku, year = 1993:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    49.3 5.42 Inf      39.8      61.2
 yes   31.3 3.90 Inf      24.5      39.9

field = Rd, year = 1993:
 spray rate   SE  df asymp.LCL asymp.UCL
```

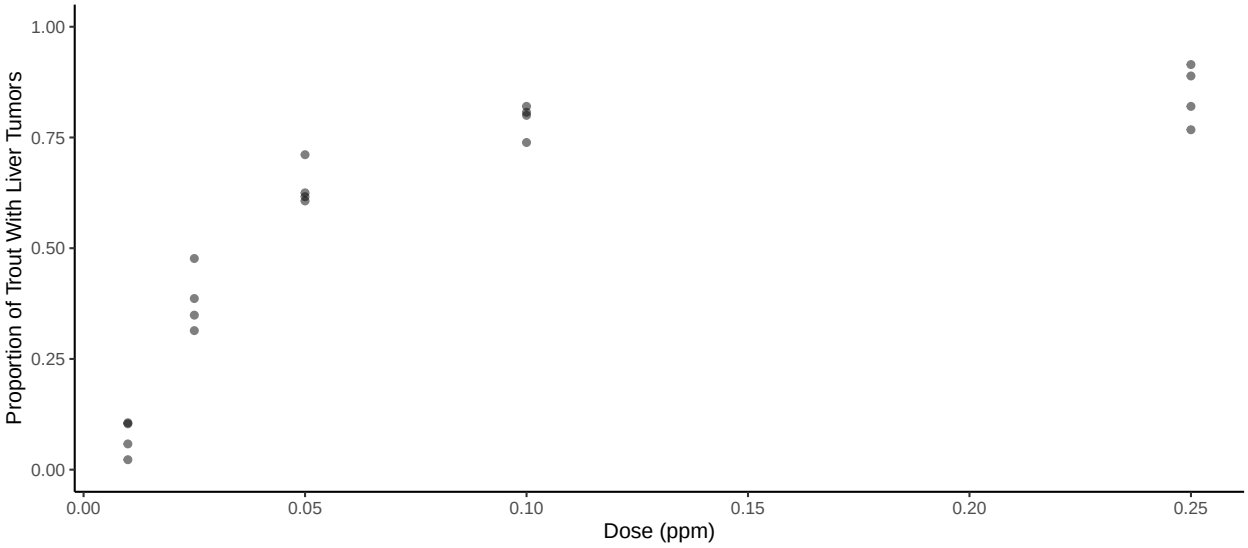
```
 no    41.0 5.37 Inf      31.8      53.0
 yes   26.0 3.45 Inf      20.1      33.7

field = Ke, year = 1994:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    32.8 4.28 Inf      25.4      42.4
 yes   20.8 2.73 Inf      16.1      26.9

field = Kr, year = 1994:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    34.5 4.50 Inf      26.7      44.5
 yes   21.8 3.11 Inf      16.5      28.9

field = Ku, year = 1994:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    33.0 4.34 Inf      25.5      42.7
 yes   20.9 3.00 Inf      15.8      27.7

field = Rd, year = 1994:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    27.4 3.74 Inf      21.0      35.8
 yes   17.4 2.39 Inf      13.3      22.8

field = Ke, year = 1995:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    42.9 5.49 Inf      33.4      55.1
 yes   27.2 3.54 Inf      21.1      35.1

field = Kr, year = 1995:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    45.0 5.07 Inf      36.1      56.1
 yes   28.5 3.63 Inf      22.2      36.6

field = Ku, year = 1995:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    43.1 4.91 Inf      34.5      53.9
 yes   27.3 3.51 Inf      21.2      35.1

field = Rd, year = 1995:
 spray rate   SE  df asymp.LCL asymp.UCL
 no    35.8 4.81 Inf      27.6      46.6
 yes   22.7 3.09 Inf      17.4      29.7

Confidence level used: 0.95
Intervals are back-transformed from the log scale
```

Aflatoxicol and Liver Tumors in Trout

The data in the data frame `ex2116` in the **Sleuth3** package are from an experiment that investigated the relationship between aflatoxicol and liver tumors in trout. The figure below shows the proportion of trout in each tank that developed liver tumors as well as the dose of aflatoxicol to which the trout were exposed. Aflatoxicol is a metabolite of Aflatoxin B1, a toxic by-product produced by a mold that infects some nuts and grains. Twenty tanks of rainbow trout embryos were exposed to one of five doses of aflatoxicol for one hour. The number of fish in each tank that developed liver tumors one year later was then observed. The

plot below shows the data.

```
library(Sleuth3)
library(ggplot2)
p <- ggplot(ex2116, aes(x = Dose, y = Tumor/Total)) +
   geom_point(alpha = 0.5) + theme_classic() + ylim(0, 1) +
   labs(x = "Dose (ppm)", y = "Proportion of Trout With Liver Tumors")
plot(p)
```



The goal here is to estimate the effect of aflatoxicol on the risk of liver tumors in trout. Here we will consider three different logistic regression models.

1. Consider the logistic regression model

$$\log\left[\frac{E(Y_i)}{1-E(Y_i)}\right] = \beta_0 + \beta_1 d_i,$$

where Y_i is the i-th observation of the proportion of trout from a tank that developed liver tumors and d_i is the corresponding dose of aflatoxicol to which those trout were exposed. This model can also be written as

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

where $\eta_i = \beta_0 + \beta_1 d_i$. Note that by definition $E(Y_i)$ is also the *probability* that a trout from a given tank will develop liver tumors, and $E(Y_i)/[1-E(Y_i)]$ is the *odds* that a trout from a given tank will develop liver tumors. Estimate this model using `glm`. You should be able to replicate the following results.

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

```
            Estimate Std. Error z value  Pr(>|z|)  2.5 %  97.5 %
(Intercept)   -0.867    0.07673   -11.3 1.321e-29 -1.019 -0.7179
Dose          14.334    0.93695    15.3 7.838e-53 12.558 16.2346
```

Next estimate the odds ratio for the effect of increasing dose by 0.05 ppm using the `contrast` function.[3] Remember to use the `tf = exp` option. You should find that increasing dose by 0.05 ppm is estimated

[3]Here e^{β_1} would be the odds ratio for the effect of increasing dose by 1 ppm. However that is probably not a realistic effect as it would be a relatively large increase in dose. The study only considered up to 0.25 ppm. Using `contrast` is convenient here to estimate the odds ratio for the effect of an arbitrary change in dose.

to increase the odds of tumor development by a factor of about 2.05 (i.e., approximately a 105% increase in the odds of tumor development). That is the odds ratio for the effect of increasing dose by 0.05 ppm.

Solution: We can estimate the model as follows.

```
m1 <- glm(cbind(Tumor, Total-Tumor) ~ Dose, family = binomial, data = ex2116)
cbind(summary(m1)$coefficients, confint(m1))
```

```
            Estimate Std. Error z value  Pr(>|z|)  2.5 %  97.5 %
(Intercept)   -0.867    0.07673   -11.3 1.321e-29 -1.019 -0.7179
Dose          14.334    0.93695    15.3 7.838e-53 12.558 16.2346
```

Now we can use `contrast` to estimate the odds ratio.

```
trtools::contrast(m1, a = list(Dose = 0.1), b = list(Dose = 0.05), tf = exp)
```

```
 estimate lower upper
    2.048 1.868 2.245
```

Note that the same odds ratio would be obtained for any two values of dose that differ by 0.05 ppm. Note that if you computed e^{β_1} you would be estimating the odds ratio for the effect of increasing dose by 1 ppm (which is quite a bit given that in the study dose varied from 0 to 0.25 ppm). Note that you can also use `contrast` to estimate the odds of tumor development at any value of dose. Here I will do it for the values of dose used in the study, but we could use any (reasonable) value of dose.

```
trtools::contrast(m1, a = list(Dose = c(0.01, 0.025, 0.05, 0.1, 0.25)), tf = exp,
    cnames = c("0.01 ppm","0.025 ppm","0.05 ppm","0.1 ppm","0.25 ppm"))
```

```
          estimate   lower   upper
0.01 ppm    0.4850  0.4225  0.5566
0.025 ppm   0.6013  0.5323  0.6792
0.05 ppm    0.8604  0.7736  0.9570
0.1 ppm     1.7618  1.5471  2.0062
0.25 ppm   15.1257 10.4786 21.8338
```

Note that the odds ratio we computed above is simply the ratio of the odds at 0.1 ppm and 0.05 ppm. The probability of tumor development at a given dose can also be obtained using `contrast` with the `tf = plogis` option.

```
trtools::contrast(m1, a = list(Dose = c(0.01, 0.025, 0.05, 0.1, 0.25)), tf = plogis,
    cnames = c("0.01 ppm","0.025 ppm","0.05 ppm","0.1 ppm","0.25 ppm"))
```

```
          estimate  lower  upper
0.01 ppm    0.3266 0.2970 0.3576
0.025 ppm   0.3755 0.3474 0.4045
0.05 ppm    0.4625 0.4362 0.4890
0.1 ppm     0.6379 0.6074 0.6674
0.25 ppm    0.9380 0.9129 0.9562
```

Note that `glmint` can also be used to obtain the estimated probabilities.

```
d <- data.frame(Dose = c(0.01, 0.025, 0.05, 0.1, 0.25))
cbind(d, glmint(m1, newdata = d))
```

```
   Dose    fit    low    upp
1 0.010 0.3266 0.2970 0.3576
2 0.025 0.3755 0.3474 0.4045
3 0.050 0.4625 0.4362 0.4890
4 0.100 0.6379 0.6074 0.6674
5 0.250 0.9380 0.9129 0.9562
```

2. Now consider a model where we use the base-2 logarithm of dose as the explanatory variable so that

$$\eta_i = \beta_0 + \beta_1 \log_2(d_i).$$

Recall that the function $\log_2$ is known to R as `log2`. Estimate this model using `glm`. You should be able to replicate the following results.

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

```
            Estimate Std. Error z value  Pr(>|z|)  2.5 % 97.5 %
(Intercept)   4.1634     0.2085   19.97 9.564e-89 3.7631  4.581
log2(Dose)    0.8997     0.0446   20.17 1.628e-90 0.8141  0.989
```

Here increasing the base-2 logarithm of dose by one unit is the same thing as *doubling* dose, and so the effect on the odds ratio of doubling dose will be the same regardless of what you double (e.g., 0.05 to 1 ppm, 0.1 to 0.2 ppm, etc.).[4] Confirm that the odds ratio for the effect of doubling dose is approximately 2.46, which is approximately a 146% increase in the odds of tumor development. That is, doubling dose will increase the odds of tumor development by a factor of about 2.46. Also confirm that these odds ratios do not depend on the *base* of the logarithm by trying the natural logarithm with the model $\eta_i = \beta_0 + \beta_1 \log(d_i)$.

Solution: We can estimate the model as follows.

```
m2 <- glm(cbind(Tumor, Total-Tumor) ~ log2(Dose), family = binomial, data = ex2116)
cbind(summary(m2)$coefficients, confint(m2))
```

```
            Estimate Std. Error z value  Pr(>|z|)  2.5 % 97.5 %
(Intercept)   4.1634     0.2085   19.97 9.564e-89 3.7631  4.581
log2(Dose)    0.8997     0.0446   20.17 1.628e-90 0.8141  0.989
```

Now we can use `contrast` to estimate the odds ratio.

```
trtools::contrast(m2, a = list(Dose = 0.1), b = list(Dose = 0.05), tf = exp)
```

```
 estimate lower upper
    2.459 2.253 2.684
```

Note that the same odds ratio would be obtained for any two values of dose that differ by a factor of 2.

3. Rather than trying to decide between using dose or some transformation of dose in the model, we can instead define dose as a 5-level factor. There are two ways we could specify dose as a factor. One would be to create a new variable.

```
ex2116$Dosef <- factor(ex2116$Dose)
```

The levels of `Dosef` will be the original values of `Dose` but converted to strings, which we can see if we use the `levels` function.

```
levels(ex2116$Dosef)
```

```
[1] "0.01"  "0.025" "0.05"  "0.1"   "0.25"
```

Another approach is to replace `Dose` in the model formula with `factor(Dose)`. Use the `contrast` function to estimate the odds ratio for the odds of tumor development at 0.025 ppm versus 0.01 ppm,

[4] We do not have to use the odds ratio for the effect of doubling dose just because we are using the base-2 logarithm of dose as our explanatory variable. We could also estimate the odds ratio for the effect of increasing dose from, say, 0.05 ppm to 0.1 ppm. But we would need to remember that because we are using the base-2 logarithm of dose as an explanatory variable that this would *not* be the same odds ratio as increasing dose the same amount from, say, 0.1 ppm to 0.15 ppm. Similarly for the previous model where we did not use the base-2 logarithm of dose, we could still estimate the odds ratio for the effect of doubling dose. But here we would need to remember that the odds ratio of doubling from, say, 0.05 ppm to 0.1 ppm would *not* be the same as the odds ratio for doubling from 0.1 ppm to 0.2 ppm.

0.05 ppm versus 0.01 ppm, 0.1 ppm versus 0.01 ppm, and 0.25 ppm versus 0.01 ppm.[5] You should find that these odds ratios are approximately 7.94, 22.92, 48.91, and 70.84. Thus, for example, at a dose of 0.025 ppm the odds of tumor development is about 7.94 times higher than it is at a dose of 0.01 ppm (i.e., about 694% higher).

Solution: Here is how we can estimate this model.

```
m3 <- glm(cbind(Tumor, Total-Tumor) ~ factor(Dose), family = binomial, data = ex2116)
cbind(summary(m3)$coefficients, confint(m3))
```

```
                  Estimate Std. Error z value  Pr(>|z|)  2.5 % 97.5 %
(Intercept)         -2.556     0.2076 -12.310 8.049e-35 -2.988 -2.171
factor(Dose)0.025    2.073     0.2353   8.809 1.264e-18  1.628  2.553
factor(Dose)0.05     3.132     0.2354  13.306 2.130e-40  2.688  3.614
factor(Dose)0.1      3.890     0.2453  15.857 1.252e-56  3.427  4.391
factor(Dose)0.25     4.260     0.2566  16.605 6.436e-62  3.775  4.784
```

```
trtools::contrast(m3,
    a = list(Dose = c(0.025,0.05,0.1,0.25)),
    b = list(Dose = 0.01), tf = exp,
    cnames = c("0.025 vs 0.01 ppm","0.05 vs 0.01 ppm","0.1 vs 0.01 ppm","0.25 vs 0.01 ppm"))
```

```
                  estimate lower  upper
0.025 vs 0.01 ppm    7.945  5.01  12.60
0.05 vs 0.01 ppm    22.920 14.45  36.36
0.1 vs 0.01 ppm     48.909 30.24  79.10
0.25 vs 0.01 ppm    70.840 42.84 117.13
```

Note that we do not necessarily need to put the dose values in quotes since they will be converted to labels anyway. Since the 0.01 ppm level is the reference category we can also obtain the odds ratios by applying the exponential function to the parameter estimates.

```
exp(cbind(coef(m3), confint(m3)))
```

```
                            2.5 %   97.5 %
(Intercept)         0.07764  0.05038   0.1141
factor(Dose)0.025   7.94467  5.09246  12.8513
factor(Dose)0.05   22.92031 14.70370  37.1110
factor(Dose)0.1    48.90919 30.77256  80.7191
factor(Dose)0.25   70.84000 43.61279 119.5297
```

Again note that the confidence intervals are slightly different because `contrast` computes Wald confidence intervals whereas `confint` computes profile likelihood confidence intervals. Since dose is being treated as a categorical variable, we can also use functions from the **emmeans** package to estimate these odds ratios. Using `pairs` provides odds ratios for all pairs of dose levels.

```
library(emmeans)
pairs(emmeans(m3, ~Dose, type = "response"),
  adjust = "none", infer = TRUE)
```

```
 contrast     odds.ratio      SE  df asymp.LCL asymp.UCL null z.ratio p.value
 0.01 / 0.025     0.1259 0.02961 Inf    0.0794    0.1996    1  -8.809  <.0001
 0.01 / 0.05      0.0436 0.01027 Inf    0.0275    0.0692    1 -13.306  <.0001
 0.01 / 0.1       0.0204 0.00502 Inf    0.0126    0.0331    1 -15.857  <.0001
 0.01 / 0.25      0.0141 0.00362 Inf    0.0085    0.0233    1 -16.605  <.0001
```

[5]Note that how you specify the levels of dose will depend on whether you created a new variable like `Dosef` or converted it to a factor within the model formula with `factor(Dose)`. For the latter you will need to specify dose as a *number* but if you created it to a new variable you will need to specify it as a *string* by enclosing it in quotes.

```
 0.025 / 0.05     0.3466 0.05431 Inf    0.2550    0.4712    1  -6.762  <.0001
 0.025 / 0.1      0.1624 0.02781 Inf    0.1161    0.2272    1 -10.614  <.0001
 0.025 / 0.25     0.1121 0.02097 Inf    0.0777    0.1618    1 -11.699  <.0001
 0.05 / 0.1       0.4686 0.08031 Inf    0.3349    0.6557    1  -4.423  <.0001
 0.05 / 0.25      0.3236 0.06055 Inf    0.2242    0.4669    1  -6.029  <.0001
 0.1 / 0.25       0.6904 0.13774 Inf    0.4670    1.0208    1  -1.857  0.0633

Confidence level used: 0.95
Intervals are back-transformed from the log odds ratio scale
Tests are performed on the log odds ratio scale
```

By default the odds ratios are computed here with the odds for a tumor for the larger dose in the denominator. We can reverse these odds ratios to put the odds corresponding to the larger dose in the numerator.

```
pairs(emmeans(m3, ~Dose, type = "response"), adjust = "none",
  rev = TRUE, infer = TRUE)
```

```
 contrast     odds.ratio     SE  df asymp.LCL asymp.UCL null z.ratio p.value
 0.025 / 0.01       7.94  1.869 Inf      5.01     12.60    1   8.809  <.0001
 0.05 / 0.01       22.92  5.395 Inf     14.45     36.36    1  13.306  <.0001
 0.05 / 0.025       2.88  0.452 Inf      2.12      3.92    1   6.762  <.0001
 0.1 / 0.01        48.91 11.998 Inf     30.24     79.10    1  15.857  <.0001
 0.1 / 0.025        6.16  1.054 Inf      4.40      8.61    1  10.614  <.0001
 0.1 / 0.05         2.13  0.366 Inf      1.53      2.99    1   4.423  <.0001
 0.25 / 0.01       70.84 18.176 Inf     42.84    117.13    1  16.605  <.0001
 0.25 / 0.025       8.92  1.668 Inf      6.18     12.86    1  11.699  <.0001
 0.25 / 0.05        3.09  0.578 Inf      2.14      4.46    1   6.029  <.0001
 0.25 / 0.1         1.45  0.289 Inf      0.98      2.14    1   1.857  0.0633

Confidence level used: 0.95
Intervals are back-transformed from the log odds ratio scale
Tests are performed on the log odds ratio scale
```

Using the `contrast` function from the **emmeans** package (which is different from the function of the same name in the **trtools** package) we can also obtain odds ratios for comparing against a given level, such as the lowest dose.

```
contrast(emmeans(m3, ~Dose, type = "response"), method = "trt.vs.ctrl",
  ref = 1, adjust = "none", infer = TRUE)
```

```
 contrast     odds.ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 0.025 / 0.01       7.94  1.87 Inf      5.01      12.6    1   8.809  <.0001
 0.05 / 0.01       22.92  5.39 Inf     14.45      36.4    1  13.306  <.0001
 0.1 / 0.01        48.91 12.00 Inf     30.24      79.1    1  15.857  <.0001
 0.25 / 0.01       70.84 18.18 Inf     42.84     117.1    1  16.605  <.0001

Confidence level used: 0.95
Intervals are back-transformed from the log odds ratio scale
Tests are performed on the log odds ratio scale
```

The `ref = 1` indicates that we want odds ratios relative to the first level.

4. Use `contrast` to estimate the odds and probability of tumor development at each value of dose used in the study for any of the three models.

   Solution: I will use the model in which dose was specified as a factor, but the syntax would be the same for any of the models. The estimated odds can be computed as follows.

```
trtools::contrast(m3, a = list(Dose = c(0.01, 0.025, 0.05, 0.1, 0.25)),
  tf = exp, cnames = c("0.01 ppm","0.025 ppm","0.05 ppm","0.1 ppm","0.25 ppm"))
```

```
          estimate   lower  upper
0.01 ppm   0.07764 0.05168 0.1166
0.025 ppm  0.61682 0.49654 0.7662
0.05 ppm   1.77953 1.43188 2.2116
0.1 ppm    3.79730 2.93938 4.9056
0.25 ppm   5.50000 4.09298 7.3907
```

The estimated probabilities can be computed as follows.

```
trtools::contrast(m3, a = list(Dose = c(0.01, 0.025, 0.05, 0.1, 0.25)),
  tf = plogis, cnames = c("0.01 ppm","0.025 ppm","0.05 ppm","0.1 ppm","0.25 ppm"))
```

```
          estimate   lower  upper
0.01 ppm   0.07205 0.04914 0.1044
0.025 ppm  0.38150 0.33179 0.4338
0.05 ppm   0.64023 0.58880 0.6886
0.1 ppm    0.79155 0.74615 0.8307
0.25 ppm   0.84615 0.80365 0.8808
```

Here is a tip. We took advantage of the fact that the R function `plogis` is the mathematical function $e^z/(1+e^z)$. But what if we wanted to use a function unknown to R. We can define the function as the argument.

```
trtools::contrast(m3, a = list(Dose = c(0.01, 0.025, 0.05, 0.1, 0.25)),
  tf = function(z) exp(z)/(1 + exp(z)),
  cnames = c("0.01 ppm","0.025 ppm","0.05 ppm","0.1 ppm","0.25 ppm"))
```

```
          estimate   lower  upper
0.01 ppm   0.07205 0.04914 0.1044
0.025 ppm  0.38150 0.33179 0.4338
0.05 ppm   0.64023 0.58880 0.6886
0.1 ppm    0.79155 0.74615 0.8307
0.25 ppm   0.84615 0.80365 0.8808
```

Alternatively you could define the function prior to using `contrast`.

```
myfunction <- function(z) {
  exp(z)/(1 + exp(z))
}
trtools::contrast(m3, a = list(Dose = c(0.01, 0.025, 0.05, 0.1, 0.25)),
  tf = myfunction, cnames = c("0.01 ppm","0.025 ppm","0.05 ppm","0.1 ppm","0.25 ppm"))
```

```
          estimate   lower  upper
0.01 ppm   0.07205 0.04914 0.1044
0.025 ppm  0.38150 0.33179 0.4338
0.05 ppm   0.64023 0.58880 0.6886
0.1 ppm    0.79155 0.74615 0.8307
0.25 ppm   0.84615 0.80365 0.8808
```

It is also useful to note that you can use `glmint` to estimate the probabilities.

```
d <- data.frame(Dose = c(0.01,0.025,0.05,0.1,0.25))
glmint(m3, newdata = d)
```

```
      fit     low    upp
1 0.07205 0.04914 0.1044
```

```
2 0.38150 0.33179 0.4338
3 0.64023 0.58880 0.6886
4 0.79155 0.74615 0.8307
5 0.84615 0.80365 0.8808
```

```
cbind(d, glmint(m3, newdata = d))
```

```
   Dose     fit     low    upp
1 0.010 0.07205 0.04914 0.1044
2 0.025 0.38150 0.33179 0.4338
3 0.050 0.64023 0.58880 0.6886
4 0.100 0.79155 0.74615 0.8307
5 0.250 0.84615 0.80365 0.8808
```

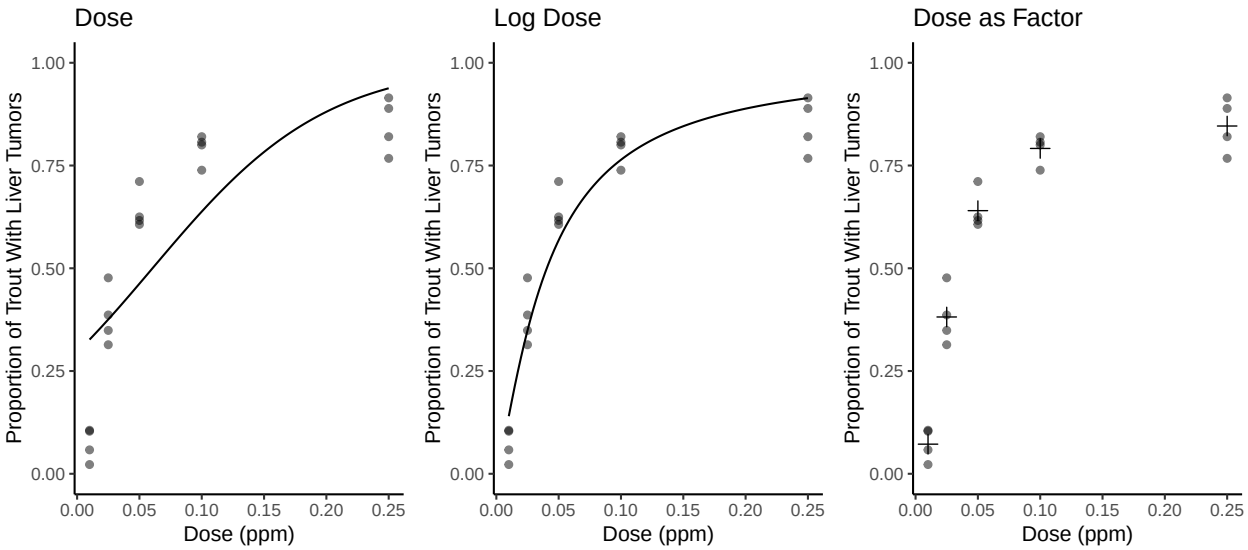
One last way to do this is using the **emmeans** package. It works well when the explanatory variable(s) is/are all categorical (i.e., factors), but is much more limited when the response variable(s) is/are quantitative.

```
library(emmeans)
emmeans(m, ~ Dose, type = "response")
```

```
 Dose  prob     SE  df asymp.LCL asymp.UCL
0.010 0.072 0.0139 Inf    0.0491     0.104
0.025 0.382 0.0261 Inf    0.3318     0.434
0.050 0.640 0.0255 Inf    0.5888     0.689
0.100 0.791 0.0216 Inf    0.7462     0.831
0.250 0.846 0.0196 Inf    0.8037     0.881

Confidence level used: 0.95
Intervals are back-transformed from the logit scale
```

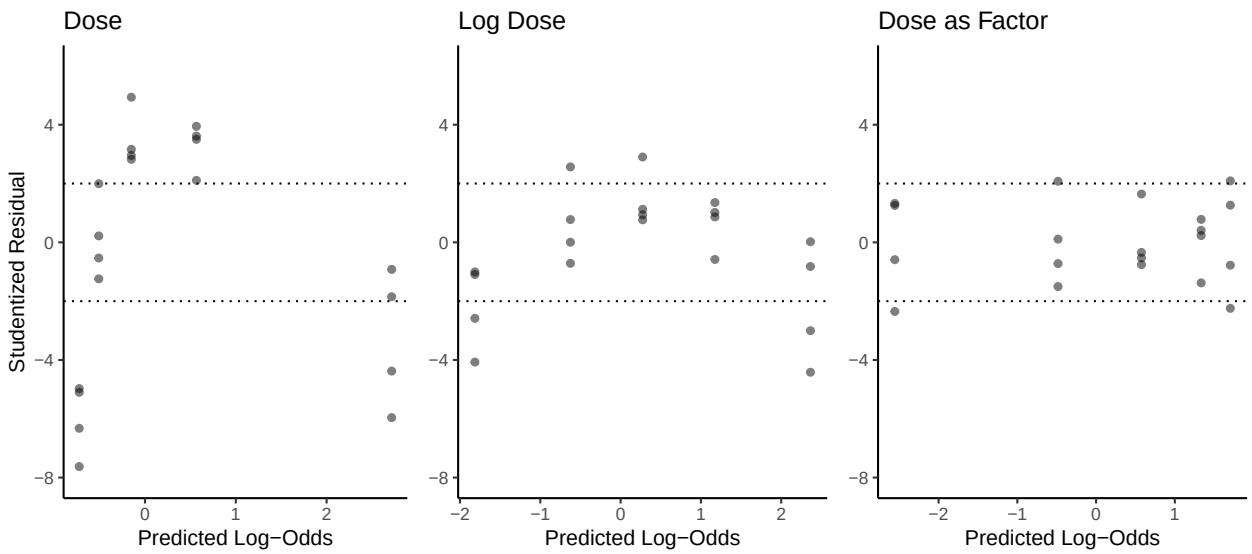
Here are plots of the three models we considered.

```
p <- ggplot(ex2116, aes(x = Dose, y = Tumor/Total)) +
   geom_point(alpha = 0.5) + theme_classic() + ylim(0, 1) +
   labs(x = "Dose (ppm)", y = "Proportion of Trout With Liver Tumors")

m <- glm(cbind(Tumor, Total-Tumor) ~ Dose, family = binomial, data = ex2116)
d <- data.frame(Dose = seq(0.01, 0.25, length = 100))
d$yhat <- predict(m, newdata = d, type = "response")
p1 <- p + geom_line(aes(y = yhat), data = d) + ggtitle("Dose")

m <- glm(cbind(Tumor, Total-Tumor) ~ log2(Dose), family = binomial, data = ex2116)
d <- data.frame(Dose = seq(0.01, 0.25, length = 100))
d$yhat <- predict(m, newdata = d, type = "response")
p2 <- p + geom_line(aes(y = yhat), data = d) + ggtitle("Log Dose")

m <- glm(cbind(Tumor, Total-Tumor) ~ factor(Dose), family = binomial, data = ex2116)
d <- data.frame(Dose = unique(ex2116$Dose))
d$yhat <- predict(m, newdata = d, type = "response")
p3 <- p + geom_point(aes(y = yhat), data = d, pch = 3, size = 3) + ggtitle("Dose as Factor")

cowplot::plot_grid(p1, p2, p3, nrow = 1)
```



Note that the three models do not appear to fit the data equally well. Using the logarithm of dose as an explanatory variable appears to be a better fit than using dose, but both models appear to systematically over-estimate or under-estimate the probability of tumor development. Treating dose as a factor may be a better model here. This is even more clear when looking at residual plots.

```
m1 <- glm(cbind(Tumor, Total-Tumor) ~ Dose, family = binomial, data = ex2116)
m2 <- glm(cbind(Tumor, Total-Tumor) ~ log2(Dose), family = binomial, data = ex2116)
m3 <- glm(cbind(Tumor, Total-Tumor) ~ factor(Dose), family = binomial, data = ex2116)

d1 <- ex2116
d1$yhat <- predict(m1)
d1$residual <- rstudent(m1)

d2 <- ex2116
d2$yhat <- predict(m2)
d2$residual <- rstudent(m2)

d3 <- ex2116
d3$yhat <- predict(m3)
d3$residual <- rstudent(m3)

p <- ggplot(d1, aes(x = yhat, y = residual)) + theme_classic()
p <- p + geom_point(alpha = 0.5) + ylim(-8,6) + geom_hline(yintercept = c(-2,2), linetype = 3)
p <- p + labs(x = "Predicted Log-Odds", y = "Studentized Residual")
p1 <- p + ggtitle("Dose")

p <- ggplot(d2, aes(x = yhat, y = residual)) + theme_classic()
p <- p + geom_point(alpha = 0.5) + ylim(-8,6) + geom_hline(yintercept = c(-2,2), linetype = 3)
p <- p + labs(x = "Predicted Log-Odds", y = NULL)
p2 <- p + ggtitle("Log Dose")

p <- ggplot(d3, aes(x = yhat, y = residual)) + theme_classic()
p <- p + geom_point(alpha = 0.5) + ylim(-8,6) + geom_hline(yintercept = c(-2,2), linetype = 3)
p <- p + labs(x = "Predicted Log-Odds", y = NULL)
p3 <- p + ggtitle("Dose as Factor")

cowplot::plot_grid(p1, p2, p3, nrow = 1)
```



Based on the residuals, the model with dose as a factor appears to provide the best fit to the data. But there may be another model that uses dose as a quantitative explanatory variable (i.e., not a factor) that would be a good fit to these data.