Poisson and Logistic Regression

Statistics 516, Homework 3 (Solutions)

Toxicity of Sodium Bromide on Daphnia magna Reproduction

The data frame `nabr` in the **trtools** package is from a study of the toxicity of sodium bromide (NaBr) on the reproduction of Daphnia magna.[1] Sodium bromide is commonly used in oil and gas drilling and as an antiseptic. It can cause ecological problems if it finds its way into water systems. This study exposed Daphnia magna to different concentrations of sodium bromide over 23 days. The number of offspring per adult over that period was observed. As shown below the mean number of offspring decreased with the concentration of sodium bromide.

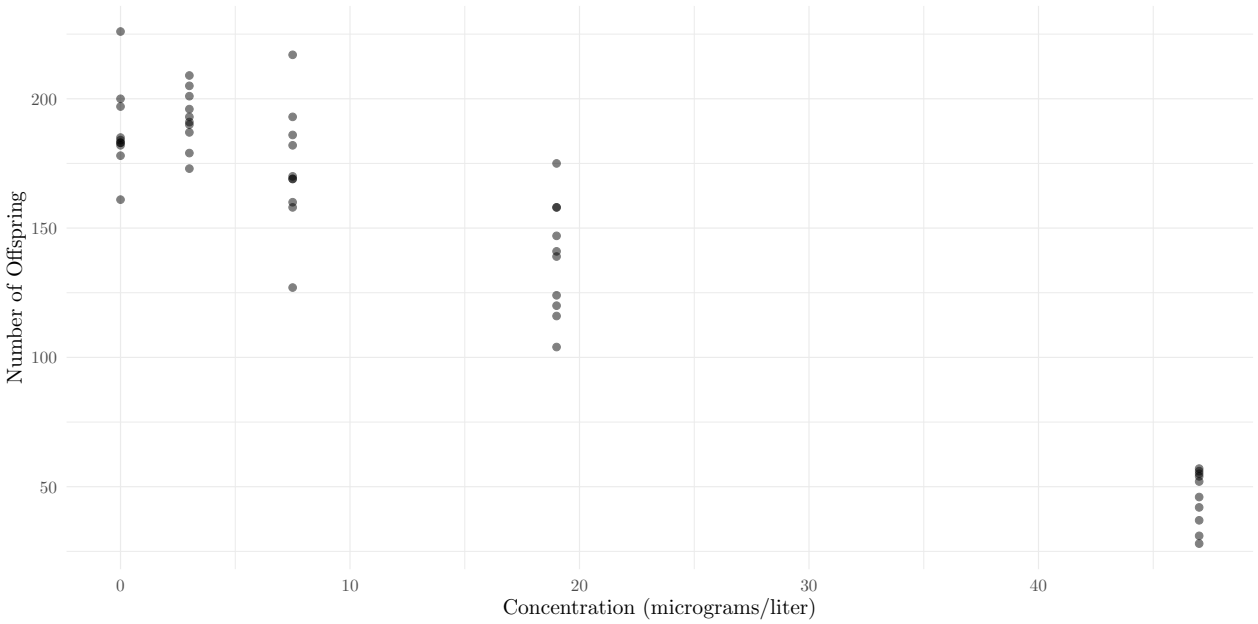
```
library(trtools)
library(dplyr)
nabr %>% group_by(concentration) %>% summarize(mean = mean(young))
```

```
# A tibble: 5 x 2
  concentration  mean
          <dbl> <dbl>
1           0   188.
2           3   192.
3           7.5 173.
4          19   138.
5          47    45.8
```

The figure below shows a plot of the data.

```
library(ggplot2)
p <- ggplot(nabr, aes(x = concentration, y = young)) +
  theme_minimal() + geom_point(alpha = 0.5) +
  labs(x = "Concentration (micrograms/liter)", y = "Number of Offspring")
plot(p)
```

[1] Maul, A., El-Shaarawi, A. H., & Férard, J. F. (1991). Application of negative binomial regression models to the analysis of quantal bioassay data. *Environmetrics*, *2*, 253–261.



Since the response variable is a count these data could maybe be modeled using Poisson regression.

1. Estimate two Poisson regression models with concentration as the explanatory variable and number of offspring as the response variable. For one model treat concentration as a quantitative explanatory variable (i.e., as is), and for the other treat concentration as a categorical explanatory variable by converting it to a factor (either by converting it to a factor within the model formula using `factor(concentration)` or by creating a new variable with something like `nabr$concentrationf <- factor(nabr$concentration)`). Show the parameter estimates and their standard errors for each model using the `summary` function.

   Solution: Here I will estimate the two models and show the parameter estimates and standard errors for each.

```
mq <- glm(young ~ concentration, family = poisson, data = nabr)
summary(mq)$coefficients
```

```
              Estimate Std. Error z value   Pr(>|z|)
(Intercept)    5.33024  0.0143830  370.59  0.000e+00
concentration -0.02845  0.0009355  -30.41 4.329e-203
```

```
mf <- glm(young ~ factor(concentration), family = poisson, data = nabr)
summary(mf)$coefficients
```

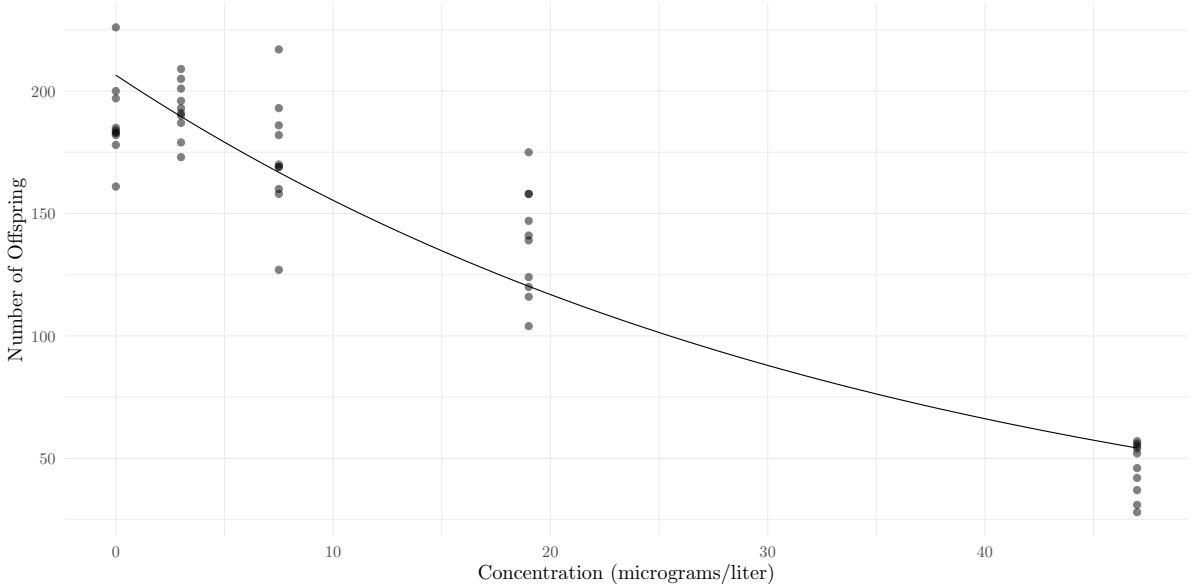
```
                        Estimate Std. Error  z value   Pr(>|z|)
(Intercept)              5.23591    0.02307 226.9633  0.000e+00
factor(concentration)3   0.02367    0.03243   0.7297  4.656e-01
factor(concentration)7.5 -0.08204   0.03332  -2.4626  1.380e-02
factor(concentration)19  -0.30721   0.03544  -8.6691  4.355e-18
factor(concentration)47  -1.41163   0.05211 -27.0886 1.342e-161
```

2. Create a plot showing the estimated expected number of offspring as a function of concentration for the model you estimated in the previous problem where concentration was treated as a *quantitative* explanatory variable.

   Solution: Here is a plot of the model that treats concentration as a quantitative variable.

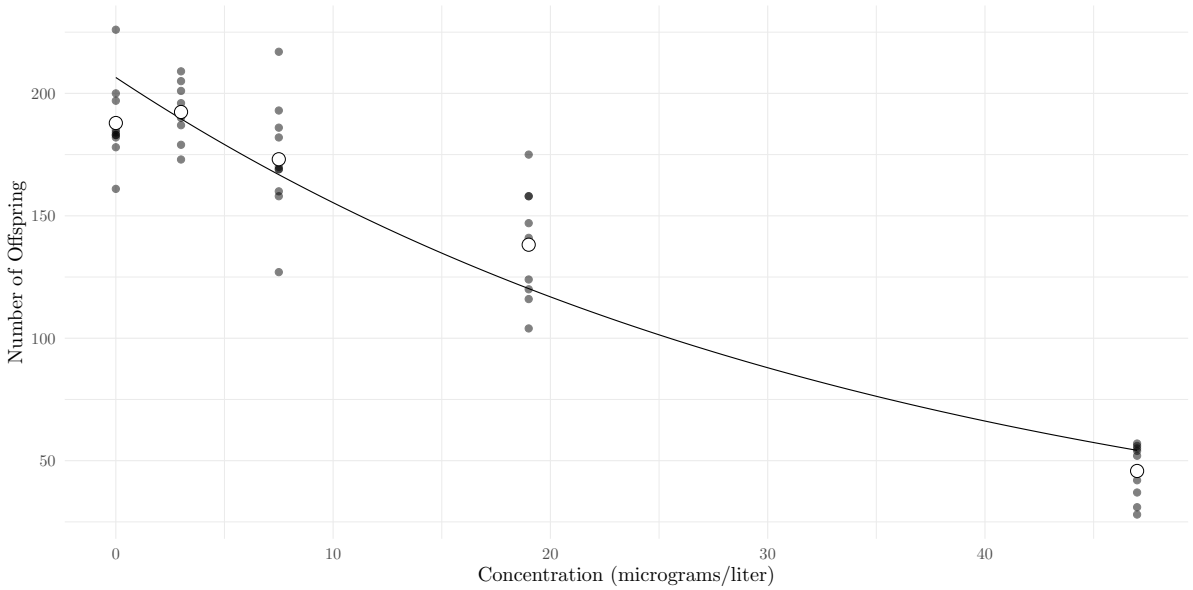
```
d <- data.frame(concentration = seq(0, 47, length = 100))
d$yhat <- predict(mq, newdata = d, type = "response")
```

```
p <- p + geom_line(aes(y = yhat), data = d)
plot(p)
```



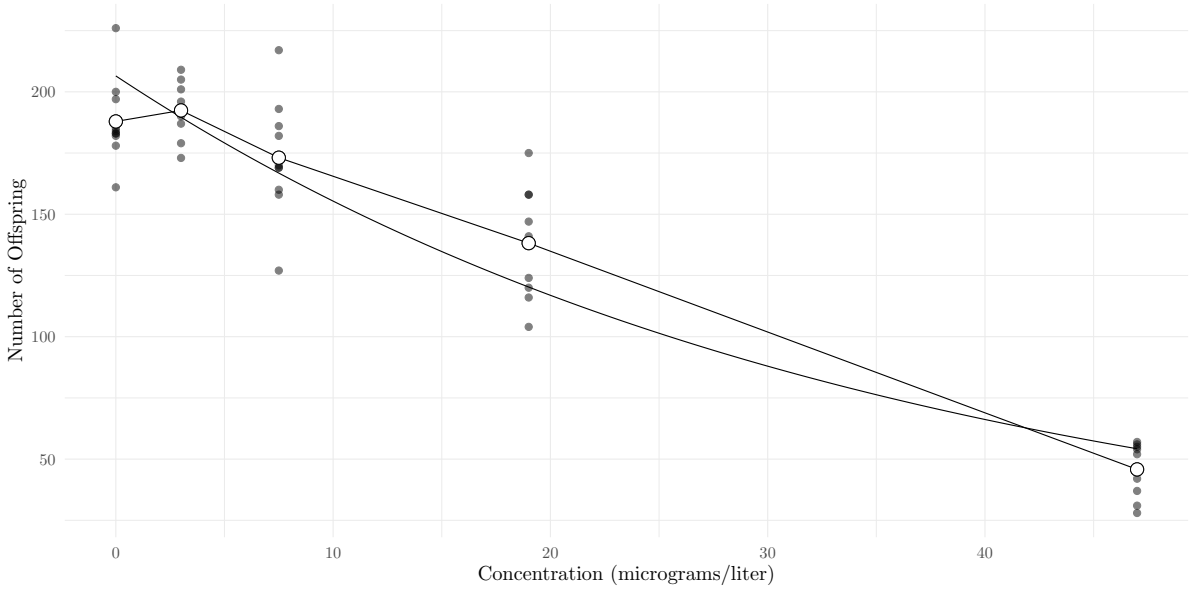
If you wanted to plot the model where concentration is treated as a factor, you could do this several ways. One is to just add points denoting the estimated expected response for each concentration.

```
d <- data.frame(concentration = unique(nabr$concentration))
d$yhat <- predict(mf, newdata = d, type = "response")
p <- p + geom_point(aes(y = yhat), data = d,
  shape = 21, fill = "white", size = 3)
plot(p)
```



Note that I made these points larger and filled with white to make them stand out a bit. We could joint them by line segments to show a trend.

```
p <- p + geom_line(aes(y = yhat), data = d) +
  geom_point(aes(y = yhat), data = d,
  shape = 21, fill = "white", size = 3)
plot(p)
```



Note that I did not necessarily need to "add" the points again, but I did so anyway because if I did not then the line would be on top of the points rather than underneath.

3. Using the `contrast` function, estimate four rate ratios for comparing the expected number of offspring at concentrations of 3, 7.5, 19, and 47 micrograms/liter to the expected number of offspring at a concentration of zero micrograms/liter. Do this for *both* of the models you estimated earlier. Write a sentence to interpret each rate ratio in terms of what it shows about the effect of a given concentration relative to a zero concentration on the expected number of offspring.

   Solution: First I will estimate the rate ratios for the first model.

```
trtools::contrast(mq, tf = exp,
  a = list(concentration = c(3,7.5,19,47)),
  b = list(concentration = 0),
  cnames = c("3 vs 0","7.5 vs 0","19 vs 0","47 vs 0"))
```

```
         estimate  lower  upper
3 vs 0     0.9182 0.9132 0.9233
7.5 vs 0   0.8079 0.7968 0.8191
19 vs 0    0.5825 0.5625 0.6031
47 vs 0    0.2626 0.2409 0.2863
```

   This shows that at a concentration of 3 micrograms/liter the expected number of offspring is about 0.92 times what it is at a concentration of zero (i.e., about 8% lower). At concentrations of 7.5, 19, and 47 micrograms/liter the expected number of offspring is about 0.81, 0.58, and 0.26 times what it is at a concentration of zero, respectively (i.e., the expected counts are about 19%, 42%, and 74% lower, respectively). We can estimate and interpret the rate ratios for the second model similarly.

```
trtools::contrast(mf, tf = exp,
  a = list(concentration = c(3,7.5,19,47)),
  b = list(concentration = 0),
```

```
  cnames = c("3 vs 0","7.5 vs 0","19 vs 0","47 vs 0"))
```

```
         estimate  lower  upper
3 vs 0     1.0239 0.9609 1.0912
7.5 vs 0   0.9212 0.8630 0.9834
19 vs 0    0.7355 0.6861 0.7884
47 vs 0    0.2437 0.2201 0.2700
```

This shows that for this model at a concentration of 3 micrograms/liter the expected number of offspring is about 1.02 times what it is at a concentration of zero (i.e., about 2% higher), while at concentrations of 7.5, 19, and 47 micrograms/liter the expected number of offspring is about 0.92, 0.74, and 0.24 times what it is at a concentration of zero, respectively (i.e., the expected counts are about 8%, 26%, and 76% lower, respectively). The **emmeans** package is not as useful for estimating rate or odds ratios for quantitative explanatory variables outside what can be done using the `emtrends` function, which is limited, but we can estimate the rate ratios above for the model where concentration is a categorical variable as follows.

```
library(emmeans)
emmeans::contrast(emmeans(mf, ~ concentration), method = "trt.vs.ctrl",
  ref = 1, type = "response", adjust = "none", infer = TRUE)
```

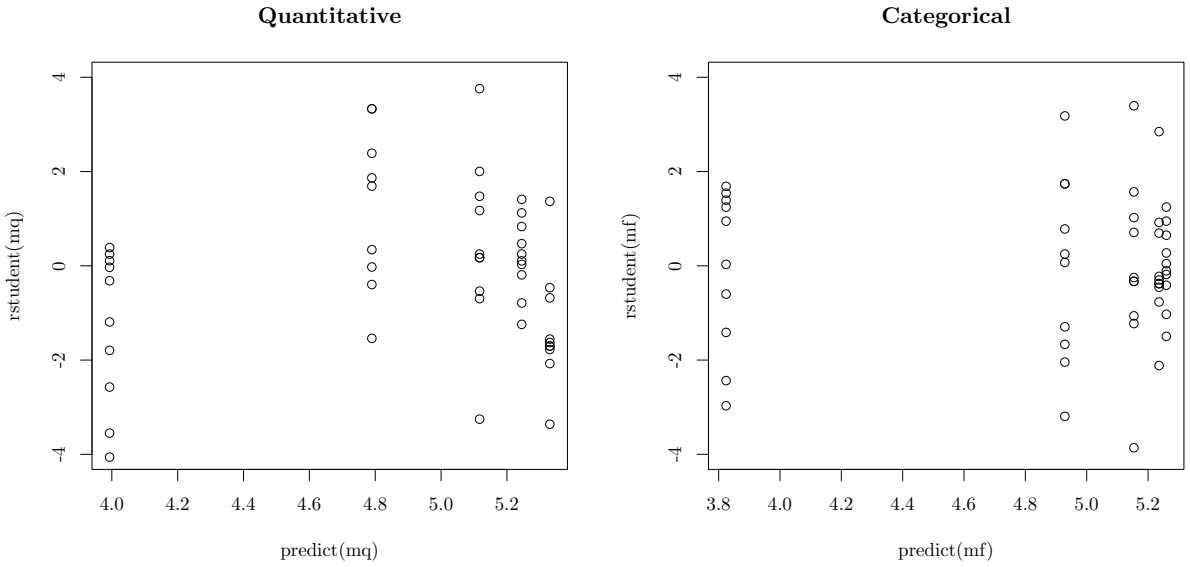
```
 contrast ratio     SE  df asymp.LCL asymp.UCL null z.ratio p.value
 3 / 0    1.024 0.0332 Inf     0.961     1.091    1   0.730  0.4656
 7.5 / 0  0.921 0.0307 Inf     0.863     0.983    1  -2.463  0.0138
 19 / 0   0.736 0.0261 Inf     0.686     0.788    1  -8.669  <.0001
 47 / 0   0.244 0.0127 Inf     0.220     0.270    1 -27.089  <.0001

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

Note that the apparent increase in the expected number offspring between concentrations of zero and 3 micrograms/liter is not statistically significant.

4. For each of the two models you estimated earlier, create a residual plot of standardized or studentized residuals against the predicted values. Based on these residual plots, which model do you think better fits the data. Explain your reasoning.

   Solution: Here are my residual plots.

```
par(mfcol = c(1,2))
plot(predict(mq), rstudent(mq), ylim = c(-4,4), main = "Quantitative")
plot(predict(mf), rstudent(mf), ylim = c(-4,4), main = "Categorical")
```



Based on these residual plots, I would say that the model that treats concentration as a *categorical* variable better fits the data. Note that in the other model the residuals are not uniformly distributed around zero, particularly for lower (log) predicted values (corresponding to higher concentrations).

5. Consider the model you estimated earlier where concentration was treated as a quantitative variable. That model can be written as $\log E(Y_i) = \beta_0 + \beta_1 x_i$, where Y_i and x_i are the i-th observations of the number of offspring and concentration, respectively. We sometimes call this a log-linear model since the log of the expected response is a linear function. Now consider two other models: a *linear* Poisson regression model which can be written as $E(Y_i) = \beta_0 + \beta_1 x_i$, and a *quadratic* polynomial log-linear model which can be written as $\log E(Y_i) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$.[2] The linear model can be specified by using an "identity" rather than a log link function (i.e., use `link = identity` rather than `link = log` in the `glm` function). The polynomial log-linear model can be specified just like other Poisson regression models but using either the `I` inhibit function or the `poly` function to specify the polynomial (see the discussion of polynomial regression). Estimate each of these models using the `glm` function, reporting the parameter estimates and their standard errors using `summary`. Also plot the standardized or studentized residuals against the predicted values for each model. Based on these residual plots, how do the the four Poisson regression models that you have now estimated compare in terms of their fit to the data? Explain your reasoning. **Note**: This problem is *extra credit* for students in Stat 436, but is *required* for students in Stat 516.

Solution: Here I will estimate the linear Poisson regression and the quadratic log-linear model.

```
ml <- glm(young ~ concentration,
  family = poisson(link = identity), data = nabr)
summary(ml)$coefficients
```

```
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   196.329    2.48908   78.88        0
concentration  -3.193    0.07429  -42.98        0
```

```
mp <- glm(young ~ poly(concentration, 2),
  family = poisson, data = nabr)
```

[2]Typically when using Poisson regression a log link function is implied. But other link functions can be used, and it would still be appropriate to call such a model a Poisson regression model since it still assumes that the response variable has a Poisson distribution. Even though we are using a linear model, we are still assuming a Poisson distribution for the response variable, and the pattern of heteroscedasticity is still that of a Poisson distribution. But when we specify a logistic regression model the link function is a logit (i.e., log-odds) link function, and the assumed distribution of the observed count is assumed to be binomial. If we change the link function but still assume a binomial distribution (which we will consider in a future lecture) the model is no longer a logistic regression model. Instead we might call it a *binomial* (or *binary*) regression model.

```
summary(mp)$coefficients
```

```
                        Estimate Std. Error z value   Pr(>|z|)
(Intercept)               4.8805    0.01347   362.3  0.000e+00
poly(concentration, 2)1  -3.7659    0.12899   -29.2 2.212e-187
poly(concentration, 2)2  -0.6767    0.09022    -7.5  6.367e-14
```

Note that using the `poly` function will apply a linear transformation to the explanatory variable to obtain what are called an orthogonal polynomial unless you include the `raw = TRUE` option.[3] Whether or not you use orthogonal polynomials will affect the parameterization and change the parameter estimates, but it will not change the model in the sense that the estimated expected response for a given concentration will be the same. Here are a couple of ways to estimate the model without using orthogonal polynomials so that the explanatory variables are simply concentration and squared concentration.

```
mp <- glm(young ~ concentration + I(concentration^2),
  family = poisson, data = nabr)
summary(mp)$coefficients
```

```
                     Estimate Std. Error z value  Pr(>|z|)
(Intercept)         5.2551669  1.782e-02 294.970 0.000e+00
concentration      -0.0080501  2.852e-03  -2.823 4.758e-03
I(concentration^2) -0.0004771  6.361e-05  -7.500 6.367e-14
```

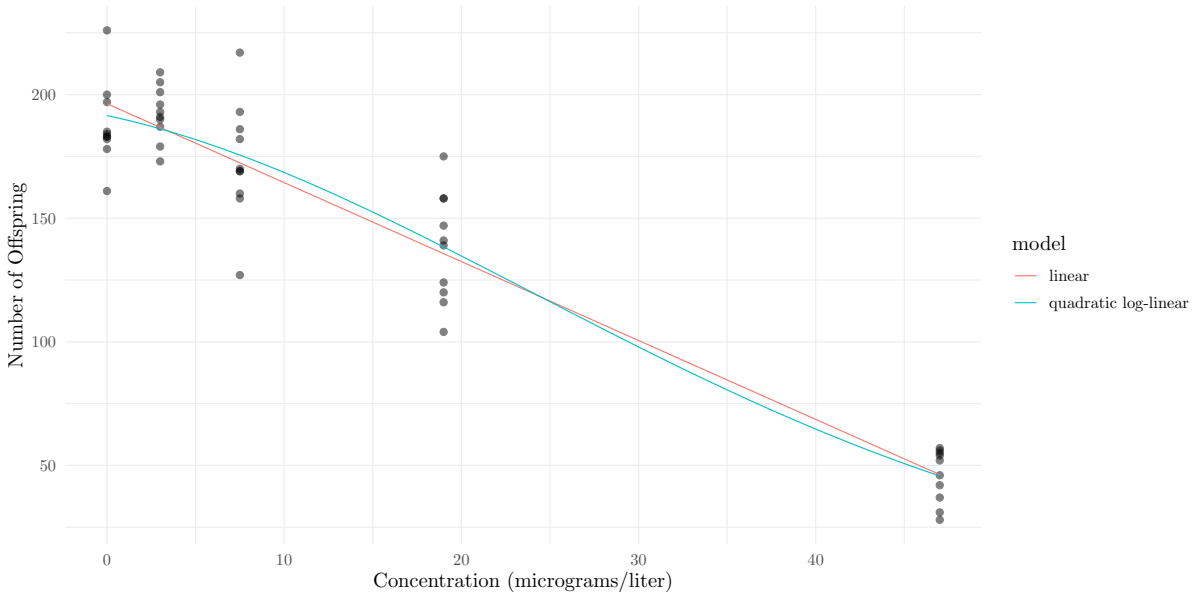
```
mp <- glm(young ~ poly(concentration, 2, raw = TRUE),
  family = poisson, data = nabr)
summary(mp)$coefficients
```

```
                                    Estimate Std. Error z value  Pr(>|z|)
(Intercept)                        5.2551669  1.782e-02 294.970 0.000e+00
poly(concentration, 2, raw = TRUE)1 -0.0080501  2.852e-03  -2.823 4.758e-03
poly(concentration, 2, raw = TRUE)2 -0.0004771  6.361e-05  -7.500 6.367e-14
```

I did not ask you to plot the models, although I should have done so. Here is what those models look like when plotted.

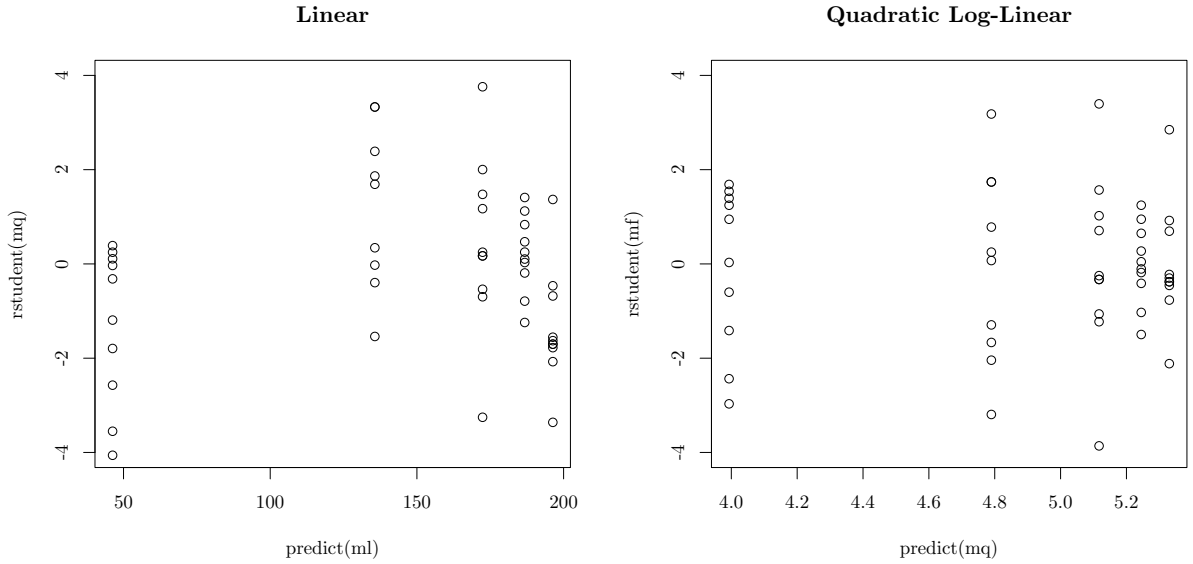
```
dl <- data.frame(concentration = seq(0, 47, length = 100),
  model = "linear")
dl$yhat <- predict(ml, newdata = dl, type = "response")
dp <- data.frame(concentration = seq(0, 47, length = 100),
  model = "quadratic log-linear")
dp$yhat <- predict(mp, newdata = dp, type = "response")

d <- rbind(dl, dp) # merge data frames by row

p <- ggplot(nabr, aes(x = concentration, y = young)) +
  theme_minimal() + geom_point(alpha = 0.5) +
  labs(x = "Concentration (micrograms/liter)", y = "Number of Offspring") +
  geom_line(aes(y = yhat, color = model), data = d)
plot(p)
```

[3] Using orthogonal polynomials is polynomial regression is sometimes useful to avoid numerical problems. But in many cases for well designed software they are not necessary.



These two models are fairly similar with respect to the relationship between the expected number of offspring and concentration. Here are the residual plots.

```
par(mfcol = c(1,2))
plot(predict(ml), rstudent(mq), ylim = c(-4,4), main = "Linear")
plot(predict(mq), rstudent(mf), ylim = c(-4,4), main = "Quadratic Log-Linear")
```



I would say that both of these models look better than the original log-linear model where concentration was treated as a quantitative variable. The linear model looks like there may be a bit of non-linearity that it is not quite capturing as can be seen by the residuals for the lower predicted values. I would say that either the model where concentration was treated as a categorical variable or the quadratic log-linear model would be the best models of the four.

Swedish Speed Limit Study

The data frame `nabr` in the **SMPracticals** package contains data from a observational study of the effects of speed limits on the number of traffic accidents.[4] This study was carried out in Sweden during the summers of 1961 and 1962 during comparable days (e.g., if an observation was made during the first Monday of July in 1961, an observation was also made during the first Monday of July in 1962). Periods of no speed limits were alternated with a posted speed limit of 90 km/h or 100 km/h. The number of traffic accidents with personal injuries that occurred and were reported each day during the study was recorded. The data are in "wide form" with each row showing observations from both 1961 and 1962.

```
library(SMPracticals)
head(limits)
```

```
  day lim1 lim2 y1 y2
1   1    0    0  9  9
2   2    0    0 11 20
3   3    0    0  9 15
4   4    0    0 20 14
5   5    0    0 31 30
6   6    0    0 26 23
```

Here `lim` and `lim2` are indicator variables for if a speed limit was posted on a given day in 1961 and 1962, respectively, and `y1` and `y2` are the number of traffic accidents on a given day in 1961 and 1962, respectively. For plotting and modeling it is useful to put the data into "long form" where each row is an observation from a given day in a given year.[5]

```
library(dplyr)
library(tidyr)
limitstudy <- limits %>%
  rename(limit_1961 = lim1, limit_1962 = lim2, y_1961 = y1, y_1962 = y2) %>%
  pivot_longer(cols = -day, names_to = c(".value", "year"), names_sep = "_") %>%
  mutate(limit = factor(limit, levels = c(0,1), labels = c("no","yes")))
head(limitstudy)
```

```
# A tibble: 6 x 4
  day   year  limit     y
  <fct> <chr> <fct> <int>
1 1     1961  no        9
2 1     1962  no        9
3 2     1961  no       11
4 2     1962  no       20
5 3     1961  no        9
6 3     1962  no       15
```

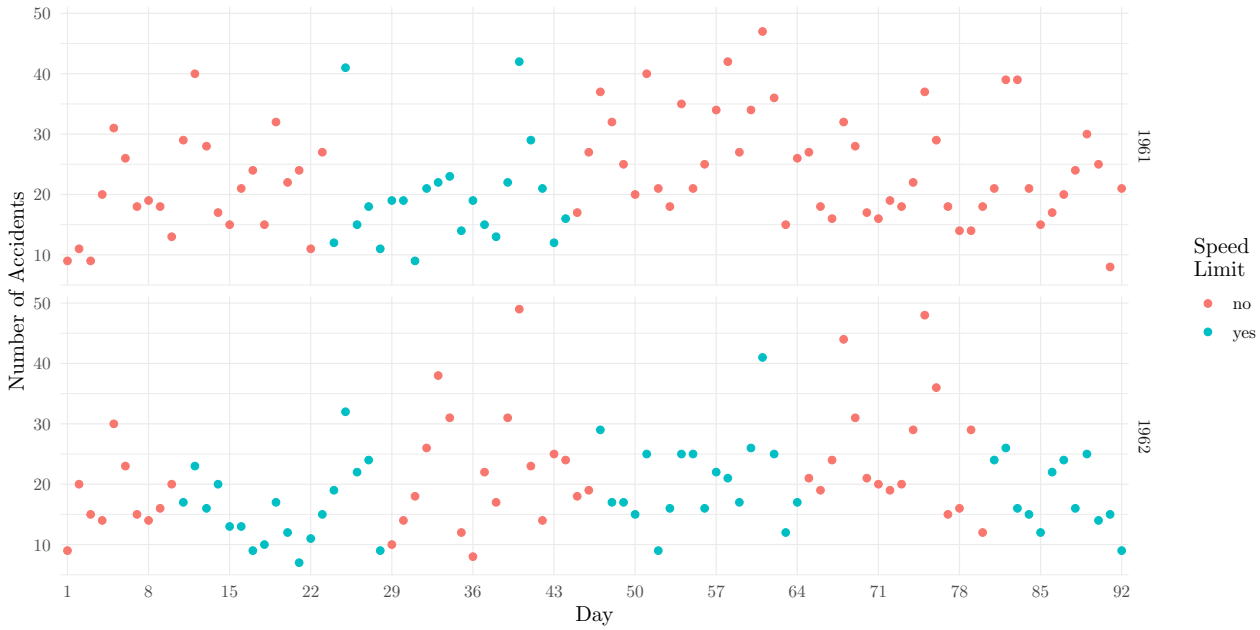
Compare the original data in `limits` to the new data frame `limitstudy` and you can see how the data have been "reshaped" by the code above. Also note that I formatted the `limit` into a factor with more clear level labels. Here is a plot of the data showing the number of accidents each day by year and whether or not a speed limit was posted.[6]

[4] Svensson, A. (1981). On a goodness-of-fit test for multiplicative Poisson models. *Annals of Statistics*, *9*, 697–704.

[5] Using `pivot_longer` effectively takes several columns that represent observations of the same variable and arranges them so that each observation is in a single row. This is relatively simple for a single variable, but gets a bit more complicated when there are two or more variables like there are here (i.e., the speed limit indicator and the number of accidents). I will admit the syntax is a bit cryptic. I actually copied this from an example that you can see if you use the command `vignette("pivot")`. The trick is to create variable names that can be parsed effectively by the `pivot_longer` function.

[6] The "\n" in the label for color is an escape character which adds a new line so that the label "Speed Limit" spans two lines of text instead of one.

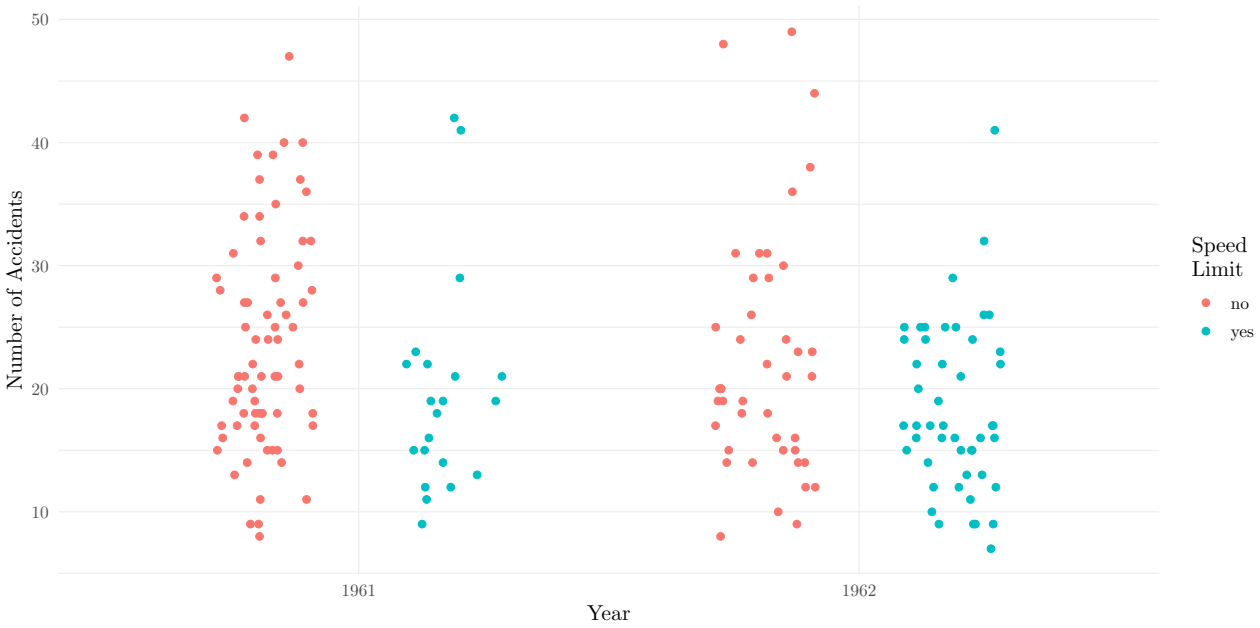
```
library(ggplot2)
p <- ggplot(limitstudy, aes(x = day, y = y, color = limit)) +
  theme_minimal() + geom_point() + facet_grid(year ~ .) +
  scale_x_discrete(breaks = seq(1, 92, by = 7)) +
  labs(x = "Day", y = "Number of Accidents", color = "Speed\nLimit")
plot(p)
```



It might be important to account for the effect of day since the risk of accidents may vary over time, and the figure above shows that the speed limits were not randomly or uniformly distributed over days. But for this problem you will ignore any effect of day and just focus on how the expected number of accidents varies by year and by whether or not a speed limit was posted.[7] The table below shows the data without accounting for day.

```
set.seed(111)
p <- ggplot(limitstudy, aes(x = year, y = y, color = limit)) +
  theme_minimal() + geom_point(position = position_jitterdodge()) +
  labs(x = "Year", y = "Number of Accidents", color = "Speed\nLimit")
plot(p)
```

[7] We may return to these data and see how we might account for the effect of day using a fixed or random effects model.



The sample statistics show that the number of accidents was, on average, lower when speed limits were posted.

```
limitstudy %>% group_by(year, limit) %>% summarize(mean = mean(y))
```

```
# A tibble: 4 x 3
# Groups:   year [2]
  year  limit  mean
  <chr> <fct> <dbl>
1 1961  no     23.7
2 1961  yes    19.7
3 1962  no     22.2
4 1962  yes    18.4
```

In this problem you will consider using regression models for inferences concerning the relationship between the posting of a speed limit and the expected number of accidents. The main focus here is on how posting a speed limit is related to the expected number of accidents, but controlling for year by including it as an explanatory variable is important since the accident rate may have varied by year and year is partially confounded with speed limit since speed limits were posted more often in 1962 than in 1961. Be sure you use the data frame `limitstudy` created above for your model.

1. Estimate a Poisson regression model with the number of accidents as the response variable and the speed limit (yes or no) and year (1961 or 1962) as explanatory variables. Do not include an interaction in your model. Year here should be treated as a categorical variable (i.e., a factor) but it is not necessary to convert it to a factor since it is stored in the data frame as a character variable and not a number, so R will automatically interpret it as a factor when it is used as an explanatory variable.[8]

   Solution: Here is how to estimate the model.

   ```
   m <- glm(y ~ limit + year, family = poisson, data = limitstudy)
   summary(m)$coefficients
   ```

   ```
               Estimate Std. Error z value  Pr(>|z|)
   (Intercept)  3.16513    0.02294 137.977 0.0000000
   limityes    -0.18893    0.03547  -5.327 0.0000001
   ```

[8] You can see that `year` is a character variable when you look at the data frame. At the top is it labeled as type `<chr>` which identifies it as a character variable. You can also see this if you use `str(limitstudy)`

```
year1962    -0.06394    0.03335  -1.917 0.0551916
```

2. Using either `contrast` or functions from the **emmeans** package, produce estimates and confidence intervals for the expected number of accidents with and without a posted speed limit in 1961, and again in 1962.

   Solution: Here is how to estimate the expected number of accidents using `contrast`.

```
trtools::contrast(m, tf = exp,
  a = list(limit = c("no","yes","no","yes"),
    year = c("1961","1961","1962","1962")),
  cnames = c("no,1961","yes,1961","no,1962","yes,1962"))
```

```
         estimate lower upper
no,1961     23.69 22.65 24.78
yes,1961    19.61 18.28 21.04
no,1962     22.22 21.01 23.51
yes,1962    18.40 17.36 19.50
```

   Here is how you can do this using the **emmeans** package.

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

```
 limit year rate    SE  df asymp.LCL asymp.UCL
 no    1961 23.7 0.543 Inf      22.6      24.8
 yes   1961 19.6 0.704 Inf      18.3      21.0
 no    1962 22.2 0.637 Inf      21.0      23.5
 yes   1962 18.4 0.547 Inf      17.4      19.5

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

   Here is another way to do it. This approach will be useful for the next problem where rate ratios are computed.

```
emmeans(m, ~limit|year, type = "response")
```

```
year = 1961:
 limit rate    SE  df asymp.LCL asymp.UCL
 no    23.7 0.543 Inf      22.6      24.8
 yes   19.6 0.704 Inf      18.3      21.0

year = 1962:
 limit rate    SE  df asymp.LCL asymp.UCL
 no    22.2 0.637 Inf      21.0      23.5
 yes   18.4 0.547 Inf      17.4      19.5

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

3. Using either `contrast` or functions from the **emmeans** package, estimate the rate ratio for the expected number of accidents when a speed limit was posted versus when it was not. Note that while you can estimate a separate rate ratio for 1961 and another for 1962, they should be equal since the model does not include an interaction. Report the rate ratio and its confidence interval, and write a sentence that interprets the value of the rate ratio in terms of the the expected number of accidents when a speed limit was posted versus when it was not.

   Solution: Here is how to estimate the rate ratios using `contrast`.

```
trtools::contrast(m,
  a = list(limit = "yes", year = c("1961","1962")),
  b = list(limit = "no", year = c("1961","1962")),
  cnames = c("1961","1962"), tf = exp)
```

```
     estimate  lower  upper
1961   0.8278 0.7722 0.8874
1962   0.8278 0.7722 0.8874
```

Here is how these rate ratios can be obtained using the **emmeans** package.

```
pairs(emmeans(m, ~limit|year, type = "response"), infer = TRUE, reverse = TRUE)
```

```
year = 1961:
 contrast ratio     SE  df asymp.LCL asymp.UCL null z.ratio p.value
 yes / no 0.828 0.0294 Inf     0.772     0.887    1  -5.327  <.0001

year = 1962:
 contrast ratio     SE  df asymp.LCL asymp.UCL null z.ratio p.value
 yes / no 0.828 0.0294 Inf     0.772     0.887    1  -5.327  <.0001

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

Note that by default `pairs` will compute the rate ratio for yes over no rather than no over yes, so I used `reverse = TRUE` to reverse the rate ratio to make it consistent with what I obtained using `contrast`. The estimated ratio ratio is about 0.83 with a confidence interval of about (0.77,0.89). We would interpret this as showing that when speed limits were posted the expected number of accidents decreased by a factor of about 0.83 (i.e., the expected number of accidents was about 17% lower when a speed limit was posted). Alternatively, we could compute the following rate ratios.

```
trtools::contrast(m,
  a = list(limit = "no", year = c("1961","1962")),
  b = list(limit = "yes", year = c("1961","1962")),
  cnames = c("1961","1962"), tf = exp)
```

```
     estimate lower upper
1961    1.208 1.127 1.295
1962    1.208 1.127 1.295
```

```
pairs(emmeans(m, ~limit|year, type = "response"), infer = TRUE)
```

```
year = 1961:
 contrast ratio     SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.21 0.0428 Inf      1.13      1.29    1   5.327  <.0001

year = 1962:
 contrast ratio     SE  df asymp.LCL asymp.UCL null z.ratio p.value
 no / yes  1.21 0.0428 Inf      1.13      1.29    1   5.327  <.0001

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

The estimated rate ratio is about 1.21 with a confidence interval of about (1.13, 1.29). This shows that when no speed limit was posted the expected number of accidents is larger by a factor of about 1.21

(i.e., about 21% higher).

Modeling Death Rates from Cervical Cancer

The data frame `cervical` in the **GLMsData** package is an observational study of the death rates due to cervical cancer by age group in four countries/regions in Europe.[9] The data are shown below. Note that using `data(cervical)` is necessary here to make the data frame accessible.[10]

```
library(GLMsData)
data(cervical)
cervical
```

```
    Country    Age Deaths Wyears
1  EngWales 25to34    192 153999
2  EngWales 35to44    860  14268
3  EngWales 45to54   2762  15450
4  EngWales 55to64   3035  15142
5   Belgium 25to34      8   2328
6   Belgium 35to44     81   2557
7   Belgium 45to54    242   2268
8   Belgium 55to64    268   2253
9    France 25to34     96  15324
10   France 35to44    477  16186
11   France 45to54    998  14432
12   France 55to64   1117  13201
13    Italy 25to34     45  19115
14    Italy 35to44    255  18811
15    Italy 45to54    621  16234
16    Italy 55to64    839  15246
```

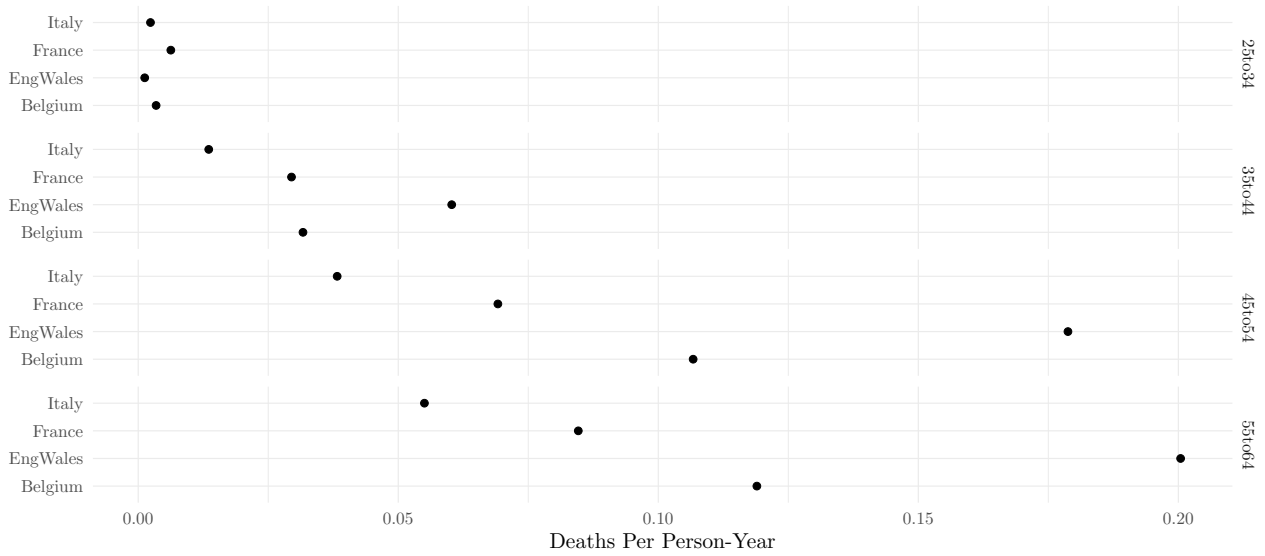
The observed death rate can be computed by dividing the number of deaths (`Deaths`) by the number of woman-years (`Wyears`). The latter, which is usually called "person-years" in gender-neutral applications, is a unit of measurement that takes into account both the number of people and the amount of time they are being observed.[11] For example, if we had ten people each observed for five years then that would be 50 person-years. Or if we had one person observed for five years and another observed for two that would be a total of seven person-years. The rate can then be defined in terms of number of deaths due to cervical cancer per person-year, or per person per year. The plot below shows the observed rate of deaths to cervical cancer by country and age group.

```
library(ggplot2)
p <- ggplot(cervical, aes(x = Deaths / Wyears, y = Country)) +
   theme_minimal() + geom_point() + facet_grid(Age ~ .) +
   labs(y = NULL, x = "Deaths Per Person-Year")
plot(p)
```

[9] Whittermore, A. S. & Gong, G. (1991). Poisson regression with misclassified counts: Applications to cervical cancer mortality rates. *Applied Statistics*, *40(1)*, 81–93.

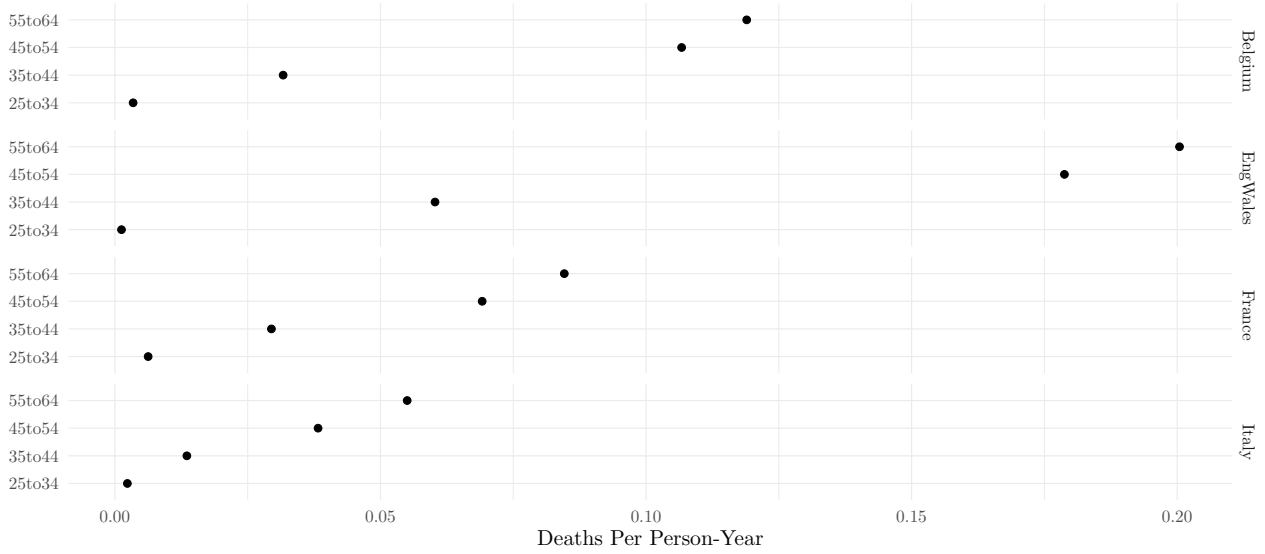
[10] The `data` function was originally intended to allow users to not have to load large data frames into memory until needed. But that is no longer necessary because packages can now use what is called lazy loading meaning that a data frame will only be loaded into memory if and when it is used. Package developers are encouraged to set their packages to use lazy loading so use of `data` is not necessary, but not everyone does this.

[11] We can use the more common gender-neutral term "person-year" here with the understanding that the rate is among people that have a cervix and thus are susceptible to cervical cancer.



Alternatively we could plot the observed rates by grouping them by country.

```
p <- ggplot(cervical, aes(x = Deaths / Wyears, y = Age)) +
   theme_minimal() + geom_point() + facet_grid(Country ~ .) +
   labs(y = NULL, x = "Deaths Per Person-Year")
plot(p)
```



The goal here is to model the rate of deaths due to cervical cancer to compare the rates between countries/regions and also between age groups.

1. Estimate a Poisson regression model for the rate of deaths due to cervical cancer that produces the following output when using `summary`.

```
                Estimate Std. Error z value  Pr(>|z|)
(Intercept)      -6.7040    0.06859 -97.744 0.000e+00
CountryEngWales   0.5105    0.04267  11.964 5.481e-33
CountryFrance    -0.3119    0.04518  -6.904 5.071e-12
CountryItaly     -0.8704    0.04730 -18.400 1.318e-75
Age35to44         3.3888    0.05990  56.574 0.000e+00
```

```
Age45to54          4.4201     0.05651  78.223 0.000e+00
Age55to64          4.5905     0.05625  81.612 0.000e+00
```

Note that since the number of person-years varies by country/region and age group you will need to use an offset variable. Report the parameter estimates and their standard errors using `summary`.

Solution: Looking at the output of `summary` above we can see that `Country` and `Age` are specified as explanatory variables since there are indicator variables for the levels of these variables. There is no interaction specified. Here is how to estimate that model.

```
m <- glm(Deaths ~ offset(log(Wyears)) + Country + Age,
  family = poisson, data = cervical)
summary(m)$coefficients
```

```
                Estimate Std. Error z value  Pr(>|z|)
(Intercept)      -6.7040    0.06859 -97.744 0.000e+00
CountryEngWales   0.5105    0.04267  11.964 5.481e-33
CountryFrance    -0.3119    0.04518  -6.904 5.071e-12
CountryItaly     -0.8704    0.04730 -18.400 1.318e-75
Age35to44         3.3888    0.05990  56.574 0.000e+00
Age45to54         4.4201    0.05651  78.223 0.000e+00
Age55to64         4.5905    0.05625  81.612 0.000e+00
```

2. Using either `contrast` or functions from the **emmeans** package, estimate three ratios that compare the expected death rate for the three older age groups with the youngest age group. Summarize each rate ratio in a sentence that describes clearly how the age groups compare with respect to the death rate.

   Solution: We can estimate the rate ratios using `contrast` as follows.

```
trtools::contrast(m,
  a = list(Age = c("35to44","45to54","55to64"), Country = "EngWales", Wyears = 1),
  b = list(Age = "25to34", Country = "EngWales", Wyears = 1),
  tf = exp, cnames = c("35-44 vs 25-34","45-54 vs 25-34","55-64 vs 25-34"))
```

```
               estimate lower  upper
35-44 vs 25-34    29.63 26.35  33.32
45-54 vs 25-34    83.11 74.39  92.84
55-64 vs 25-34    98.54 88.26 110.03
```

Note that which country we specify does not affect the rate ratios since there is no interaction in the model. Here is how we could do this using the **emmeans** package.

```
library(emmeans)
emmeans::contrast(emmeans(m, ~Age|Country), method = "trt.vs.ctrl", ref = 1,
  adjust = "none", infer = TRUE, type = "response")
```

```
Country = Belgium:
 contrast        ratio   SE  df asymp.LCL asymp.UCL null z.ratio p.value
 35to44 / 25to34  29.6 1.77 Inf      26.4      33.3    1  56.570  <.0001
 45to54 / 25to34  83.1 4.70 Inf      74.4      92.8    1  78.220  <.0001
 55to64 / 25to34  98.5 5.54 Inf      88.3     110.0    1  81.610  <.0001

Country = EngWales:
 contrast        ratio   SE  df asymp.LCL asymp.UCL null z.ratio p.value
 35to44 / 25to34  29.6 1.77 Inf      26.4      33.3    1  56.570  <.0001
 45to54 / 25to34  83.1 4.70 Inf      74.4      92.8    1  78.220  <.0001
 55to64 / 25to34  98.5 5.54 Inf      88.3     110.0    1  81.610  <.0001

Country = France:
```

```
 contrast        ratio   SE  df asymp.LCL asymp.UCL null z.ratio p.value
 35to44 / 25to34  29.6 1.77 Inf      26.4      33.3    1  56.570  <.0001
 45to54 / 25to34  83.1 4.70 Inf      74.4      92.8    1  78.220  <.0001
 55to64 / 25to34  98.5 5.54 Inf      88.3     110.0    1  81.610  <.0001

Country = Italy:
 contrast        ratio   SE  df asymp.LCL asymp.UCL null z.ratio p.value
 35to44 / 25to34  29.6 1.77 Inf      26.4      33.3    1  56.570  <.0001
 45to54 / 25to34  83.1 4.70 Inf      74.4      92.8    1  78.220  <.0001
 55to64 / 25to34  98.5 5.54 Inf      88.3     110.0    1  81.610  <.0001

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

So we estimate that the expected number of deaths per person per year for the 35 to 44 age group is about 30 times higher than the 25 to 34 age group. The 45 to 54 and 55 to 64 age groups have rates about 83 and 99 times higher, respectively, than the 25 to 34 age group. Note that we can also use the reciprocals of these rate ratios.

```
emmeans::contrast(emmeans(m, ~Age|Country), method = "trt.vs.ctrl", ref = 1,
  adjust = "none", infer = TRUE, type = "response", reverse = TRUE)
```

```
Country = Belgium:
 contrast         ratio       SE  df asymp.LCL asymp.UCL null z.ratio p.value
 25to34 / 35to44 0.0338 0.002022 Inf   0.03001    0.0379    1 -56.570  <.0001
 25to34 / 45to54 0.0120 0.000680 Inf   0.01077    0.0134    1 -78.220  <.0001
 25to34 / 55to64 0.0101 0.000571 Inf   0.00909    0.0113    1 -81.610  <.0001

Country = EngWales:
 contrast         ratio       SE  df asymp.LCL asymp.UCL null z.ratio p.value
 25to34 / 35to44 0.0338 0.002022 Inf   0.03001    0.0379    1 -56.570  <.0001
 25to34 / 45to54 0.0120 0.000680 Inf   0.01077    0.0134    1 -78.220  <.0001
 25to34 / 55to64 0.0101 0.000571 Inf   0.00909    0.0113    1 -81.610  <.0001

Country = France:
 contrast         ratio       SE  df asymp.LCL asymp.UCL null z.ratio p.value
 25to34 / 35to44 0.0338 0.002022 Inf   0.03001    0.0379    1 -56.570  <.0001
 25to34 / 45to54 0.0120 0.000680 Inf   0.01077    0.0134    1 -78.220  <.0001
 25to34 / 55to64 0.0101 0.000571 Inf   0.00909    0.0113    1 -81.610  <.0001

Country = Italy:
 contrast         ratio       SE  df asymp.LCL asymp.UCL null z.ratio p.value
 25to34 / 35to44 0.0338 0.002022 Inf   0.03001    0.0379    1 -56.570  <.0001
 25to34 / 45to54 0.0120 0.000680 Inf   0.01077    0.0134    1 -78.220  <.0001
 25to34 / 55to64 0.0101 0.000571 Inf   0.00909    0.0113    1 -81.610  <.0001

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

Thus, for example, the expected rate for the youngest age group of 25 to 34 is about 0.03 times that of the next youngest age group of 35 to 44 (i.e., about 97% less).

3. Using either `contrast` or functions from the **emmeans** package, estimate three ratios that compare the expected death rate in England and Wales with the three other countries. Summarize each rate ratio in

a sentence that describes clearly how the countries/regions compare with respect to the death rate.[12]

Solution: We can estimate the rate ratios using `contrast` as follows.

```
trtools::contrast(m,
  a = list(Country = "EngWales", Age = "25to34", Wyears = 1),
  b = list(Country = c("Belgium","France","Italy"), Age = "25to34", Wyears = 1),
  tf = exp, cnames = c("EngWales vs Belgium","EngWales vs France","EngWales vs Italy"))
```

```
                   estimate lower upper
EngWales vs Belgium    1.666 1.533 1.812
EngWales vs France     2.276 2.176 2.381
EngWales vs Italy      3.979 3.775 4.193
```

Note that which age group we specify does not affect the rate ratios since there is no interaction in the model. Here is how we could do this using the **emmeans** package.

```
emmeans::contrast(emmeans(m, ~Country|Age), method = "trt.vs.ctrl", ref = 2,
  adjust = "none", infer = TRUE, type = "response", reverse = TRUE)
```

```
Age = 25to34:
 contrast           ratio     SE  df asymp.LCL asymp.UCL null z.ratio p.value
 EngWales / Belgium  1.67 0.0711 Inf      1.53      1.81    1  11.960  <.0001
 EngWales / France   2.28 0.0521 Inf      2.18      2.38    1  35.940  <.0001
 EngWales / Italy    3.98 0.1067 Inf      3.77      4.19    1  51.470  <.0001

Age = 35to44:
 contrast           ratio     SE  df asymp.LCL asymp.UCL null z.ratio p.value
 EngWales / Belgium  1.67 0.0711 Inf      1.53      1.81    1  11.960  <.0001
 EngWales / France   2.28 0.0521 Inf      2.18      2.38    1  35.940  <.0001
 EngWales / Italy    3.98 0.1067 Inf      3.77      4.19    1  51.470  <.0001

Age = 45to54:
 contrast           ratio     SE  df asymp.LCL asymp.UCL null z.ratio p.value
 EngWales / Belgium  1.67 0.0711 Inf      1.53      1.81    1  11.960  <.0001
 EngWales / France   2.28 0.0521 Inf      2.18      2.38    1  35.940  <.0001
 EngWales / Italy    3.98 0.1067 Inf      3.77      4.19    1  51.470  <.0001

Age = 55to64:
 contrast           ratio     SE  df asymp.LCL asymp.UCL null z.ratio p.value
 EngWales / Belgium  1.67 0.0711 Inf      1.53      1.81    1  11.960  <.0001
 EngWales / France   2.28 0.0521 Inf      2.18      2.38    1  35.940  <.0001
 EngWales / Italy    3.98 0.1067 Inf      3.77      4.19    1  51.470  <.0001

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

So the expected number of deaths per person per year in England and Wales is about 1.67 times that in Belgium (i.e., about 67% higher), about 2.28 times that in France (i.e., about 128% higher), and about 3.98 times that in Italy (i.e., about 298% higher). We can also use the reciprocals of these rate ratios.

```
emmeans::contrast(emmeans(m, ~Country|Age), method = "trt.vs.ctrl", ref = 2,
  adjust = "none", infer = TRUE, type = "response")
```

[12]Not too much should be made of these rate ratios. The paper that reported these data pointed out that there was evidence that death rates due to cervical cancer were under-reported in some countries.

```
Age = 25to34:
 contrast           ratio      SE  df asymp.LCL asymp.UCL null z.ratio p.value
 Belgium / EngWales 0.600 0.02561 Inf     0.552     0.652    1 -11.960  <.0001
 France / EngWales  0.439 0.01005 Inf     0.420     0.460    1 -35.940  <.0001
 Italy / EngWales   0.251 0.00674 Inf     0.238     0.265    1 -51.470  <.0001

Age = 35to44:
 contrast           ratio      SE  df asymp.LCL asymp.UCL null z.ratio p.value
 Belgium / EngWales 0.600 0.02561 Inf     0.552     0.652    1 -11.960  <.0001
 France / EngWales  0.439 0.01005 Inf     0.420     0.460    1 -35.940  <.0001
 Italy / EngWales   0.251 0.00674 Inf     0.238     0.265    1 -51.470  <.0001

Age = 45to54:
 contrast           ratio      SE  df asymp.LCL asymp.UCL null z.ratio p.value
 Belgium / EngWales 0.600 0.02561 Inf     0.552     0.652    1 -11.960  <.0001
 France / EngWales  0.439 0.01005 Inf     0.420     0.460    1 -35.940  <.0001
 Italy / EngWales   0.251 0.00674 Inf     0.238     0.265    1 -51.470  <.0001

Age = 55to64:
 contrast           ratio      SE  df asymp.LCL asymp.UCL null z.ratio p.value
 Belgium / EngWales 0.600 0.02561 Inf     0.552     0.652    1 -11.960  <.0001
 France / EngWales  0.439 0.01005 Inf     0.420     0.460    1 -35.940  <.0001
 Italy / EngWales   0.251 0.00674 Inf     0.238     0.265    1 -51.470  <.0001

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

So the estimated expected death rate in England and Wales is about 0.6 times that in Belgium (about 40% lower higher), about 0.44 times that in France (about 56% lower), and about 0.25 times that in Italy (about 75% lower). Note also that to put these rate ratios into perspective it is useful to also consider the estimated rates for each country and age group. These can be obtained as follows using the **emmeans** package.

```
emmeans(m, ~Country*Age, type = "response", offset = 0)
```

```
 Country  Age       rate       SE  df asymp.LCL asymp.UCL
 Belgium  25to34 0.00123 0.000084 Inf   0.00107   0.00140
 EngWales 25to34 0.00204 0.000111 Inf   0.00184   0.00227
 France   25to34 0.00090 0.000052 Inf   0.00080   0.00101
 Italy    25to34 0.00051 0.000031 Inf   0.00046   0.00058
 Belgium  35to44 0.03632 0.001692 Inf   0.03316   0.03980
 EngWales 35to44 0.06052 0.001578 Inf   0.05751   0.06370
 France   35to44 0.02659 0.000782 Inf   0.02510   0.02817
 Italy    35to44 0.01521 0.000495 Inf   0.01427   0.01621
 Belgium  45to54 0.10189 0.004328 Inf   0.09375   0.11073
 EngWales 45to54 0.16976 0.002850 Inf   0.16427   0.17544
 France   45to54 0.07458 0.001670 Inf   0.07138   0.07793
 Italy    45to54 0.04267 0.001129 Inf   0.04051   0.04494
 Belgium  55to64 0.12081 0.005087 Inf   0.11124   0.13120
 EngWales 55to64 0.20130 0.003195 Inf   0.19513   0.20766
 France   55to64 0.08844 0.001944 Inf   0.08471   0.09233
 Italy    55to64 0.05059 0.001317 Inf   0.04808   0.05324

Confidence level used: 0.95
```

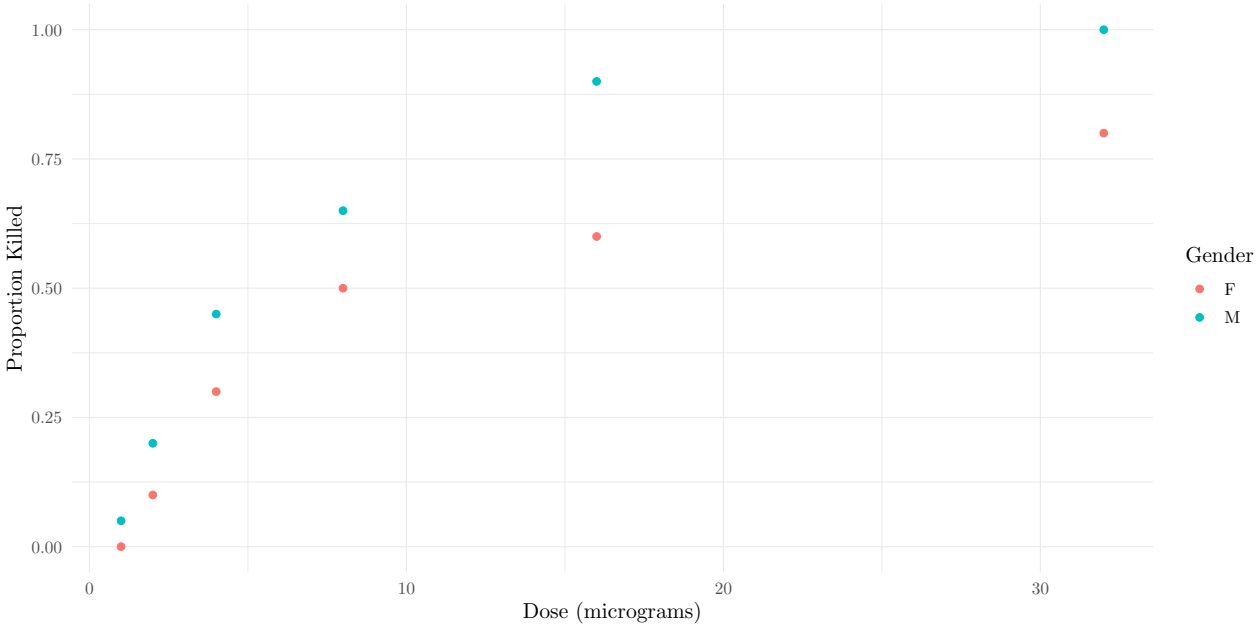
```
Intervals are back-transformed from the log scale
```

Note that when using `emmeans` you do need to specify the offset value. Here I have specified `offset = 0` which is the same as `Wyears = 1` when using `contrast` (note that log of one equals zero, and you could also just write `offset = log(1)`). So what is estimated is the estimated number of deaths per person per year. This is not necessary for rate ratios because the unit of the offset cancels-out in the rate ratio.

Toxicity of Trans-Cypermetrin for Tobacco Budworm

The data frame `budworm` in the **GLMsData** package is from a study of the toxicity of the pyrethroid trans-cypermethrin in tobacco budworm (*Heliothis virescens*) which are responsible for considerable damage to cotton crops in North and South America.[13]

```
library(ggplot2)
library(GLMsData)
data(budworm)

p <- ggplot(budworm, aes(x = Dose, y = Killed/Number, color = Gender)) +
  theme_minimal() + geom_point() +
  labs(x = "Dose (micrograms)", y = "Proportion Killed")
plot(p)
```



In this problem you will use logistic regression to investigate the effect of dose on mortality.

1. Estimate *two* logistic regression models: one using dose and gender as explanatory variables, and a second using the *logarithm* of dose and gender as explanatory variables. For the model using the logarithm of dose as an explanatory variable, include the transformation within the model formula as `log(dose)` rather than creating a new variable in the data frame. Both models should include an interaction between dose (or log of dose) and gender. Report the parameter estimates and their

[13] Holloway, J. W. (1989). A comparison of the toxicity of the pyrethroid trans-cypermethrin, with and without the synergist piperonyl butoxide, to adult moths from two strains of *Helios virescens*. Final year dissertation, Department of Pure and Applied Zoology, University of Reading, UK. Batches of twenty male or female moths were exposed to each of six doses of the pyrethroid, and the number that were killed after 72 hours of exposure was observed. The plot below shows the proportion of moths killed by gender and dose.

standard errors for both models using the `summary` function, and plot both models by adding curves to the plot above to show the estimated expected proportion of dead budworms as a function of dose and gender. You can either make one plot showing both models, or one plot for each model.

Solution: Here is how to estimate the two logistic regression models.

```
m.dose <- glm(cbind(Killed, Number - Killed) ~ Dose * Gender,
  family = binomial, data = budworm)
summary(m.dose)$coefficients
```

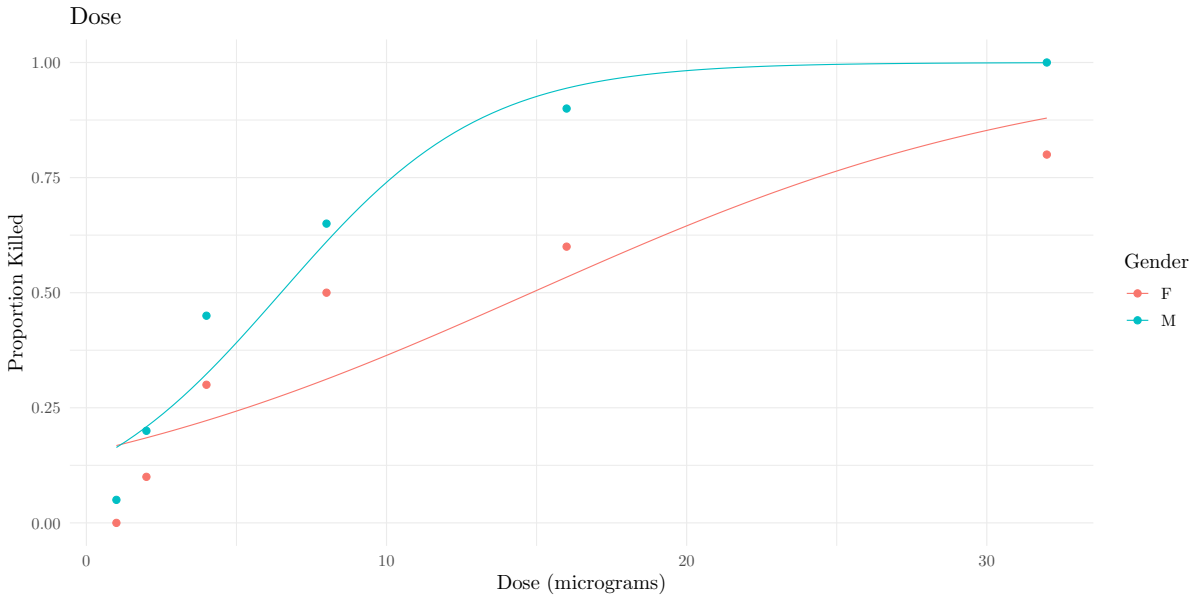
```
             Estimate Std. Error z value  Pr(>|z|)
(Intercept)   -1.7158    0.32233 -5.3230 1.020e-07
Dose           0.1157    0.02379  4.8627 1.158e-06
GenderM       -0.2119    0.51523 -0.4113 6.808e-01
Dose:GenderM   0.1816    0.06692  2.7132 6.663e-03
```

```
m.logd <- glm(cbind(Killed, Number - Killed) ~ log(Dose) * Gender,
  family = binomial, data = budworm)
summary(m.logd)$coefficients
```

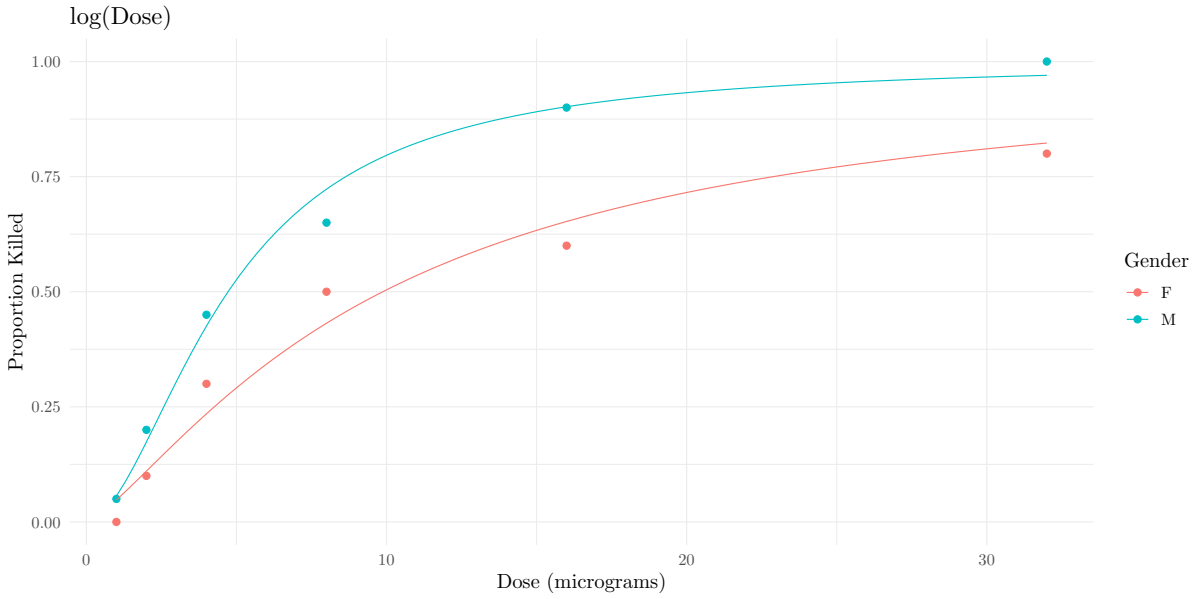
```
                  Estimate Std. Error z value  Pr(>|z|)
(Intercept)        -2.9935     0.5527 -5.4162 6.087e-08
log(Dose)           1.3071     0.2411  5.4221 5.891e-08
GenderM             0.1750     0.7783  0.2248 8.221e-01
log(Dose):GenderM   0.5091     0.3895  1.3071 1.912e-01
```

Here is how we can plot the two models.

```
d <- expand.grid(Dose = seq(1, 32, length = 100), Gender = c("F","M"))
d$yhat <- predict(m.dose, newdata = d, type = "response")
p <- ggplot(budworm, aes(x = Dose, y = Killed/Number, color = Gender)) +
  theme_minimal() + geom_point() +
  labs(x = "Dose (micrograms)", y = "Proportion Killed") +
  geom_line(aes(y = yhat), data = d) + ggtitle("Dose")
plot(p)
```



```
d$yhat <- predict(m.logd, newdata = d, type = "response")
p <- ggplot(budworm, aes(x = Dose, y = Killed/Number, color = Gender)) +
  theme_minimal() + geom_point() +
  labs(x = "Dose (micrograms)", y = "Proportion Killed") +
  geom_line(aes(y = yhat), data = d) + ggtitle("log(Dose)")
plot(p)
```



2. For the model without the log transformation of dose, estimate the odds ratio for each gender for the effect of increasing dose by one unit (i.e., one microgram), and for the model with the log transformation of dose, estimate the odds ratio for each gender for the effect of doubling the dose. Use the `contrast` function to estimate the odds ratios. Summarize each odds ratio in a sentence that describes clearly the effect of increasing the dose for a given gender of tobacco budworm.

   Solution: Here are the odds ratios for the model without the log transformation of dose.

```
trtools::contrast(m.dose,
  a = list(Dose = 2, Gender = c("F","M")),
  b = list(Dose = 1, Gender = c("F","M")),
  tf = exp, cnames = c("F","M"))
```

```
  estimate lower upper
F    1.123 1.071 1.176
M    1.346 1.191 1.522
```

   We estimate that by increasing dose by one unit (i.e., one microgram) the odds of death increases by a factor of about 1.12 (i.e., about 12%) for female budworms, and by a factor of about 1.35 (i.e., about 35%) for male budworms. Here are the odds ratios for the model with the log transformation of dose.

```
trtools::contrast(m.logd,
  a = list(Dose = 2, Gender = c("F","M")),
  b = list(Dose = 1, Gender = c("F","M")),
  tf = exp, cnames = c("F","M"))
```

```
  estimate lower upper
F    2.474 1.783 3.433
M    3.522 2.324 5.337
```

We estimate that if we *double* the dose then the odds of death increases by a factor of about 2.47 (i.e., about 147%) for female budworms, and by a factor of about 3.52 (i.e., about 252%) for male budworms. You can also uses the `emtrends` function in the **emmeans** package to estimate odds ratios for a quantitative explanatory variable, but it will only compute odds ratios for a one unit increase.

```
library(emmeans)
emtrends(m.dose, ~Gender, var = "Dose",
  type = "response", tran = "log")
```

```
 Gender prob     SE  df asymp.LCL asymp.UCL
 F      1.12 0.0267 Inf      1.07      1.18
 M      1.35 0.0842 Inf      1.19      1.52

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

This is a little confusing because what you are supplying with the `tran` argument is the inverse function of the transformation function you are specifying with the `contast` function. One neat thing we can do with this (which can also be done with `contrast` but it is more tedious) is to make inferences to compare the two odds ratios (e.g., are they significantly different). This is what you are doing later when you test the interaction, but we can do it here with `pairs`.

```
pairs(emtrends(m.dose, ~Gender, var = "Dose",
  type = "response", tran = "log"))
```

```
 contrast ratio     SE  df null z.ratio p.value
 F / M    0.834 0.0558 Inf    1  -2.713  0.0067

Tests are performed on the log scale
```

What this means is that the odds ratio for females is about 17% lower than that for males. Note that the test statistic equals (except for the sign) the Wald test statistic for the interaction!

3. The plots show that the estimated probability of mortality is higher for male tobacco budworms (except perhaps at very low doses). Use either the `contrast` function or functions from the **emmeans** package to estimate the odds ratio for the effect of gender (i.e., comparing males to females) at a dose of five micrograms, and again at a dose of 10 micrograms. Summarize each of the four odds ratios in a sentence that describes clearly the comparison between male and female budworms at those two doses.

   Solution: First I will estimate the odds ratios for the model without the log transformation.

```
trtools::contrast(m.dose,
  a = list(Dose = c(5,10), Gender = "M"),
  b = list(Dose = c(5,10), Gender = "F"),
  tf = exp, cnames = c("@5","@10"))
```

```
    estimate lower  upper
@5     2.005 1.007  3.995
@10    4.971 2.044 12.091
```

```
pairs(emmeans(m.dose, ~Gender, at = list(Dose = 5), type = "response"),
  infer = TRUE, reverse = TRUE)
```

```
 contrast odds.ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 M / F          2.01 0.705 Inf      1.01         4    1   1.979  0.0478

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

```
pairs(emmeans(m.dose, ~Gender, at = list(Dose = 10), type = "response"),
  infer = TRUE, reverse = TRUE)
```

```
 contrast odds.ratio   SE  df asymp.LCL asymp.UCL null z.ratio p.value
 M / F          4.97 2.25 Inf      2.04      12.1    1   3.536  0.0004

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

So at a dose of five micrograms the odds of death for male budworms is about twice that of female budworms (i.e., about 101% higher), while at a dose of ten micrograms the odds of death for male budworms is about 4.98 times that of female budworms (i.e., about 398% higher). Next I will estimate the odds ratios for the model with the log transformation.

```
trtools::contrast(m.logd,
  a = list(Dose = c(5,10), Gender = "M"),
  b = list(Dose = c(5,10), Gender = "F"),
  tf = exp, cnames = c("@5","@10"))
```

```
    estimate lower upper
@5     2.703 1.326 5.512
@10    3.847 1.713 8.638
```

```
pairs(emmeans(m.logd, ~Gender, at = list(Dose = 5), type = "response"),
  infer = TRUE, reverse = TRUE)
```

```
 contrast odds.ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
 M / F           2.7 0.983 Inf      1.33      5.51    1   2.736  0.0062

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

```
pairs(emmeans(m.logd, ~Gender, at = list(Dose = 10), type = "response"),
  infer = TRUE, reverse = TRUE)
```

```
 contrast odds.ratio   SE  df asymp.LCL asymp.UCL null z.ratio p.value
 M / F          3.85 1.59 Inf      1.71      8.64    1   3.265  0.0011

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

So based on this model, the odds of death at a dose of five micrograms is about 2.7 times higher (i.e., about 170% higher) for male budworms, while at a dose of ten micrograms the odds of death for male budworms is about 3.85 times higher (i.e., about 285% higher).

4. In the previous problem you estimated odds ratios to compare male and female tobacco budworms at five and 10 micrograms of dose. Use either the `contrast` function or functions from the **emmeans** package to estimate the *probability* and the *odds* of death of male and female tobacco budworms at doses of five and 10 micrograms.

   Solution: We can estimate the probabilities as follows, first for the model without the log transformation.

```
trtools::contrast(m.dose, tf = plogis,
  a = list(Dose = c(5,5,10,10), Gender = c("F","M","F","M")),
  cnames = c("F@5","M@5","F@10","M&10"))
```

```
     estimate  lower  upper
F@5    0.2428 0.1645 0.3431
M@5    0.3914 0.2833 0.5112
F@10   0.3638 0.2709 0.4681
M&10   0.7397 0.5665 0.8608
```

```
emmeans(m.dose, ~Gender, at = list(Dose = 5), type = "response")
```

```
 Gender  prob     SE  df asymp.LCL asymp.UCL
 F      0.243 0.0458 Inf     0.164     0.343
 M      0.391 0.0591 Inf     0.283     0.511

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

```
emmeans(m.dose, ~Gender, at = list(Dose = 10), type = "response")
```

```
 Gender  prob     SE  df asymp.LCL asymp.UCL
 F      0.364 0.0509 Inf     0.271     0.468
 M      0.740 0.0763 Inf     0.567     0.861

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

And now for the model with the log transformation.

```
trtools::contrast(m.logd, tf = plogis,
  a = list(Dose = c(5,5,10,10), Gender = c("F","M","F","M")),
  cnames = c("F@5","M@5","F@10","M&10"))
```

```
     estimate  lower  upper
F@5    0.2912 0.1988 0.4048
M@5    0.5261 0.4016 0.6475
F@10   0.5041 0.3900 0.6177
M&10   0.7963 0.6683 0.8835
```

```
emmeans(m.logd, ~Gender, at = list(Dose = 5), type = "response")
```

```
 Gender  prob     SE  df asymp.LCL asymp.UCL
 F      0.291 0.0531 Inf     0.199     0.405
 M      0.526 0.0640 Inf     0.402     0.647

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

```
emmeans(m.logd, ~Gender, at = list(Dose = 10), type = "response")
```

```
 Gender  prob     SE  df asymp.LCL asymp.UCL
 F      0.504 0.0591 Inf     0.390     0.618
 M      0.796 0.0549 Inf     0.668     0.883

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

Next we can estimate the odds of death.

```
trtools::contrast(m.dose, tf = exp,
  a = list(Dose = c(5,5,10,10), Gender = c("F","M","F","M")),
  cnames = c("F@5","M@5","F@10","M&10"))
```

```
     estimate  lower  upper
F@5    0.3207 0.1968 0.5224
M@5    0.6430 0.3953 1.0460
F@10   0.5718 0.3715 0.8801
M&10   2.8423 1.3066 6.1830
```

```
trtools::contrast(m.logd, tf = exp,
  a = list(Dose = c(5,5,10,10), Gender = c("F","M","F","M")),
  cnames = c("F@5","M@5","F@10","M&10"))
```

```
     estimate  lower  upper
F@5    0.4107 0.2481 0.6801
M@5    1.1103 0.6712 1.8366
F@10   1.0164 0.6394 1.6156
M&10   3.9102 2.0151 7.5873
```

You can also use `emmeans` to estimate odds. We can "trick" the function into computing odds by specifying a logarithmic transformation function. Note that this is the opposite of what we specify with `contrast`. The **emmeans** package functions allow you to specify if there is a transformation of the response variable, but in this case when included with the `type = "response"` option we are effectively specifying that we are modeling the logarithm of the odds, so the function will "undo" the logarithm to give us odds (it is admittedly a bit confusing but it works as you can see when you compare it to what is reported by `contrast`). Unfortunately we can only specify one dose value at a time. Also note that the estimates are reported as `prob` but are in fact odds.

```
emmeans(m.dose, ~Gender, at = list(Dose = 5), type = "response", tran = "log")
```

```
 Gender  prob     SE  df asymp.LCL asymp.UCL
 F      0.321 0.0799 Inf     0.197     0.522
 M      0.643 0.1596 Inf     0.395     1.046

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

```
emmeans(m.dose, ~Gender, at = list(Dose = 5), type = "response", tran = "log")
```

```
 Gender  prob     SE  df asymp.LCL asymp.UCL
 F      0.321 0.0799 Inf     0.197     0.522
 M      0.643 0.1596 Inf     0.395     1.046

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

```
emmeans(m.logd, ~Gender, at = list(Dose = 5), type = "response", tran = "log")
```

```
 Gender  prob    SE  df asymp.LCL asymp.UCL
 F      0.411 0.106 Inf     0.248      0.68
 M      1.110 0.285 Inf     0.671      1.84

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

```
emmeans(m.logd, ~Gender, at = list(Dose = 5), type = "response", tran = "log")
```

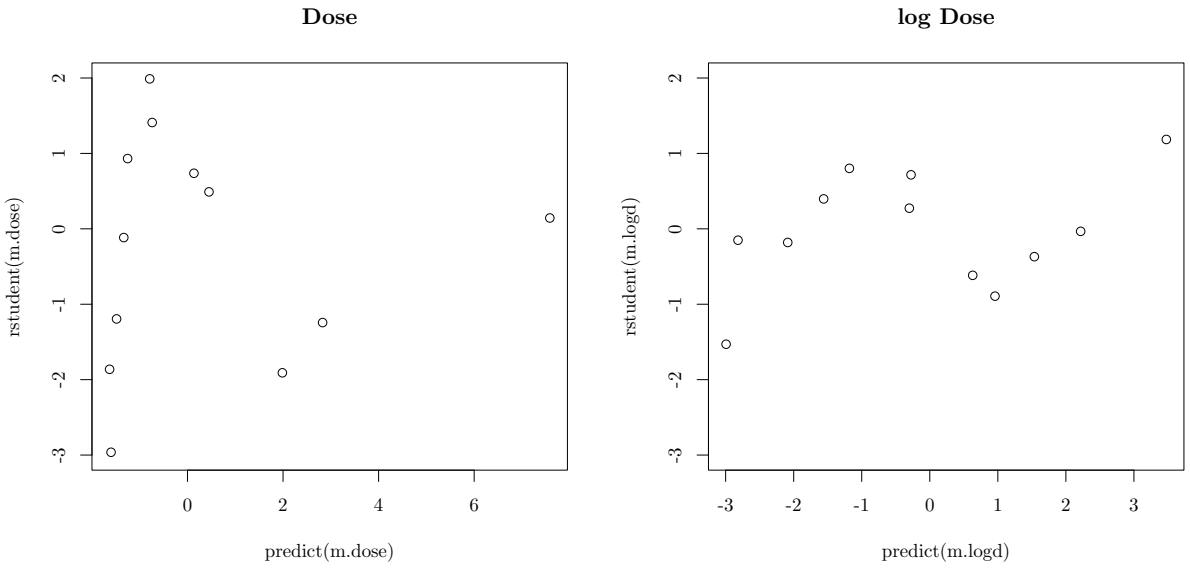
```
 Gender  prob    SE  df asymp.LCL asymp.UCL
 F      0.411 0.106 Inf     0.248      0.68
 M      1.110 0.285 Inf     0.671      1.84
```

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

5. The two models you estimated and used in the previous problems are not equivalent. We might want to assess which model is a better fit to the data. There are several ways that this could be done. One is to inspect the plots you made earlier to compare the estimated expected proportions to the observed proportions. A second approach is to inspect a residual plot of the predicted values against standardized or studentized residuals. And a third approach is to look at the residual deviance of each model, noting that the residual deviance can be viewed as a measure of the "lack of fit" relative to a hypothetical best-fitting model. Use all three methods to compare the three models and decide which of the two models would be a better fit to the data and explain your decision.

   Solution: Here are the residual plots for each model.

```
par(mfcol = c(1,2))
plot(predict(m.dose), rstudent(m.dose), ylim = c(-3,2), main = "Dose")
plot(predict(m.logd), rstudent(m.logd), ylim = c(-3,2), main = "log Dose")
```



   The residual deviance values can be seen by using `summary` or just using the `deviance` function.

```
deviance(m.dose)
```

```
[1] 18.16
```

```
deviance(m.logd)
```

```
[1] 4.994
```

   Based on model and residual plots and the residual deviances I would say that the model with the log transformation of dose is a better fit to the data. The plotted model appears to better represent the trends in the observed proportions. The residual plot for the model without the transformation shows larger (in absolute value) residuals overall and with a pattern that suggests that the nonlinear relationship between the observed and estimated expected proportions is not quite captured by the model. Finally the model with the transformation has a much lower residual deviance.

6. The models with an interaction allow the odds ratio for one explanatory variable to depend on the value of the other explanatory variable (i.e., the odds ratio for the effect of dose can depend on gender, and the odds ratio for the effect of gender can depend on dose). The *estimated* odds ratio for one explanatory variable does depend on the value of the other, but suppose we want to know if this is

statistically significant. This can be tested using a likelihood ratio test or a Wald test. Using the model you selected in the previous problem, estimate the same model except *without* the interaction and use this model with the original model with the interaction to conduct a likelihood ratio test of the interaction. Also conduct a Wald test of the interaction using `summary`. You should be able to identify the parameter responsible for the interaction relatively easily. For each test report the test statistic and the p-value as well as the decision assuming a significance level of 0.05.

Solution: The Wald test can be seen in the output of `summary`.

```
summary(m.logd)$coefficients
```

```
                  Estimate Std. Error z value  Pr(>|z|)
(Intercept)        -2.9935     0.5527 -5.4162 6.087e-08
log(Dose)           1.3071     0.2411  5.4221 5.891e-08
GenderM             0.1750     0.7783  0.2248 8.221e-01
log(Dose):GenderM   0.5091     0.3895  1.3071 1.912e-01
```

The values of the Wald test statistic is $z \approx 1.31$ which yields a p-value of about 0.19, which is not statistically significant. Next we can compute the likelihood ratio test statistic.

```
m.null <- glm(cbind(Killed, Number - Killed) ~ log(Dose) + Gender,
  family = binomial, data = budworm)
anova(m.null, m.logd, test = "LRT")
```

```
Analysis of Deviance Table

Model 1: cbind(Killed, Number - Killed) ~ log(Dose) + Gender
Model 2: cbind(Killed, Number - Killed) ~ log(Dose) * Gender
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         9       6.76
2         8       4.99  1     1.76     0.18
```

The test statistic is $X^2 \approx 1.76$ which yields a p-value of about 0.18, which is not statistically significant. Note that the Wald and likelihood ratio test statisics are on different scales. For a test of one parameter the Wald test statistic is usually reported as that with a distribution of a standard normal distribution whereas a likelihood ratio test statistic is reported as one with a chi-squared distribution. But if you square a test statistics that has a standard normal distribution then the result will have a chi-squared distribution, so the test statistics are actually closer than the look since $1.31^2 \approx 1.72$.

Case-Control Study of Peptic Ulcers and Aspirin Use

The data frame `ulcer` in the **dobson** package contains data from a study of the relationship between peptic ulcers and aspirin use.[14] There is evidence that non-steroidal anti-inflammatory drugs (NSAID) like aspirin are risk factors for peptic ulcers. This study used a retrospective case-control design. This design involves identifying a sample of "cases" (e.g., people with a peptic ulcer) and a sample of "controls" (e.g., people without a peptic ulcer) and then comparing them with respect to prior risk factors (e.g., regular use of aspirin). This particular study formed case and control groups for two different kinds of cases corresponding to two different kinds of peptic ulcers: *duodenal* ulcers (i.e., ulcers in the first part of the upper intestines), and *gastric* ulcers (i.e., ulcers in the stomach).

Data from case-control studies are often modeled using logistic regression where the status (i.e., case or control) is used as the response variable. It is actually the *wrong* model for the design. The likelihood function for logistic regression assumes that the individual binary observations are independent, but this cannot be true in a retrospective case-control design where the number of cases and controls are determined by the researchers. It can be shown, however, that using logistic regression will result in consistent estimators

[14] Duggan, J. M., Dobson, A. J., Johnson, H., & Fahey, P. P. (1986). Peptic ulcer and non-steroidal anti-inflammatory agents. *Gut*, *27*, 929–933.

for all parameters *except* β_0.[15] The β_0 parameter in a retrospective case-control design depends also on the probabilities of cases and controls being included in the study (which are generally unknown) so what is being estimated by β_0 in a logistic regression model for a retrospective case-control study is not the same as what is being estimated by β_0 for a prospective study where, for example, subjects were first classified based aspirin use and then we waited to see who developed peptic ulcers. This implies that the model for the retrospective case-control design *cannot* produce valid inferences for the *probability* of an event since the number of cases and controls are determined *by design*. But it can be shown that the inferences for the *odds ratio* for any explanatory variable are valid. Retrospective case-control designs are often used for fairly rare events (e.g., ulcers) where it is easier to identify a sample of subjects with the condition *after* it has happened.[16]

The data from this study are stored in terms of frequencies of each combination of ulcer type, case or control status, and aspirin use.

```
library(dobson)
ulcer
```

```
# A tibble: 8 x 4
  ulcer    `case-control` aspirin  frequency
  <chr>    <chr>          <chr>        <dbl>
1 gastric  control        non-user        62
2 gastric  control        user             6
3 gastric  case           non-user        39
4 gastric  case           user            25
5 duodenal control        non-user        53
6 duodenal control        user             8
7 duodenal case           non-user        49
8 duodenal case           user             8
```

For modeling it is useful to reshape the data so that each row gives the number of cases and controls.

```
library(dplyr)
library(tidyr)
ulcer <- dobson::ulcer %>%
  pivot_wider(names_from = `case-control`, values_from = frequency)
ulcer
```

```
# A tibble: 4 x 4
  ulcer    aspirin  control  case
  <chr>    <chr>      <dbl> <dbl>
1 gastric  non-user      62    39
2 gastric  user           6    25
3 duodenal non-user      53    49
4 duodenal user           8     8
```

We can also compute the proportion of regular users of aspirin and non-users that are cases for each type of ulcer — i.e., the proportion of participants in the study for each combination of ulcer type and aspirin use that have an ulcer.

```
ulcer %>% group_by(ulcer, aspirin) %>%
  mutate(proportion = case / (case + control))
```

```
# A tibble: 4 x 5
# Groups:   ulcer, aspirin [4]
```

[15] The property of consistency is quite technical, but what it essentially means that as the sample size increases the estimator will have a tendency to produces estimates that are closer to the parameter being estimated.

[16] Because probabilities and odds are very similar for very small probabilities, an odds ratio for a model for a rare event (like an ulcer) can be viewed as a good approximation to the relative risk which is a ratio of probabilities instead of odds. So if the odds ratio is, say, two, then this means that the odds are twice as high but also the probability is nearly twice as high.

```
  ulcer    aspirin  control  case proportion
  <chr>    <chr>      <dbl> <dbl>      <dbl>
1 gastric  non-user      62    39      0.386
2 gastric  user           6    25      0.806
3 duodenal non-user      53    49      0.480
4 duodenal user           8     8      0.5
```

This illustrates why a logistic regression model for a retrospective case-control design cannot be used to infer probabilities. For example, the percent of regular aspirin users that have a gastric ulcer is over 80%, but certainly the probability of a regular aspirin user having a gastric ulcer is not that high. As stated earlier, a logistic regression model for a retrospective case-control design can be used to estimate odds ratios, but not (interpretable) probabilities.

1. Estimate a logistic regression model for the proportion of subjects that are cases using ulcer type and aspirin use as explanatory variables. Include an interaction in your model. Report the parameter estimates and their standard errors using the `summary` function.

   Solution: Here is how we can estimate the model.

   ```
   m <- glm(cbind(case, control) ~ aspirin * ulcer,
     family = binomial, data = ulcer)
   summary(m)$coefficients
   ```

   ```
                           Estimate Std. Error z value Pr(>|z|)
   (Intercept)              -0.07847     0.1982 -0.3960  0.69214
   aspirinuser               0.07847     0.5378  0.1459  0.88400
   ulcergastric             -0.38510     0.2847 -1.3527  0.17614
   aspirinuser:ulcergastric  1.81222     0.7333  2.4714  0.01346
   ```

2. Using either the `contrast` function or functions from the **emmeans** package, estimate the odds ratio for the effect of regular use of aspirin for gastric and duodenal ulcers (i.e., one odds ratio for each type of ulcer). Report the odds ratios and their confidence intervals. Interpret each odds ratio in writing to explain what it means about the relationship between having a particular type of ulcer and regular aspirin use.

   Solution: Here is how we can estimate the odds ratios.

   ```
   trtools::contrast(m,
     a = list(aspirin = "user", ulcer = c("gastric","duodenal")),
     b = list(aspirin = "non-user", ulcer = c("gastric","duodenal")),
     tf = exp, cnames = c("gastric","duodenal"))
   ```

   ```
            estimate  lower  upper
   gastric     6.624 2.4937 17.595
   duodenal    1.082 0.3769  3.104
   ```

   ```
   library(emmeans)
   pairs(emmeans(m, ~aspirin|ulcer, type = "response"), infer = TRUE, reverse = TRUE)
   ```

   ```
   ulcer = duodenal:
    contrast          odds.ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
    user / (non-user)       1.08 0.582 Inf     0.377       3.1    1   0.146  0.8840

   ulcer = gastric:
    contrast          odds.ratio    SE  df asymp.LCL asymp.UCL null z.ratio p.value
    user / (non-user)       6.62 3.302 Inf     2.494      17.6    1   3.793  0.0001

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

```
Tests are performed on the log odds ratio scale
```

We estimate that the odds of a duodenal ulcer is about 1.08 times higher (i.e., about 8% higher) for an aspirin user than for a non-user (although this is not statistically signficiant). We also estimate that the odds of gastric ulcer is about 6.62 times higher (i.e., about 562% higher) for an aspirin user than for a non-user.

3. A test of the interaction is a test of whether or not the odds ratios are different for the two types of ulcers. Conduct either a likelihood ratio test or a Wald test of the interaction. Report your test statistic, p-value, and decision using a significance level of 0.05.

   Solution: Here are the Wald and likelihood ratio tests.

```
summary(m)
```

```
Call:
glm(formula = cbind(case, control) ~ aspirin * ulcer, family = binomial,
    data = ulcer)

Deviance Residuals:
[1]  0  0  0  0

Coefficients:
                         Estimate Std. Error z value Pr(>|z|)
(Intercept)               -0.0785     0.1982   -0.40    0.692
aspirinuser                0.0785     0.5378    0.15    0.884
ulcergastric              -0.3851     0.2847   -1.35    0.176
aspirinuser:ulcergastric   1.8122     0.7333    2.47    0.013 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance:  1.7698e+01  on 3  degrees of freedom
Residual deviance: -1.6209e-14  on 0  degrees of freedom
AIC: 24.8

Number of Fisher Scoring iterations: 3
```

```
m.null <- glm(cbind(case, control) ~ aspirin + ulcer,
  family = binomial, data = ulcer)
anova(m.null, m, test = "LRT")
```

```
Analysis of Deviance Table

Model 1: cbind(case, control) ~ aspirin + ulcer
Model 2: cbind(case, control) ~ aspirin * ulcer
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         1       6.28
2         0       0.00  1     6.28    0.012 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The Wald test statistic is $z \approx 2.47$ and the likelihood ratio test statistic is $X^2 \approx 6.28$, yield p-values of 0.013 and 0.012, respectively. Both are statistically significant.