Topic 7 Generalised Linear Models

Steve Paterson

2024-06-03

Up until now we've worked largely with data that is normally distributed, which is what t-tests, anova and linear-models expect in order to meet their assumptions. (More formally, it's the residuals for these models that should be normally distributed). Real data, though, does not have to be normal. Here we will consider two approaches where data are not normal.

The first approach is to transform the response variable to a form where it is (approximately) normally distributed.

The second approach is to use generalised linear models (GLMs) that are explicitly designed for non-normal distributions.

Transforming the response variable

We've touched on this approach already. In Topic 2 we used a data set on vertebrate age as an example. Find the file 'vertebrate_age.csv` and read it in again.

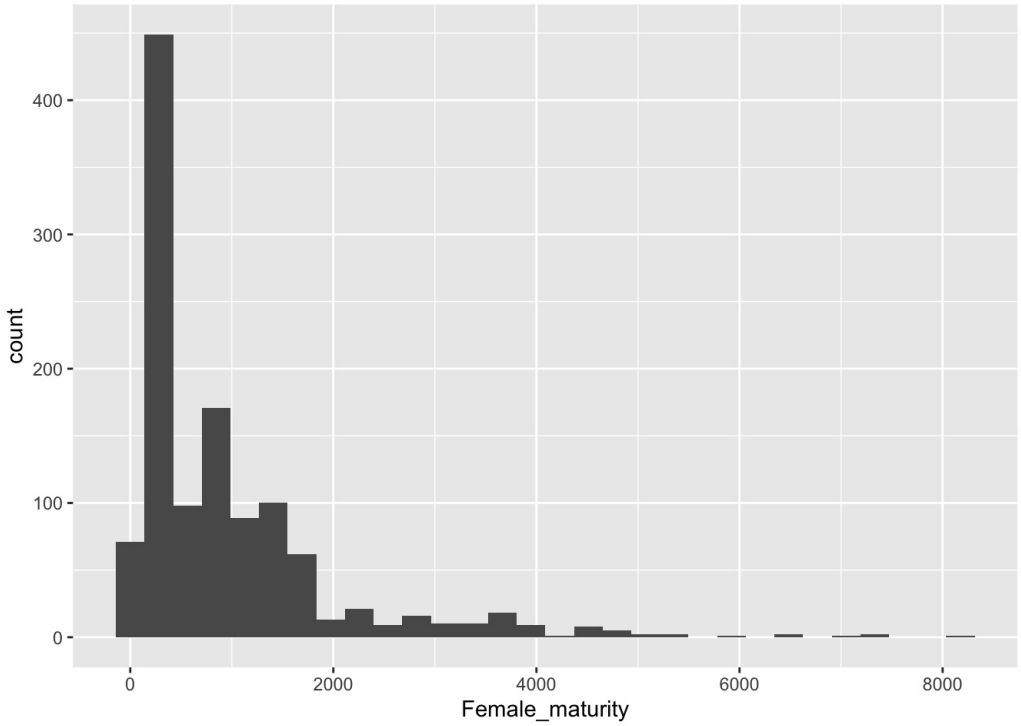
Specify the 'Class' column (birds, mammals, etc) as a factor.

```
vertebrate_age$Class <- factor(vertebrate_age$Class)

#now check this has worked
summary(vertebrate_age$Class)
```

```
##  Amphibia      Aves  Mammalia  Reptilia Teleostei
##       125       679       660       260       446
```

In Topic 2, we looked at female maturity as a trait and used histograms to view distributions.

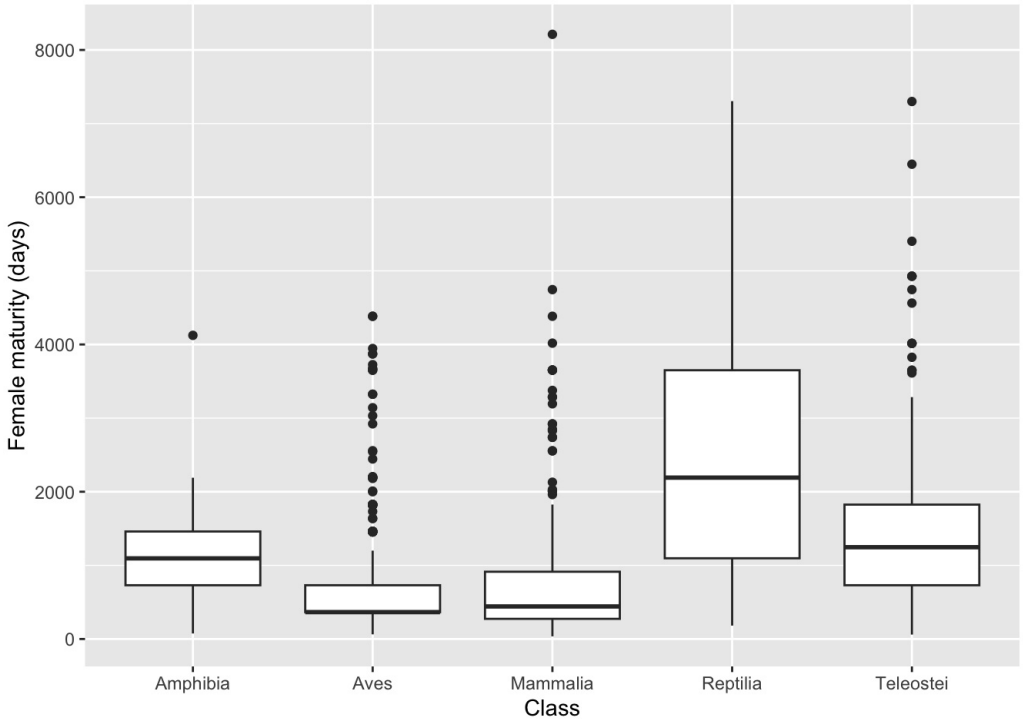
```
p.fm1 <- ggplot(data=vertebrate_age,aes(x=Female_maturity))

p.fm1 + geom_histogram()
```



Here you see that there's clump of values near zero and a tail of values out towards the right. This is one indication that the data are not normal.

Similarly, you can see this plot from Topic 2 and the tail of a few species in each class with a long maturation time.

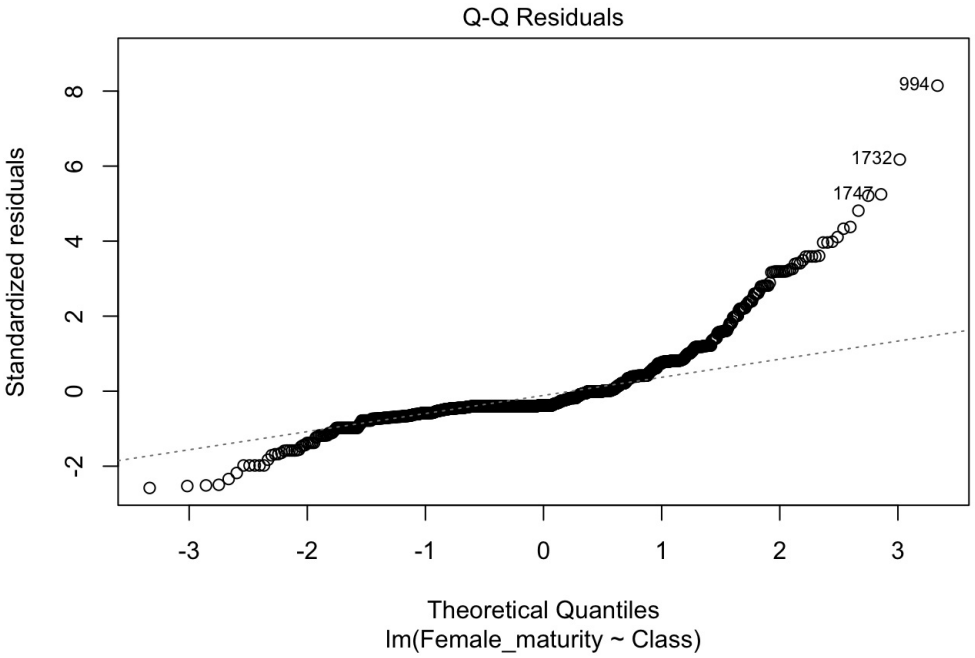
```
p.fm2 <- ggplot(data=vertebrate_age,aes(x=Class, y=Female_maturity))

p.fm2 + geom_boxplot() +
   ylab("Female maturity (days)")
```



One thing we might be interested in is testing whether female maturity differs between vertebrate classes. Applying a linear model to the raw data is easy enough to do (but may not be the right thing to do).

```
mod.fm1 <- lm(Female_maturity ~ Class, data = vertebrate_age)
```

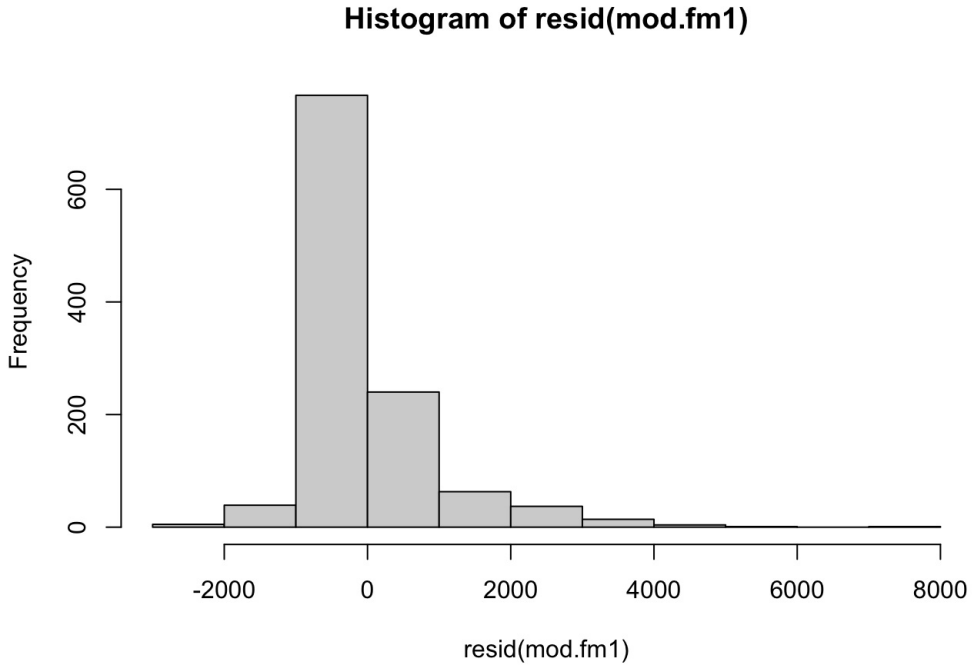
Now do the diagnostic plot with `plot(mod.fm1)`. The qqplot looks really odd with a big curve up off the line on the right-hand side.



While one never expects real data to fall exactly on the line, this looks really far off the line in places. Essentially its saying that there are some much bigger residuals than you would expect.

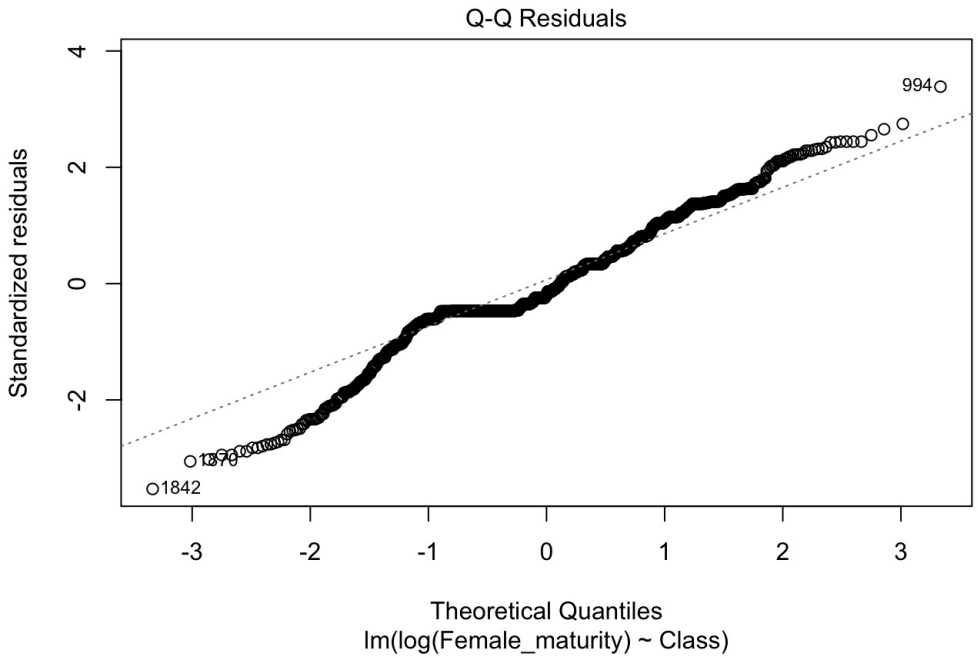
This becomes apparent if you plot a histogram of the residuals directly. For a reasonably well-fitted model, you should get something symmetrical.

```
hist(resid(mod.fm1))
```

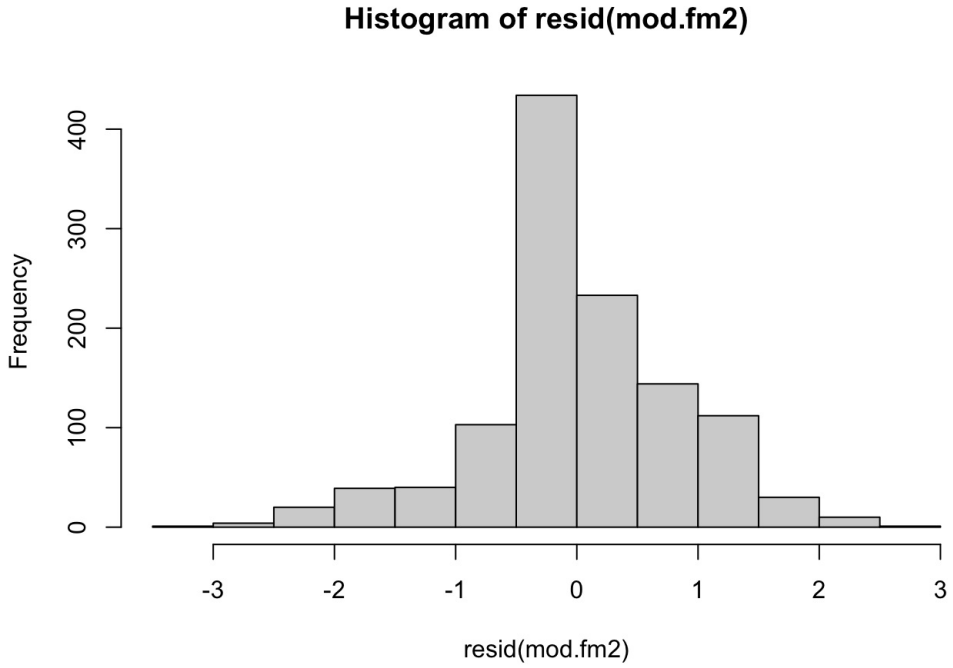


A log transform is often called for. In fact when plotting the data in Topic 2, we often plotted these on a log axis. Let's put log-transformed values for female maturity into a linear model. You can either create another column in your data frame with the log-transformed values, or just do this on the fly within `lm`.

```
mod.fm2 <- lm(log(Female_maturity) ~ Class, data = vertebrate_age)
plot(mod.fm2, which =2)
```




```
hist(resid(mod.fm2))
```



A linear model using log transform therefore looks a lot better than on the raw data, even if the lower part of the line is still a bit off the line. There are different approaches to how to deal with non-normal data like this. The strictest approach is to is to say that if you can't get your residuals to be completely normal then you throw your whole approach out and do something else. Real data is never completely normal so we need to be a bit more pragmatic. Fortunately, a linear model is fairly robust to how the residuals look, so long as they're not completely skewed (as in the qqplot or histogram for the untransformed data). See Kneif and Fortmeier (2021) (https://doi.org/10.3758/s13428-021-01587-5) for a comprehensive study of the problem. So in our example it's probably OK to use the log-transformed data here in a linear model or anova, provided that you have the caveat in mind that the model isn't perfect. This is probably most important in interpreting the p-value that results, where a p-value may not be completely accurate and needs a bit of caution if it's only just below 5%. In this case though, there doesn't seem much doubt that class significantly affects female maturation.

```
drop1(mod.fm2,test="F")
```

```
## Single term deletions
##
## Model:
## log(Female_maturity) ~ Class
##        Df Sum of Sq     RSS     AIC F value    Pr(>F)
## <none>               858.78 -353.12
## Class   4    224.44 1083.23  -89.23  76.184 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod.fm2)
```

```
##
## Call:
## lm(formula = log(Female_maturity) ~ Class, data = vertebrate_age)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0203 -0.4037 -0.1697  0.5122  2.9035
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.8915     0.1108  62.201  < 2e-16 ***
## ClassAves       -0.5880     0.1190  -4.943 8.82e-07 ***
## ClassMammalia   -0.7817     0.1172  -6.668 4.00e-11 ***
## ClassReptilia    0.6643     0.1531   4.339 1.55e-05 ***
## ClassTeleostei   0.2231     0.1311   1.702    0.089 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8582 on 1166 degrees of freedom
##   (999 observations deleted due to missingness)
## Multiple R-squared:  0.2072, Adjusted R-squared:  0.2045
## F-statistic: 76.18 on 4 and 1166 DF,  p-value: < 2.2e-16
```

TASK

The file chlorophyll.txt contains data on chlorophyll levels (mg/L) on a number of lakes, plus information on pH, water alkalinity (mg/L carbonate levels) and calcium concentration (mg/L).

Generate and present a minimal model that identifies the main driver(s) of chlorophyll levels. This will involve what you've learnt in the previous model selection topic as well as what you've learnt here in transforming data.

Note that there is more than one way to tackle this, and no single right answer.

Generalised linear models

Sometimes the nature of the data mean that it's fundamentally not normal or able to be transformed. This is most apparent for count data, which can only be an integer ≥ 0.

For this reason, the linear model can be extended to the generalized linear model (GLM), using the linear predictor η and the link function $g(\mu)$. Here the errors around the linear predictor η are normal and so lets us use the same framework.

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n g(\mu) = \eta$$

You will need to read Chapter 13 of Crawley 'The R Book' to fully understand the concepts (wikipedia also has an OK explanation (https://en.wikipedia.org/wiki/Generalized_linear_model)).

Some of the common types of data you might see are:

Continuous data. This is most familiar to you from previous topics and includes quantitative data where the residuals from fitting a model are normally distributed. This also includes data that can be transformed, eg logged, to allow it to conform to a normal distribution (more on this in the next topic). These can be fitted using a linear model (`lm`).

Discrete data. Data are from counts and can either be bound on the left side only $\{0, 1, 2, \ldots, \infty\}$ for which a Poisson distribution is appropriate, or bounded on both the left and right hand side $\{0, 1, 2, \ldots, n\}$ for which a binomial distribution is appropriate. (Note though that if your counts are in the hundreds, as you might get from a large sample, a normal distribution may provide a good approximation.)

Task

Do the quiz on different types of data and how to analyse them

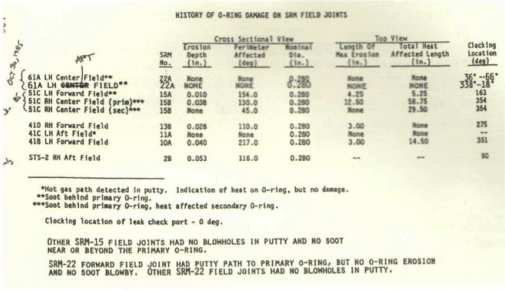
The Challenger example

Let's look at a famous set of data from the Challenger disaster. Failure of the O-rings on the booster rockets led to an explosion and the loss of 7 crew.

Previous to the launch, the company (Morton-Thiokol) that made the boosters, had the necessary data to show that launching in cold weather would likely be catastrophic. But they were unable to convince NASA to delay the launch – at least in part because their time to prepare a presentation to NASA was so rushed they presented a disparate series of graphs rather than one clear graph. We will follow an example modified from Faraway (2006).

The next three figures give the data as presented by the engineers, can we do any better?

O-ring data sheet 1



HISTORY OF O-RING DAMAGE ON SRM FIELD JOINTS

SRM No.	Cross Sectional View: Erosion Depth (in.)	Cross Sectional View: Per Heater Affected (deg)	Cross Sectional View: Nominal Dia. (in.)	Top View: Length Of Max Erosion (in.)	Top View: Total Heat Affected Length (in.)	Clocking Location (deg)
61A LH Center Field**	22A	None	None	0.280	None	36°-66°
61A LH CENTER FIELD**	22A	NONE	NONE	0.280	NONE	338°-18°
51C LH Forward Field**	16A	0.010	154.0	0.280	4.25	163
51C RH Center Field (prim)***	15B	0.038	130.0	0.280	12.50	58.75	354
51C RH Center Field (sec)***	15B	None	45.0	0.280	None	29.50	354
41D RH Forward Field	13B	0.028	110.0	0.280	3.00	None	275
41C LH Aft Field*	11A	None	None	0.280	None	None	--
41B LH Forward Field	10A	0.040	217.0	0.280	3.00	14.50	351
STS-2 RH Aft Field	2B	0.053	116.0	0.280	--	--	90

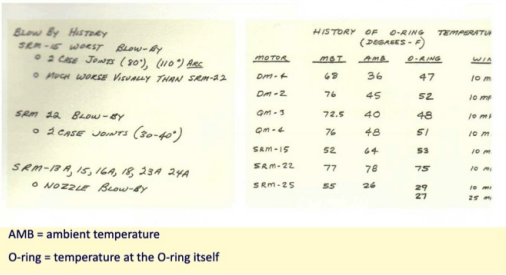
*Hot gas path detected in putty. Indication of heat on O-ring, but no damage.
**Soot behind primary O-ring.
***Soot behind primary O-ring, heat affected secondary O-ring.

Clocking location of leak check port - 0 deg.

OTHER SRM-15 FIELD JOINTS HAD NO BLOWHOLES IN PUTTY AND NO SOOT NEAR OR BEYOND THE PRIMARY O-RING.

SRM-22 FORWARD FIELD JOINT HAD PUTTY PATH TO PRIMARY O-RING, BUT NO O-RING EROSION AND NO SOOT BLOWBY. OTHER SRM-22 FIELD JOINTS HAD NO BLOWHOLES IN PUTTY.

O-ring data sheet 2



Blow By History
SRM-15 Worst Blow-By
- 2 Case Joints (30°), (110°) Asc
- Much Worse Visually than SRM-22

SRM 22 Blow-By
- 2 Case Joints (30-40°)

SRM-13A, 15, 16A, 18, 23A 24A
- Nozzle Blow-By

HISTORY OF O-RING (DEGREES - F) TEMPERATU

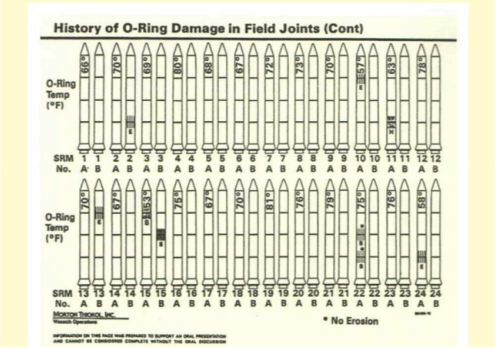
MOTOR	MGT	AMB.	O-RING	WIN
DM-4	68	36	47	10 m
DM-2	76	45	52	10 m
QM-3	72.5	40	48	10 m
QM-4	76	48	51	10 m
SRM-15	52	64	53	10 m
SRM-22	77	78	75	10 m
SRM-25	55	26	29 27	10 m 25 m

AMB = ambient temperature

O-ring = temperature at the O-ring itself

SRM # = launch number of shuttle

O-ring data sheet 3



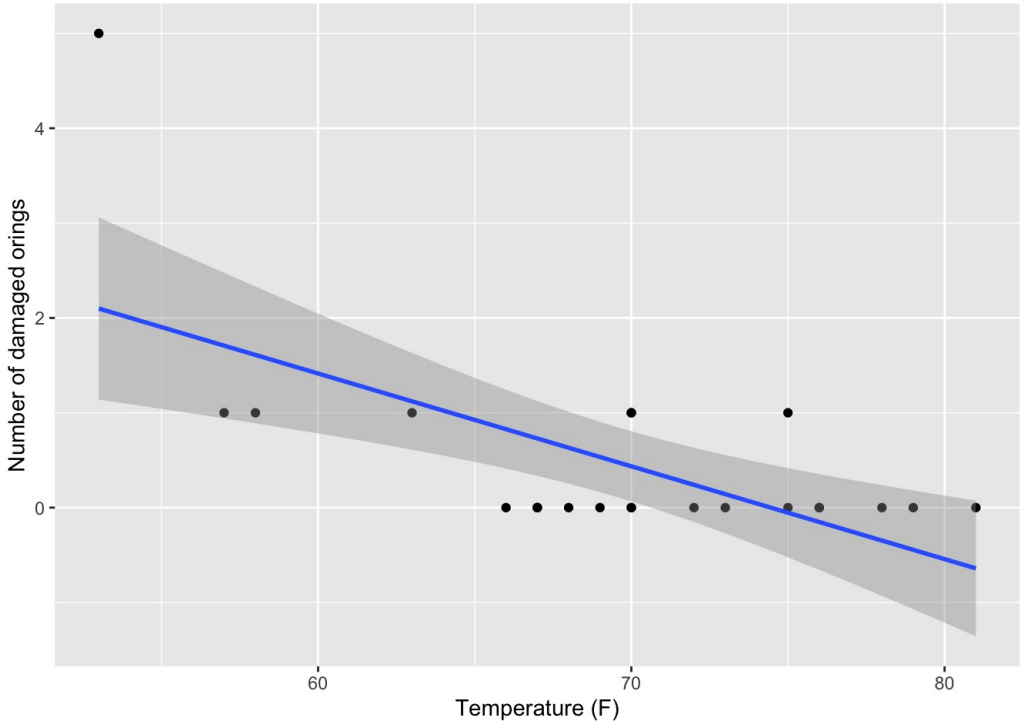
Plotting the Challenger data

Read in the orings.txt file from the canvas site and check it's read in OK.

Let's plot the data and stick a straight line through it.

```
# What data we want to plot
oring.plot1 <- ggplot(data=orings, aes(x=temp,y=damage))

# plot the points
oring.plot1 + geom_point() +

  # put on informative axes labels
  xlab("Temperature (F)") + ylab("Number of damaged orings") +

  # stick a straight line through it
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



There are 3 O-rings per booster and 2 boosters per shuttle, so this plot represents the number of the O-rings on a launch damaged with temperature in °F.

Each group of 6 O-rings is independent of each other, but each launch involves 6 O-rings so we plot them in groups of 6, rather than independently.

Most of the damaged O-rings appear at the colder temperatures, but simply plotting a straight line doesn't work well since there can never be less than zero or more than 6 O-rings damaged on a launch.

For most purposes, an O-ring is either damaged or not, rather taking some fractional value. The number of O-rings damaged is a *binomial* variable, it can take an integer value between 0 and6 and the probability of success of an individual O-ring (i.e. that it doesn't fail) takes the range $0 \leq p(success) \leq 1$.

Here we can use a logistic link function which instead of describing the probability of success of an O-ring, describes the log odds ratio of success over failure.

$$\log(oddsratio) = \log(\frac{p(success)}{p(failure)})$$

Fitting a generalized linear model in R

Previously we've used the `lm` function for linear models. Here we use a very similar syntax with `glm`. One difference is that we have to specify the `family` argument and state that the data are binomial. For binomial data, we also need to code it as two columns of successes and failures, adding up to the total number of trials. The function `cbind` essentially sticks two columns together side by side.

```
#what the binomial response looks like going into the model
with(orings,head(cbind(damage, 6 - damage)))
```

```
##      damage
## [1,]      5 1
## [2,]      1 5
## [3,]      1 5
## [4,]      1 5
## [5,]      0 6
## [6,]      0 6
```

```
#fitting the model
or1 <- glm(cbind(damage, 6 - damage) ~ temp, data = orings, family = "binomial")

#testing the significance of temperature
drop1(or1,test="Chisq")
```

```
## Single term deletions
##
## Model:
## cbind(damage, 6 - damage) ~ temp
##        Df Deviance    AIC    LRT  Pr(>Chi)
## <none>      16.912 33.675
## temp    1   38.898 53.660 21.985 2.747e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

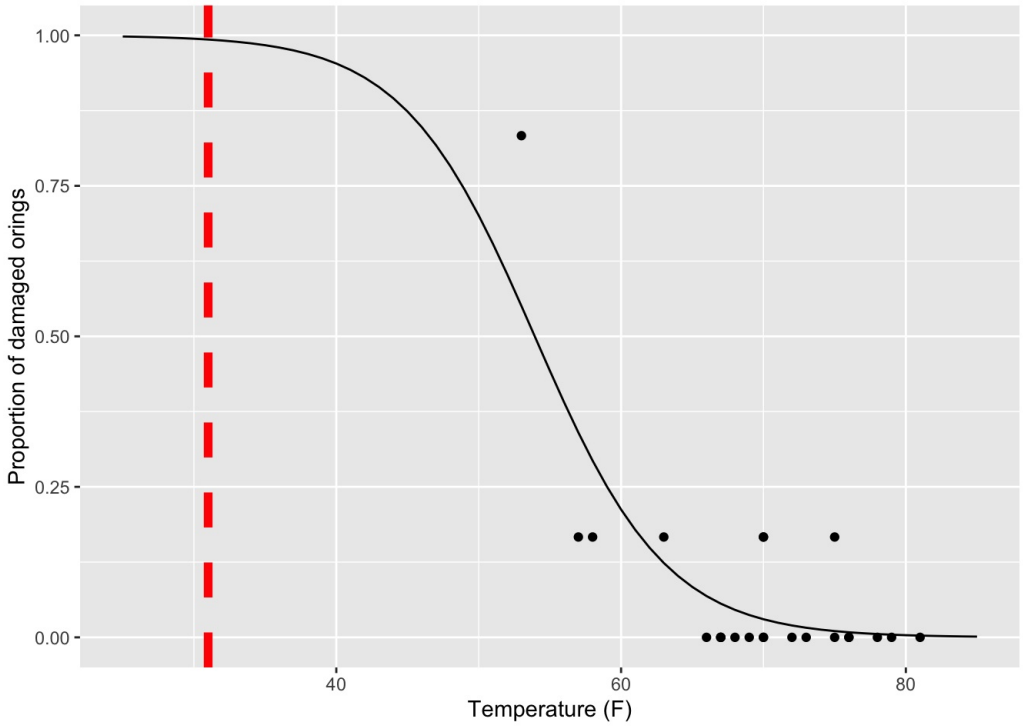
Note that, while we use an *F*-test for significance in a linear model, for a GLM we compare the test statistic (which is a log-likelihood ratio) to a Chi-square distribution, hence the use of `test="Chisq"` as an argument to the `drop1` function.

Let's replot the figure above. Instead of plotting number of orings damaged, we'll plot proportion of orings damaged. On top of this we'll lay on top the model prediction of probability of an oring being damaged.

you're combining both the temperature sequence and the predictions into a single data frame called or.predict.
This data frame will have two columns:
temp: The temperatures from 25 to 85.
pred.damage: The predicted damage values for each corresponding temperature based on the model or1.

or1 is likely a pre-trained model that you've built earlier (for example, a logistic regression or a linear model). You're now using it to predict values.
type = "response" means you're asking for the actual predicted values (not just the raw prediction values, like log-odds, if it were a logistic regression).
newdata = data.frame(temp = seq(25, 85)) means that you're asking the model to predict the outcome for each temperature in the sequence you created earlier.

```
#Create a data frame for the model prediction
# we put in temperatures between 25 and 85 F and ask R to predict the probability of damage for each temperature
or.predict <- data.frame(temp = seq(25,85),
                pred.damage=predict(or1, type = "response", newdata = data.frame(temp = seq(25,85))))


# What data we want to plot
oring.plot1 <- ggplot(data=orings, aes(x=temp,y=damage/6))

# plot the points
oring.plot1 + geom_point() +

  # put on informative axes labels
  xlab("Temperature (F)") + ylab("Proportion of damaged orings") +

  # set the limits of x and y axes beyond the range of the data (but to include the fatal launch temperature)
  ylim(c(0,1)) + xlim(c(25,85)) +

  # add a line for the probability of failure predicted by the model
  geom_line(data=or.predict,aes(x=temp,y=pred.damage)) +

  # the fatal launch was at 31F, let's plot this as a thick red line
  geom_vline(xintercept=31,col="red",linetype="dashed",linewidth=2)
```



There are many caveats around extending a prediction beyond the range of the available data, but nevertheless… if you had this plot, would you chose to launch?

TASK

Sex determination in turtles is dependent on the temperature at which eggs are incubated at. Consider the data in the turtles.txt file and describe this relationship.

TASK

Wild-type and a mutant bacterial strain were tested for their ability to gain antibiotic resistance following infection by a lysogenic phage (which integrates into the host genome and hence alter the regulation of host genes). Counts of antibiotic resistant bacteria in wildtype and mutant strains following phage infection in replicate experiments were conducted.

Consider the data in phage.txt and perform an analysis to test for a difference in antibiotic resistance between wildtype and mutant bacterial strains.