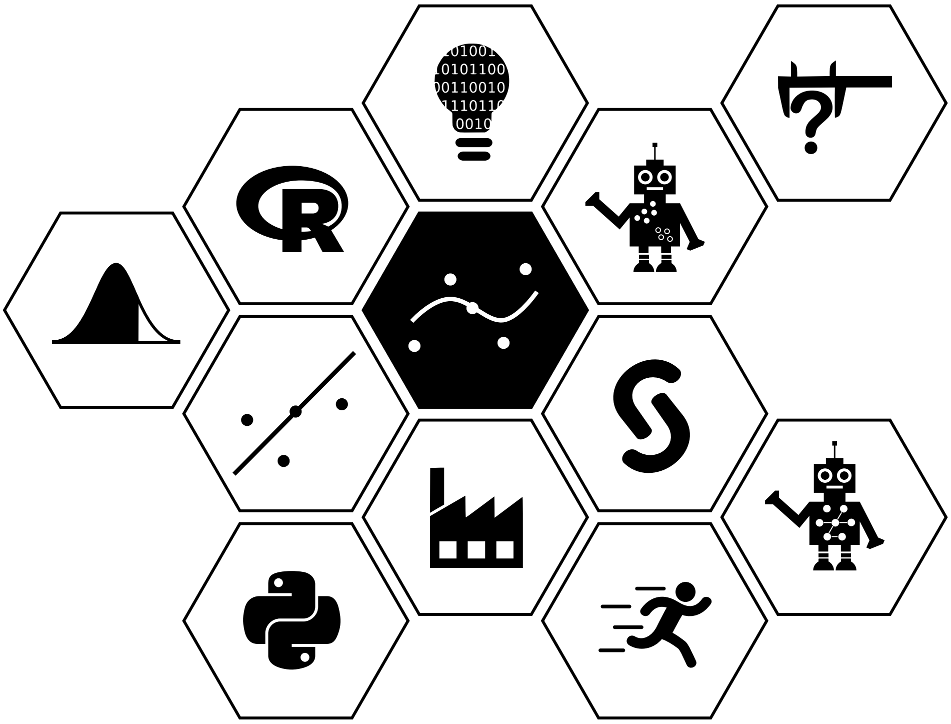
Advanced Predictive Models

Tereza Neocleous

Academic Year 2020-21

Week 3:

Models for binary/binomial responses



Models for binary/binomial response

This week we will learn how to model outcomes of interest that take one of two categorical values (e.g. yes/no, success/failure, alive/dead). The independent responses Y_i can either be

- *binary* (ungrouped), taking the value 1 (say success, with probability p_i) or 0 (failure, with probability $1 - p_i$) or
- *binomial* (grouped), where Y_i is the number of successes in a given number of trials n_i, with the probability of success being p_i and the probability of failure being $1 - p_i$.

In both cases the distribution of the Y_i is binomial, but in the first case it is Bin$(1, p_i)$ and in the second case it is Bin(n_i, p_i).

Binomial response

We will begin with models for binomial responses and we will look at exploratory plots of the data, different choices of link function and hypothesis tests about terms in the model. Finally we will examine measures of goodness of fit of the model. Let's begin with an example.

Logit model

```
m1 <- glm(beetles.mat ~ beetles$dose,
          family = binomial(link = 'logit'))
summary(m1)
Coef
              imate Std. Error z value Pr(>|z|)
(I            .717      5.181  -11.72   <2e-16 ***
bee          34.270     2.912   11.77   <2e-16 ***
           nce: 284.202  on 7  degrees of freedom
Residual deviance:  11.232  on 6  degrees of freedom
qchisq(df=6, p=0.95)

## [1] 12.59159
```

Binomial response models applied to beetle mortality data

https://youtu.be/d2nG7Y17sQw

Duration: 9m59s

Example 1 (Beetle mortality).

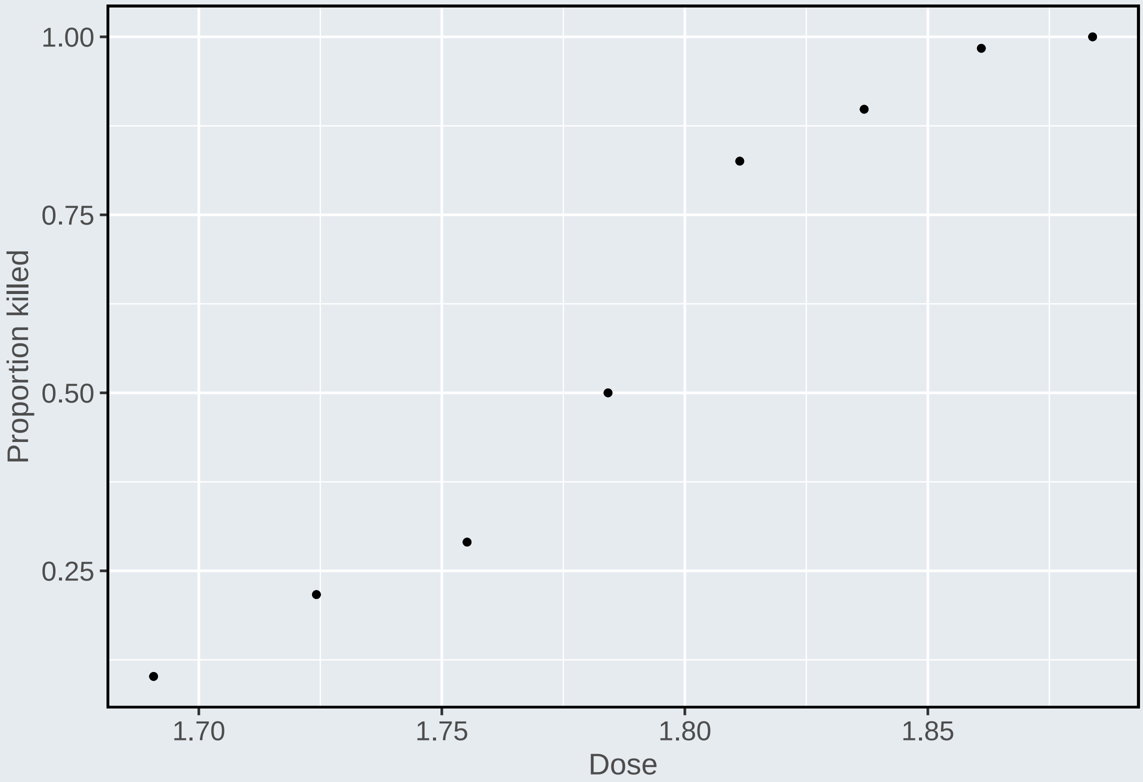
Data for this example consists of the number of beetles dead (*killed*) after five hours exposure to gaseous carbon disulphide at various concentrations (*dose*). The goal for this analysis is to model the probability of a beetle dying as a function of the carbon disulphide dose.

```
beetles <- read.csv(url("http://www.stats.gla.ac.uk/~tereza/rp/beetles.csv"))
beetles
```

dose	number	killed
1.6907	59	6
1.7242	60	13
1.7552	62	18
1.7842	56	28
1.8113	63	52
1.8369	59	53
1.8610	62	61
1.8839	60	60

Since we have grouped data (multiple beetles per dose), we can visualise the probability of the outcome of interest (beetles killed) by plotting the proportion killed for each dose against the dose. We see that the proportion killed increases with increasing dose.

```
beetles$propkilled <- beetles$killed / beetles$number
p1 <- ggplot(beetles, aes(x = dose, y = propkilled))+
      geom_point(size = 1) + xlab ("Dose") + ylab ("Proportion killed")
```



Since the response variable Y_i can only take one of two values (killed or not), we need to apply a generalised linear model. This takes the form $g(p_i) = \beta_0 + \beta_1 x_i$ where x_i is the dose for $i = 1, \ldots, 8$, $Y_i \overset{indep}{\sim}$ Bin(n_i, p_i) and p_i is the proportion of beetles killed for the ith dose. We have several choices of link function $g(p_i)$. We start by considering the *logit* link function $g(p_i) = \log\left(\frac{p_i}{1-p_i}\right)$.

This model can be fitted in R using the `glm()` function as follows:

```
beetles.mat <- cbind(beetles$killed, beetles$number-beetles$killed)
m1 <- glm(beetles.mat ~ beetles$dose, family = binomial(link = 'logit'))
```

Notice that we specify the response as a matrix with two columns, the first being the number of successes (`beetles$killed`) and the second the number of failures (`beetles$number-beetles$killed`).

The output is given below:

```
summary(m1)
```

```
Call:
glm(formula = beetles.mat ~ beetles$dose, family = binomial(link = "logit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5941  -0.3944   0.8329   1.2592   1.5940

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -60.717      5.181  -11.72   <2e-16 ***
beetles$dose   34.270      2.912   11.77   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 284.202  on 7  degrees of freedom
Residual deviance:  11.232  on 6  degrees of freedom
AIC: 41.43

Number of Fisher Scoring iterations: 4
```

From the `glm()` output we can get the estimates $\hat{\beta}_0 = -60.72$ and $\hat{\beta}_1 = 34.27$ with standard errors 5.18 and 2.91 respectively. We can test the hypothesis $H_0 : \beta_1 = 0$ by comparing $z = \frac{\hat{\beta}_1}{\text{se}(\hat{\beta}_1)} = 11.77$ with a standard normal distribution. This is called a *Wald* test. Under H_0 the probability of observing this value or an even more extreme one is very small (less than 2×10^{-16} as can be seen from the p-value in the output), suggesting that it is unlikely that the data came from a distribution with $\beta_1 = 0$. In other words, the dose coefficient is significant in the model.

The same conclusion can be reached if we compare the residual deviance (this is the deviance for the model with dose included as a predictor) and the null deviance (this is the deviance for the model with just an intercept term in it). The difference in deviances is $284.202 - 11.232 = 272.97$ which is much larger than the 95th percentile of a $\chi^2(7-6) = \chi^2(1)$ distribution:

```
qchisq(df=1, p=0.95)
```

```
[1] 3.841459
```

so we once again conclude that including `dose` in the model is worthwhile.

The value of the deviance for this model is $D = 11.23$. If the model is a good fit for the beetle data the deviance should approximately follow the $\chi^2(8-2) = \chi^2(6)$ distribution. The degrees of freedom are determined as the number of distinct covariate patterns in the data (in this case doses, thus 8) minus the number of parameters in the model (intercept and dose coefficient, thus 2). The 95th percentile of the $\chi^2(6)$ distribution is

```
qchisq(df=6, p=0.95)
```

```
[1] 12.59159
```

and since $11.23 < 12.59$, we don't have evidence of lack of fit. However, we have to be careful when using the approximate chi-squared distribution as a measure of goodness of fit, because this approximation relies on having reasonably large fitted values.

For the logit model the fitted values can be obtained by taking the predicted probabilities, $\hat{p}_i$, and multiplying them by the corresponding total number of beetles for $i = 1, \ldots, 8$:

```
p.hat <- predict(m1, type="response")
fitted <- beetles$number * p.hat

cbind(beetles$killed, round(fitted,2))
```

```
  [,1]  [,2]
1    6  3.46
2   13  9.84
3   18 22.45
4   28 33.90
5   52 50.10
6   53 53.29
7   61 59.22
8   60 58.74
```

All fitted values with the exception of the first are quite large (as a rule of thumb > 5), so in this case we can say that the chi-squared approximation seems plausible.

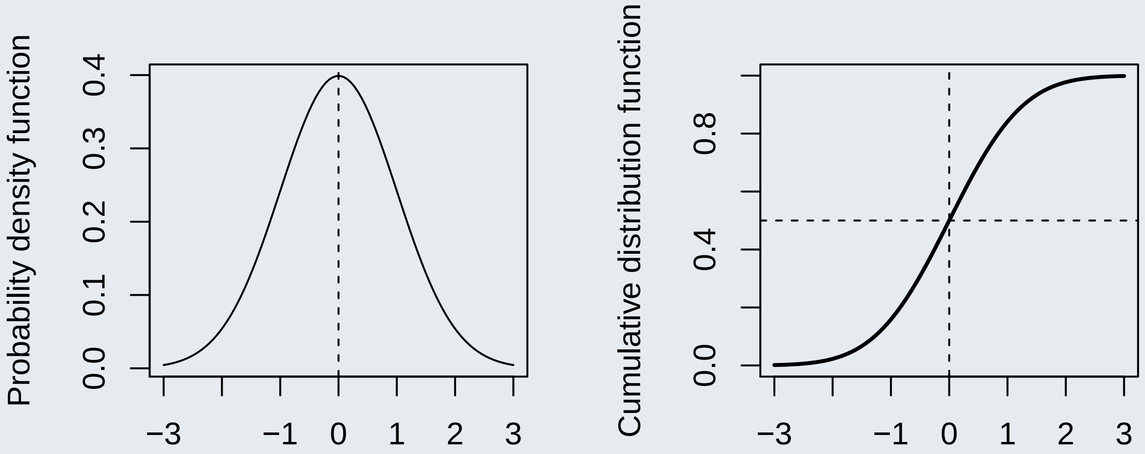
Last but not least on the logit model is the interpretation of the coefficient of dose in terms of the odds ($\frac{p_i}{1-p_i}$) of success. We usually interpret $\hat{\beta}$ in a logit model by taking $\exp(\hat{\beta})$. For the beetles this would give the *odds ratio* $\exp(\hat{\beta}_1) = \exp(34.270) = 7.643141 \times 10^{14}$. For each unit increase in dose, the odds of being killed get multiplied by this amount.

The logit link is the most commonly used link for binary/binomial data for this interpretability in terms of the odds of the outcome of interest.

However, there are situations where another link may also be suitable for a specific application. For instance here we have what is called a dose-response model in which we look at the response as a function of increasing doses of a toxic substance. In this setting, it may be quite natural to consider the *probit* link,

$$g(p_i) = \Phi^{-1}(p_i) = \beta_0 + \beta_1 x_i,$$

where Φ denotes the cumulative distribution function of the standard normal distribution. As a reminder, here are plots of the probability density function (p.d.f.) and cumulative distribution function (p.d.f.) of the standard normal distribution.



We can also write this model as $p_i = \Phi\left(\frac{x_i-\mu}{\sigma}\right)$. Then we can see that $\beta_0 = -\frac{\mu}{\sigma}$, $\beta_1 = \frac{1}{\sigma}$.

Note that at the *median lethal dose* $x_i = \mu$, the probability p_i equals 0.5, and therefore we'd expect half of the beetles to be killed. We can fit the probit model by changing the link option in the `glm` function to `probit`:

```
m2 <- glm(beetles.mat ~ beetles$dose, family = binomial(link = 'probit'))
summary(m2)
```

```
Call:
glm(formula = beetles.mat ~ beetles$dose, family = binomial(link = "probit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5714  -0.4703   0.7501   1.0632   1.3449

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -34.935      2.648  -13.19   <2e-16 ***
beetles$dose   19.728      1.487   13.27   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 284.20  on 7  degrees of freedom
Residual deviance:  10.12  on 6  degrees of freedom
AIC: 40.318

Number of Fisher Scoring iterations: 4
```

From the output we get the estimates $\hat{\beta}_1 = -34.93$ and $\hat{\beta}_2 = 19.72$ with standard errors 2.65 and 1.49 respectively. These differ from the coefficient estimates in the logit model because the model equation is totally different between the two. The interpretation of the coefficients also differs. We are still able to conduct hypothesis tests for the significance of the dose coefficient (small p-value, hence significant), and a goodness-of-fit test based on the residual deviance ($D = 10.12 < 12.59$ so no evidence of lack of fit). As the deviance is slightly lower than that of the logit model, we can say that the fit is better for the probit model, but the difference is rather small.

Finally, a third choice of link that we could consider is the *complementary log-log* link, with the GLM equation given by

$$g(p_i) = \log[-\log(1-p_i)] = \beta_0 + \beta_1 x_i.$$

Fitting this model in R is just a matter of specifying the link as follows:

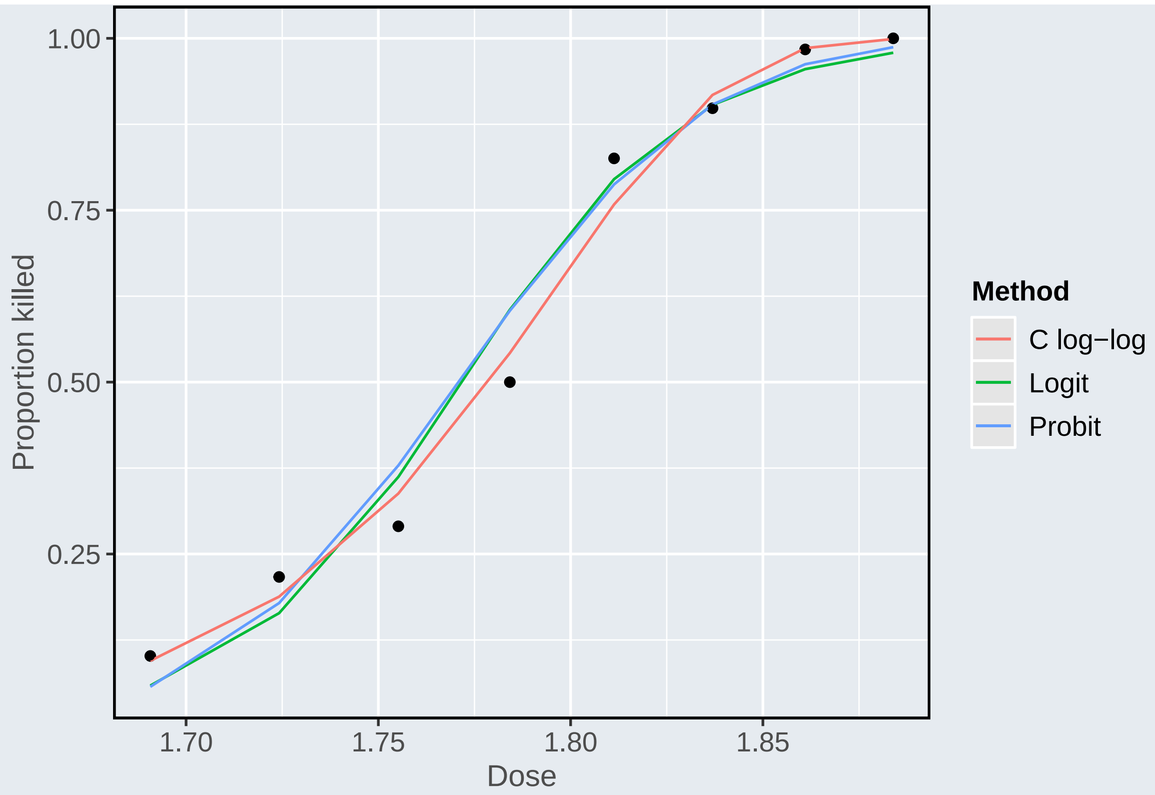
```
m3 <- glm(beetles.mat ~ beetles$dose, family = binomial(link = 'cloglog'))
summary(m3)
```

```
Call:
glm(formula = beetles.mat ~ beetles$dose, family = binomial(link = "cloglog"))

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-0.80329  -0.55135   0.03089   0.38315   1.28883  

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept)   -39.572      3.240  -12.21   <2e-16 ***
beetles$dose   22.041      1.799   12.25   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 284.2024  on 7  degrees of freedom
Residual deviance:   3.4464  on 6  degrees of freedom
AIC: 33.644

Number of Fisher Scoring iterations: 4
```

The parameter estimates are $\hat{\beta}_1 = -39.57$ and $\hat{\beta}_2 = -22.04$ with standard errors 3.24 and 1.80 respectively. The deviance is $D = 3.45$ which is quite a bit smaller than the deviances obtained with the other two link functions.

We can plot the fitted curves (on the probability scale) for each of the three regression models as follows:

```
beet_p <- data.frame(beetles = beetles,
                     logit = fitted(m1),
                     probit = fitted(m2),
                     cloglog = fitted(m3))

p2 <- ggplot(beet_p, aes(x = beetles$dose, y = beetles$propkilled)) +
      geom_point() + xlab("Dose") + ylab("Proportion killed") +
      geom_line(aes(x = beetles$dose, y = logit, colour = "Logit")) +
      geom_line(aes(x = beetles$dose, y = probit, colour = "Probit")) +
      geom_line(aes(x = beetles$dose, y = cloglog, colour = "C log-log")) +
      guides(colour = guide_legend("Method"))
```



We see that all three links give a good fit, with the complementary log-log being the best (although in practice we rarely choose the link based on fit: for one thing, the logit and probit are symmetric and can often be quite similar to each other, and for another, we tend to like the interpretability of the logit link and stick with it most of the time).

Now let us look at another example for which you will do all the work.

Example 2 (Challenger disaster).

In January 1986, the space shuttle Challenger exploded shortly after launch. It was subsequently found that the rubber O-ring seals in the rocket boosters were susceptible to breaking in low temperatures. At the time of the launch the temperature was 31 degrees Fahrenheit. Could the failure of the O-rings have been predicted? Data from the previous 23 missions shows some evidence of damage on some of the six O-rings on each shuttle, as well as the temperature during the shuttle launch. The data is available from `library(faraway)` and is called `orings`. The first column of the data gives the temperature at launch in degrees F and the second column gives the number of damage incidents out of 6 possible.

Here are the first few rows of the data:

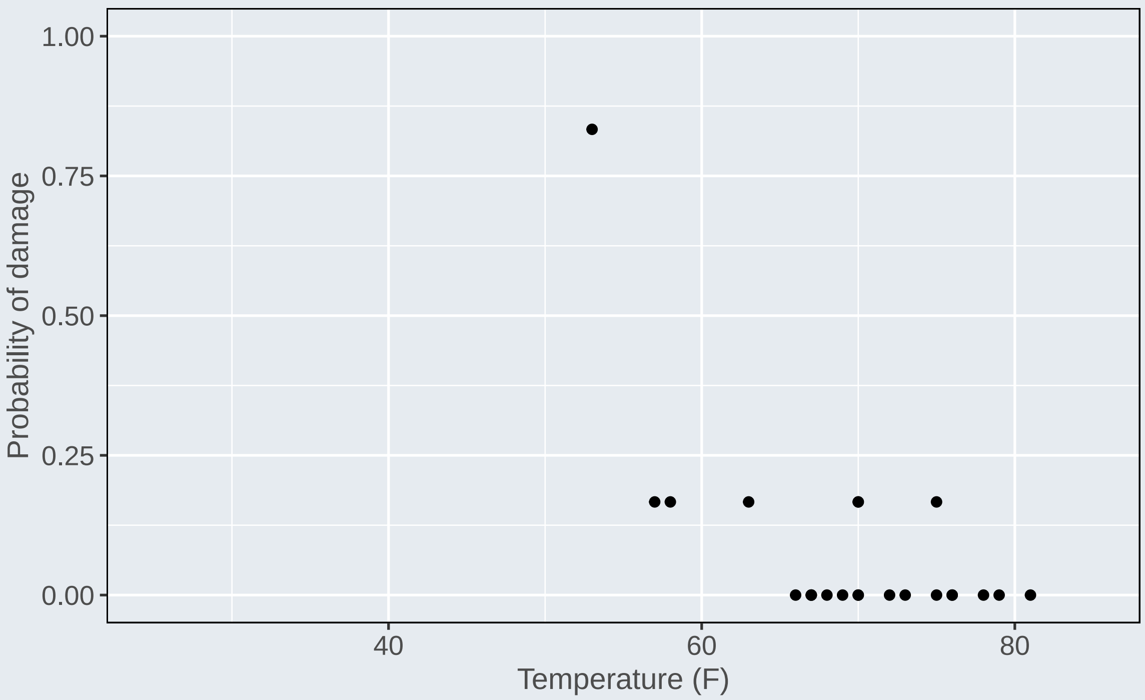
```
library(faraway)
head(orings)
```

```
  temp damage
1   53      5
2   57      1
3   58      1
4   63      1
5   66      0
6   67      0
```

We wish to use as our predictor x_i the temperature (in degrees F) during launch for the ith mission, $i = 1, \dots, 23$, with the response y_i being the number of damaged O-rings (out of 6 total). The model for the probability p_i of damage to the O-rings is $Y_i \overset{indep}{\sim} \text{Bin}(n, p_i)$ with $g(p_i) = \beta_0 + \beta_1 x_i$. Here $n = 6$.

Here is a plot of the data:

```r
p1<- ggplot(orings, aes(x=temp, y=damage/6)) +
     geom_point()+ xlim (c(25,85)) + ylim(c(0,1)) +
     xlab ("Temperature (F)") + ylab("Probability of damage")
```



Task 1.

Fit a binomial regression model to the data, trying out the logit, probit and complementary log-log options for the link function.

Task 2.

Superimpose the fitted probabilities from each of the three models on the above plot.

Task 3.

Calculate a point estimate of the probability of damage to the O-rings when the temperature is 31 degrees Fahrenheit using each of the three models.

Binary response

Now let us turn our attention to models with a binary response $Y_i \sim \text{Bin}(1, p_i)$. We will go over the main ideas through examples, starting with a light-hearted one.

Logistic regression model

```
fit <- glm(hear ~ age, family=binomial, data=yl)
summary(fit)

Coefficients:
            timate Std. Error z value Pr(>|z|)
(Int        51874    1.21032   1.255     0.21
ag          4812     0.03423  -1.406     0.16

            nce: 71.779  on 51  degrees of freedom
Residu      ance: 69.586  on 50  degrees of freedom
  (1 observation deleted due to missingness)
```

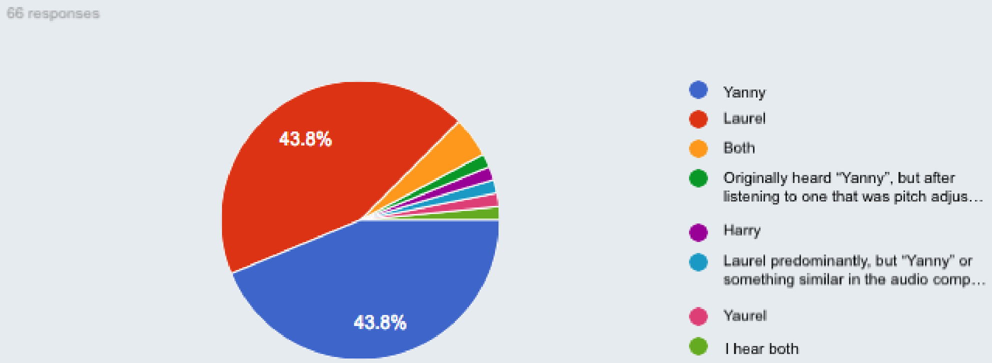
Yanny or Laurel?

https://youtu.be/iNJkAER2ZPg

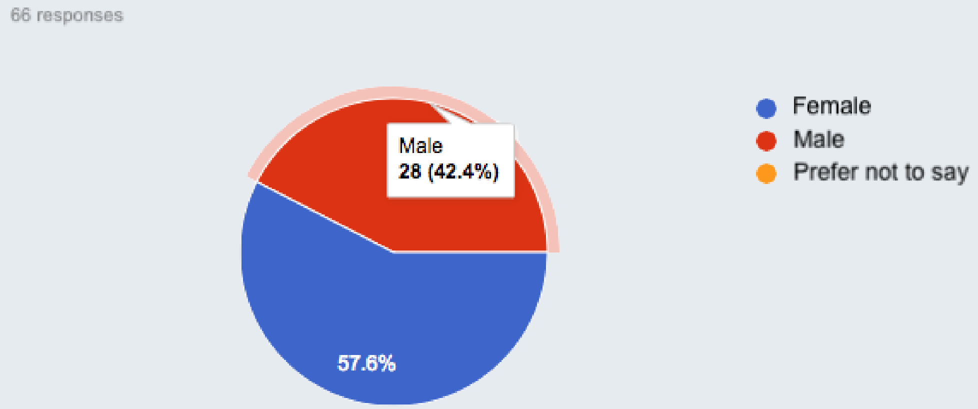
Duration: 3m19s

Example 3 (Yanny or Laurel?).

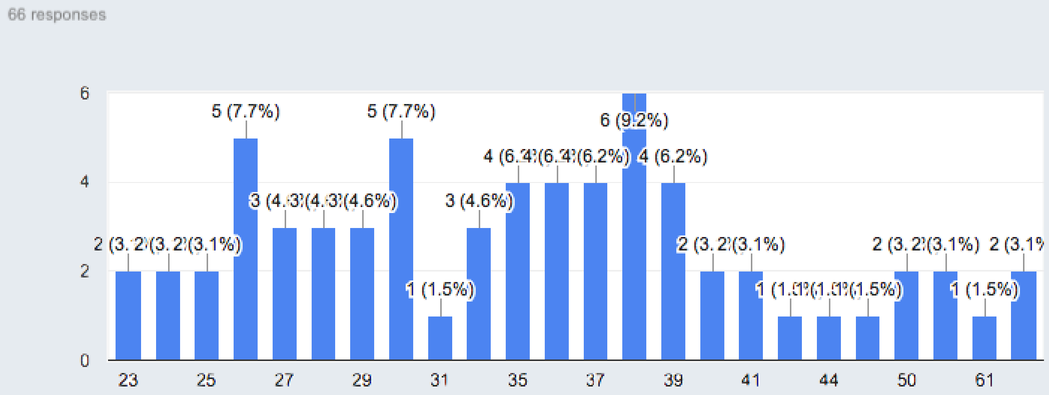
This auditory illusion first appeared on the internet in May 2018. An explanation of why people hear different things can be found in this short video, just one of many internet sources discussing the phenomenon. The main reason behind the difference appears to be that as we age we lose the ability to hear certain sounds. To see if we could find evidence of such an age effect, we asked people (mainly students on the online MSc programme, and staff and PhD students at the School of Mathematics and Statistics at the University of Glasgow) to fill out a survey on what they hear. Below you can see summaries of the first 66 responses.



What is your gender?



What is your age?



The proportions hearing "Yanny" and "Laurel" are very similar to each other, and there are some respondents who hear both or even something completely different. This may be because people do not listen to the audio file using the same device, something we couldn't control for in our online survey. Ignoring for the time being the responses that list something other than just "Yanny" or just "Laurel", we have 53 observations left. Here are the first few rows of the data:

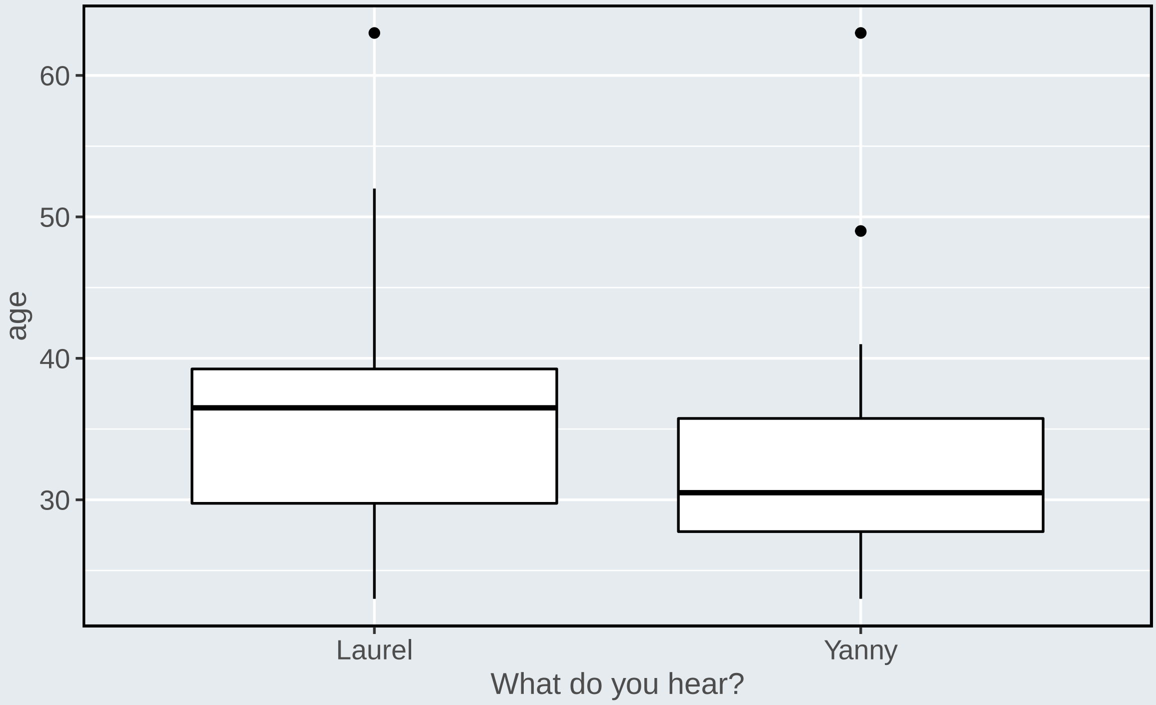
```
yl<- read.csv(url("http://www.stats.gla.ac.uk/~tereza/rp/yl53.csv"))
head(yl)
```

```
  X   hear gender age
1 1 Laurel Female  28
2 2 Laurel   Male  39
3 3 Laurel Female  25
4 4  Yanny Female  25
5 5 Laurel Female  36
6 6 Laurel Female  24
```

```
yl$hear <- factor(yl$hear)
```

For exploratory plots we can consider a boxplot for age, the continuous covariate, and a bar chart for gender, the categorical covariate.

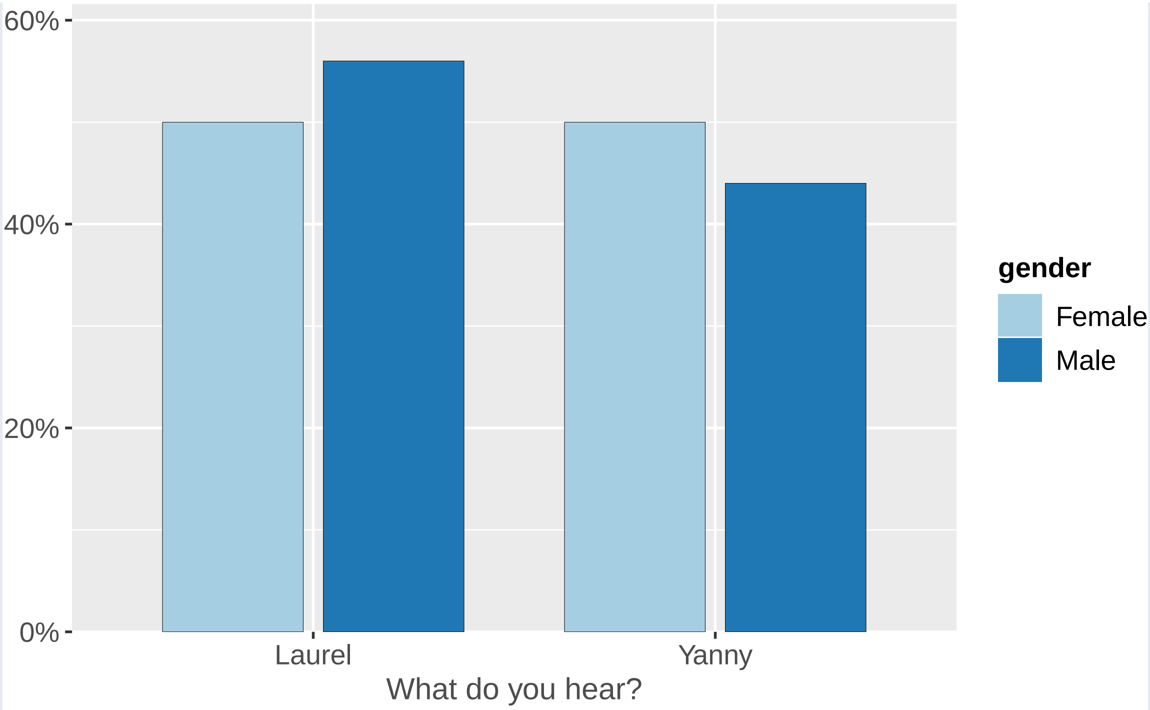
```
yl.plot1 <- ggplot(yl, aes(y=age, x=hear))

yl.plot1 + geom_boxplot()+ xlab("What do you hear?") +
           theme(panel.background = element_rect(fill = "transparent", colour = NA),
           plot.background = element_rect(fill = "transparent", colour = NA),
           panel.border = element_rect(fill = NA, colour = "black", size = 1))
```



We see in the boxplot that the people who hear "Yanny" are younger on average, but that there is a substantial overlap between the two.

The plot of the proportions against gender is shown below. There is a slightly smaller proportion of men hearing "Yanny", but the proportions look very similar overall.

```
library(sjPlot)
plot_xtab(yl$hear,yl$gender, show.values = FALSE, show.total = FALSE,
         axis.labels = c("Laurel", "Yanny"),
         axis.titles=c("What do you hear?"))
```



Let us look at a logistic regression model with age as the explanatory variable. Here $Y_i = 1$ if the ith respondent heard "Yanny" and $Y_i = 0$ if the ith respondent heard "Laurel", with x_i being the respondent's age for $i = 1, \ldots, 53$. The model we will consider is of the form

$$g(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_i$$

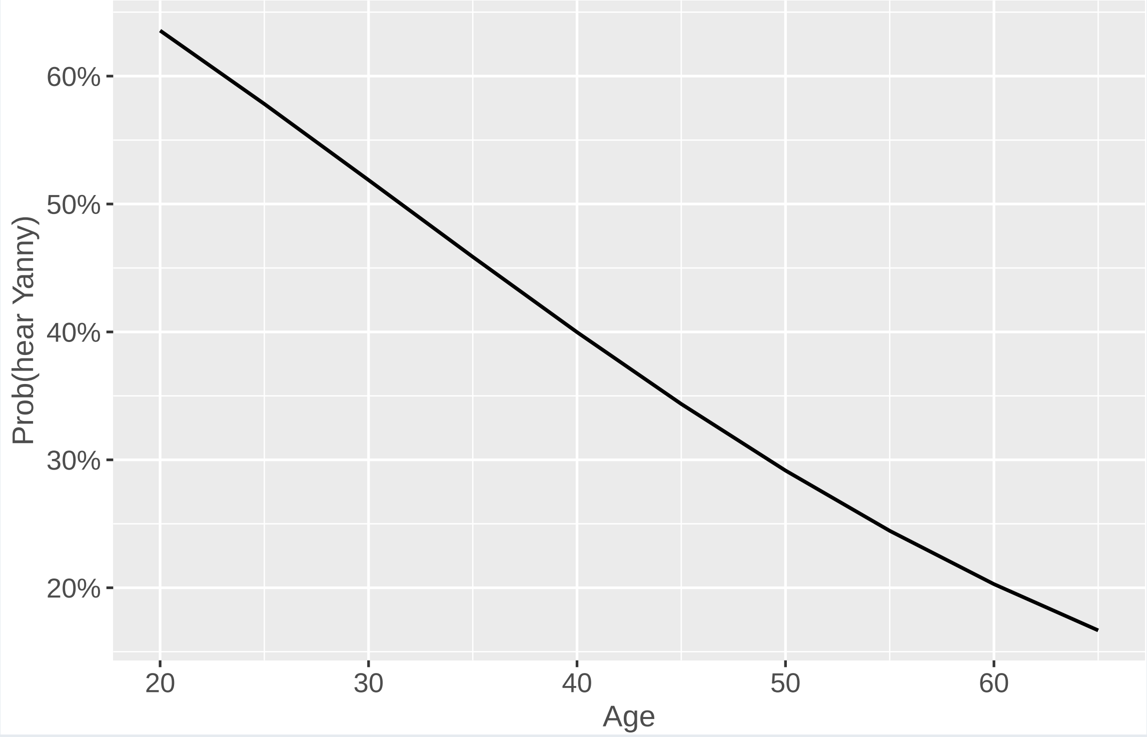
and we fit it in R as follows:

```
mod.yl <- glm(hear ~ age, family=binomial, data=yl)
summary(mod.yl)
```

```
Call:
glm(formula = hear ~ age, family = binomial, data = yl)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3566  -1.0881  -0.7971   1.1457   1.8505

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.51874    1.21032   1.255     0.21
age         -0.04812    0.03423  -1.406     0.16

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 71.779  on 51  degrees of freedom
Residual deviance: 69.586  on 50  degrees of freedom
  (1 observation deleted due to missingness)
AIC: 73.586

Number of Fisher Scoring iterations: 4
```

Notice that the age coefficient is negative, suggesting that older people are less likely to hear "Yanny", but that this coefficient is not significant (p-value of 0.16 greater than 0.05, 95% confidence interval of $-0.04812 \pm 1.96 \times 0.03423 = (-0.0115, 0.019)$ includes zero). Still, if we wanted to use the estimated coefficient to quantify the effect of age, we would need to look at `exp(-0.04812)=0.953`. This suggests that for two people who differ by one year in age, the older person's odds of hearing "Yanny" are 0.953 times those of the younger person. And the odds of hearing "Laurel" get multiplied by a factor of exp(0.04812)=1.049. If we look at a ten-year age difference the odds multiplier becomes exp(0.04812*10)=1.618. For two people who differ by ten years in age, the older person's odds of hearing "Laurel" are 1.618 times those of the younger person. Finally we can plot the predicted probabilities from this model as a function of age and, as expected, we see that the predicted probability of hearing "Yanny" decreases with age:

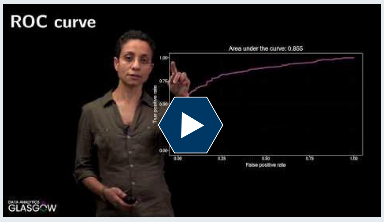
```
library(sjPlot)
plot_model(mod.yl,type="pred",terms=c("age"), axis.title=c("Age", "Prob(hear Yanny)"),
           title="", ci.lvl=NA)
```



Task 4.

Fit appropriate logistic regression models to explore if gender is related to whether people hear "Yanny" or "Laurel".

For our second example of a binary logistic regression model, let us look at another famous disaster, the sinking of the Titanic.



A logistic regression model for predicting which of the passengers of the Titanic were more likely to survive

https://youtu.be/q6gaSm-7sXE

Duration: 8m04s

Example 4 (Titanic).

On 15th April 1912, during its maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

Our goal is to build a model to predict the survival of a passenger based on information about the passenger's age, gender and ticket class. Here are the first few rows of the data:

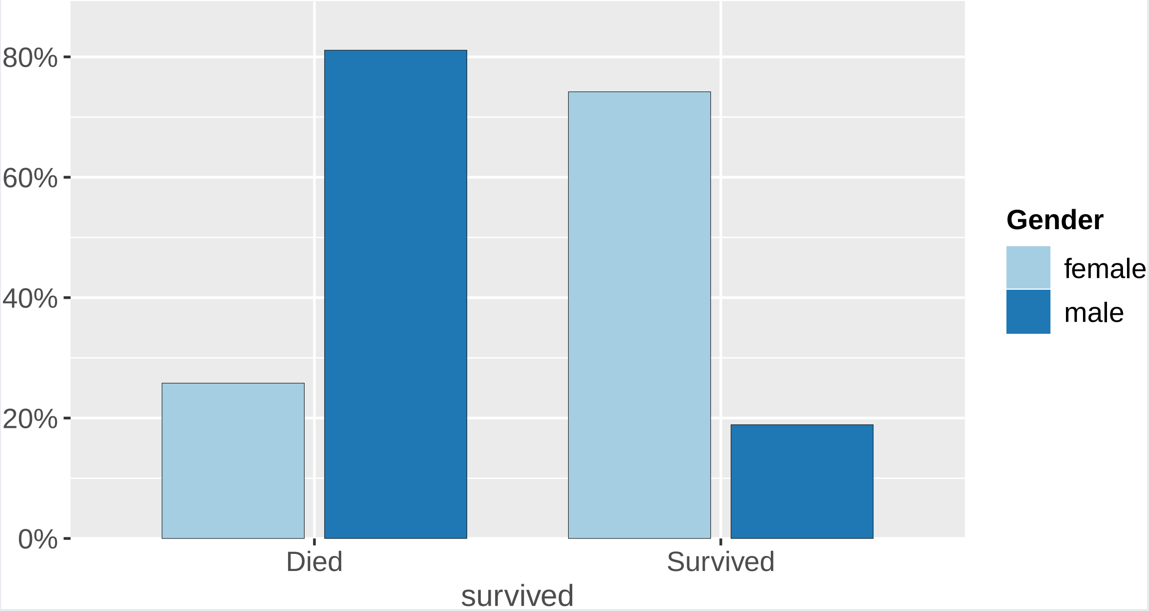
```
titanic <- read.csv(url("http://www.stats.gla.ac.uk/~tereza/rp/titanic.csv"))
titanic$passenger.class <- factor(titanic$passenger.class)
head(titanic)
```

```
  X survived passenger.class gender      age siblings.spouses.aboard parents.children.aboard fa
1 1        0               3   male 22.00000                       1                       0
2 2        1               1 female 38.00000                       1                       0
3 3        1               3 female 26.00000                       0                       0
4 4        1               1 female 35.00000                       1                       0
5 5        0               3   male 35.00000                       0                       0
6 6        0               3   male 29.69912                       0                       0
```

The response variable Y_i is the survival status for $n = 891$ passengers, taking value 1 for `survived` and 0 for `died`. Predictors include the passenger's ticket class, gender, age, fare, number of relatives on board and so on. We assume that $Y_i \overset{indep}{\sim}$ Bin$(1, p_i)$ where p_i is the probability of survival for the ith passenger. We fit a logistic regression model of the form $g(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \mathbf{x}_i^T \boldsymbol{\beta}$, where $\mathbf{x}_i$ is the vector of covariates for the ith passenger.

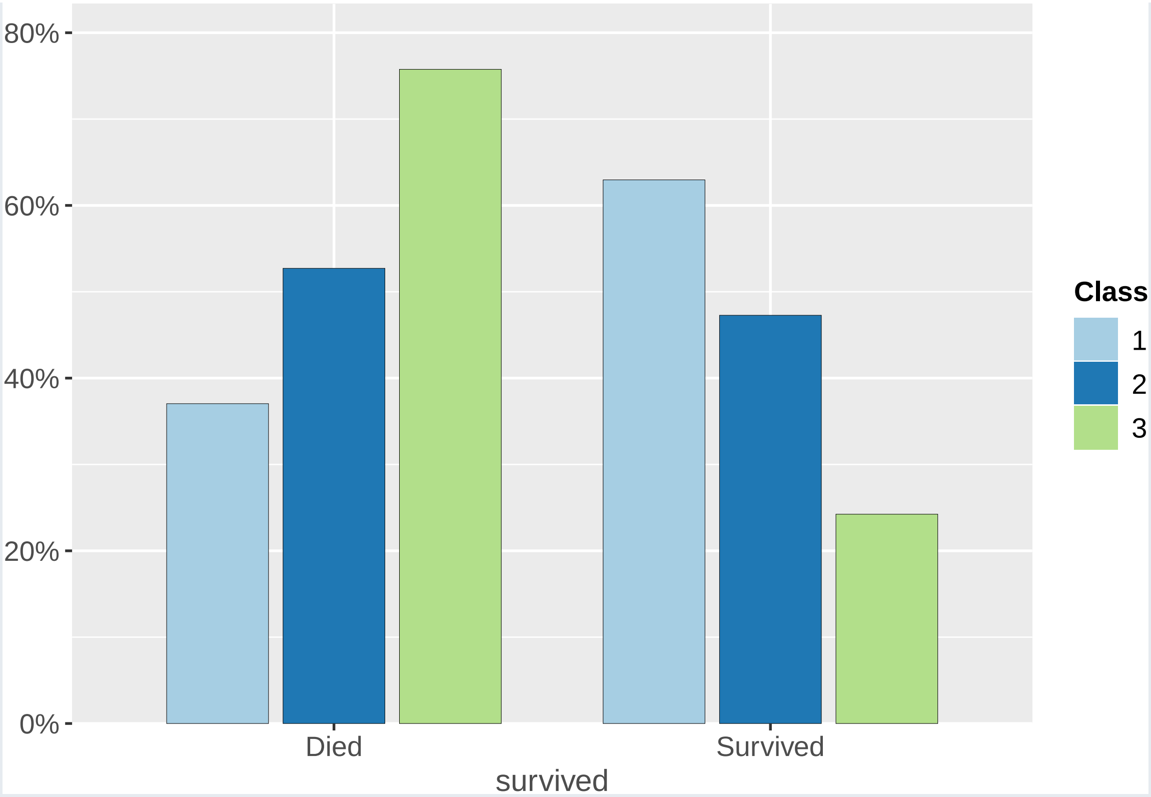
First, let us look at some exploratory plots:

```
plot_xtab(titanic$survived,titanic$gender, show.values = FALSE,
         show.total = FALSE, axis.labels = c("Died", "Survived"),
         legend.title = "Gender")
```



There is a clear pattern here with the proportion surviving much higher for women than for men.

```
plot_xtab(titanic$survived,titanic$passenger.class, show.values = FALSE,
         show.total = FALSE, axis.labels = c("Died", "Survived"),
         legend.title = "Class")
```



The largest group amongst the passengers who died were third class passengers while amongst those who survived the largest group was first class passengers.

Now let's fit a model with age, gender and passenger's ticket class as predictors:

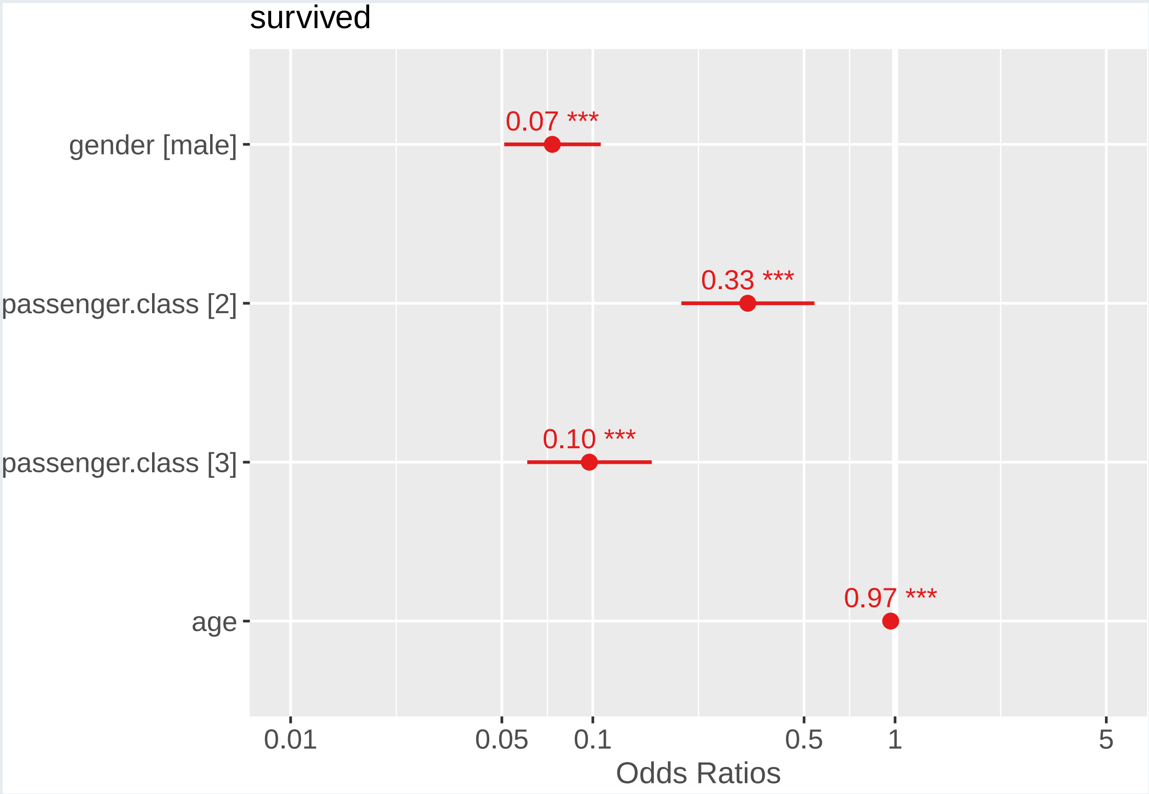
```
mod.titan <- glm(survived~gender + passenger.class + age,
                 family=binomial(link="logit"), data=titanic)
summary(mod.titan)
```

```
Call:
glm(formula = survived ~ gender + passenger.class + age, family = binomial(link = "logit"),
    data = titanic)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6490  -0.6636  -0.4198   0.6328   2.4283

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       3.54474    0.36537   9.702  < 2e-16 ***
gendermale       -2.61131    0.18671 -13.986  < 2e-16 ***
passenger.class2 -1.12216    0.25773  -4.354 1.34e-05 ***
passenger.class3 -2.32917    0.24089  -9.669  < 2e-16 ***
age              -0.03330    0.00737  -4.519 6.21e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1186.66  on 890  degrees of freedom
Residual deviance:  805.29  on 886  degrees of freedom
AIC: 815.29

Number of Fisher Scoring iterations: 5
```

We see from the output that the coefficient for males is negative, indicating a lower chance of survival for male passengers. Similarly the coefficients for second and third class are negative, with the magnitude of the third class coefficient larger than that of the second class coefficient, suggesting that second class passengers had a worse chance of survival than first class passengers, and that third class passengers had an even worse chance. Finally the age coefficient is also negative, suggesting that older people were less likely to survive.

To quantify the effect of each of these predictors, we look at *odds ratios* which can be computed as $\exp(\hat{\beta})$. These are shown in the plot below.

```
plot_model(mod.titan, show.values=TRUE)
```



We interpret the odds ratios as follows: men's odds of survival were 0.07 times those of women, third class passengers' odds of survival were 0.10 times those of first class passengers, and second class passengers' odds of survival were 0.33 times those of first class passengers. Finally, for each year increase in the passenger's age, the odds of survival decrease (get multiplied by a factor of 0.97).

Note that the plot also includes confidence intervals for the odds ratios. To illustrate how these are calculated, let's take the coefficient of gender as an example:

An approximate 95% confidence interval for the gender coefficient (this is on the *log odds* scale, hence a *log odds ratio*) is:

$$-2.61131 \pm 1.96 \times 0.18671 = (-2.977, -2.245).$$

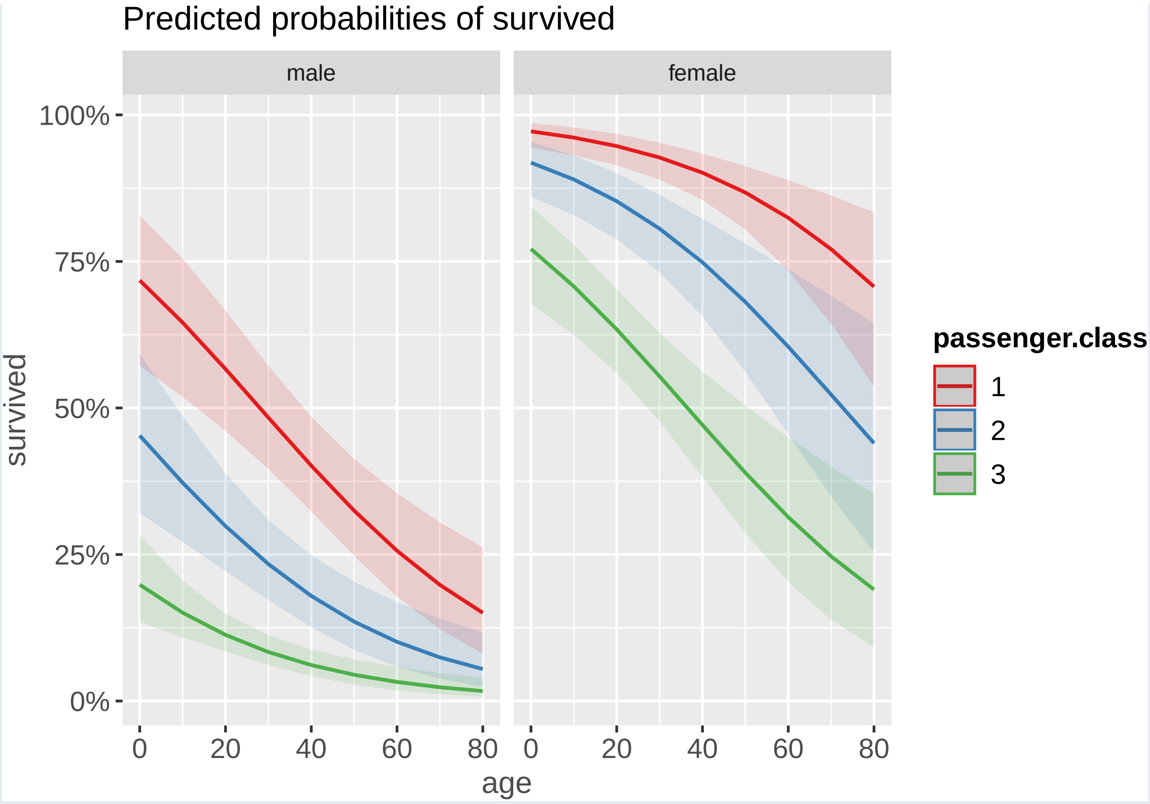
The corresponding interval on the *odds scale* is obtained by exponentiating the endpoints:

$$(\exp(-2.977), \exp(-2.245)) = (0.051, 0.106).$$

Thus the *odds ratio* comparing men to women is between 0.05 and 0.10 (point estimate of 0.07): the odds of survival for men are between 0.05 and 0.10 times the odds for women.

We can also plot the predicted probabilities of survival against the passenger's age by the passenger's gender and ticket class. The plot also shows pointwise confidence intervals for the predicted probabilities.

```
plot_model(mod.titan,type="pred",terms=c("age","passenger.class", "gender"))
```



We see the gender and class differences in survival we have already discussed, and also that survival probabilities decrease by age.

Probabilities, odds, odds multipliers and odds ratios

In logit models, we interpret coefficients in terms of the odds, and terms involving the word “odds” inevitably come up when describing the model fit. Here we present all of these terms in the same place and describe the relationships between them.

The odds are defined as $\text{Odds} = \dfrac{p}{1-p}$ where p is the probability of the outcome of interest. We can express the probability in terms of the odds as $p = \dfrac{\text{Odds}}{\text{Odds}+1}$.

In logistic regression we model the **log odds**: $\log(\text{Odds}) = \log\left(\dfrac{p}{1-p}\right) = \mathbf{x}^T\boldsymbol{\beta}$.

The β coefficients are **log odds ratios**: Suppose we have a predictor with two levels, say `gender` in the Titanic example, which is coded 1 for men and 0 for women.

This means that the gender coefficient is the difference between $\log(\text{Odds}_2) = \log\left(\frac{p_2}{1-p_2}\right)$ (the log odds for men) and $\log(\text{Odds}_1) = \log\left(\frac{p_1}{1-p_1}\right)$ (the log odds for women).

And since $\log(\text{Odds}_2) - \log(\text{Odds}_1) = \log\left(\dfrac{\text{Odds}_2}{\text{Odds}_1}\right)$, the gender coefficient is equal to the log odds ratio:

$$\beta = \log\left(\frac{\text{Odds}_2}{\text{Odds}_1}\right) = \log\left(\frac{\frac{p_2}{1-p_2}}{\frac{p_1}{1-p_1}}\right).$$

By exponentiating both sides we see that $\exp(\beta)$ is the **odds ratio** for comparing the two levels of the predictor (here men and women) in terms of the **odds** of the outcome of interest.

And since we can express this as $\text{Odds}_2 = \exp(\beta) \times \text{Odds}_1$, we also call $\exp(\beta)$ the **odds multiplier**.

For the Titanic example, a year increase in age is associated with multiplying the odds of survival by a facor of $\exp(-0.03330) = 0.97$.

If the explanatory variable x in the model is continuous rather than a factor, the odds multiplier gives the effect of an increase of one unit in x on the odds of the outcome of interest.

The following video, in which Prof. David Spiegelhalter talks about odds ratios and their interpretation, may be of further use in clarifying these concepts.

Prof. David Spiegelhalter on odds ratios.

https://youtu.be/ixKhS0SIlb4

Duration: 7m03s

Model checking and diagnostics for logistic regression

We saw that the deviance, D, is one possible goodness-of-fit statistic for GLMs. Another one is the Pearson chi-squared statistic.

Definition 1 (Pearson' s chi-squared statistic).

Pearson's chi-squared statistic is defined as

$$X^2 = \sum_{i=1}^{n} \frac{(y_i - n_i\hat{p}_i)^2}{n_i\hat{p}_i(1-\hat{p}_i)}, \quad i = 1, \ldots, n$$

where y_i represents the observed number of successes, n_i is the number of trials and$\hat{p}_i$ for the ith covariate pattern.

Theorem 1 (Sampling/asymptotic distribution of X^2).

X^2 is asymptotically equivalent to the deviance. Therefore, under H_0: the model fits the data well, X^2 is approximately distributed as $\chi^2(n-p)$ where n is the number of parameters in the saturated model (usually equal to the number of observations), and p is the number of parameters in the model of interest. This results holds for relatively large fitted values.

Example 5 (Beetle data, revisited).

Suppose that we would like to assess the fit of the logistic model in the beetle mortality example seen earlier. The data and fitted values obtained from the logit model were as follows.

x_i	n_i	y_i	$\hat{y}_i = n_i\hat{p}_i$
1.6907	59	6	3.46
1.7242	60	13	9.84
1.7552	62	18	22.45
1.7842	56	28	33.90
1.8113	63	52	50.10
1.8369	59	53	53.29
1.8610	62	61	59.22
1.8839	60	60	58.74

We wish to test H_0: the model fits the data well against H_1: the model does not fit the data well. Instead of using maximum likelihood we could estimate the model parameters by minimising the weighted sum of squares

$$S_w = \sum_{i=1}^{n} \frac{(y_i - n_i p_i)^2}{n_i p_i (1 - p_i)}$$

since $E(Y_i) = n_i p_i$ and $\mathrm{Var}(Y_i) = n_i p_i (1 - p_i)$. This turns out to be equivalent to minimising the Pearson chi-squared statistic since

$$\begin{aligned} X^2 &= \sum_{i=1}^{n} \frac{(y_i - n_i p_i)^2}{n_i p_i} + \sum_{i=1}^{n} \frac{[(n_i - y_i) - n_i(1 - p_i)]^2}{n_i(1 - p_i)} \\ &= \sum_{i=1}^{n} \frac{(y_i - n_i p_i)^2}{n_i p_i (1 - p_i)} (1 - p_i + p_i) = S_w \end{aligned}$$

When X^2 is evaluated at the estimated expected frequencies, the statistic is

$$X^2 = \sum_{i=1}^{n} \frac{(y_i - n_i \hat{p}_i)^2}{n_i \hat{p}_i (1 - \hat{p}_i)}$$

which will be approximately $\chi^2(n - p)$ if the model is true.

For the beetle mortality example, $D = 11.23$ and $X^2 = 10.03$. Both are relatively large compared with a $\chi^2(6)$ distribution, but not greater than the 95th percentile of the $\chi^2(6)$ distribution:

```
qchisq(p=0.95, df=6)
```

```
[1] 12.59159
```

Therefore there is no evidence of lack of fit for the logit model.

Note: The chi-squared approximation for the deviance and X^2 relies on having expected frequencies (fitted values) that are not too small. It should be ok to use for checking the fit of the beetle mortality model, but if each observation has a different covariate pattern so that y_i is either 0 or 1, as in the Yanny-Laurel example, then neither D nor X^2 provides a useful measure of goodness of fit. For this reason, a modification of the chi-squared test has been proposed.

Hosmer-Lemeshow goodness of fit test

Suppose that for a model for binary responses we wish to test H_0: the model fits the data well (observed and expected response frequencies are close to each other) versus H_1: the model is not a good fit for the data (observed frequencies are far from expected frequencies). The Hosmer-Lemeshow test statistic is calculated as follows:

1. Order the fitted values
2. Group the fitted values into g classes (where g is between 6 and 10) of roughly equal size.
3. Calculate the observed and expected number in each group
4. Perform a chi-squared goodness-of-fit test, with $\chi^2(g - 2)$ as the reference distribution.

Example 6 (Yanny-Laurel revisited).

Recall the Yanny-Laurel example we saw earlier, with the model of interest predicting the probability of hearing "Yanny" as a function of the age of the participant.

We can use an implementation of the Hosmer-Lemeshow test to check for evidence of lack of fit in the model.

```
source(url("http://www.chrisbilder.com/categorical/Chapter5/AllGOFTests.R"))
HLTest(mod.yl, g=10)
```

```
Warning in HLTest(mod.yl, g = 10): Some expected counts are less than 5. Use smaller number of

	Hosmer and Lemeshow goodness-of-fit test with 10 bins

data:  mod.yl
X2 = 10.026, df = 8, p-value = 0.2632
```

The large p-value indicates no lack of fit. To make sure this is not just due to the choice of g, we try a few more values, *e.g.* here

```
HLTest(mod.yl,g=6)
```

```
Warning in HLTest(mod.yl, g = 6): Some expected counts are less than 5. Use smaller number of g

    Hosmer and Lemeshow goodness-of-fit test with 6 bins

data:  mod.yl
X2 = 4.1333, df = 4, p-value = 0.3883
```

Note the warning that some expected counts are less than 5. This suggests that the chi-squared approximation may not be very reliable for these data.

Notes on the Hosmer-Lemeshow test:

- Failing to reject H_0 does not mean that the fit is good.
- The power of the test can be too small to detect lack of fit.
- How the fitted values are grouped together matters – use different values of g and see if that changes the conclusion.
- Other tests can be used in addition to the Hosmer-Lemeshow test, see http://www.chrisbilder.com/categorical/Chapter5/AllGOFTests.R for some.

Other diagnostics/model checking for binomial models

Likelihood ratio chi-squared statistic

The likelihood ratio chi-squared statistic is defined as twice the difference in maximised log-likelihood under the model of interest and under the null (minimal) model. Under the null model $\tilde{p} = \sum y_i / \sum n_i$. Let $\hat{p}$ be the MLE under the model of interest. Then

$$C = 2[l(\hat{\mathbf{p}};\mathbf{y}) - l(\tilde{\mathbf{p}},\mathbf{y})]$$

should be approximately $\chi^2(p-1)$ if all the p parameters except the intercept β_0 are zero.

We have already used this in models of the form $g(\mu) = \beta_0 + \beta 1x$ to test $H_0 : \beta_1 = 0$.

AIC and BIC

The **Akaike information criterion (AIC)** and the Schwartz or **Bayesian information criterion** (BIC) are other goodness-of-fit statistics based on the log-likelihood function with adjustment for the number, p, of parameters estimated.

$$AIC = -2l(\hat{\mathbf{p}};\mathbf{y}) + 2p$$

$$BIC = -2l(\hat{\mathbf{p}};\mathbf{y}) + 2p \times \log(\text{number of observations})$$

A small value of these statistics indicates that there is no lack of fit in the model. These statistics could be used as model selection criteria, especially when the models under comparison are not nested.

Residuals

There are two main forms of residuals for logistic regression: **deviance** and **Pearson** (or **chi-squared**) residuals. These are the contributions to D and X^2 respectively of each distinct covariate pattern. Suppose there are m distinct covariate patterns and that Y_k, n_k and $\hat{p}_k$ are the number of successes, the number of trials and the estimated probability of success for the kth covariate pattern.

Definition 2 (Pearson residuals).

The Pearson or chi-squared residual is

$$X_k = \frac{y_k - n_k \hat{p}_k}{\sqrt{n_k \hat{p}_k (1 - \hat{p}_k)}}.$$

The standardised Pearson residual is

$$r_{Pk} = \frac{X_k}{\sqrt{1 - h_k}},$$

where h_k is the leverage which is obtained from the hat matrix.

Definition 3 (Deviance residuals).

The deviance residual is

$$d_k = \text{sign}(y_k - n_k \hat{p}_k) \times \left\{2\left[y_k \log\left(\frac{y_k}{n_k \hat{p}_k}\right) + (n_k - y_k)\log\left(\frac{n_k - y_k}{n_k - n_k \hat{p}_k}\right)\right]\right\}^{1/2}.$$

The standardised deviance residual is

$$r_{Dk} = \frac{d_k}{\sqrt{1 - h_k}}.$$

The residuals can be plotted against continuous covariates to check the linearity assumption, and in the order of the measurements to check for serial correlation. Normal probability plots could also be used as the residuals should be approximately $N(0, 1)$ provided the numbers of observations for each covariate pattern are not too small.

However, the residuals are not informative if the response is binary of if n_k is small for most covariate patterns. So residual plots wouldn't be useful for the Yanny-Laurel data where the outcome variable is binary and the predictor (age) is continuous, but they could be used for the beetle data.

Task 5.

Compare residual plots from the Yanny-Laurel model with those from the logit model used for the beetle data.

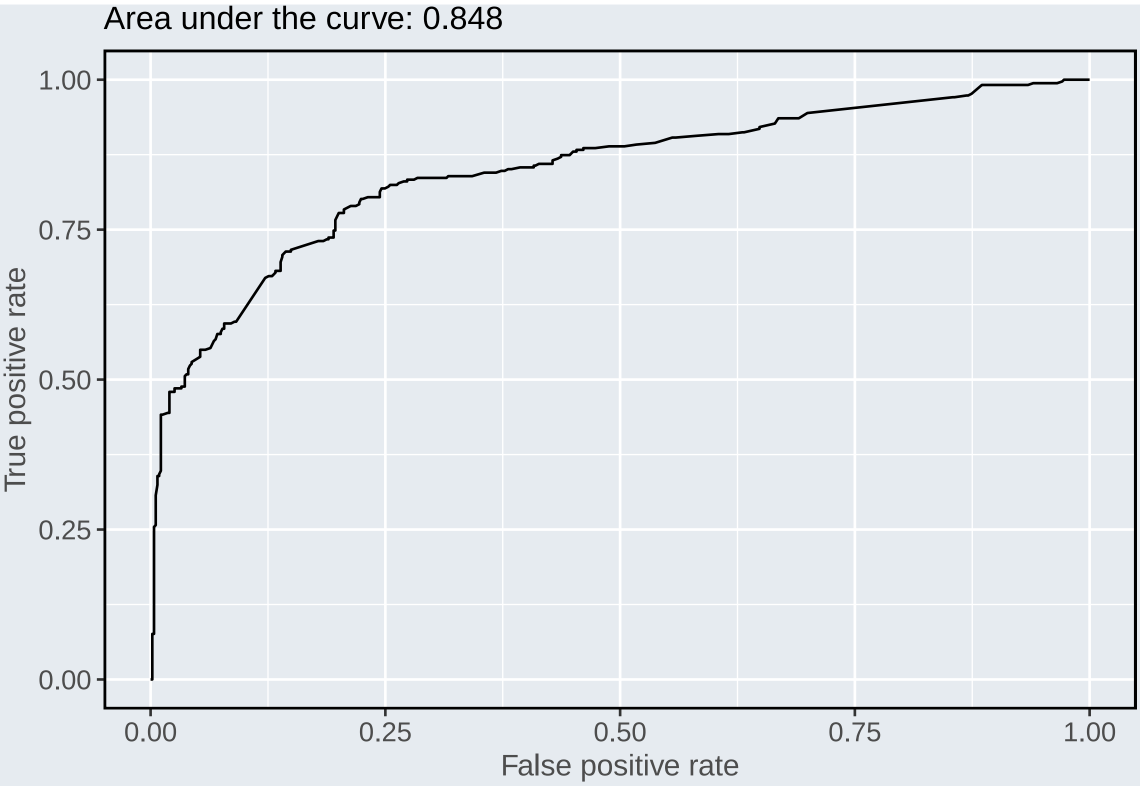
Logistic regression as a classifier

A way to assess the predictive power of a model is to look at the receiver operating characteristic (ROC) curve, which is a measure of classifier performance. Using the proportion of positive data points that are correctly predicted as positive (true positive rate) and the proportion of negative data points that are incorrectly predicted as positive (false positive rate), one can generate a graph that shows the trade off between the rate at which the model predicts the response correctly versus predicting it incorrectly. On the horizontal axis of the ROC curve we have the false positive rate and on the vertical axis the true positive rate. The area under the ROC curve, known as AUC, is used as a measure of a diagnostic test's discriminatory power. An AUC value of 0.5 indicates that the predictive model is of no discriminative value. We would like models to perform better than a random guess, so we would like the AUC to be greater than 0.5. We can also compare the ROC curves for different models to help us choose between them.

Example 7.

Here is one way to produce the ROC curve and AUC for the model fitted to the Titanic data:

```
library(ROCR)
titanic$Prid <- predict(mod.titan, titanic, type="response")
score <- prediction(titanic$Prid,titanic$survived)
perf <- performance(score,"tpr","fpr")
auc <- performance(score,"auc")
perfd <- data.frame(x= perf@x.values[1][[1]], y=perf@y.values[1][[1]])
p4<- ggplot(perfd, aes(x= x, y=y)) + geom_line() +
       xlab("False positive rate") + ylab("True positive rate") +
       ggtitle(paste("Area under the curve:", round(auc@y.values[[1]], 3)))
```



Here `perf` is the performance of the predictions from our model in terms of the true positive and false positive rates.

The area under the curve is about 0.85, which is reasonable considering that we have only used three of the predictors available in the data.

We can also change the decision boundary according to some criterion. Let's say that we want to keep the false positive rate lower than 20%, in other words we don't want to incorrectly predict that a passenger survived with a probability of more than 0.2. In that case we can find the cutoff point as follows:

```
cutoffs <- data.frame(cut=perf@alpha.values[[1]], fpr=perf@x.values[[1]],
                      tpr=perf@y.values[[1]])
cutoffs <- cutoffs[order(cutoffs$tpr, decreasing=TRUE),]
head(subset(cutoffs, fpr < 0.2))
```

```
          cut       fpr       tpr
153 0.4365077 0.1967213 0.7660819
152 0.4422038 0.1967213 0.7631579
151 0.4447161 0.1967213 0.7543860
149 0.4461146 0.1948998 0.7485380
150 0.4460822 0.1967213 0.7485380
148 0.4474317 0.1948998 0.7426901
```

Note that `library(ROCR)` is just one package that produces ROC curves. For more recent packages you can also try `library(plotROC)` or `library(ROCit)`.

As a final note, there is a lot more that one can do with the Titanic dataset. We have only used three explanatory variables and one could try to improve predictive performance by adding more terms to the model. In fact there is a prediction competition on this dataset, with the data we've used as the training set and also a test set available on which the predictions are made.

Now let's look at another example where logistic regression can help us predict an outcome of interest.

Example 8 (Trump tweets).

Donald J. Trump was the 45th President of the United States, inaugurated on 20th January, 2017. Trump was an avid user of twitter, an online news and social networking service where users post and interact with messages, called "tweets". In the beginning of his presidency there was some question as to who wrote the tweets from Trump's official twitter account @realDonaldTrump: Trump himself, or his team?

Donald Trump was known to have used a Samsung Galaxy smartphone to write his tweets, whereas his team used iOS devices. The device used to send a tweet can be retrieved using the Twitter API, so it used to be easy to determine whether a tweet was written by Donald Trump himself or his team. However, in March 2017, Donald Trump switched to an iPhone, after which point there was no way of telling whether Donald Trump had written a tweet himself.

The dataset provided contains a set of 3329 tweets from January 1st 2016 to his inauguration on 20th January 2017. During this period Donald Trump used an Android phone throughout.

The data file `trump.csv` contains the following columns:

- `source` - device used: "Android" (Trump) or "iOS" (team)
- `text` - text of the tweet
- `nwords` - number of words in the tweet
- `contains_url` - whether the tweet contains a URL
- `sentiment` - sentiment obtained from a simple sentiment analysis
- `dow` - day of the week ("Monday" to "Sunday")
- `day` - day (in days since 1 January 2016)
- `hour` - decimal hour in the day

Let us read in the data:

```
trump <- read.csv(url("http://www.stats.gla.ac.uk/~tereza/rp/trump.csv"))
```

We define a new variable, `Source`, to be the outcome variable such that when the tweet is written from an iOS device (i.e. by the team) this is set to 0 and when the tweet is written by Trump himself, this is set to 1.

```
trump$Source <- 0
trump$Source[trump$source == "Android"] <- 1
```

We also add a "none" level to `sentiment` to replace `NA` values:

```
trump$sentiment[is.na(trump$sentiment)] <- "none"
trump$sentiment <- factor(trump$sentiment)
```

Task 6.

Fit a logistic regression model with `Source` as the response and `nwords`, `contains_url`, `sentiment`, `dow`, `day`, and `hour` as predictors. Are all the terms significant in the model?

Task 7.

From the fitted model with all terms in it, is a tweet with a url in it more likely to have been written by Trump or his team? What about the number of words? Is a longer tweet more likely to have been written by Trump or his team?

Task 8.

Assess the predictive performance of the above model by producing a ROC curve and obtaining the area under the curve.

Other issues with models for binary/binomial data

Separation/perfect prediction

```
dat

##    y x1 x2
## 1 1 29 62
## 2 1 30 83
## 3 1 31 74
## 4 1 32 68
## 5 0 29 41
## 6 0 30 44
## 7 0 31 21
## 8 0 32 50
## 9 0 33 33
```

Separation in logistic regression

https://youtu.be/H4UpxY3dpIM

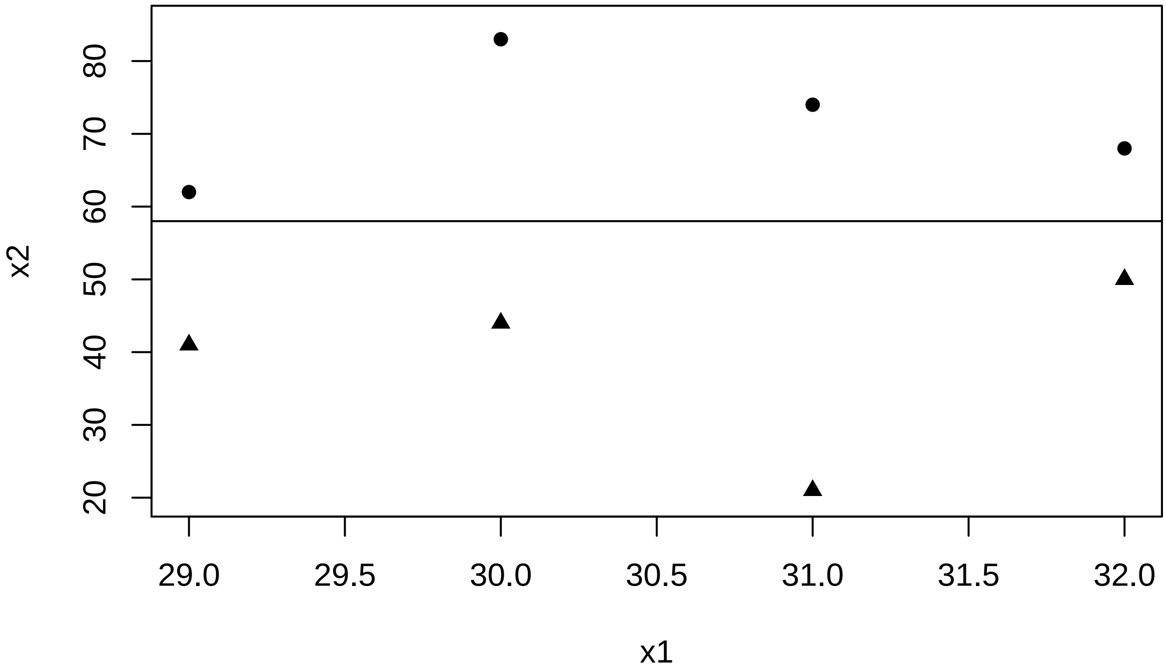
Duration: 1m21s

Separation occurs in logistic regression models when a hyperplane exists that perfectly separates responses from non-responses. In that case the MLE $\hat{\beta}$ does not exist. Consider the following illustration:

```
dat<- read.table(url("http://www.stats.gla.ac.uk/~tereza/rp/separation.txt"), header=TRUE)
dat
```

```
  y x1 x2
1 1 29 62
2 1 30 83
3 1 31 74
4 1 32 68
5 0 29 41
6 0 30 44
7 0 31 21
8 0 32 50
9 0 33 33
```

Here the binary response y can be perfectly predicted from the value of explanatory variable x_2, as can be seen in the following plot.



When we fit the model using `glm` we get a warning:

```
mod.sep <- glm(y~x1+x2, family="binomial", data=dat)
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

And the output looks strange:

```
summary(mod.sep)
```

```
Call:
glm(formula = y ~ x1 + x2, family = "binomial", data = dat)

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-1.610e-05  -1.058e-06  -2.110e-08   2.110e-08   1.456e-05

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.604e+01  1.727e+06   0.000        1
x1          -5.808e+00  5.372e+04   0.000        1
x2           2.541e+00  4.292e+03   0.001        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1.2365e+01  on 8  degrees of freedom
Residual deviance: 5.0320e-10  on 6  degrees of freedom
AIC: 6

Number of Fisher Scoring iterations: 24
```

Notice the large standard errors, large p-values and zero deviance!

For more information on separation in logistic regression and on ways to deal with this problem follow this link. In particular, Firth's logistic regression implemented in the function `logistf()` in `library(logistf)`, `bayesglm()` from `library(arm)` and `glmnet()` from `library(glmnet)` may be useful here.

Overdispersion

In a model for binomial responses Y_i, we model the mean $E(Y_i)$ as a function of explanatory variables. And since the variance of a binomial distribution is $\mathrm{Var}(Y_i) = n_i p_i(1 - p_i)$, it is related to the mean, so by fitting a model for the mean of binomial responses we are implicitly also modelling the variance.

Observations y_i may have observed variance greater than the binomial variance $n_i p_i(1 - p_i)$. This is called *overdispersion*. Similarly, *underdispersion* occurs when $\mathrm{Var}(Y_i)$ is much smaller than $n_i p_i(1 - p_i)$. This could be caused by omission of important explanatory variables, correlated Y_i, misspecification of the link function or other data complexities. Overdispersion can be detected if the deviance is much greater than its degree of freedom of $n-p$. One approach to correct for overdispersion is to include an extra dispersion parameter ϕ in the model so that $\mathrm{Var}(Y_i) = \phi n_i p_i(1-p_i)$. We will see more on how to deal with overdispersed data later, in the context of count regression.

Rare events

Logistic regression is frequently used to model the probability of a rare event such as a disease, an equipment failure or natural disaster. The problem is that if there are only a few instances of the rare event in the data, the logistic regression model estimates will be biased, leading to poor prediction outcomes. This means that if we have very few 1s in our data, any logistic regression model we fit will do a poor job of predicting the 1s. One strategy to try and deal with this problem is to sample all of the 1s in the data but only some of the 0s before fitting the logistic regression model.

Additional resources

You can read more about models for binomial data in Chapter 2 from *Extending linear models with R: generalized linear, mixed effects and nonparametric regression models by Julian Faraway*:

http://encore.lib.gla.ac.uk/iii/encore/record/C__Rb2939999?lang=eng

For more examples in R, along with a summary of issues relating to logistic regression, see this data analysis example from UCLA's Institute for Digital Research and Education. The same source also provides an example of probit regression.

Week 3 learning outcomes

- Recognise the types of data that require a model for binary(ungrouped)/binomial (grouped) response
- Distinguish which type of model, binomial or binary, is appropriate for a given dataset

- Use appropriate plots to explore the relationship between the response and explanatory variables (e.g. boxplots for continuous explanatory variables, and barcharts for categorical explanatory variables)
- Fit a GLM for binary/binomial outcomes using any of the three link functions presented here (logit, probit, complementary log-log)
- Recognise the model being fit from the R code used to fit it
- Interpret logistic regression (logit model) coefficients in terms of the odds of the outcome of interest
- Test hypotheses about regression coefficients using Wald tests and deviances
- Interpret logistic regression output presented in the form of odds ratios
- Check the goodness of fit of a binomial model using the deviance, but only if the fitted values are relatively large (in particular do not use the deviance for goodness of fit of a binary logistic regression model)
- Obtain predicted probabilities and fitted values from a GLM for binary/binomial responses
- Use logistic regression as a classification tool
- Use ROC curves and the area under curve to assess classification performance
- Be aware of the difficulty with trying to predict rare events and the resulting poor classification performance
- Recognise the warnings that separation (perfect prediction) may be occurring in a logistic regression model and be familiar with possible remedies
- Be aware of the issue of overdispersion in logistic regression models

Answers to tasks

Answer to Task 1. Logit link:

```
lmod <- glm(cbind(damage, 6-damage) ~ temp, family=binomial, data=orings)
summary(lmod)
```

Probit link:

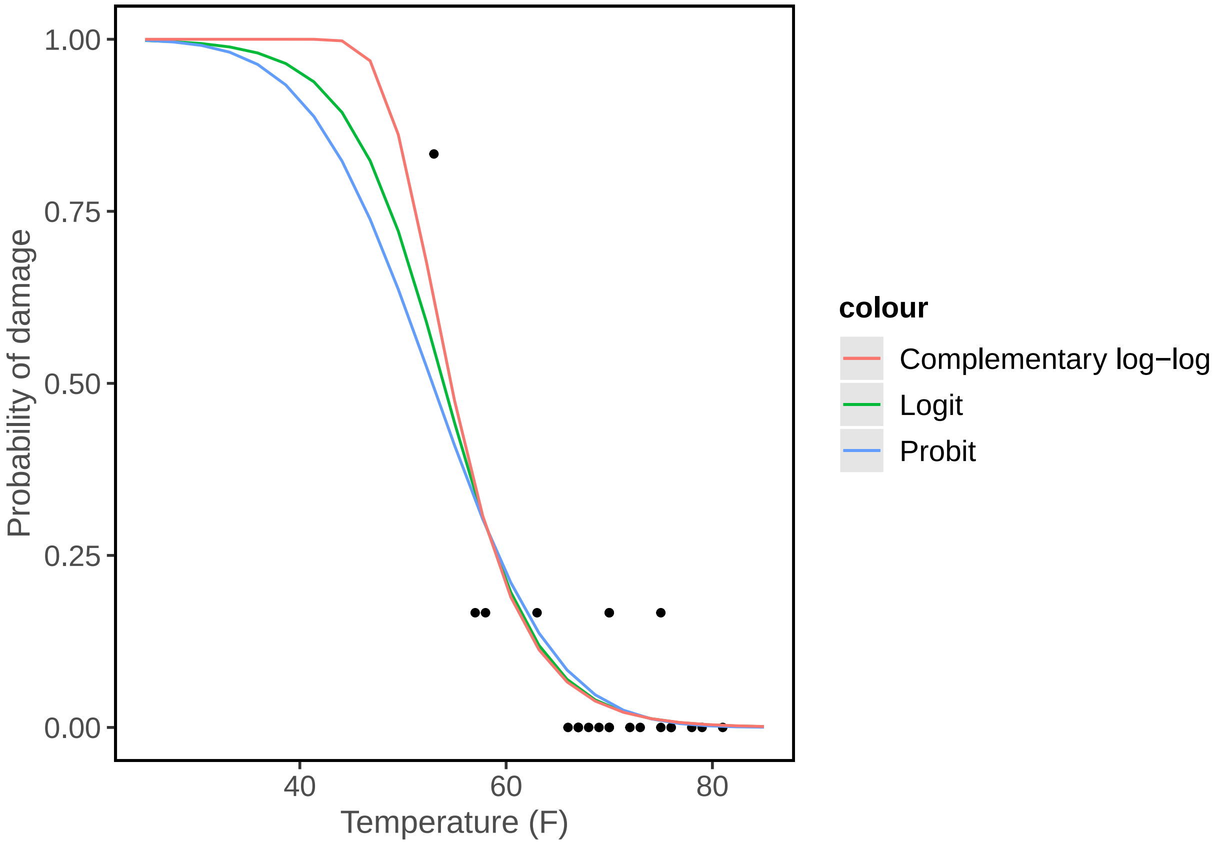
```
pmod <- glm(cbind(damage, 6-damage) ~ temp, family=binomial(link="probit"),
            data=orings)
summary(pmod)
```

Complementary log-log link:

```
cmod <- glm(cbind(damage, 6-damage) ~ temp, family=binomial(link="cloglog"), data=orings)
summary(cmod)
```

Answer to Task 2. Here is some code for plotting the three fits.

```
pred1 <- predict(lmod, newdata=data.frame(temp=seq(25,85,le=23)), type="response")
pred2 <- predict(pmod, newdata=data.frame(temp=seq(25,85,le=23)), type="response")
pred3 <- predict(cmod, newdata=data.frame(temp=seq(25,85,le=23)), type="response")
pred <- data.frame(logit = pred1, probit= pred2, cloglog=pred3, px = seq(25,85,le=23),orings)
p1.1 <- ggplot(pred, aes(x=orings$temp, y= orings$damage/6)) +
        geom_point(size = 1)+ xlim (c(25,85)) + ylim(c(0,1)) +
        xlab ("Temperature (F)") + ylab("Probability of damage") +
        geom_line(aes(x = px, y = logit, color = "Logit")) +
        geom_line(aes(x = px, y = probit, color = "Probit"))+
  geom_line(aes(x = px, y = cloglog, color = "Complementary log-log"))
```



Answer to Task 3. We can obtain the predicted probabilities using the model equation:

```
exp(11.6630-0.2162*31)/(1+exp(11.6630-0.2162*31))
```

```
[1] 0.9930414
```

We can get the same answer using the `predict()` function as follows:

```
predict(lmod, newdata=data.frame(temp=31), type="response")
```

```
        1
0.9930342
```

Similarly, we can obtain the prediction for the probit model using the cumulative distribution function of a normal distribution:

```
pnorm(5.5915-0.1058*31)
```

```
[1] 0.9896029
```

or by using the `predict()` function:

```
predict(pmod, newdata=data.frame(temp=31), type="response")
```

```
        1
0.9895983
```

Finally for the complementary log-log model the predicted probability is

```
predict(cmod, newdata=data.frame(temp=31), type="response")
```

```
1
1
```

The predicted probability of damage is very high for all models.

Answer to Task 4. We can fit a model with just `gender` as a predictor:

```
mod.yl2 <- glm(hear ~ gender, family=binomial, data=yl)
summary(mod.yl2)
```

```
Call:
glm(formula = hear ~ gender, family = binomial, data = yl)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-1.177   -1.177   -1.077    1.177    1.281

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.754e-16  3.780e-01   0.000    1.000
genderMale  -2.412e-01  5.524e-01  -0.437    0.662

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 73.304  on 52  degrees of freedom
Residual deviance: 73.113  on 51  degrees of freedom
AIC: 77.113

Number of Fisher Scoring iterations: 3
```

or we can add `gender` to the model with `age`:

```
mod.yl3 <- glm(hear ~ gender+age, family=binomial, data=yl)
summary(mod.yl3)
```

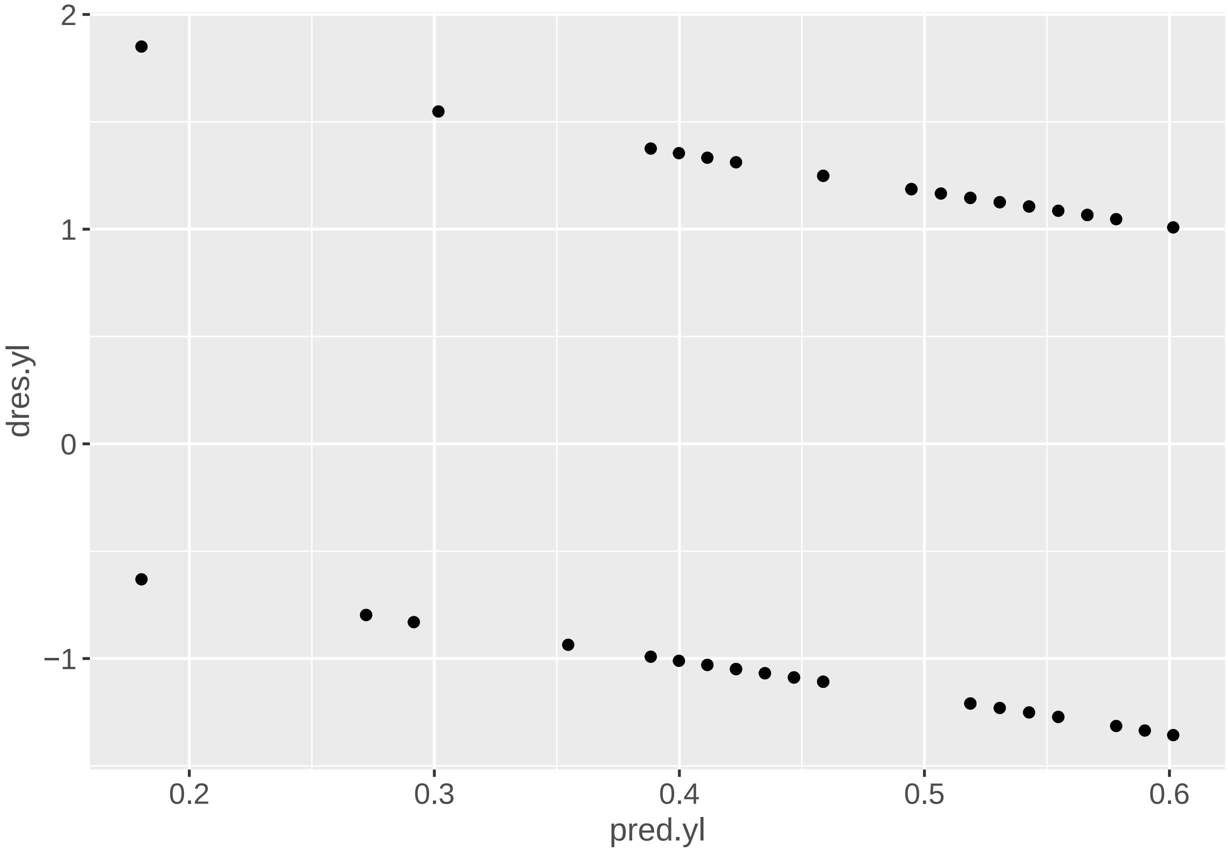
```
Call:
glm(formula = hear ~ gender + age, family = binomial, data = yl)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-1.381   -1.099   -0.776    1.154    1.809

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept)  1.62792    1.24392   1.309    0.191
genderMale  -0.20637    0.56935  -0.362    0.717
age         -0.04839    0.03404  -1.422    0.155

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 71.779  on 51  degrees of freedom
Residual deviance: 69.454  on 49  degrees of freedom
  (1 observation deleted due to missingness)
AIC: 75.454

Number of Fisher Scoring iterations: 4
```

In both cases we see that there is no significant gender effect.

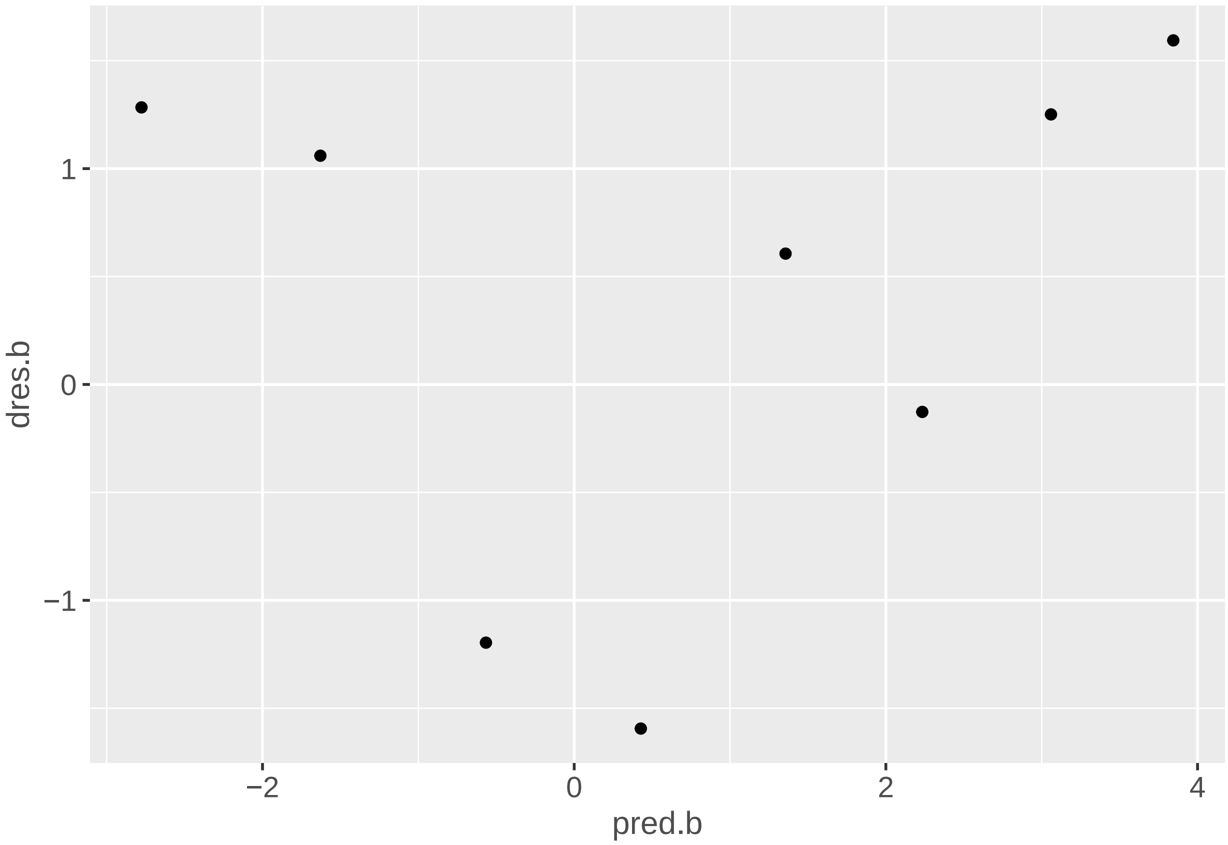
Answer to Task 5. Residual plots for the Yanny-Laurel model:

```
dres.yl <- resid(mod.yl, type="deviance") # Deviance residuals
pres.yl <- resid(mod.yl, type="pearson") # Pearson residuals
pred.yl <- predict(mod.yl, type="response") # Fitted probabilities
d.yl <- data.frame(pred.yl=pred.yl,dres.yl=dres.yl, pres.yl=pres.yl)

ggplot(d.yl, aes(x = pred.yl, y = dres.yl)) +   geom_point()
```



and for the beetles logit model:

```
m1.b <- glm(beetles.mat ~ beetles$dose, family = binomial(link = 'logit'))
dres.b <- resid(m1.b, type="deviance")
pres.b <- resid(m1.b, type="pearson")
pred.b <- predict(m1.b)

d.b <- data.frame(pred.b=pred.b,dres.b=dres.b, pres.b=pres.b)

ggplot(d.b, aes(x = pred.b, y = dres.b)) +   geom_point()
```



We can see that the beetle model residuals scatter around zero in a similar way to a linear regression model, while for the Yanny-Laurel model they follow a distinct pattern.

Answer to Task 6. Fitting a model with the required set of predictors gives the following results:

```
mod.trump <- glm(Source ~ nwords + contains_url + sentiment + dow + day + hour,
 data = trump, family = binomial)
summary(mod.trump)
```

```
Call:
glm(formula = Source ~ nwords + contains_url + sentiment + dow +
    day + hour, family = binomial, data = trump)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4702  -0.1076  -0.0559   0.6100   3.4563

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)           -0.0648737  0.3480450  -0.186  0.85214
nwords                 0.0725729  0.0107829   6.730 1.69e-11 ***
contains_urlTRUE      -6.0212045  0.4209382 -14.304  < 2e-16 ***
sentimentanticipation  0.2996729  0.2088897   1.435  0.15140
sentimentdisgust      -0.1026294  0.3989978  -0.257  0.79701
sentimentfear         -0.2352392  0.2866442  -0.821  0.41184
sentimentjoy           0.7641286  0.3307041   2.311  0.02085 *
sentimentnegative      0.0219771  0.1936794   0.113  0.90966
sentimentnone          0.1635874  0.2268069   0.721  0.47075
sentimentpositive     -0.1317144  0.1810389  -0.728  0.46689
sentimentsadness       2.9297573  3.7618886   0.779  0.43610
sentimentsurprise     -0.0094329  0.5236602  -0.018  0.98563
sentimenttrust         0.1131445  0.3038813   0.372  0.70965
dowMonday              0.4235713  0.2022209   2.095  0.03621 *
dowSaturday            0.2963022  0.1931703   1.534  0.12506
dowSunday              1.0348119  0.2124598   4.871 1.11e-06 ***
```

```
dowThursday            0.0677405  0.2043373   0.332  0.74026    
dowTuesday             0.6180228  0.2000779   3.089  0.00201 ** 
dowWednesday           0.4546226  0.2059776   2.207  0.02730 *  
day                    0.0026619  0.0005212   5.107 3.28e-07 ***
hour                  -0.0986860  0.0099297  -9.938  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4403.8  on 3228  degrees of freedom
Residual deviance: 2028.5  on 3208  degrees of freedom
AIC: 2070.5

Number of Fisher Scoring iterations: 8
```

We can look at what happens to the residual deviance as we add each term by using

```
anova(mod.trump)
```

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: Source

Terms added sequentially (first to last)


             Df Deviance Resid. Df Resid. Dev
NULL                          3228     4403.8
nwords        1  1059.81      3227     3344.0
contains_url  1  1129.71      3226     2214.3
sentiment    10     9.46      3216     2204.8
dow           6    38.99      3210     2165.9
day           1    33.19      3209     2132.7
hour          1   104.18      3208     2028.5
```

We see that the largest reduction in residual deviance comes when adding `contains_url` and the smallest when adding `sentiment`. We could try a model without `sentiment` as the resulting reduction in deviance (10.47) is smaller than the 95th percentile of a $\chi^2(9)$ distribution:

```
qchisq(df=9,p=0.95)
```

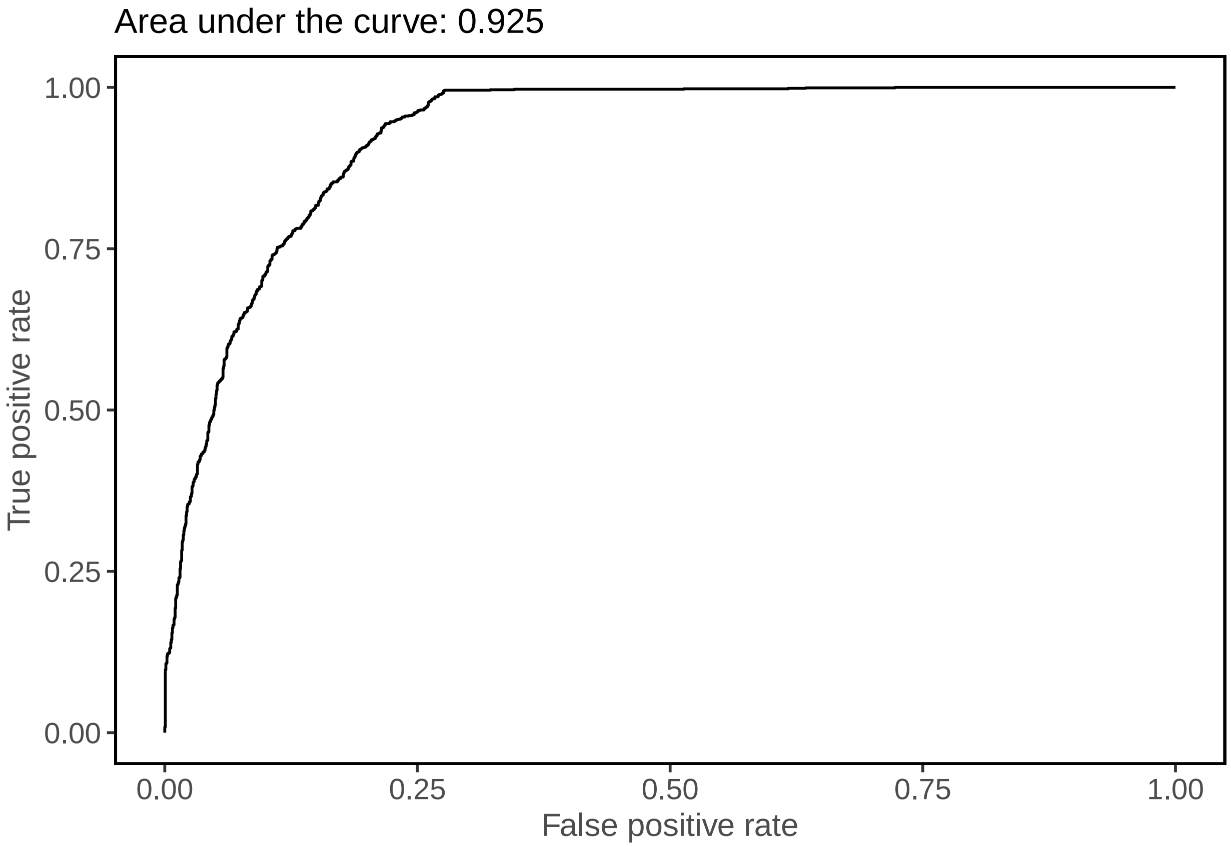
```
[1] 16.91898
```

Alternatively one can use a stepwise procedure to select which terms to include in a final model, using the *Akaike Information Criterion (AIC)* or similar as the criterion to optimise.

Answer to Task 7. The coefficient of `contains_url` is negative, suggesting that such a tweet is less likely to have been written by Trump. Conversely, the coefficient of `nwords` is positive, suggesting that a longer tweet is more likely to have been written by Trump.

Answer to Task 8. We can produce a ROC curve and calculate the area under the curve as follows:

```
library(ROCR)
trump$pred <- predict(mod.trump, trump, type="response")
score <- prediction(trump$pred,trump$Source)
perf <- performance(score,"tpr","fpr")
auc <- performance(score,"auc")
perfd <- data.frame(x= perf@x.values[1][[1]], y=perf@y.values[1][[1]])
roc.trump<- ggplot(perfd, aes(x= x, y=y)) + geom_line() +
       xlab("False positive rate") + ylab("True positive rate") +
       ggtitle(paste("Area under the curve:", round(auc@y.values[[1]], 3)))
```



The area under the curve is 0.91. It may be possible to improve the predictive performance of the model by adding more predictors or even by removing some of the terms currently in the model.