SCIE4002: Experimental Design and Data Analysis

L05 - Generalized linear models - Linear, logistic and Poisson regression

John Ormerod and Ellis Patrick
10 March, 2024

THE UNIVERSITY OF SYDNEY

Outline

- Simple Linear regression (revision).
- Multiple Linear regression (revision - hopefully).
- Relationship between linear regression with ANOVA and variants.
- Logistic regression.
- Poisson regression.

Linear regression (revision)

In terms of models so far we have covered

- t-tests
- One way ANOVA
- Two way ANOVA

Simple linear regression - Motivating data

Consider the following dataset.

```
dat <- read.csv("data/heightWeight.csv")
head(dat)
```

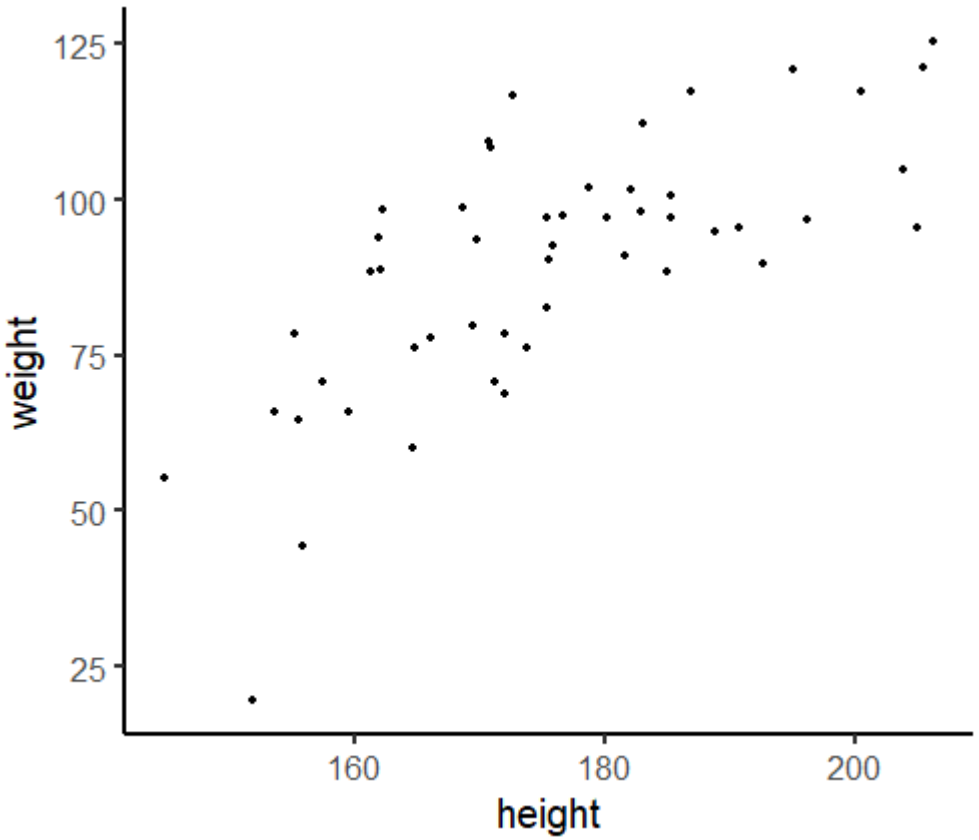
```
##     height    weight diet age
## 1 161.1871  88.45414 meat  60
## 2 178.6927 101.89500 meat  50
## 3 176.6476  97.50369 meat  61
## 4 205.3532 121.12928 meat  14
## 5 200.5163 117.49053 meat  44
## 6 162.0562  88.62529 meat  32
```

Why has someone collected this data?

Simple Linear regression

Let's plot a scatterplot of y=weight vs x= height

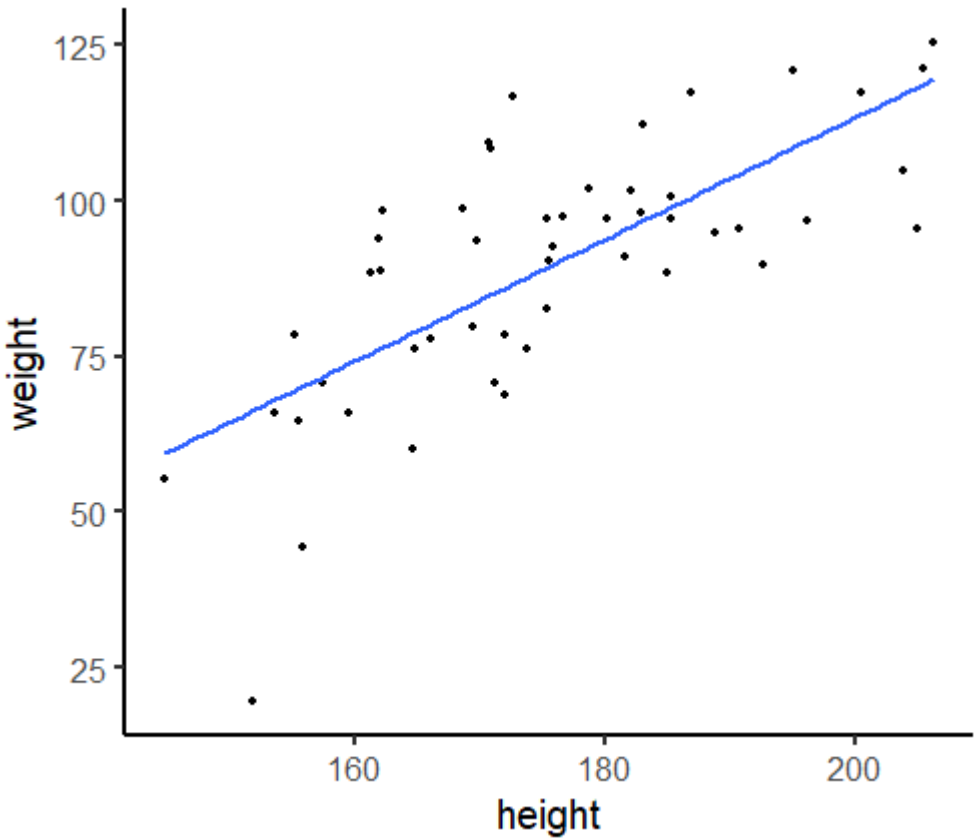
```
ggplot(dat, aes(x = height, y = weight)) +
  geom_point()
```



Simple Linear regression

A linear relationship looks like a good initial model for our data.

```
ggplot(dat, aes(x = height, y = weight)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```



Simple Linear regression

The equation of the fitted line is

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad i = 1, \ldots, n,$$

where $\varepsilon_i \sim N(0, \sigma^2)$ is random noise.

For our example

- The y_i's correspond to each individual's weight in the dataset.
- The x_i's correspond to each individual's height in the dataset.
- β_0 is the population intercept parameter.
- β_1 is the population slope parameter.

The predicted value of y_i for a given x_i.

$$\hat{y}_i = \beta_0 + \beta_1 x_i$$

Simple Linear regression - Assumptions

Linearity:

The relationship between y_i and x_i is linear

$$y_i = \beta_0 + \beta_1 x_i$$

An alternative model might be

$$y_i = \beta_0 + \beta_1 x_i^2$$

or

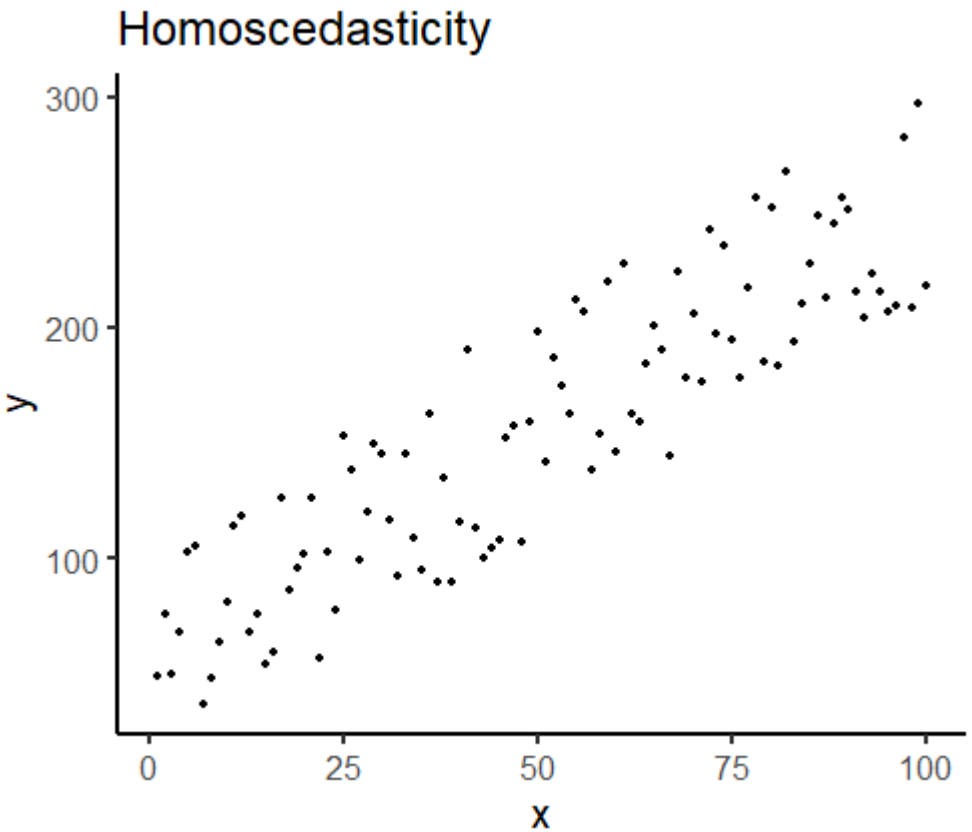
$$y_i = \beta_0 + \beta_1 e^{x_i}$$

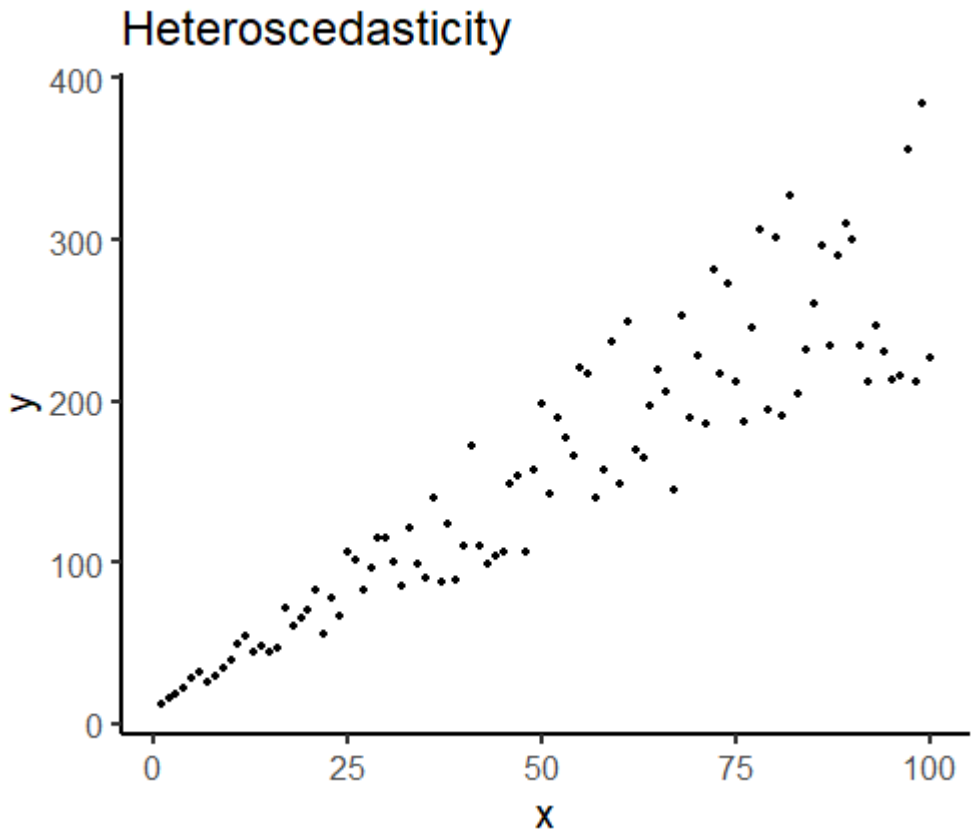
in which case the relationship between y_i and x_i is not linear.

Simple Linear regression - Assumptions

Homoscedasticity or constant variance:

- The value of σ^2 does not depend on the value of x_i (or any other predictor in our dataset).





Simple Linear regression - Other assumptions

Some other assumptions are:

- Independence: Each point pair $[x_i, y_i]$ do not depend on ε_i or any other point pair.
- Normality: The errors $\varepsilon_1, \ldots, \varepsilon_n$ follow a normal distribution.

Note: We do not need to assume that the x_i's are normally distributed.

Simple Linear regression - Fitting the model

If we have mean zero errors, independence, linearity, and homoscedasticity (not necessariy normality!) then estimating the coefficients β_0 and β_1 using least squares is "best" (Gauss-Markov theorem).

We choose β_0 and β_1 to minimize

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$$

In this course we won't concern ourselves about how this calculation is done (but can be done using high school calculus).

Simple Linear regression - Fitted values

```
mod <- lm(weight~height, data=dat)
mod
```

```
##
## Call:
## lm(formula = weight ~ height, data = dat)
##
## Coefficients:
## (Intercept)       height
##    -82.2887       0.9786
```

From the above R code we see that the fitted values for the intercept and slope are

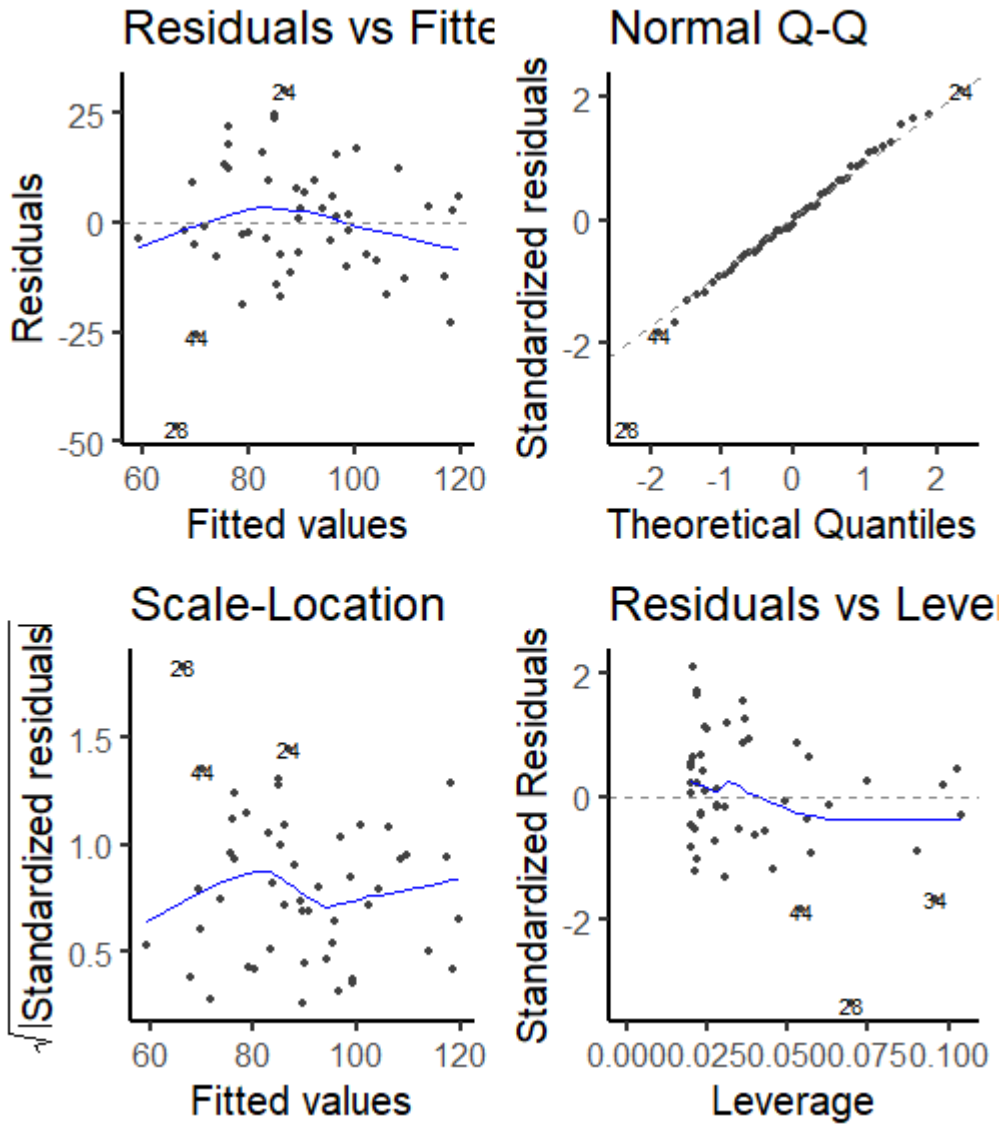
- $\widehat{\beta}_0 = -82.2887$ and
- $\widehat{\beta}_1 = 0.9786$ respectively.

We have placed "hats" on the parameters to indicate that these are fitted values from the sample (and are no longer population parameters).

Simple Linear regression - Assumptions - R code

```
mod <- lm(weight~height, data=dat)

library(ggfortify)
autoplot(mod)
```



Simple Linear regression - Fitted model and intepretation

The fitted model is

$$\hat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_i = -82.2887 + 0.9786 \times x_i$$

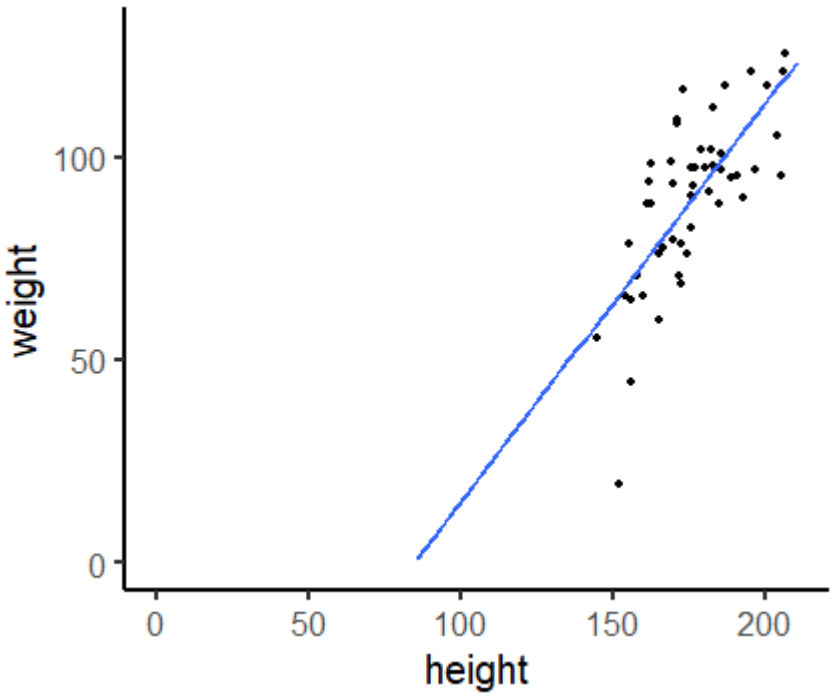
- For every unit increase in x_i the y_i values increase by 0.9786 kilograms.
- When $x_i = 0$ (zero height - no one is this tall) then the predicted y_i value is -82.2887 kilograms (no one has the weight).

This highlights the dangers of extrapolating outside the range of our data.

Simple Linear regression - Extrapolation

We should only use this model inside the range of the observed data. Otherwise we can make non-sensical predictions - this is called extrapolation.

```
ggplot(dat, aes(x = height, y = weight)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, fullrange = TRUE) +
  xlim(0, 210) + ylim(0, 130)
```



Simple Linear regression - Inference

```
mod <- lm(weight~height, data=dat)
summary(mod)
```

```
##
## Call:
## lm(formula = weight ~ height, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.777  -7.734  -0.055   9.203  30.123
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -82.2887    23.9323  -3.438  0.00122 **
## height        0.9786     0.1358   7.207 3.55e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.45 on 48 degrees of freedom
## Multiple R-squared:  0.5197,    Adjusted R-squared:  0.5097
## F-statistic: 51.94 on 1 and 48 DF,  p-value: 3.547e-09
```

Simple Linear regression - Inference

- The 5 number summary statistics for the residuals $r_i = y_i - \hat{y}_i \; i = 1, \ldots, n$ are

Min	1st quartile	Median	3rd Quartile	Max
-46.777	7.734	-0.055	9.203	30.123

- The median being close to 0 indicates the data are not very skewed.
- $\widehat{\beta}_0 = -82.2887$ and $\widehat{\beta}_1 = 0.9786$
- $\text{se}(\widehat{\beta}_0) = 23.9323$ and $\text{se}(\widehat{\beta}_1) = 0.1358$
- T-values are t-statisitcs: $\widehat{\beta}_j / \text{se}(\widehat{\beta}_j)$.
- Hypothesis testing $H_0 : \beta_0 = 0$ the p-value is 0.00122 - β_0 is significantly different from 0.
- Hypothesis testing $H_0 : \beta_1 = 0$ the p-value is 3.55×10^{-9} - β_1 is significantly different from 0.
- The fitted value of $\widehat{\sigma}$ is 14.45.
- F-statistic is 51.94 (testing that the intercept model is adequate) has p-value 3.55×10^{-9} suggesting we reject the model containing the intercept only in favor of the model with an intercept and slope.

Simple Linear regression - R-squared

- Let the Residual sum of squares for the full model be

$$\mathrm{RSS}_1 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i)^2$$

- Let the Residual sum of squares for the intercept model be

$$\mathrm{RSS}_0 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{\beta}_0)^2$$

- Then the R-squared value is

$$R^2 = \frac{\mathrm{RSS}_0 - \mathrm{RSS}_1}{\mathrm{RSS}_0}$$

- The R-squared value is between 0 and 1 and can be interpreted as the amount of variability explained by the linear regression model and is a measure of goodness of fit.
- The R-squared value for this model is 0.5197

Multiple Linear regression - Motivating data

Let's consider again the same dataset we considered when looking at simple linear regression.

```
dat <- read.csv("data/heightWeight.csv")
head(dat)
```

```
##     height    weight diet age
## 1 161.1871  88.45414 meat  60
## 2 178.6927 101.89500 meat  50
## 3 176.6476  97.50369 meat  61
## 4 205.3532 121.12928 meat  14
## 5 200.5163 117.49053 meat  44
## 6 162.0562  88.62529 meat  32
```

Multiple Linear regression - The Model

When going for simple linear regression to multiple linear regression we can add additional predictors to our model. Our model becomes

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \varepsilon_i$$

where $\varepsilon_i \sim N(0, \sigma^2)$, $i = 1, \ldots, n$.

Instead of having two coefficients and one variance parameter to estimate we now have $p + 2$ parameters where p is the number of predictors.

Multiple Linear regression - Fitting the Model

Suppose that we add diet to our model

```
mod <- lm(weight~height+diet, data=dat)
mod
```

```
##
## Call:
## lm(formula = weight ~ height + diet, data = dat)
##
## Coefficients:
## (Intercept)       height      dietveg
##     -72.036        0.942       -7.622
```

Since diet is a factor with two categories, the model that R fits is

$$y_i = \beta_0 + \beta_1 \mathbf{x}_{i1} + \beta_2 \mathbf{x}_{i2} + \varepsilon_i$$

where

- y_i corresponds to the ith person's weight.
- x_{i1} corresponds to the ith person's height.
- x_{i2} is 0 if ith person's diet is "meat" and 1 if the ith persons diet is "veg".

Multiple Linear regression - Intepretting our model

The fitted model from our previous slide is

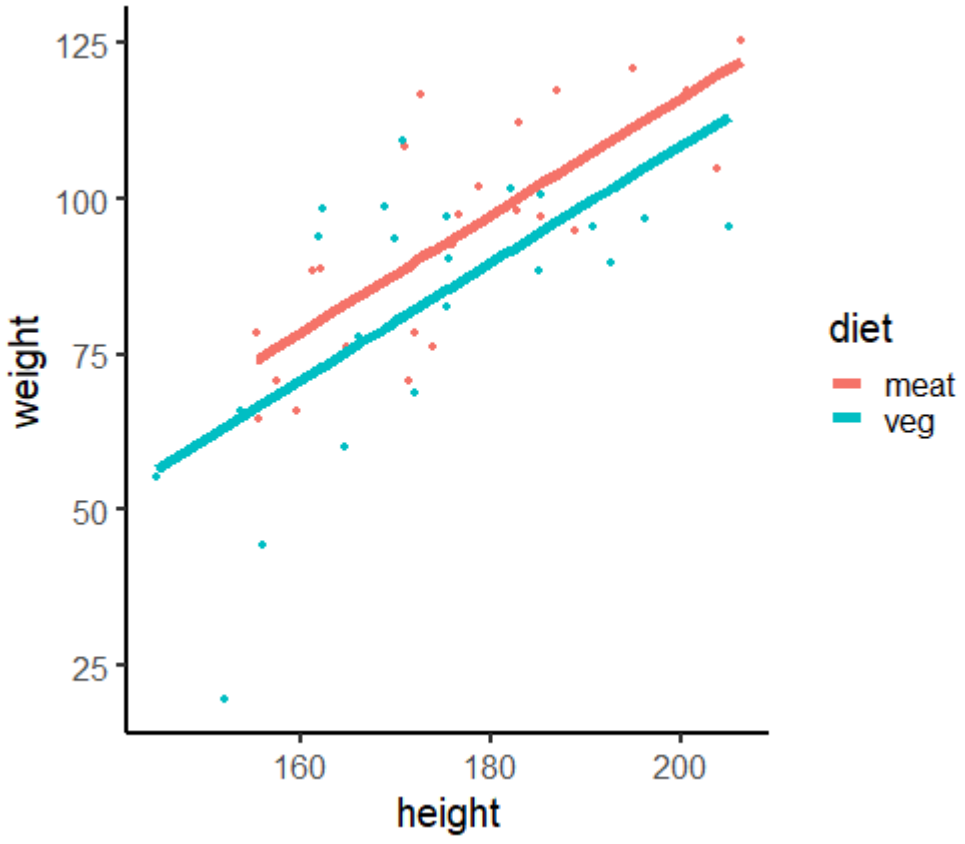
$$\text{weight}_i = -72.036 + 0.942 \times \text{height}_i - 7.622 \times I(\text{diet}_i = veg)$$

The interpretation is...

- On average, for every 1 unit increase in height we would predict weight to increase by 0.942kg
- On average, if the person's diet is "veg" we would predict their weight to be 7.622 less than if their diet was "meat" (which is the baseline).

Multiple Linear regression - Plotting the Model

```
dat |> mutate(fit = predict(mod)) |>
ggplot(aes(x = height, y = weight, colour = die
  geom_point()+
  geom_line(aes(height, fit), linewidth = 2)
```



Multiple linear regression - 2 continuous and 1 binary predictor

Next we are going to add "age" to our model.

```
mod <- lm(weight~height+diet+age, data=dat)
mod
```

```
##
## Call:
## lm(formula = weight ~ height + diet + age, data = dat)
##
## Coefficients:
## (Intercept)       height      dietveg          age
##    -57.4078       0.8948      -8.2640      -0.1298
```

The fitted model from our previous slide is

$$\text{weight}_i = -72.036 + 0.942 \times \text{height}_i - 7.622 \times I(\text{diet}_i = veg)$$

However, one might ask the question: Does height or age play a bigger role in determining weight?

Multiple linear regression - 2 continuous and 1 binary predictor

Looking at the model summary...

```
summary(mod)
```

```
##
## Call:
## lm(formula = weight ~ height + diet + age, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.109  -8.937  -1.371   8.326  28.897
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -57.4078    26.6620  -2.153   0.0366 *
## height        0.8948     0.1385   6.460 5.89e-08 ***
## dietveg      -8.2640     4.0384  -2.046   0.0465 *
## age          -0.1298     0.1065  -1.219   0.2290
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.01 on 46 degrees of freedom
## Multiple R-squared:  0.5677,    Adjusted R-squared:  0.5395
## F-statistic: 20.14 on 3 and 46 DF,  p-value: 1.765e-08
```

Categorization of linear regression

The table below summarises the types of models that we have considered.

Name	R Formula	Comments
t-test	lm(y~x)	y continuous, x binary
1-way ANOVA	lm(y~x)	y continuous, x factor
2-way ANOVA	lm(y~x1*x2)	y continuous, x1 & x2 factors
simple linear regression	lm(y~x)	y continuous, x continuous
multiple linear regression	lm(y~x1+x2+x3)	y continuous, x1,x2 & x3 continuous

In the first three weeks we have covered the first three model types.

Non-continuous responses

- Often the response variable cannot take any continuous value.
- The response could be
  - Positive continuous, e.g., height, blood pressure, time to respond to treatment.
  - Counts: Number of seizures, number of people in a household.
  - Binary: Did the treatment work? Does the person have cancer?
  - Ordinal: How did you rate the movie out of 5 stars?
  - Categorical: Eye colour, book genre.
- In your projects what is the variable of interest?

Models for non-continuous responses

We use different distributions to model different response types:

- Binary: Bernoulli.
- Counts: Poisson, Negative-Binomial.
- Positive continuous: Gamma, inverse-gamma, log-normal distributions.
- Ordinal: Beyond the scope of this course.
- Categorical: Multinomial.

In this course we will only focus on the first two cases, i.e., Bernoulli and Poisson.

Bernoulli distribution

The Bernoulli distribution has one parameter $\rho \in [0, 1]$ so that the probability that the response Y takes a binary value $y = 0$ or $y = 1$ is

$$P(Y = y) = \rho^y (1 - \rho)^{1-y} \tag{1}$$

So for example when $\rho = 0.7$

$$P(Y = 1) = \rho^1 (1 - \rho)^{1-1} = \rho = 0.7 \tag{2}$$
$$P(Y = 0) = \rho^0 (1 - \rho)^{1-0} = 1 - \rho = 0.3 \tag{3}$$

Expected value or mean is

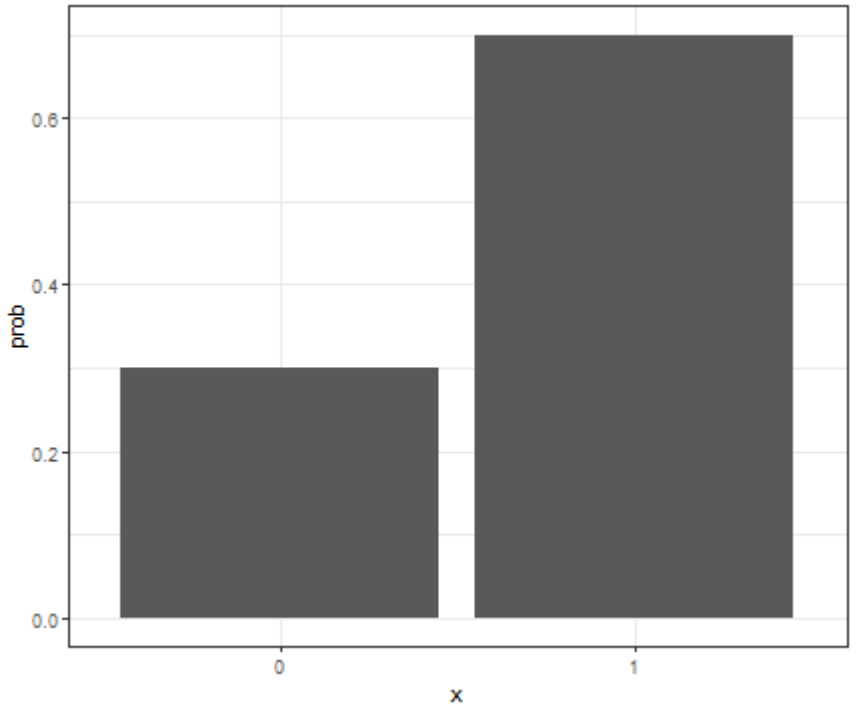
$$\mathbb{E}(Y) = \sum_{y=0}^{\infty} y P(Y = y) = \rho$$

and the variance is

$$\text{Var}(Y) = \rho(1 - \rho)$$

Bernoulli distribution

A histogram of the probabilities when $\rho = 0.7$ is below.



A sequence of $n = 10$ Bernoulli distributed values with $\rho = 0.7$ is given below

```
n <- 10
rho <- 0.7
set.seed(51773)
rbinom(n,1,rho)
```

```
##  [1] 1 1 1 1 0 0 1 1 1 1
```

Increasing ρ increases the average proportion of 1's to 0's, and decreasing ρ decreases the average proportion of 1's to 0's. The probabilities can be calculated using

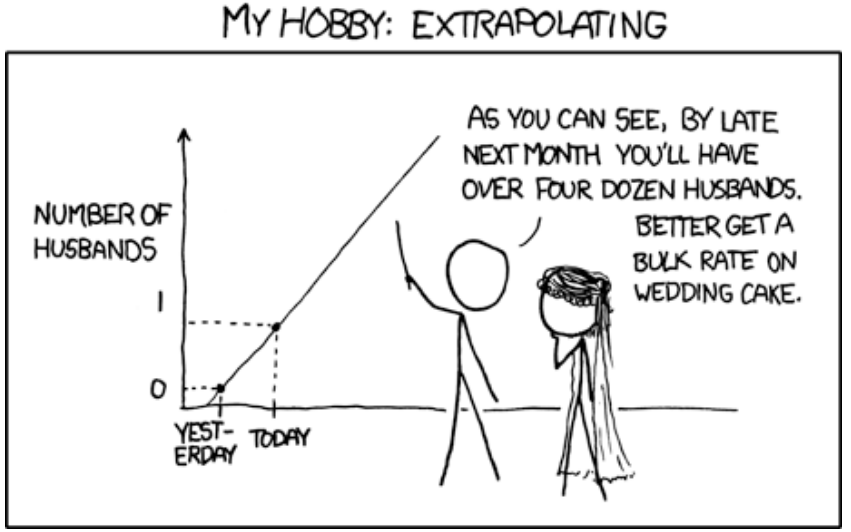
```
dbinom(0:1,1,rho)
```

```
## [1] 0.3 0.7
```

Using linear models for non-continuous responses

Two main problems can occur when using linear models to model non-continuous responses

- The uncertainty (variance) associated with parameter estimates are not calculated correctly. Consequently, the p-values from hypothesis test for each regression coefficient will not be correct (and so not valid).
- Extrapolating the linear outside the domain of the observed data can be problematic.



Example

For the 23 space shuttle flights that occurred before the Challenger disaster of 1986 the table below gives the temperature (in degrees Fahrenheit) at the time of launch and a code of 0-1 where 1 denotes at least one O-ring suffered thermal distress (TD).

Thermal distress of O-rings was blamed for the disaster. Seven people died in the incident.

Flight	Temp	TD	Flight	Temp	TD
1	66	0	13	67	0
2	70	1	14	53	1
3	69	0	15	67	0
4	68	0	16	75	0
5	67	0	17	70	0
6	72	0	18	81	0
7	73	0	19	76	0
8	70	0	20	79	0
9	57	1	21	75	1
10	63	1	22	76	0
11	70	1	23	58	1
12	78	0			

R code

We have two variables

- $y_i = \text{TD}_i$ an indicator of thermal distress.
- $x_i = \text{temp}_i$ the temperature (in degrees Fahrenheit).

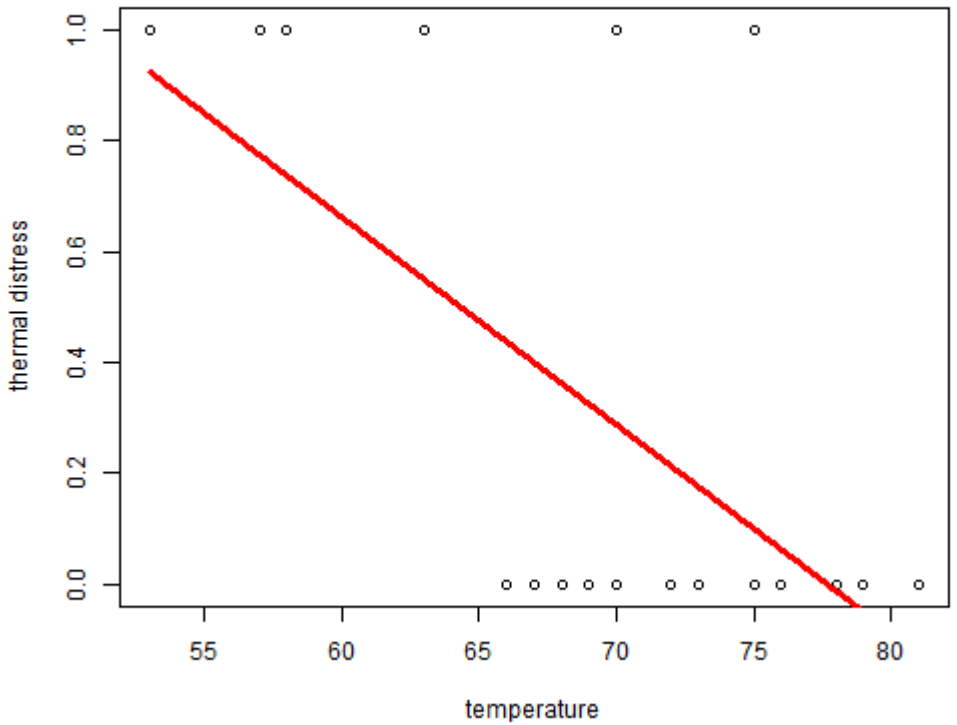
We can enter the data into R via

```
TD   <- c(0,1,0,0,0,0,0,0,1,1,1,0,0,1,0,0,0,0,0,0,1,0,1)
temp <- c(66,70,69,68,67,72,73,70,57,63,70,78,67,53,67,75,70,81,76,79,75,76,58)
dat  <- data.frame(TD,temp)
```

```
res4 <- lm(TD~temp, data=dat)
summary(res4)
```

```
##
## Call:
## lm(formula = TD ~ temp, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43762 -0.30679 -0.06381  0.17452  0.89881
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.90476    0.84208   3.450  0.00240 **
## temp        -0.03738    0.01205  -3.103  0.00538 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3987 on 21 degrees of freedom
## Multiple R-squared:  0.3144,    Adjusted R-squared:  0.2818
## F-statistic:  9.63 on 1 and 21 DF,  p-value: 0.005383
```

Using linear regression with binary data



The linear regression fit gives estimated probabilities of events being less than 0 and greater than one for certain values of the predictor.

From linear regression to logistic regression

If we are dealing with $y_i \in \{0, 1\}$ a Bernoulli model might be more appropriate than a linear model, say,

$$y_i | x_i \overset{ind}{\sim} \text{Bernoulli}(\rho_i)$$

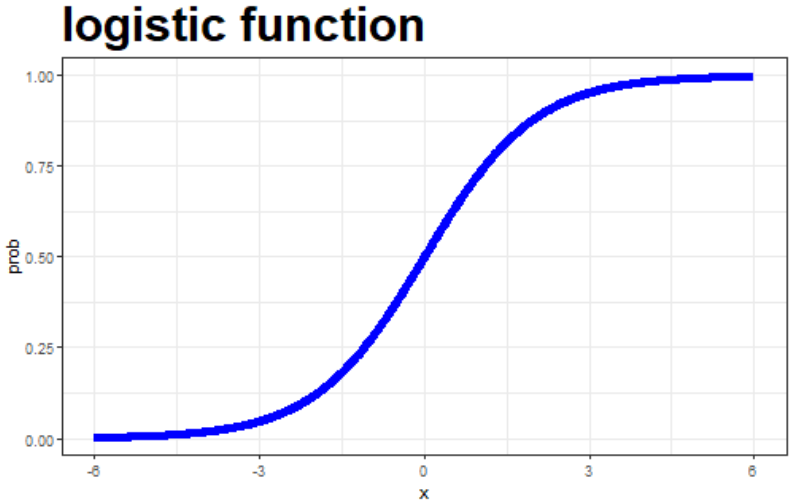
where p_i is some function of $\mathbf{x}_i$. Since $p_i \in [0, 1]$ we want this function to map a value of $\mathbf{x}_i$ to the unit interval. This most common choice is

$$\rho_i = \frac{\exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)}$$

or equivalently

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

The LHS is the log-odds.



This leads to **logistic regression** since $\log\left(\frac{x}{1-x}\right)$ is sometimes referred to as the logistic function.

Odds

The **odds** are an alternative way of quantifying the probability of an event.

For some event E,

$$\text{odds}(E) = \frac{P(E)}{1 - P(E)}.$$

If we are told the odds of E are a to b, then

$$\text{odds}(E) = \frac{a}{b} = \frac{a/(a+b)}{b/(a+b)},$$

which implies $P(E) = a/(a+b)$.

Odds feature in logistic regression.

Probability, Odds, and log-odds

Let's look at some examples for probability, odds, and log-odds to get a feel for them.

$P(X)$	$P(X)/(1-P(X))$	$\log[P(X)/(1-P(X)]$
0.01	0.0101	-4.5951
0.1	0.1111	-2.1972
0.5	1	0
0.9	9	2.1972
0.99	99	4.5951

Logistic regression

So instead of

```
res4 <- lm(TD~temp, data=dat)
```

for linear models. For generalized linear models for binary data we use almost identical syntax...

```
res5 <- glm(TD~temp, data=dat, family=binomial)
```

noting that the Bernoulli distribution is a special case of the binomial distribution.

So we are fitting a model of the form

$$y_i \sim \text{Bernoulli}(p_i) \quad \text{with} \qquad \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 \times \text{temp}_i$$

```
res5 <- glm(TD~temp, data=dat, family=binomial)
summary(res5)
```

```
##
## Call:
## glm(formula = TD ~ temp, family = binomial, data = dat)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  15.0429     7.3786   2.039   0.0415 *
## temp         -0.2322     0.1082  -2.145   0.0320 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28.267  on 22  degrees of freedom
## Residual deviance: 20.315  on 21  degrees of freedom
## AIC: 24.315
##
## Number of Fisher Scoring iterations: 5
```

Logistic regression - The fitted model

The fitted model is

$$y_i \sim \text{Bernoulli}(\hat{p}_i)$$

$$\log\left(\frac{\hat{p}_i}{1-\hat{p}_i}\right) = \widehat{\beta}_0 + \widehat{\beta}_1 \times \text{temp}_i$$

or equivalently

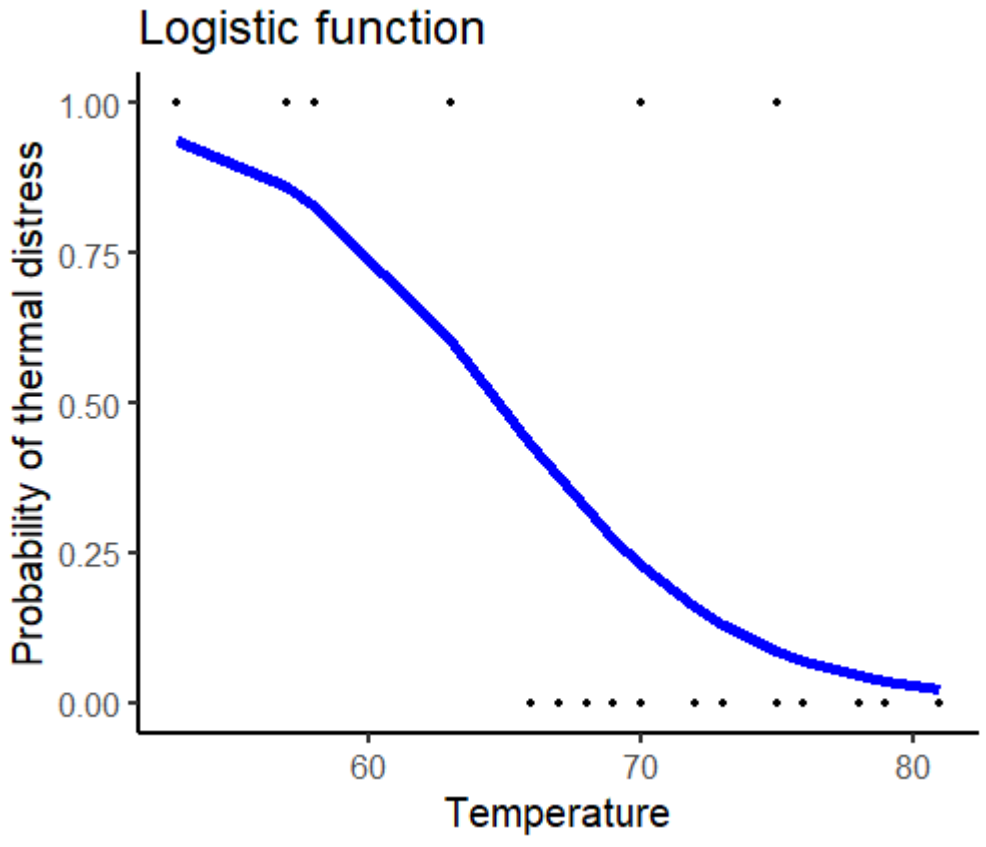
$$\hat{p}_i = \frac{\exp(\widehat{\beta}_0 + \widehat{\beta}_1 \times \text{temp}_i)}{1+\exp(\widehat{\beta}_0 + \widehat{\beta}_1 \times \text{temp}_i)}$$

where

- $\widehat{\beta}_0 = 15$ and
- $\widehat{\beta}_1 = -0.23$.

Plot the fit

```
df <- dat |> mutate(fit = fitted(res5, "respons
g <- ggplot(df, aes(x= temp,y=TD)) +
  geom_point() +
  geom_line(aes(x= temp,y=fit), linewidth=2,col
  labs(title = "Logistic function",
       x = "Temperature",
       y = "Probability of thermal distress")
g
```



Logistic regression - Interpretation of coefficients

The fitted model is

$$\log\left(\frac{\hat{p}_i}{1-\hat{p}_i}\right) = \widehat{\beta}_0 + \widehat{\beta}_1 \times \text{temp}_i$$

It can be shown that

$$\beta_i = \log(\text{odds}(p|X = x_0 + 1)) - \log(\text{odds}(p|X = x_0))$$

and so

$$e^{\beta_i} = \frac{\text{odds}(p|X = x_0 + 1)}{\text{odds}(p|X = x_0)}$$

With our coefficients as $\widehat{\beta}_0 = 15$ and $\widehat{\beta}_1 = -0.23$. We have $e^{\widehat{\beta}_0} = 3.4 \times 10^6$ and $e^{\widehat{\beta}_1} = 0.79$

- The odds of TD decrease by 21% for each degree increase in temperature.

Titanic survival

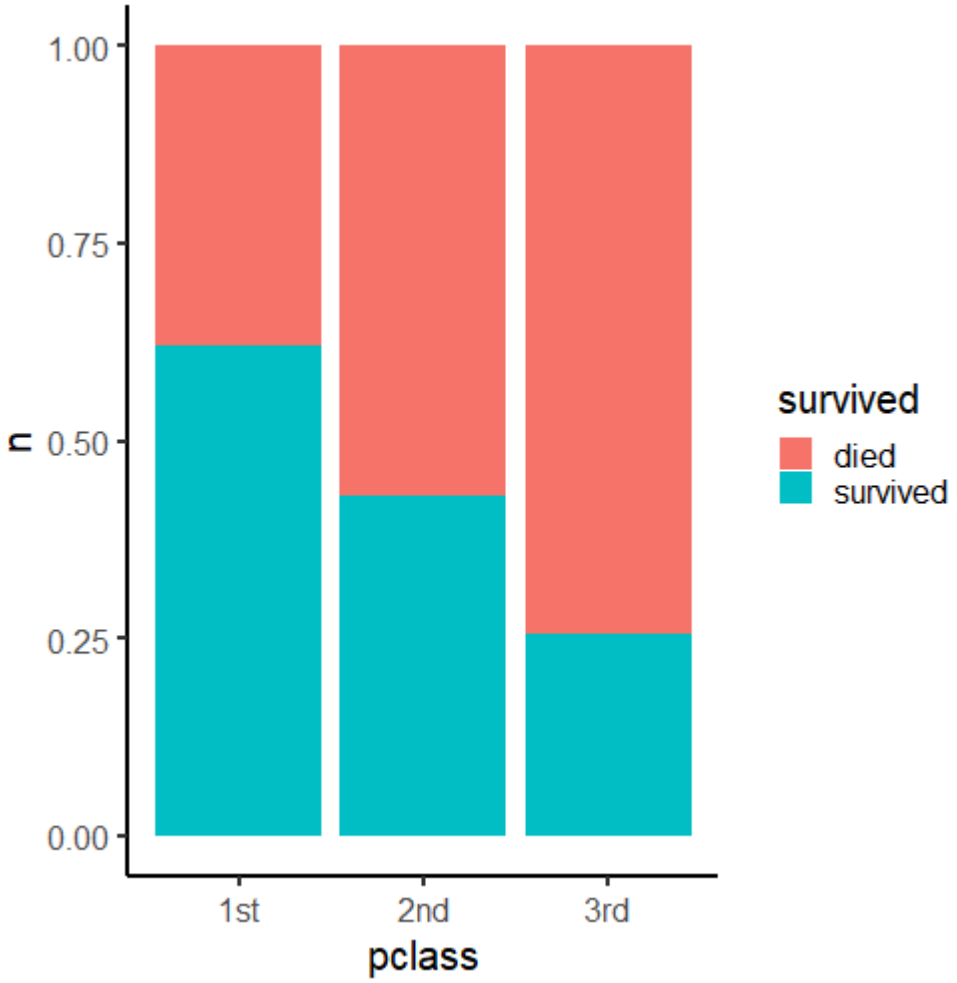
Data on passengers on the RMS Titanic, excluding the crew and some individual identifier variables.

- **pclass** a factor with levels `1st 2nd 3rd`
- **survived** a factor with levels `died survived`
- **sex** a factor with levels `female male`
- **age** passenger age in years (or fractions of a year, for children), a numeric vector; age is missing for 263 of the passengers
- **sibsp** number of siblings or spouses aboard, integer, 0 to 8
- **parch** number of parents or children aboard, integer, 0 to 6

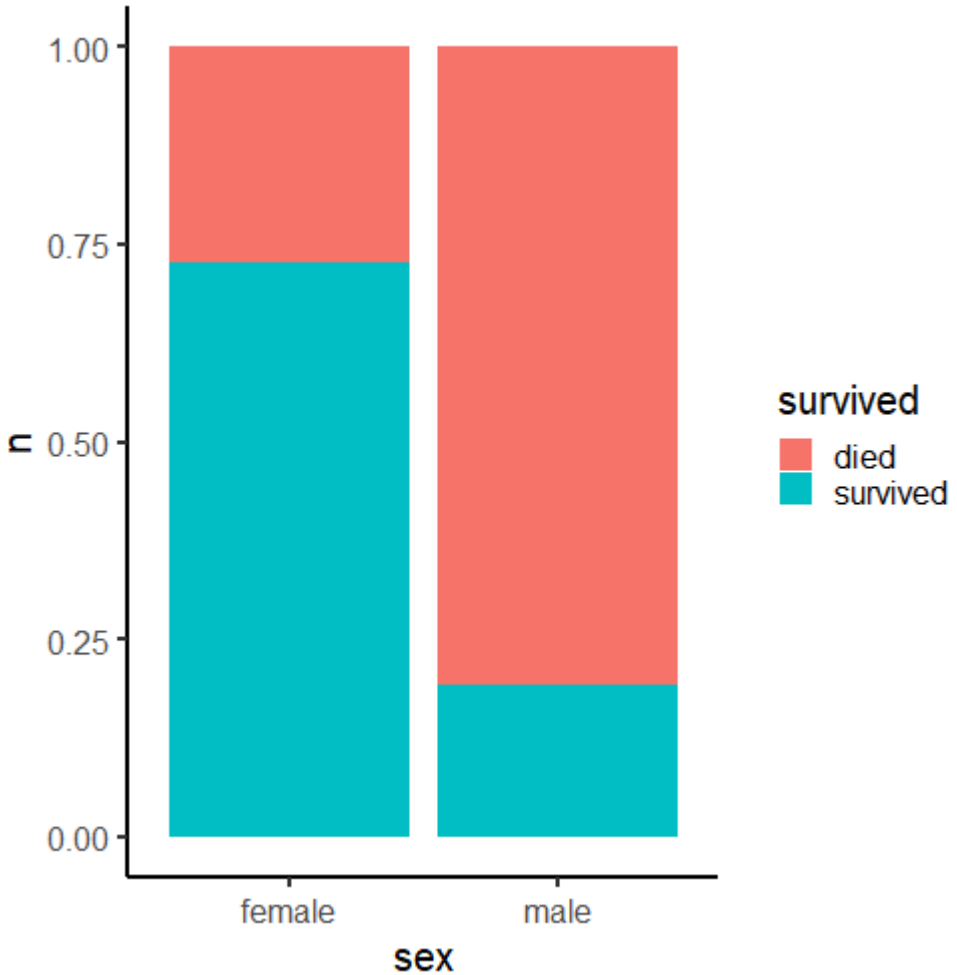
```r
# install.packages(vcdExtra)
data("Titanicp", package = "vcdExtra")
glimpse(Titanicp)
```

```
## Rows: 1,309
## Columns: 6
## $ pclass   <fct> 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1…
## $ survived <fct> survived, survived, died, died, died, survived, survived, die…
## $ sex      <fct> female, male, female, male, female, male, female, male, femal…
## $ age      <dbl> 29.0000, 0.9167, 2.0000, 30.0000, 25.0000, 48.0000, 63.0000, …
## $ sibsp    <dbl> 0, 1, 1, 1, 1, 0, 1, 0, 2, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1…
## $ parch    <dbl> 0, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1…
```

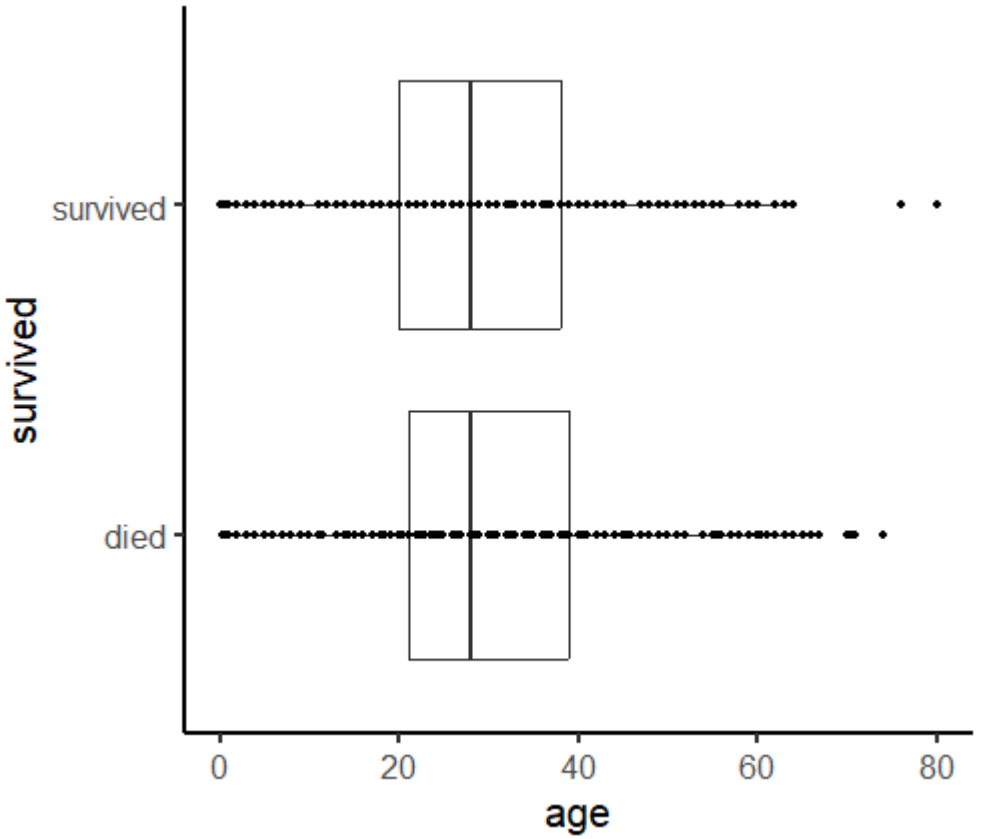
```
Titanicp |> group_by(survived, pclass) |> count() |>
  ggplot(aes(x = pclass, y = n, fill = survived)) +
  geom_bar(stat = "identity", position = "fill")
```



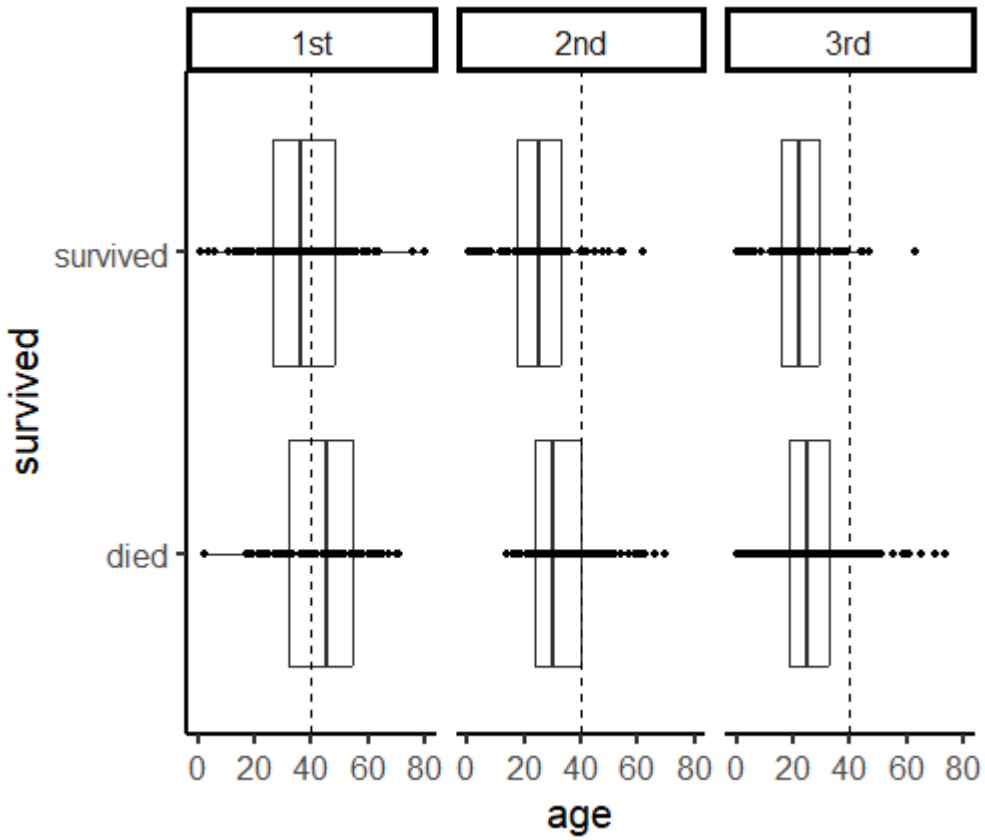
```
Titanicp |> group_by(survived, sex) |> count() |>
  ggplot(aes(x = sex, y = n, fill = survived)) +
  geom_bar(stat = "identity", position = "fill")
```



```
Titanicp |>
  ggplot() +
  aes(x = age, y = survived) +
  geom_boxplot()+
  geom_point()
```




```
Titanicp |>
  ggplot(aes(x = age, y = survived)) +
  geom_boxplot()+
  geom_point() +
  facet_grid(~pclass) +
  geom_vline(xintercept = 40, linetype = 2)
```



Logistic regression

- A logistic regression model begins with,

$$y_i|\boldsymbol{x}_i \sim \text{Bernoulli}\left(\frac{\exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})}{1 + \exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})}\right).$$

- If we had a new observation vector $(x_{i1}, \ldots, x_{ip})$ and we knew the $(\beta_1, \ldots, \beta_p)$ vector, we could calculate the probability that the corresponding $Y = 1$

$$P(Y = 1|\boldsymbol{x}) = \frac{\exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})}{1 + \exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})}.$$

- If this probability is greater than 0.5, we would make the prediction $\hat{Y} = 1$, otherwise we would predict $\hat{Y} = 0$.

Modelling the titanic data

- Start by converting survival to 0/1 (numeric) variable

```
x = Titanicp |> mutate(survived = ifelse(survived == "survived", 1, 0))
glimpse(x)
```

```
## Rows: 1,309
## Columns: 6
## $ pclass   <fct> 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1…
## $ survived <dbl> 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1…
## $ sex      <fct> female, male, female, male, female, male, female, male, femal…
## $ age      <dbl> 29.0000, 0.9167, 2.0000, 30.0000, 25.0000, 48.0000, 63.0000, …
## $ sibsp    <dbl> 0, 1, 1, 1, 1, 0, 1, 0, 2, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1…
## $ parch    <dbl> 0, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1…
```

- We treat `survived` and `died` as successes and failures from a Bernoulli (binomial) distribution where the probability of success is given by a transformation of a linear model of the predictors.

Fit a logistic regression model

```
glm1 = glm(survived ~ pclass + sex + age, family = binomial, data = x)
summary(glm1)
```

```
##
## Call:
## glm(formula = survived ~ pclass + sex + age, family = binomial,
##     data = x)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.522074   0.326702  10.781  < 2e-16 ***
## pclass2nd   -1.280570   0.225538  -5.678 1.36e-08 ***
## pclass3rd   -2.289661   0.225802 -10.140  < 2e-16 ***
## sexmale     -2.497845   0.166037 -15.044  < 2e-16 ***
## age         -0.034393   0.006331  -5.433 5.56e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1414.62  on 1045  degrees of freedom
## Residual deviance:  982.45  on 1041  degrees of freedom
##   (263 observations deleted due to missingness)
## AIC: 992.45
##
## Number of Fisher Scoring iterations: 4
```

Checking for significance

Before we start to interpret our model and make predictions, we might want to know if we can drop any of the variables from the model. This is equivalent to testing

$$H_0 \colon \beta_j = 0$$

against the alternative

$$H_1 \colon \beta_j \neq 0$$

We test that $\beta_j = 0$ if the esimated value for β_j, that is $\widehat{\beta}_j$ is large (in absolute) magnitude we say that β_j is significantly different from 0. Formally we do a test

$$Z = \frac{\widehat{\beta}_j}{\mathrm{SE}(\widehat{\beta}_j)} \overset{\text{approx}}{\sim} N(0,1)$$

Where we find the test statistic and p-value in the summary output.

Test statistics will be approximately $N(0,1)$ distributed.

```
##
## Call:
## glm(formula = survived ~ pclass + sex + age, family = binomial,
##     data = x)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6399  -0.6979  -0.4336   0.6688   2.3964
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.522074   0.326702  10.781  < 2e-16 ***
## pclass2nd   -1.280570   0.225538  -5.678 1.36e-08 ***
## pclass3rd   -2.289661   0.225802 -10.140  < 2e-16 ***
## sexmale     -2.497845   0.166037 -15.044  < 2e-16 ***
## age         -0.034393   0.006331  -5.433 5.56e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1414.62  on 1045  degrees of freedom
## Residual deviance:  982.45  on 1041  degrees of freedom
##   (263 observations deleted due to missingness)
## AIC: 992.45
```

P-values: Convention is, if the p-value is below 5% then the coefficient is significantly different from 0, i.e., reject H_0 in favour of H_1.

```
##
## Call:
## glm(formula = survived ~ pclass + sex + age, family = binomial,
##     data = x)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6399  -0.6979  -0.4336   0.6688   2.3964
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.522074   0.326702  10.781  < 2e-16 ***
## pclass2nd   -1.280570   0.225538  -5.678 1.36e-08 ***
## pclass3rd   -2.289661   0.225802 -10.140  < 2e-16 ***
## sexmale     -2.497845   0.166037 -15.044  < 2e-16 ***
## age         -0.034393   0.006331  -5.433 5.56e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1414.62  on 1045  degrees of freedom
## Residual deviance:  982.45  on 1041  degrees of freedom
##   (263 observations deleted due to missingness)
## AIC: 992.45
```

Write down the fitted model

```
glm1
```

```
##
## Call:  glm(formula = survived ~ pclass + sex + age, family = binomial,
##     data = x)
##
## Coefficients:
## (Intercept)    pclass2nd    pclass3rd      sexmale          age
##     3.52207     -1.28057     -2.28966     -2.49784     -0.03439
##
## Degrees of Freedom: 1045 Total (i.e. Null);  1041 Residual
##   (263 observations deleted due to missingness)
## Null Deviance:        1415
## Residual Deviance: 982.5     AIC: 992.5
```

$$\text{logit}(p) = 3.5 - 1.3\,\text{pclass2nd} - 2.3\,\text{pclass3rd} - 2.5\,\text{sexmale} - 0.03\,\text{Age}$$

What's this logit function?

The **logit** function is our **link** from a linear combination of the predictors to the probability of the outcome being equal to 1.

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

- It's the log-odds!
- Our estimated coefficients are therefore interpreted as changes in the **log-odds**.
- I.e. we can write out fitted model as

$$\log\left(\frac{p}{1-p}\right) = 3.5 - 1.3\,\text{pclass2nd} - 2.3\,\text{pclass3rd} - 2.5\,\text{sexmale} - 0.03\,\text{Age}$$

The logistic function (introduced earlier) is the inverse of the logit function.

Interpreting our coefficients

$$\log\left(\frac{p}{1-p}\right) = 3.5 - 1.3\,\text{pclass2nd} - 2.3\,\text{pclass3rd} - 2.5\,\text{sexmale} - 0.03\,\text{age}$$

- **Intercept** the log-odds of survival for an individual travelling in 1st class who is female and aged zero years old.
- Holding sex and age constant, the `pclass2nd` coefficient represents the **difference** in the log-odds between someone travelling in 1st class and someone travelling in 2nd class. In this case, it's **negative**, so we're saying that your odds of survival were lower if you travelled in second class, relative to those who travelled in first class.
- Holding class and age constant, the `sexmale` coefficient represents the **difference** in the log-odds between males and females. It is **negative,** so we can say that if you were a male, your odds of survival were **lower** than if you were a female.
- The `age` coefficient is also negative, which implies that older people had lower odds of survival than younger people. Specifically, on average, for each additional year older you are, the log-odds of survival decreased by 0.03, holding class and sex constant.

Interpreting our coefficients

$$\log\left(\frac{p}{1-p}\right) = 3.5 - 1.3\,\text{pclass2nd} - 2.3\,\text{pclass3rd} - 2.5\,\text{sexmale} - 0.03\,\text{age}$$

```
coef(glm1) |>
  exp() |>
  signif(2)
```

```
## (Intercept)   pclass2nd   pclass3rd     sexmale         age
##      34.000       0.280       0.100       0.082       0.970
```

What do our predictions mean?

$$\log\left(\frac{p}{1-p}\right) = 3.5 - 1.3\,\text{pclass2nd} - 2.3\,\text{pclass3rd} - 2.5\,\text{sexmale} - 0.03\,\text{Age}$$

We can predict the log-odds for a newborn male travelling in first class

- `pclass2nd = 0`, `pclass3rd = 0`, `sexmale = 1`, `age = 0`

$$\log\left(\frac{p}{1-p}\right) = 3.5 - 1.3 \times 0 - 2.3 \times 0 - 2.5 \times 1 - 0.03 \times 0 = 3.5 - 2.5 = 1$$

So the odds of survival for a newborn male travelling in first class are 1.

```
new_data = data.frame(pclass = "1st", sex = "male", age = 0)
predict(glm1, newdata = new_data, type = "link")
```

```
##        1
## 1.024229
```

Can we work out the estimated probability of survival for a newborn male travelling in first class?

$$
\begin{aligned}
\log\left(\frac{p}{1-p}\right) &= 1 \\
\left(\frac{p}{1-p}\right) &= \exp(1) \\
p &= \exp(1) - p\exp(1) \\
p + p\exp(1) &= \exp(1) \\
p &= \frac{\exp(1)}{1+\exp(1)} \approx 0.73
\end{aligned}
$$

```
new_data = data.frame(pclass = "1st", sex = "male", age = 0)
predict(glm1, newdata = new_data, type = "response")
```

```
##         1
## 0.7357956
```

Note that we've used the **logistic** function to transform back to obtain an estimate of the **probability** (from the output of our model which is an estimate of the log-odds).

Extensions to Logistic regression

- Multinomial logistic regression: When the response variable is an unordered category. This type of model can be fit using the vglm() function in the VGAM package.
- Ordinal logistic regression: When the response variable is an unordered category. This type of model can be fit using the polr() function in the MASS package.

Using linear regression with count data

In the following example we look at the number of awards earned by students at one high school, y_i with $n = 200$ students. Predictors of the number of awards earned include the type of program in which the student was enrolled (e.g., vocational, general or academic) and the score on their final exam in math.

```
head(dat)
```

```
##    id num_awards       prog math
## 1  45          0 Vocational   41
## 2 108          0    General   41
## 3  15          0 Vocational   44
## 4  67          0 Vocational   42
## 5 153          0 Vocational   40
## 6  51          0    General   42
```

```
summary(dat)
```

```
##        id        num_awards          prog          math      
##  1      :  1   Min.   :0.00   General   : 45   Min.   :33.0  
##  2      :  1   1st Qu.:0.00   Academic  :105   1st Qu.:45.0  
##  3      :  1   Median :0.00   Vocational: 50   Median :52.0  
##  4      :  1   Mean   :0.63                    Mean   :52.6  
##  5      :  1   3rd Qu.:1.00                    3rd Qu.:59.0  
##  6      :  1   Max.   :6.00                    Max.   :75.0  
##  (Other):194
```

Let's look at `math` as a predictor in a linear model.

```
res1 <- lm(num_awards~math, data=dat)
summary(res1)
```

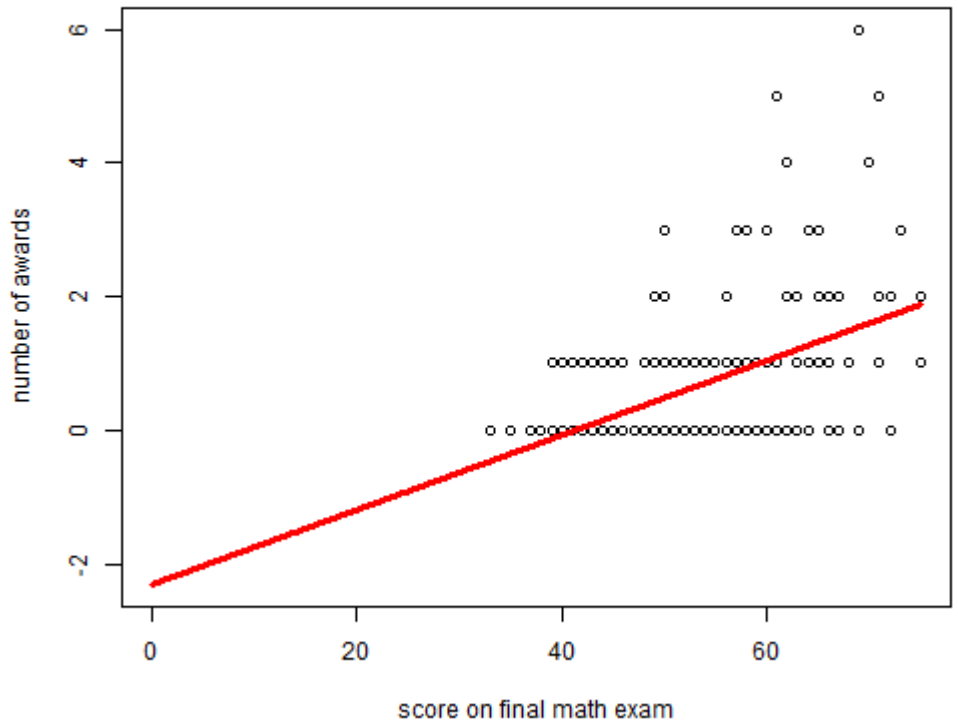
```
##
## Call:
## lm(formula = num_awards ~ math, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7113 -0.5940 -0.0968  0.2901  4.4563
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.311023   0.370566  -6.236 2.65e-09 ***
## math         0.055865   0.006931   8.061 7.06e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9159 on 198 degrees of freedom
## Multiple R-squared:  0.2471,    Adjusted R-squared:  0.2433
## F-statistic: 64.97 on 1 and 198 DF,  p-value: 7.059e-14
```

The fitted model is $\hat{y}_i = -2.3110231 + 0.0558652 \times \text{math}_i$ with both coefficients statistically different from 0 at the 5% level.

Using linear regression with count data

If $\text{math}_i = 0$ what goes wrong?

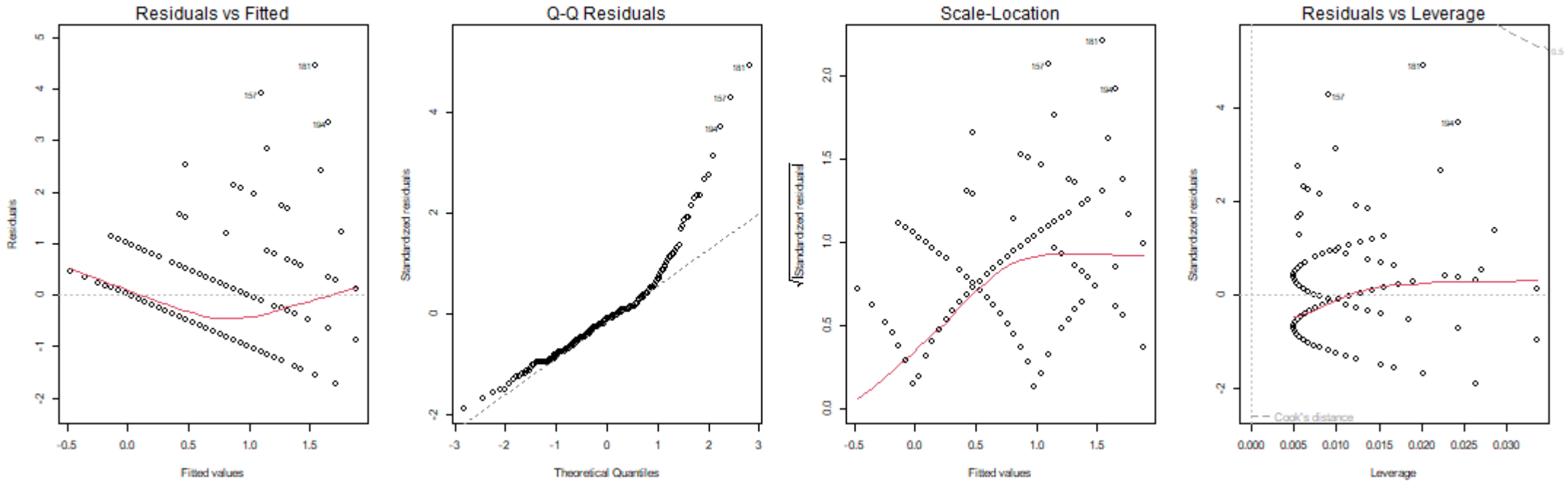
```
x    <- seq(0,max(dat$math),,1000)
yhat <- res1$coef[1] + res1$coef[2]*x
plot(dat$math,
     dat$num_awards,
     xlim=range(x),
     ylim=range(c(dat$num_awards,yhat)),
     xlab="score on final math exam",
     ylab="number of awards")
lines(x,yhat,col="red",lwd=3)
```



- If we go below a math mark of about 30 we start predicting a negative number of awards.
- In this context it might not matter much.
- However, in other contexts making positive predictions might be vital.

Using linear regression with count data - Diagnostics

```
par(mfrow = c(1, 4))
plot(res1)
```



The diagnostic plots look like a disaster with outliers, the tail distribution of the residuals reveal non-normally distribution of errors.

Poisson distribution

The Poisson distribution has one parameter $\lambda > 0$ so that the probability that the response Y takes the count value $y = 0, 1, 2, \ldots$ is

$$P(Y = k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{4}$$

So for example when $\lambda = 3$

$$P(Y = 0) = \frac{3^0 e^{-3}}{0!} = e^{-3} \approx 0.0498 \tag{5}$$

$$P(Y = 1) = \frac{3^1 e^{-3}}{1!} = 3e^{-3} \approx 0.1494 \tag{6}$$

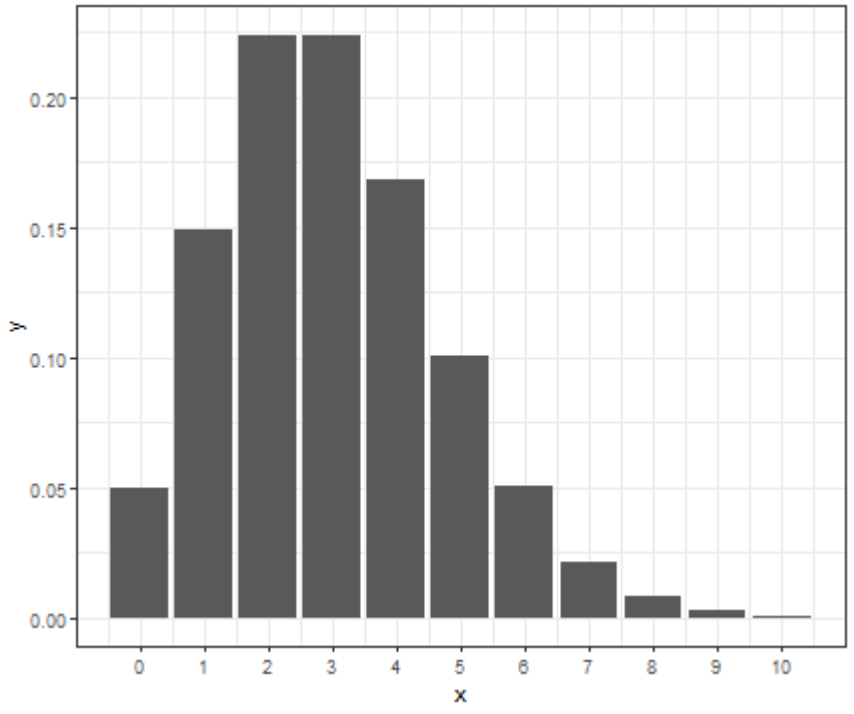
$$P(Y = 2) = \frac{3^2 e^{-3}}{2!} = \frac{9e^{-3}}{2} \approx 0.2240 \tag{7}$$

Expected value or mean and variance are respectively

$$\mathbb{E}(Y) = \sum_{y=0}^{\infty} yP(Y = y) = \lambda \qquad \text{and} \qquad \text{Var}(Y) = \lambda.$$

Poisson distribution

A histogram of the probabilities when $\lambda = 3$ is below.



A sequence of $n = 10$ Poisson distributed values with $\lambda = 3$ is given below

```
n <- 10
lambda <- 3
rpois(n,lambda)
```

```
##  [1] 5 5 2 2 1 1 5 2 3 3
```

Increasing λ means on average higher counts, and decreasing λ means on average that the counts are lower. The probabilities can be calculated using

```
round(dpois(0:10,lambda),4)
```

```
##  [1] 0.0498 0.1494 0.2240 0.2240 0.1680 0.1008 0.0
## [11] 0.0008
```

From linear models to generalized linear models

For the Poisson distribution we had that

$$P(Y = k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{8}$$

The parameter $\lambda > 0$ controls the magnitude of the counts.

Suppose that we instead use

$$\log(\lambda_i) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} \tag{9}$$

- Now the mean of the Poisson distribution is guaranteed, through construction, to have a positive mean!!!!
- Because log of the parameter λ is used, this is referred to as using the "log"-link.
- Other links are possible, but the explanation is technical, and beyond the scope of this course.

From linear models to generalized linear models

However, we can't use the R function `lm()` to fit this model any more.

Instead we use the R function `glm()`.

For comparison we used

```
res1 <- lm(num_awards~math, data=dat)
```

for linear models. For generalized linear models for Poisson data we use almost identical syntax...

```
res2 <- glm(num_awards~math, data=dat, family = poisson)
```

So we are fitting a model of the form

$$y_i \sim \text{Poisson}(\lambda_i) \quad \text{with} \qquad \lambda_i = \exp(\beta_0 + \beta_1 \times \text{math}_i)$$

From linear models to generalized linear models

```
res2 <- glm(num_awards~math,
            data=dat,
            family = poisson)
summary(res2)
```

```
##
## Call:
## glm(formula = num_awards ~ math, family = poisson, data = dat)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.333532   0.591261  -9.021   <2e-16 ***
## math         0.086166   0.009679   8.902   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 204.02  on 198  degrees of freedom
## AIC: 384.08
##
## Number of Fisher Scoring iterations: 6
```

So the fitted model is

$$y_i \sim \text{Poisson}(\widehat{\lambda}_i)$$

with

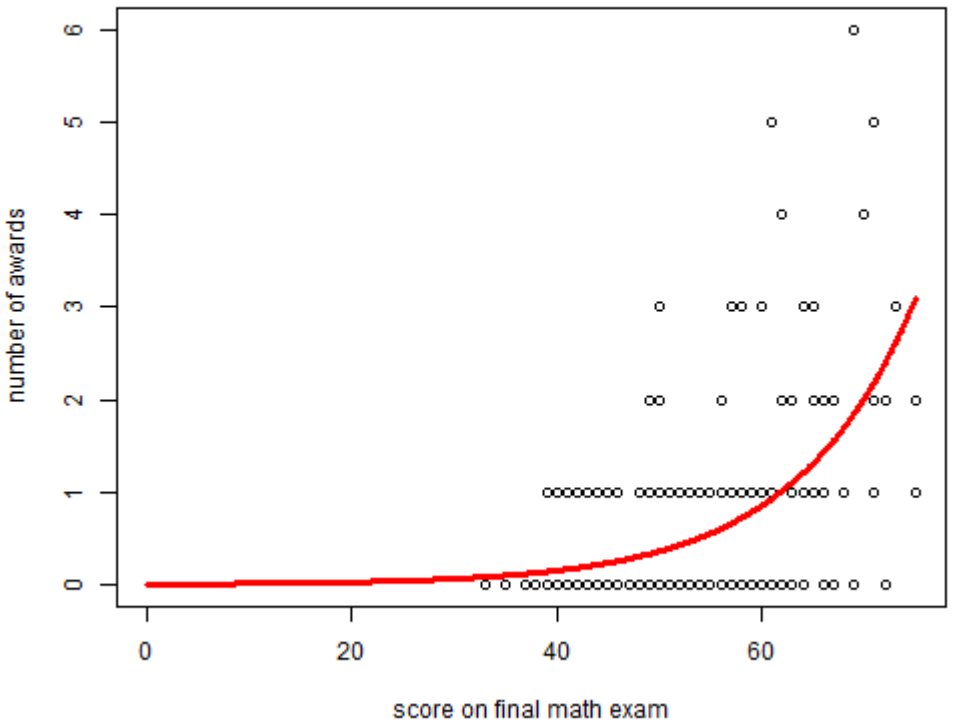
$$\widehat{\lambda}_i = \exp(-5.3335321 + 0.0861656 \times \text{math}_i)$$

and both regression coefficients being statistically different to 0 at the 5 percent level.

Plotting the results

```
x <- seq(0,max(dat$math),,1000)
df <- data.frame(math=x)
yhat <- predict(res2,
                newdata=df,
                type="response")

plot(dat$math,
     dat$num_awards,
     xlim=range(x),
     ylim=range(c(dat$num_awards,yhat)),
     xlab="score on final math exam",
     ylab="number of awards")
lines(x,yhat,col="red",lwd=3)
```



Interpretation - Poisson linear model

So the predicted mean of the fit was of the form

$$\widehat{\lambda}_i = \exp(-5.3335321 + 0.0861656 \times \text{math}_i)$$

We can interpret the coefficient $\widehat{\beta}_1 = 0.0861656$

If we increase the value of the predictor $x = \text{math}_i$ by 1 unit, then the mean will increase by a **factor** of $\exp(0.0861656) = 1.0899868$.

Poisson linear model with multiple predictors

We are going to now fit a slightly more complicated model where we use both `math` and `prog` as predictors

The complication here is that `prog` is a categorical variable with 3 levels ("General", "Academic", "Vocational").

In R we use the syntax

```
res3 <- glm(num_awards ~ math + prog,
            data=dat,
            family = poisson)
```

to fit this model. But what model are we actually fitting?

Note that

```
levels(dat$prog)
```

```
## [1] "General"    "Academic"   "Vocational"
```

Poisson linear model with multiple predictors

Since `prog` is categorical with $K = 3$ factors, R creates two dummy variables

- $I(\text{prog}_i = \text{Academic})$ and
- $I(\text{prog}_i = \text{Vocational})$.

where $I(\text{prog}_i = \text{General})$ is treated as the base category.

The fitted model

$$y_i \sim \text{Poisson}(\lambda_i)$$

with

$$\lambda_i = \exp(\beta_0 + \beta_1 \text{math}_i + \beta_2 I(\text{prog}_i = \text{Academic}) + \beta_3 I(\text{prog}_i = \text{Vocational}))$$

Let's fit this model in R.

```
res3 <- glm(num_awards~math + prog, data=dat, family = poisson)
summary(res3)
```

```
##
## Call:
## glm(formula = num_awards ~ math + prog, family = poisson, data = dat)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -5.24712    0.65845  -7.969 1.60e-15 ***
## math            0.07015    0.01060   6.619 3.63e-11 ***
## progAcademic    1.08386    0.35825   3.025  0.00248 **
## progVocational  0.36981    0.44107   0.838  0.40179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 189.45  on 196  degrees of freedom
## AIC: 373.5
##
## Number of Fisher Scoring iterations: 6
```

The fitted model

The fitted model uses

$$\widehat{\lambda}_i = \exp(-5.247 + 0.07\text{math}_i + 1.084I(\text{prog}_i = \text{Academic}) + 0.37I(\text{prog}_i = \text{Vocational}))$$

with

- $\widehat{\beta}_0 = -5.247$
- $\widehat{\beta}_1 = 0.07$
- $\widehat{\beta}_2 = 1.084$
- $\widehat{\beta}_3 = 0.37$

where β_0 (p-value 1.60×10^{-15}), β_1 (p-value 3.63×10^{-11}), and β_2 (p-value 0.00248) being significantly different from 0 at the 5% level, and β_3 (p-value 0.40179) not being significantly different from 0 at the 5% level.

Interpreting the fitted coefficients

$$\widehat{\lambda}_i = \exp(-5.3335321 + 0.0861656 \times \text{math}_i)$$

- For every unit increase of math_i the mean number of awards increases by a factor of $\exp(0.07) = 1.073$.
- If $\text{prog}_i = \text{Academic}$ the mean number of awards increases by a factor of $\exp(1.084) = 2.956$ compared to if $I(\text{prog}_i = \text{General})$.
- If $\text{prog}_i = \text{Vocational}$ the mean number of awards increases by a factor of $\exp(0.37) = 1.447$ compared to if $I(\text{prog}_i = \text{General})$.

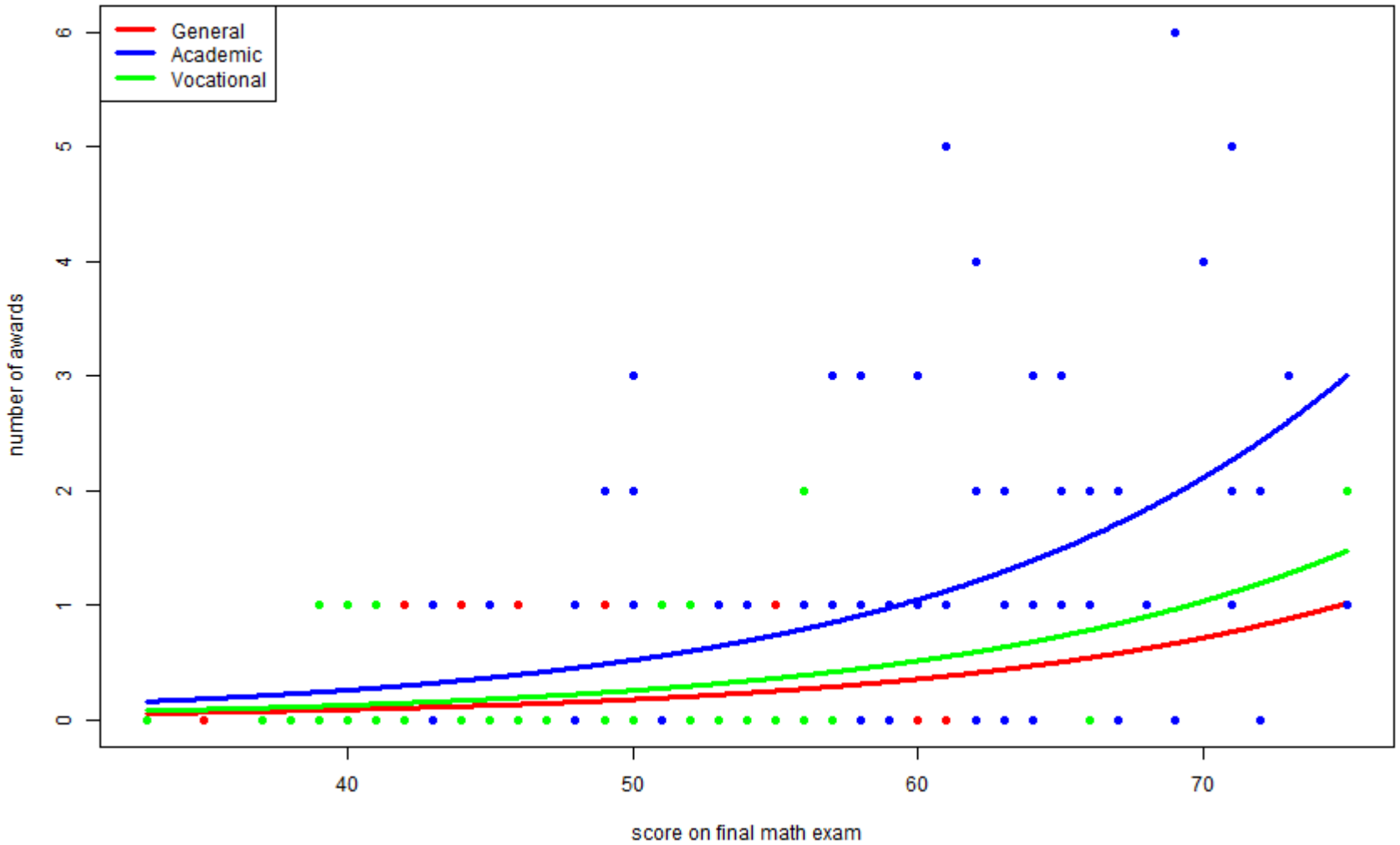
Prediction

We can make predictions if we have all of the covariates.

We need to enter all of the new covariates into a new data.frame object and pass that to the predict function

```
new_data = data.frame(math = 50, prog="Vocational")
predict(res3, newdata = new_data, type = "link")
```

```
##          1
## -1.369695
```


General
Academic
Vocational
number of awards
0
1
2
3
4
5
6
40
50
60
70
score on final math exam

Model selection

There are two commonly used criterion for model selection.

The Akaike Information Criterion (AIC)

```
AIC(res2)
```

```
## [1] 384.0762
```

```
AIC(res3)
```

```
## [1] 373.5045
```

and the Bayesian Information Criterion (BIC)

```
BIC(res2)
```

```
## [1] 390.6728
```

```
BIC(res3)
```

```
## [1] 386.6978
```

- For both of these Criterion the lower value are better.
- So both would choose the model with the `math` and `prog` as predictors as being the better model compared to the model with only `math` as a predictor.

Poisson regression variants

Other variants of regression for count data include:

- Negative binomial regression: Introduces an additional parameter which can be used when the counts are overdispersed (mean < variance). This type of model can be fit using the glm.nb() function in the MASS package or the manyglm() function in the mvabund package.
- Zero inflated Poisson: Some count data contains an excessive number of zeroes. The package pscl has the function zeroinfl() for dealing with this.

Created using R Markdown with flair by **xaringan**