ANCOVA

Kevin Healy

2023-11-06

In this script we will learn more about fitting linear models. In particular, what if we want to fit a line to multiple groups. Lets use the iris data to do this.

```
iris_data <- iris
```

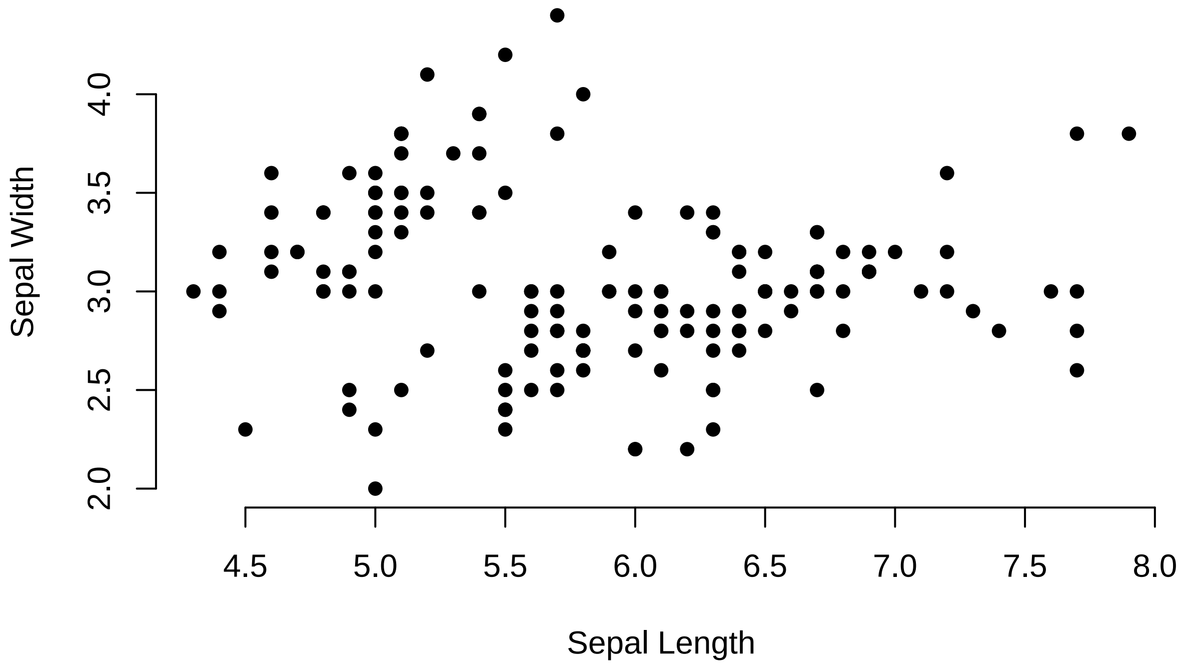
We will also use the dose.p() function later on so we will load it up.

```
library(MASS)
```

R exploring the data

Lets say we were interested in how the sepals on iris plants grow, in particular I might want to know whether longer sepals are also wider. The first thing I could do is plotthe data.

```
plot(iris_data$Sepal.Width ~ iris_data$Sepal.Length,
     pch = 16,
     bty = "n",
     xlab = "Sepal Length",
     ylab = "Sepal Width")
```



It doesn’t look like there is a clear relationship (think back to the R-squared game). Lets fit a simple model anyway.

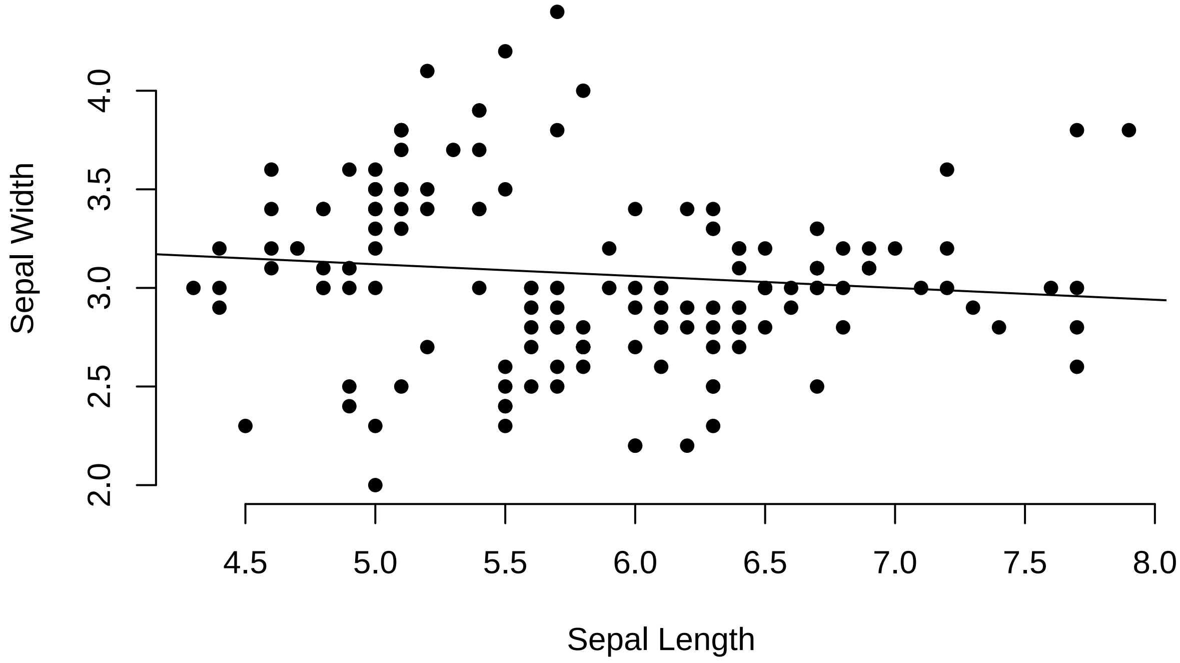
```
mod_1 <- lm(Sepal.Width ~ Sepal.Length,
            data = iris_data)

summary(mod_1)
```

```
##
## Call:
## lm(formula = Sepal.Width ~ Sepal.Length, data = iris_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1095 -0.2454 -0.0167  0.2763  1.3338
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.41895    0.25356   13.48   <2e-16 ***
## Sepal.Length -0.06188    0.04297   -1.44    0.152
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.4343 on 148 degrees of freedom
## Multiple R-squared:  0.01382,    Adjusted R-squared:  0.007159
## F-statistic: 2.074 on 1 and 148 DF,  p-value: 0.1519
```

We can see from the summary that while the intercept is significantly different to zero, which in this case is not biologically interesting, the slope is not significantly different to zero. This means we cannot reject the NULL that there is no relationship between Sepal length and Sepal width. Also notice how low the R-squared value is suggesting the data is not very close to our fitted line.

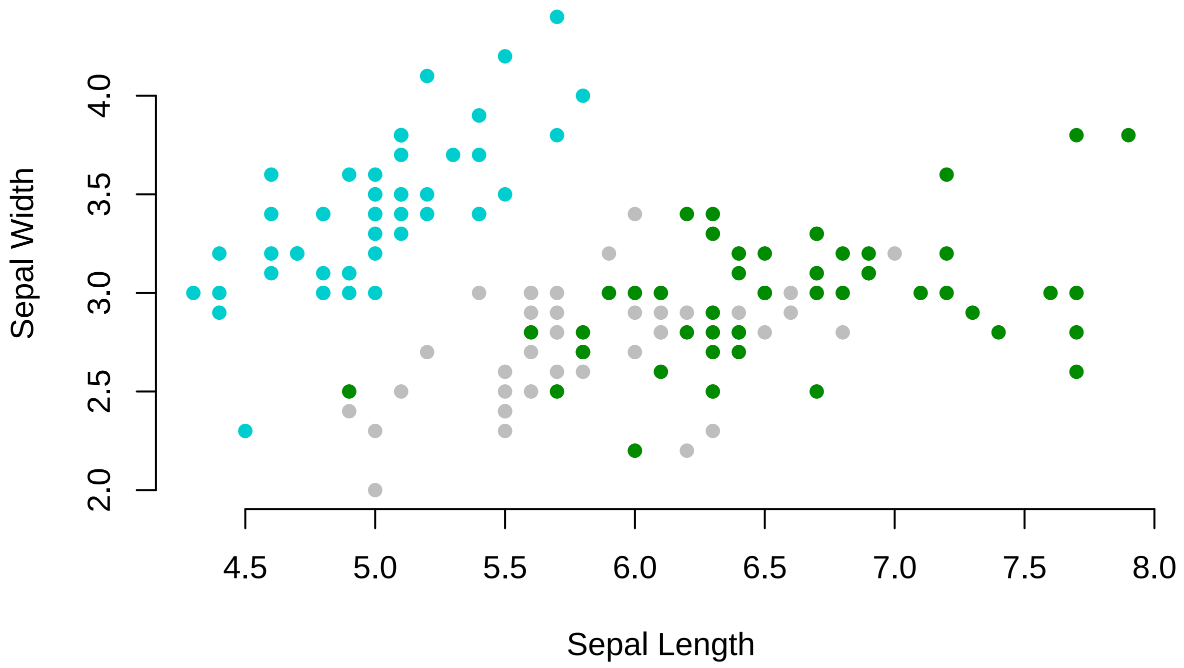
If we plot this model on the scatter plot it will look like a straight line.

```
plot(iris_data$Sepal.Width ~ iris_data$Sepal.Length,
     pch = 16,
     bty = "n",
     xlab = "Sepal Length",
     ylab = "Sepal Width")

abline(3.42, -0.06)
```



However, lets plot the same graph but now we will colour the species in.

```
plot(iris_data$Sepal.Width ~ iris_data$Sepal.Length,
     bty = "n",
     xlab = "Sepal Length",
     ylab = "Sepal Width",
     col = "white")

# colour the points for setosa
points(iris_data[iris_data$Species == "setosa", "Sepal.Width"] ~
         iris_data[iris_data$Species == "setosa", "Sepal.Length"],
       col = "cyan3",
       pch = 16)


# colour the points for versicolor
points(iris_data[iris_data$Species == "versicolor", "Sepal.Width"] ~
         iris_data[iris_data$Species == "versicolor", "Sepal.Length"],
       col = "grey",
       pch = 16)


# colour the points for virginica
points(iris_data[iris_data$Species == "virginica", "Sepal.Width"] ~
         iris_data[iris_data$Species == "virginica", "Sepal.Length"],
       col = "green4",
       pch = 16)
```



When we plot it like this it looks like the there is more going on than before and fitting a single line won't quite capture the relationship between sepal length ad width. Instead, it might make more sense to fit a seperate line to each species seperatly.

We can do that in linear models by adding the catagorical varible into the model. By adding this R will fit a line for each of the levels in catagorical varible added. Before fitting lets check what the three species are and also what order they are in.

```
levels(iris_data$Species)
```

```
## [1] "setosa"     "versicolor" "virginica"
```

In this case we can see that there are three species so we will be fitting 3 lines. we can also see that "setosa" appears first and so this will be the first line fitted or the baseline. Whe testing the NUll in this model we will compare the other lines to this baseline. Lets fit the mnodel by adding the explanitory varaible + Species

```
mod_2 <- lm(Sepal.Width ~ Sepal.Length + Species,
            data = iris_data)

summary(mod_2)
```

```
##
## Call:
## lm(formula = Sepal.Width ~ Sepal.Length + Species, data = iris_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95096 -0.16522  0.00171  0.18416  0.72918
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.67650    0.23536   7.123 4.46e-11 ***
## Sepal.Length       0.34988    0.04630   7.557 4.19e-12 ***
## Speciesversicolor -0.98339    0.07207 -13.644  < 2e-16 ***
## Speciesvirginica  -1.00751    0.09331 -10.798  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.289 on 146 degrees of freedom
## Multiple R-squared:  0.5693, Adjusted R-squared:  0.5604
## F-statistic: 64.32 on 3 and 146 DF,  p-value: < 2.2e-16
```

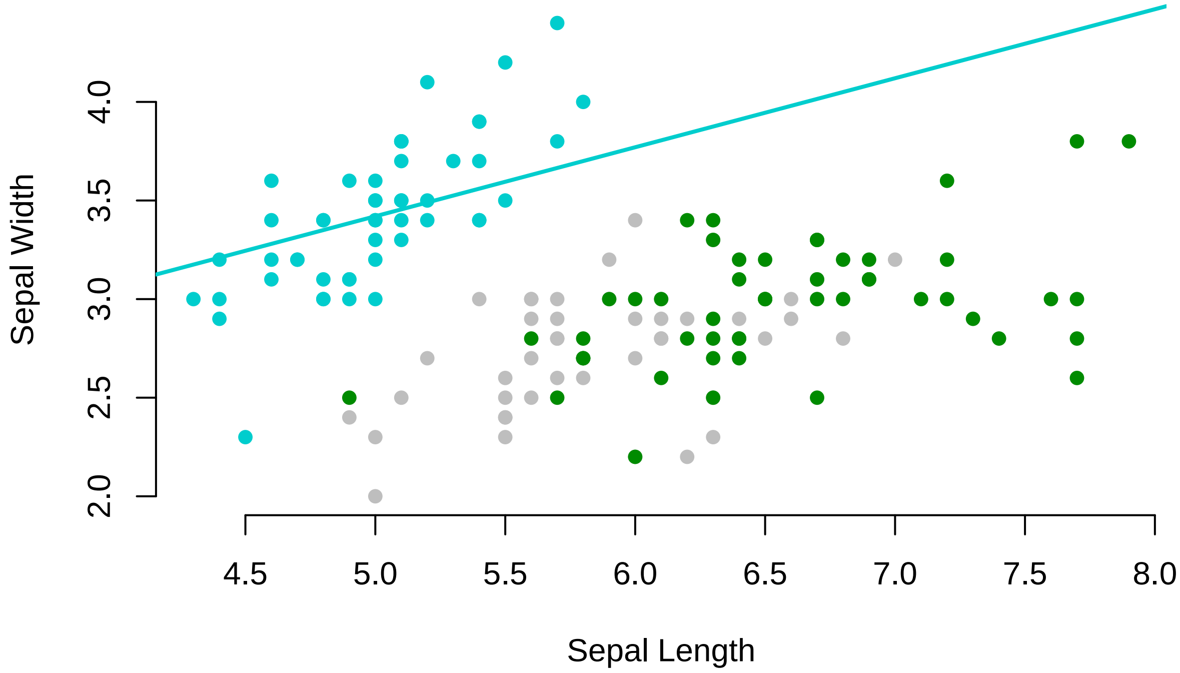
Notice how now we see very different results and straight away we can see the R-squared is much higher suggesting this model is much better compared to our previous model. lets break it down a little.

The line with Intercept is now the estimated intercept value for the baseline which we know is for the species setosa as its the first in the list of levels(iris_data$Species). We can see the intercept for this line is 1.67 and that as the p-value is less than zero its significantly different to zero.

The next line is the slope for the base line, hence its the slope for the line fitted for setosa. The estimate for the slope is 0.35, meaning that for every cm we increase in sepal length the width increases by 0.35cm for the species setosa.
We can also see its significantly different from the NUll which is a line of the slope zero.

Before looking at the rest lets plot this line.

```r
plot(iris_data$Sepal.Width ~ iris_data$Sepal.Length,
     bty = "n",
     xlab = "Sepal Length",
     ylab = "Sepal Width",
     col = "white")

# colour the points for setosa
points(iris_data[iris_data$Species == "setosa", "Sepal.Width"] ~
         iris_data[iris_data$Species == "setosa", "Sepal.Length"],
       col = "cyan3",
       pch = 16)


# colour the points for versicolor
points(iris_data[iris_data$Species == "versicolor", "Sepal.Width"] ~
         iris_data[iris_data$Species == "versicolor", "Sepal.Length"],
       col = "grey",
       pch = 16)


# colour the points for virginica
points(iris_data[iris_data$Species == "virginica", "Sepal.Width"] ~
         iris_data[iris_data$Species == "virginica", "Sepal.Length"],
       col = "green4",
       pch = 16)


abline(1.67, 0.35,
       col = "cyan3",
       lty = 1,
       lwd = 2)
```



lets look at the model again and see how the other liens are fitted.

```
summary(mod_2)
```

```
##
## Call:
## lm(formula = Sepal.Width ~ Sepal.Length + Species, data = iris_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95096 -0.16522  0.00171  0.18416  0.72918
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.67650    0.23536   7.123 4.46e-11 ***
## Sepal.Length       0.34988    0.04630   7.557 4.19e-12 ***
## Speciesversicolor -0.98339    0.07207 -13.644  < 2e-16 ***
## Speciesvirginica  -1.00751    0.09331 -10.798  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.289 on 146 degrees of freedom
## Multiple R-squared:  0.5693, Adjusted R-squared:  0.5604
## F-statistic: 64.32 on 3 and 146 DF,  p-value: < 2.2e-16
```

You will notice there are now two extra lines compared to the first model Speciesversicolor and Speciesvirginica. These two lines give the estimates of how differnt the intercepts are for these lines compared to the baseline.

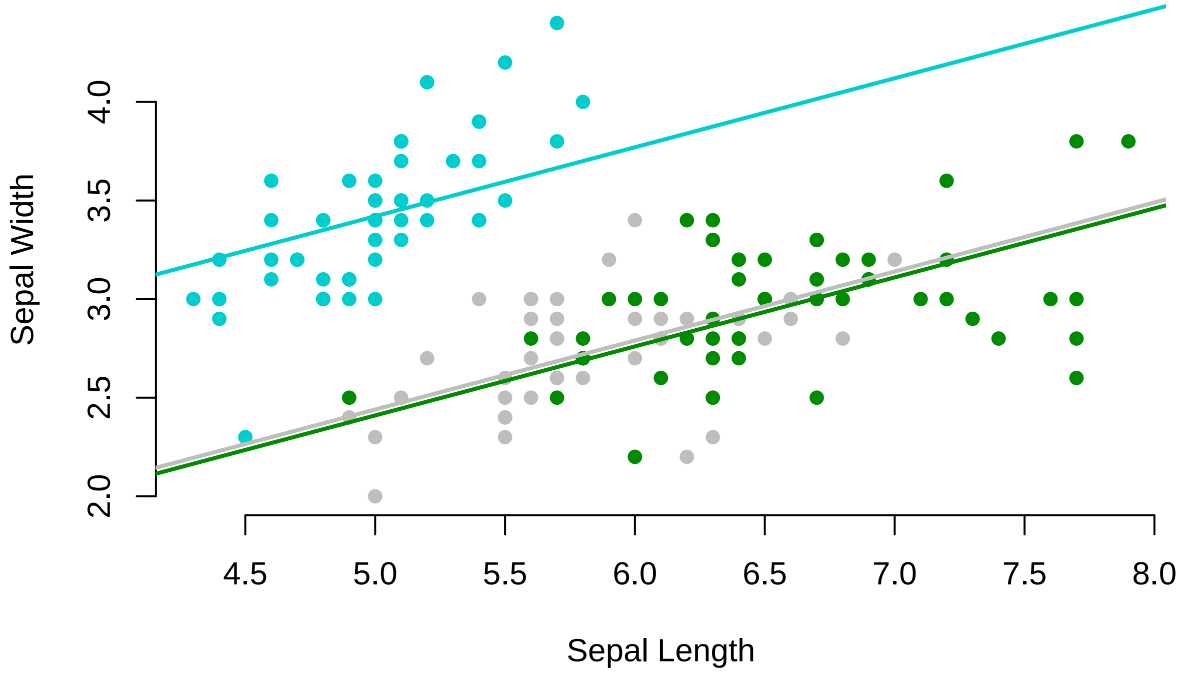
For example, for Speciesversicolor, which represents the line for versicolor has an estimate of -0.98. This value is how differnt the intercept for this line is compared to the baseline. Hence, since the baseline of setosa has an intercept of 1.68, this value tells us the line of versicolor is -0.98 lower than that line and so has an intercept of 1.68 - 0.98 = 0.7. We can also see that this -0.98 is significant. This actualyl means that the intercept for this group is significantly differnt when compared to the baseline (the line fit for setosa).

We can see something similar for virginica which is -1.01 below the baseline and is also signifcantly differnt compared to that baseline.

As we never specified to the model we want differnt slopes for each line in this model all lines have the same slope which we saw was 0.35.

Lets plot these models.

```r
#Set up the graph
plot(iris_data$Sepal.Width ~ iris_data$Sepal.Length,
     bty = "n",
     xlab = "Sepal Length",
     ylab = "Sepal Width",
     col = "white")

# colour the points for setosa
points(iris_data[iris_data$Species == "setosa", "Sepal.Width"] ~
         iris_data[iris_data$Species == "setosa", "Sepal.Length"],
       col = "cyan3",
       pch = 16)


# colour the points for versicolor
points(iris_data[iris_data$Species == "versicolor", "Sepal.Width"] ~
         iris_data[iris_data$Species == "versicolor", "Sepal.Length"],
       col = "grey",
       pch = 16)


# colour the points for virginica
points(iris_data[iris_data$Species == "virginica", "Sepal.Width"] ~
         iris_data[iris_data$Species == "virginica", "Sepal.Length"],
       col = "green4",
       pch = 16)

#fit the line for setosa
abline(1.67, 0.35,
       col = "cyan3",
       lty = 1,
       lwd = 2)

#fit the line for versicolor
abline(1.67 -0.98, 0.35,
       col = "grey",
       lty = 1,
       lwd = 2)

#fit the line for virginica
abline(1.67 -1.01, 0.35,
       col = "green4",
       lty = 1,
       lwd = 2)
```



We can see from the plot that this model captures that setosa has wider sepals but it also captures that indaviduals with longer sepals have naturally wider sepals (rember the varible on the x-axis drives the thing on the y-axis.)

Like before we can make predictions for this. Say I want to estimate the expected sepal width for an indavidual from the species versicolor with a sepal length of 6. I can use the equation of the line Y = MX + C but remeber I need to do it for the correct line so in this case Y = 0.35X + 1.67 -0.98 which gives Y = 0.35X + 0.69 and putting in for X; Y = 0.35(6) + 0.69 = 2.79cm

Interaction terms

However, here we assume the slope is the same for each species meaning we think sepal length and width has the same relationship for all three species. If we don't want to make that assumption we can include an interaction term.

Interaction terms are were we think there is some added effect associated with differnt groups. For example, we might think that longer sepals mean wider sepals but an interaction terms would be that sepals increase in width faster with length depending on a species. Here the interaction term is between sepal length and species and we can add it into our model using length:species which will estimate the interaction term for each species compared to the baseline.

```
mod_3 <- lm(Sepal.Width ~ Sepal.Length + Species + Sepal.Length:Species,
            data = iris_data)

summary(mod_3)
```

```
##
## Call:
## lm(formula = Sepal.Width ~ Sepal.Length + Species + Sepal.Length:Species,
##     data = iris_data)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.72394 -0.16327 -0.00289  0.16457  0.60954 
## 
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                     -0.5694     0.5539  -1.028 0.305622    
## Sepal.Length                     0.7985     0.1104   7.235 2.55e-11 ***
## Speciesversicolor                1.4416     0.7130   2.022 0.045056 *  
## Speciesvirginica                 2.0157     0.6861   2.938 0.003848 ** 
## Sepal.Length:Speciesversicolor  -0.4788     0.1337  -3.582 0.000465 ***
## Sepal.Length:Speciesvirginica   -0.5666     0.1262  -4.490 1.45e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 0.2723 on 144 degrees of freedom
## Multiple R-squared:  0.6227, Adjusted R-squared:  0.6096 
## F-statistic: 47.53 on 5 and 144 DF,  p-value: < 2.2e-16
```

Like in the last model we have an intercept and slope for the baseline which is for the species setosa. In this case the base line has an intercept of -0.57 and a slope of 0.80 and both are signifcantly different to zero.

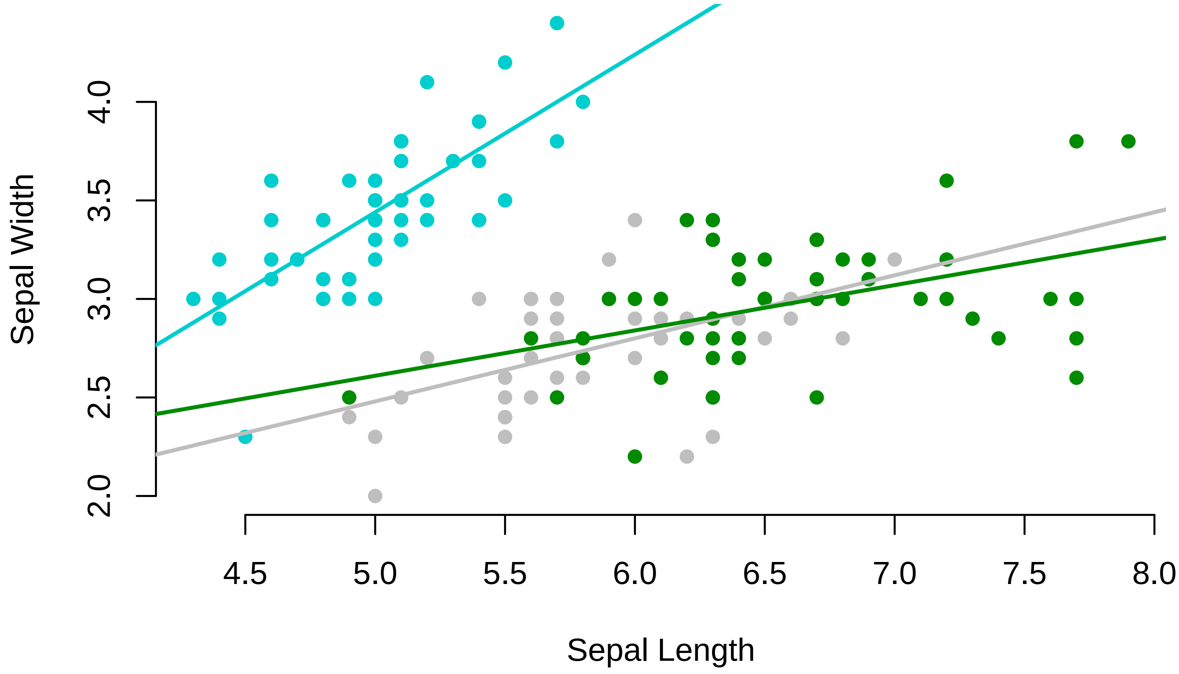
We also have the two contrast values with versicolor having an intercept 1.44 higher than setosa and virginica having an intercept 2.02 higher than setosa with both of these differnces signifcantly differnt to the baseline intercept.

The next line Sepal.Length:Speciesversicolor is the estimate for the interaction between Sepal.Length and being the group versicolor. This value of -0.48 is the difference in slope for this line compared to the base line. So for versicolor the slope between sepal lenght and width is 0.80 (baseline slope) - 0.48 = 0.32. This means that the width of versicolor sepals only increase by 0.32cm, for every cm increase in length compared to setosa where the sepal width increase by 0.80cm for every cm increase in sepal length. This interaction term of - 0.48 is also signifcant meaning that it is differnt to the slope of the baseline (setosa).

We can see something simlar for virginica where the interaction term of -0.57 is also signifcantly different to the baseline.

Lets plot these.

```
#Set up the graph
plot(iris_data$Sepal.Width ~ iris_data$Sepal.Length,
     bty = "n",
     xlab = "Sepal Length",
     ylab = "Sepal Width",
     col = "white")

# colour the points for setosa
points(iris_data[iris_data$Species == "setosa", "Sepal.Width"] ~
         iris_data[iris_data$Species == "setosa", "Sepal.Length"],
       col = "cyan3",
       pch = 16)


# colour the points for versicolor
points(iris_data[iris_data$Species == "versicolor", "Sepal.Width"] ~
         iris_data[iris_data$Species == "versicolor", "Sepal.Length"],
       col = "grey",
       pch = 16)


# colour the points for virginica
points(iris_data[iris_data$Species == "virginica", "Sepal.Width"] ~
         iris_data[iris_data$Species == "virginica", "Sepal.Length"],
       col = "green4",
       pch = 16)

#fit the line for setosa
abline(-0.56, 0.80,
       col = "cyan3",
       lty = 1,
       lwd = 2)

#fit the line for versicolor
abline(-0.56 + 1.44, 0.80 -0.48,
       col = "grey",
       lty = 1,
       lwd = 2)

#fit the line for virginica
abline(-0.56 + 2.02, 0.80 -0.57,
       col = "green4",
       lty = 1,
       lwd = 2)
```



We now have a much clearer picture of the relationship between sepal length and width for the differnt species of iris. We can check residuals of the model like before using qq-plots (see previous week for how to do that) or by using plot(mod_3). These look ok to me so we will move on.

As an aside, notice that we only compare each group to the baseline so we never test if virginica and versicolor are differnt. This is a drawback of using contrasts as its important what baseline you pick and that you set up a clear hypothesis.

Model selection

Now we have three differnt models for the same data so now we need to decide which is the best model in terms of explaining your data. One way is to set up you hypothesis and model and just sticking with it. For example, I could have started by stating that my hypothesis is that individuals with longer sepals have wider sepals but that this relationship is species specific. This would in effect be model 3 and even if I did not find any differnce I would simply report all the results.

However, another approach/philosophy is to compare the models and how well they are fit. For example we could compare the R-squared values of our three models.

```
#R squared can also just be found at the bottom of the summary(mod_1) results.
summary(mod_1)$adj.r.squared
```

```
## [1] 0.007159294
```

```
#R squared for model 2
summary(mod_2)$adj.r.squared
```

```
## [1] 0.5604018
```

```
#R squared for model 3
summary(mod_3)$adj.r.squared
```

```
## [1] 0.609608
```

This tells us that the residuals are closer to the fitted values in model 1. This makes sense as what that means is that each data is closer to its fitted line. However, by fitting more lines will always reduce the distance of the data from the fitted values (fitted values just means the fitted line, so any data point for the species setosa the line fit for that species is its corresponding line).

To account for this problem we use what is called AIC. This is a value to looks at how well fit the line is (i.e. how good the R-squared value is) but then also takes into account how many parameters are in the model (each estimate is a parameter, so each intercept, slope, interaction terms is a parameter). The better the fit of te model the lower the AIC value is, however, the more parameters there are the higher the AIC value. We can use these values to compare our models and pick the model with the lowest AIC value.

```
#AIC
AIC(mod_1)
```

```
## [1] 179.4644
```

```
#AICfor model 2
AIC(mod_2)
```

```
## [1] 59.21722
```

```
#AICfor model 3
AIC(mod_3)
```

```
## [1] 43.34175
```

From this we can see that model 3 has an AIC value of 43.3 which is the lowest. Hence, we can pick this as our best model. Typically if the values are within 2 AIC values of each other you can pick the model with less parameters or report both models.

Generalised Linear models

Up until now we have been running linear models. Generalised Linear models or glms are very similar, but as the name suggests are generalised to include more types of models. The main differnce is the family term here. This tells the model how the residuals (sometimes refered to as the error term) are expected to be spread around the line. By default we expect normal distributions which is gaussian. Apart from that running it is exactly the same, for example we can run the interaction model as before and get the same results.

```
mod_4 <- glm(Sepal.Width ~ Sepal.Length + Species + Sepal.Length:Species,
             family = "gaussian",
            data = iris_data)

summary(mod_4)
```

```
##
## Call:
## glm(formula = Sepal.Width ~ Sepal.Length + Species + Sepal.Length:Species,
##     family = "gaussian", data = iris_data)
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -0.5694     0.5539  -1.028 0.305622
## Sepal.Length                     0.7985     0.1104   7.235 2.55e-11 ***
## Speciesversicolor                1.4416     0.7130   2.022 0.045056 *
## Speciesvirginica                 2.0157     0.6861   2.938 0.003848 **
## Sepal.Length:Speciesversicolor  -0.4788     0.1337  -3.582 0.000465 ***
## Sepal.Length:Speciesvirginica   -0.5666     0.1262  -4.490 1.45e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.07416645)
##
##     Null deviance: 28.307  on 149  degrees of freedom
## Residual deviance: 10.680  on 144  degrees of freedom
## AIC: 43.342
##
## Number of Fisher Scoring iterations: 2
```

What makes the glm so powerful is that it can deal with otherwise tricky data.

For example, if the response variable is binary (it has only two states like 0 and 1 or yes no or alive dead) we can tell the model that the data is not normal and to treat it accordingly.

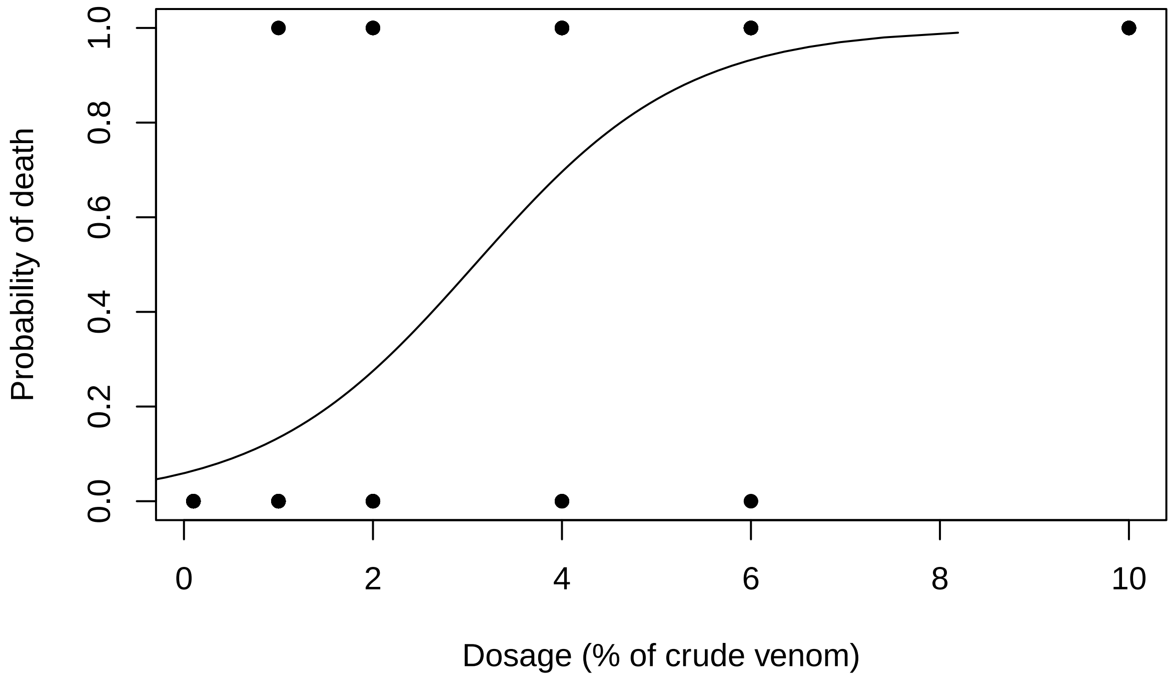
As an example, upload the data called tox_data.csv. This will include data of whether crickets are alive or dead at the end of a series experiments testing how potent different concentrations of spider venom are.

We can run this analysis in a very similar way as before with our response variable telling us if an individual died (1) or survived the trail with dosage our explanatory variable.

```
model_logistic <- glm(dead ~ dose_percent_venom,
             binomial(link = logit),
             data = tox_data)
```

Here the estimate refers to what is called the log-odds (don't worry about this but if you interested you can find more here (link)). In short it tells us how the odds of death change as we increase the dosages. We can plot this model and see that we fit a s like curve as the data is bounded between 0 (alive) and 1 (dead).

```
#this just generates a sequence of numbers
ypredicts <- seq(0, 1, 0.01)
xpredicts <- dose.p(model_logistic,p=c(ypredicts))


plot(dead ~ dose_percent_venom,
     ylab = "Probability of death",
     xlab = "Dosage (% of crude venom)",
     pch = 16,
     data = tox_data)

lines(unlist(xpredicts), ypredicts)
```



Notice that since our response variable is binary the data points form two line one for each state (dead/alive). The model fits a s like line following the estimated change in the probability of death. This is super useful as we can do things like predict the probability of death of a given dosage or how big a dose we would need to cause different probabilities of death. For example, we can estimate what dosage of this spider venom we would need to cause a 50% mortality rate using the dose.p() function and our model.

```
#the dose.p function is found in the MASS package
# The p=c(0.5) part asks for the value on the x-axis when the value is 0.5
# on the y-axis. Here 0.5 means 50% chance of mortality.

LD50 <- dose.p(model_logistic,p=c(0.5))
```

We can see here that a venom dose of 3.08 with a standard error of 0.32.

This is just an example of the flexibility of glm() models.