Model Selection II: Hypothesis Testing

Bob O'Hara

Contents

Now you have learned about why model selection is needed, you should be ready to find out how hypothesis testing is really model selection, and how to actually do it in R.

Hypothesis Testing

Hypothesis Testing is asymmetrical: we do not see the models being compared as equal. Hypothesis tests ask the question "Is the model without the effect sufficient to explain the data?", so the model with the effect is only declared the winner if the one without isn't good enough.

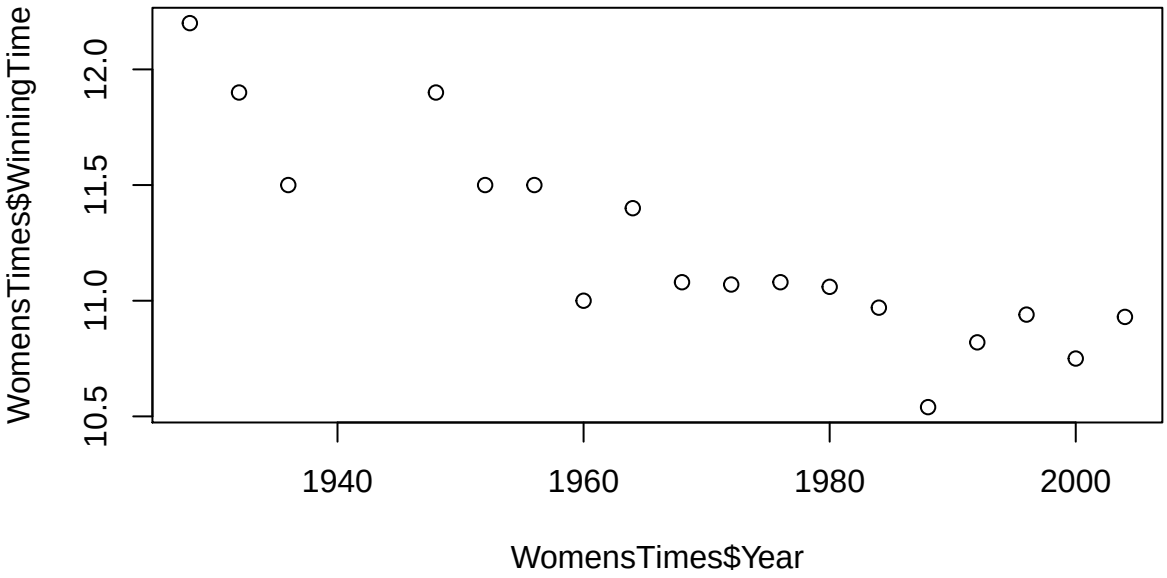
This is how to do statistical hypothesis testing

1. get a *null hypothesis* (i.e. without the effect)
2. get an *alternative hypothesis* (i.e. with the effect)
3. Chose a *test statistic* (e.g. the likelihood)
4. calculate the distribution of the test statistic if the null hypothesis was true
5. ask if the observed value of the statistic falls within the null distribution
6. if it does not, declare that the data were unlikely if the null hypothesis were true (and then make whatever biological inference seems reasonable).

As an example, we can look at the regression model we were using a few weeks ago.

```
Lk <- "https://www.math.ntnu.no/emner/ST2304/2019v/Week1/Olymp100m.csv"
# Lk <- "../Data/Olymp100m.csv"

Times100m <- read.csv(Lk)
Use <- Times100m$Sex=="Women" & !is.na(Times100m$WinningTime)
WomensTimes <- Times100m[Use,]
plot(WomensTimes$Year, WomensTimes$WinningTime)
```



The question is whether the times are changing. This is the same as asking if the slope is not zero, which is the opposite as asking if the slope is zero. Here is the process for this problem.

For each of these, in relation to the running times, try to work out what you need to do, and then click on "Answer" to see if you are right.

1. Decide on a *null hypothesis*

Answer

The slope is zero

2. Decide on an *alternative hypothesis*

Answer

The slope is not zero

3. Chose a *test statistic*

Answer

The slope is the obvious choice, and is correct. But below we will use the likelihood, because it can be used for more problems.

4. calculate the distribution of the test statistic if the null hypothesis was true

This will be your job, but you can wait until later to see how it is actually done. For now, how do you think you could do it?

Answer

You could simulate the data lots of times from the null model, or do lots of maths to work out what the distribution should be. Don't worry, below you will do the former.

5. ask if the observed value of the statistic falls within the null distribution

This will be your job below

Exercise: Testing the Hypothesis

Is the likelihood from the data likely if the null hypothesis is true?

Use this code & compare the model.H1 likelihood with the null distribution. Be aware that the final line (`Lhood <- replicate(...)`) takes a bit of time to run. If you want to check that it is working, change the `1e5` to a smaller number (e.g. `1e2`).

```
# Null Hypothesis
model.H0 <- lm(WinningTime ~ 1, data=WomensTimes)
# Alternative Hypothesis
model.H1 <- lm(WinningTime ~ Year, data=WomensTimes)

# This function simulates data from a model, and returns the likelihood
SimNullModel <- function(mod, X) {
  Sim <- simulate(mod) # simulate data from model
  model.test <- lm(Sim[,1] ~ X)
  logLik(model.test) # extract log-likelihood
}
# This line repeatedly simulates data from the null model, so retuens a distribution of the likelihood
Lhood <- replicate(1e5, SimNullModel(mod=model.H0,
                                     X=WomensTimes$Year))
```

Is the data likely to have been generated by the null hypothesis model?

Hint

You could plot the distribution of likelihoods with `hist()`. To get a p-value, you need to calculate what proportion of the distribution is more extreme than the likelihood for the alternative hypothesis (which you can get with `logLik(model.H1)`). A trick to do that is to use `mean()`, e.g. `mean(c(1,2,5,6)>4)` returns 0.5

Answer

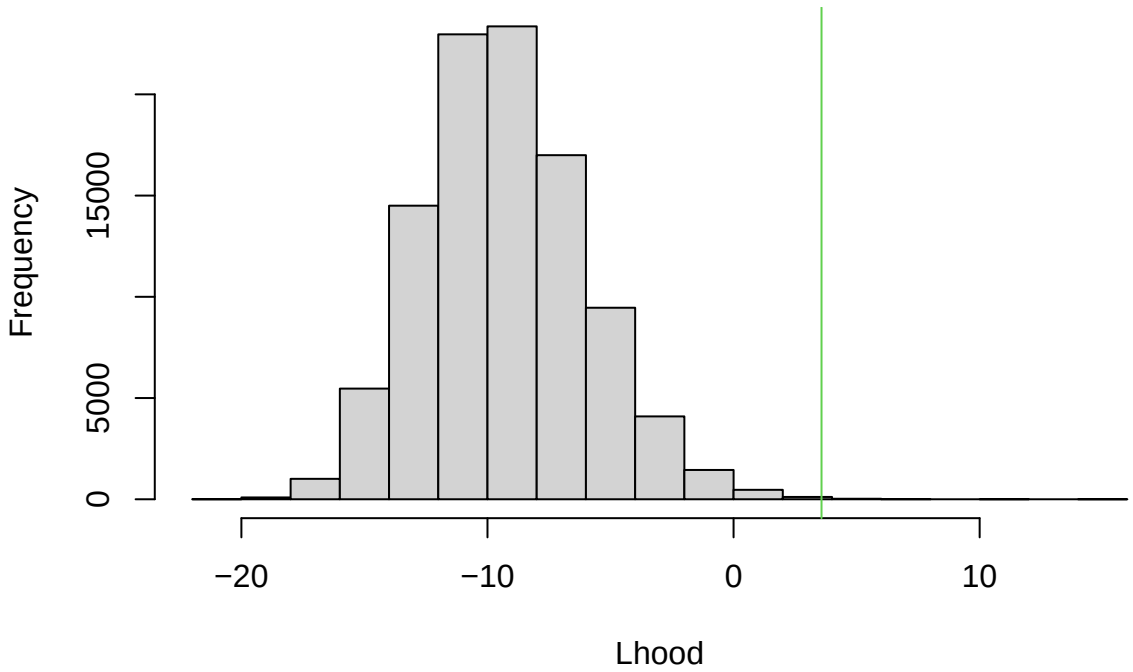
For those who want a video click here or watch:

First, we can run the code and plot a histogram of the simulated likelihoods:

```
# Null Hypothesis
model.H0 <- lm(WinningTime ~ 1, data=WomensTimes)
# Alternative Hypothesis
model.H1 <- lm(WinningTime ~ Year, data=WomensTimes)

# This function simulates data from a model, and returns the likelihood
SimNullModel <- function(mod, X) {
  Sim <- simulate(mod) # simulate data from model
  model.test <- lm(Sim[,1] ~ X)
  logLik(model.test) # extract log-likelihood
}
# This line repeatedly simulates data from the null model, so retuens a distribution of the likelihood
Lhood <- replicate(1e5, SimNullModel(mod=model.H0,
                                     X=WomensTimes$Year))

hist(Lhood, main="Simulated Likelihoods")
abline(v=logLik(model.H1), col=3)
```



The absolute values of the log likelihood do not matter: what is important is their comparison with the value we get from the data. We can see from the histogram that almost all simulations from the null distribution (i.e. the distribution if the null hypothesis was true) have likelihoods smaller than the one we see in the data.

Indeed, for this set of simulations, only 46 simulations had a larger likelihood (your numbers might be a bit different because the simulations are random). This equates to 0.05% of the simulations.

You can get these numbers in the following way. First, create a vector that tests if the simulated likelihood is larger than the likelihood in the data: `Lhood>logLik(model.H1)`. Then to count how many are larger, add up the values: `sum(Lhood>logLik(model.H1))`. This works because `Lhood>logLik(model.H1)` returns a logical vector, i.e. a vector with values `TRUE` or `FALSE`. Then using `sum()` R first converst the `TRUE`s to 1 and `FALSE`s to 0. Then summing these sums up the `TRUE`s, i.e. counts them. To get the proportion, we need to divide by the total length of the vector. But this is what `mean()` does, so we use that: `mean(Lhood>logLik(model.H1))`

What conclusion can you make about how times have changed in the women's 100m?

Hint

There is a strict answer to this, which is correct, and the easy answer which is probably what is going on (and is a scientific inference from the statistical results).

Answer

Strictly, from this test, all we can say is that we are very unlikely to have got this data if times were not, in fact, changing.

The 'easy' answer is that the times are decreasing over time, i.e. women have been getting faster.

Why Use the Likelihood?

We could use any reasonable test statistic to test whether women's times have changed in the 100m. The slope parameter might be a more direct statistic to use, because it is a direct measure of the change. So why do we use the likelihood?

The likelihood is a measure of overall model fit: it is $Pr(Data|parameters)$. This means it should be useful for many problems (i.e. we don't have to decide on a new statistic for each problem). Also, because it summarises whole model, it can be used to test more complicated hypotheses, such as when several treatments are used in an experiment, it can test whether any of the treatments work. So, for example, in the example from the last couple of weeks about ferilisers, we can ask whether any fertilisation has an effect. Or whether there are any interctions between date and feriliser (this latter test is useful because if there are no interactions, the model gets easier to interpret).

The likelihood also has some nice statistical properties, which makes it easier to use in practice. So we don't have to do simulations, because we know what the distribution of the statistic will be under the null hypothesis (at least approximately, but the approximation is usually pretty good).

The main "nice statistical property" is that we know the distribution for the difference between likelihoods when the models are nested. Nested moels are those where one is a subset of another. In particular, if we set some of the parameters of the larger model to zero, it becomes the smaller model. We then say that the smaller model is nested with in the larger one. For example, the model $y_i = \alpha + \epsilon_i$ is nested within $y_i = \alpha + \beta x_i + \epsilon_i$: we can set β to 0 and they become the same.

It turn out that twice the difference in log likelihoods follows a χ^2 distribution:

$$-2(log(L_1) - log(L_0)) \sim \chi^2_p$$

where p is the difference in the number of parameters between the models. We call this the *degrees of freedom*. They are the amount of information used to estimate the distribution: we "spend" p pieces of information to estimate p parameters. We use $-2log(L)$ a lot in statistics, so we call it the **deviance**.

We will use this later: for GLMs we can use the deviance to compare models. But for the normal distribution things are a bit more complicated. The log-likelihood is

$$log(L) \propto -\frac{\sum(y_i - \hat{\mu}_i)^2}{2\sigma^2}$$

so it also needs σ^2. The χ^2 distribution is only correct if we fix σ^2, but in practice we estimate it and this adds some extra error to the statistic. Instead we use

$$F = \frac{-2(log(L_1) - log(L_0))}{s^2}\frac{n-p}{p}$$

where s^2 is the estimate of the residual variance, which also follows a χ^2 distribution. We know from statistical theory that the ratio of χ^2 distributions divided by their degrees of freedom follows an F distribution (which is why we call this F). The numerator of F is the difference in likelihoods, and this has p degrees of freedom. The denominator, s^2, is the estimator of σ^2, the residual variation. We use the estimate of this from the most complex model, so it is

$$s^2 = \frac{1}{n-p}\sum(y_i - \hat{\mu}_i)^2$$

We use $n - p$ because we have n data points, but we "spend" p of them to estimate the model (i.e. $\hat{\mu}_i$), so we have $n - p$ left to estimate s^2.

Degrees of Freedom

I suspect statisticians have spent far too much time worrying about degres of freedom, but for the models we are looking at in this course they can be quite useful, because they summarise how we use the information in the data.

We can think of each data point we collect as one "piece" of information. If we want to estimate a parameter, we need one piece of information to do that. Once we have estimated all of the parameters we want to, we use the rest to estimate the residual variance.

Our uncertainty about a parameter estimate depends on the residual variance, so the better we estimate that, the more confident we can be about our paramters. Thus we want to use lots of degrees of freedom to estimate the residual variance, because that improves all of the estimates. In other words, we want lots of data. But it also means we don't want to estimate too many parameters, because they use up degrees of freedom.

What We Do In Practice

As a result of all of that theory, we find that to compare models we just need the sums of squares and degrees of freedom. Of course R will do the hard work for us:

```
model.H1 <- lm(WinningTime ~ Year, data=WomensTimes)
anova(model.H1)
```

```
Analysis of Variance Table

Response: WinningTime
          Df  Sum Sq Mean Sq F value    Pr(>F)
Year       1 2.64645 2.64645   59.76 8.626e-07 ***
Residuals 16 0.70855 0.04428
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The first column gives the degrees of freedom (for the difference in models and, in the final row, for the residual variance). The second gives the sums of squares, e.g. $\sum(y_i - \hat{\mu}_i)^2$. The third column is the "mean square error", i.e. the sum of squares divided by the degrees of freedom. The fourth column gives the F statistic, i.e. the mean squares divided by the residual mean square error.

The final column is the p-value for the F statistic, i.e. the probability that we would get the observed value or one more extreme if the null hypothesis (i.e. the samller model) was correct, if we re-sampled the data lots of times.

I mentioned the null hypothesis in the previous paragraph, but it doesn't appear in the table. That is because the null model is `WinningTime ~ 1`, and R implicitly uses this. Later, when we have more than one hypothesis, we will see that the null hypothesis for each test comes from the previous line in the `anova()` output.

We can also make the comparison of models more explicit:

```
model.H0 <- lm(WinningTime ~ 1, data=WomensTimes)
model.H1 <- lm(WinningTime ~ Year, data=WomensTimes)
(an <- anova(model.H0, model.H1))
```

```
Analysis of Variance Table

Model 1: WinningTime ~ 1
Model 2: WinningTime ~ Year
  Res.Df    RSS Df Sum of Sq     F    Pr(>F)
1     17 3.3550
2     16 0.7086  1    2.6465 59.76 8.626e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Can you work out what the output means? (hint: compare this to the output from `anova(model.H1)`)

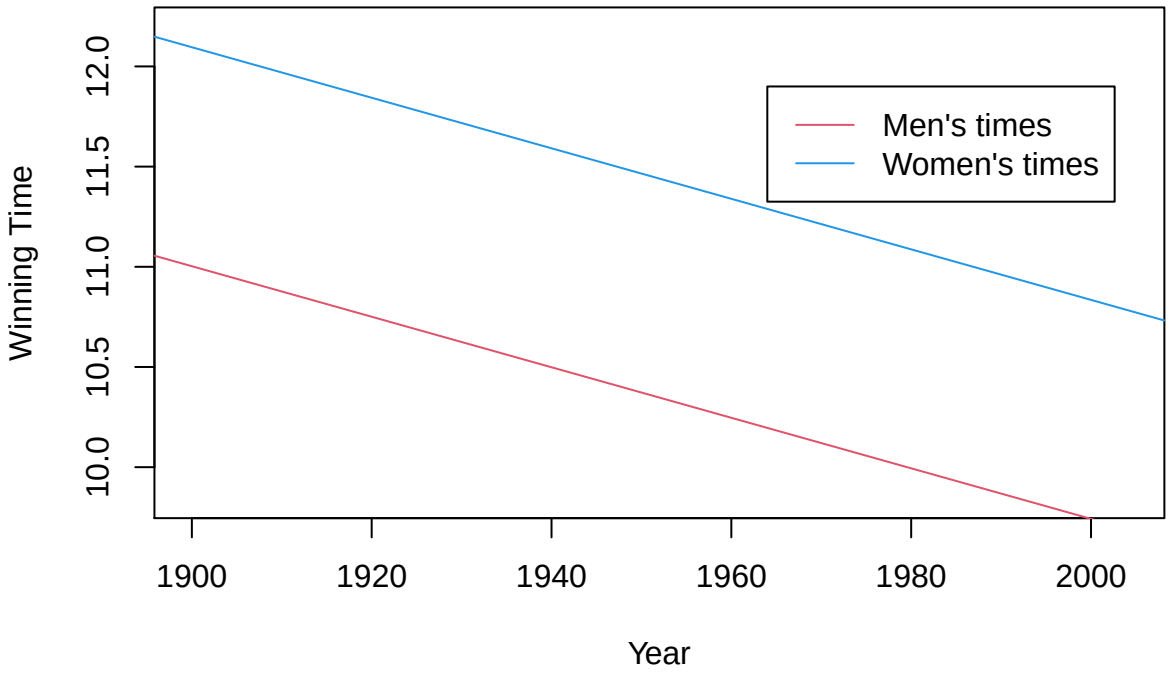
Answer

Here we see that R explicitly gives a line for each model, and gives the residual sum of squares ("RSS") for that model. The difference in the RSS between models is another way of calculating the sums of squares for the difference in the models. The other columns are the same (compare the numbers in the two tables if you're not sure).

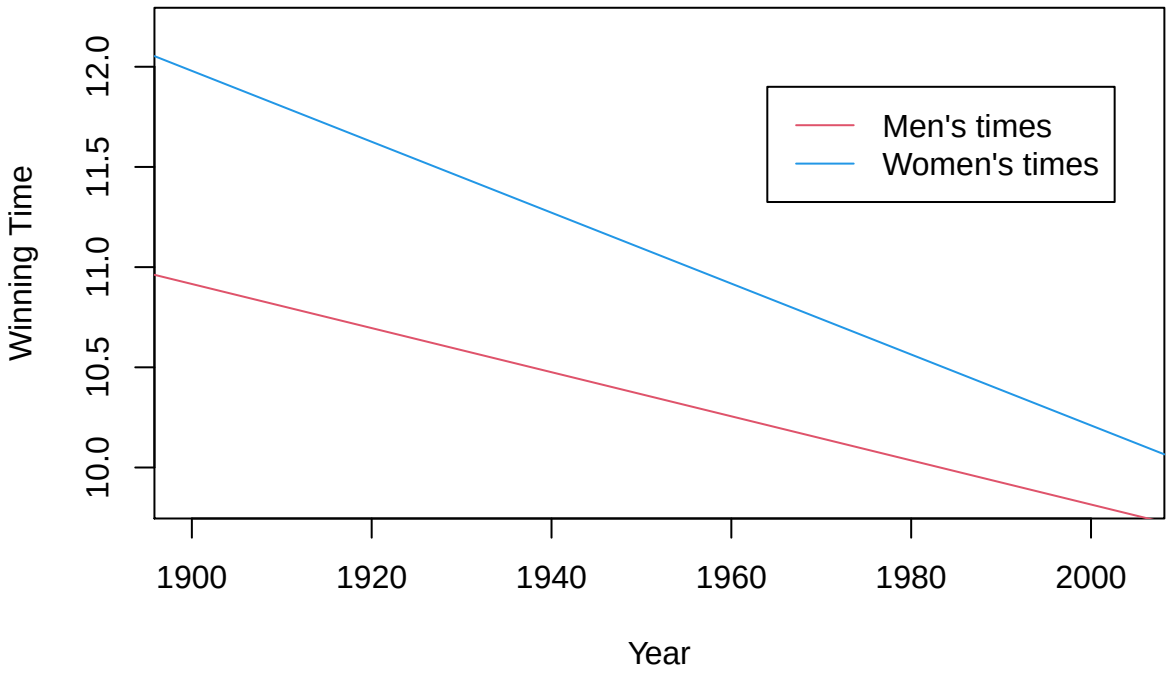
A more complex problem

The full data of 100m times has both men and women. The main reason to look at the data was to ask whether women's times were getting quicker at a faster rate than men's (and thus would women eventually run the same times as men).

The null model is thus that the change in times is the same for men and women. We can also assume that the mean times are different, i.e. the intercepts are different. This would be the null model:



The alternative hypothesis is that the lines are different:



We thus want to compare the model `WinningTime ~ Sex+Year` with `WinningTime ~ Sex*Year`

```
NullModel <- lm(WinningTime ~ Sex+Year, data=Times100m)
FullModel <- lm(WinningTime ~ Sex*Year, data=Times100m)

anova(NullModel, FullModel)
```

```
## Analysis of Variance Table
##
## Model 1: WinningTime ~ Sex + Year
## Model 2: WinningTime ~ Sex * Year
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     39 1.3369
## 2     38 1.1078  1   0.22918 7.8615 0.007911 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So we can see that the data are unlikely if there was no difference in times. But we have to look at the estimates to know what direction the difference is:

```
summary(FullModel)
```

```
##
## Call:
## lm(formula = WinningTime ~ Sex * Year, data = Times100m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37579 -0.05460  0.00738  0.08276  0.32234
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   31.826453   2.128910  14.950  < 2e-16 ***
## SexWomen      12.520596   4.076141   3.072  0.00392 **
```

```
## Year          -0.011006   0.001089 -10.104 2.56e-12 ***
## SexWomen:Year -0.005817   0.002074  -2.804  0.00791 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1707 on 38 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.9275, Adjusted R-squared:  0.9218
## F-statistic: 162.1 on 3 and 38 DF,  p-value: < 2.2e-16
```

(I will let you work out how to interpret this. If in doubt, plot the data)

We can also get the test like this:

```
anova(FullModel)
```

```
## Analysis of Variance Table
##
## Response: WinningTime
##           Df Sum Sq Mean Sq  F value    Pr(>F)
## Sex        1 8.5566  8.5566 293.5207 < 2.2e-16 ***
## Year       1 5.3937  5.3937 185.0198 3.507e-16 ***
## Sex:Year   1 0.2292  0.2292   7.8615  0.007911 **
## Residuals 38 1.1078  0.0292
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We want to look at the final line, the `Sex:Year` effect: this is the same as the test above. But what about the other terms?

`anova()` with several terms

`anova()` can be useful when we want to look at several models. However this also makes in dangerous.

If `anova()` is given a single model, it will look at the models that are needed to build it up by adding single terms, and test each of these. So it looks at the `Sex * Year` model and sees that it can be built up from the following models:

- `1` (i.e. a constant)
- `Sex` (add the `Sex` effect)
- `Sex + Year` (add the `Year` effect to the model with `Sex`)
- `Sex * Year` (add the interaction)

It then compares these *in the order given*, so the first column of the table tells us which explanatory variable has been added. For example, the `Year` row compares the `Sex + Year` model with the `Sex` model.

Why is this dangerous? For two reasons:

1. The test results can depend on the order the terms are added into the model. There are examples where the first variable you enter into the model is not significant, but the second is. And it doesn't matter which variable is put in first. The Schey example from multiple regression is an example of this.
2. If you have interactions, the tests of main effects are almost meaningless **and should be ignored**. For example, the Sex test tests whether Males and Females have different times *at the origin*. If we move the origin (e.g. by standardising the data), we can get different test results. Similarly, the `Year` effect tests whether there is an overall Year effect. but if this differs between treatments, it can be that it is positive in some, and negative in others. The test then depends on if there is more data in treatments that are positive or negative.

When you are testing a specific hypotheses, you should be able to explicitly fit the null and alternative models, and so should be able to use `anova()` to compare them. If you have a lot of possible models, you are not testing hypotheses, so should probably not be using `anova()`.

Why is ANOVA called ANOVA

ANOVA = Analysis of Variance

The approach was first developed to analyse field trials. When the data cooperate[1], the mean squared error is an estimate of the variance explained by that effect.

Unfortunately, ANOVA now has two meanings:

- linear model with categorical factors
- model testing for linear models

Your Turn

This is the Yields data from a couple of weeks ago.

```
Yields <-  read.csv("https://www.math.ntnu.no/emner/ST2304/2021v/Week08/Hoosfield_Yields.csv")
```

Is there an interation between treatment and date (i.e. before/after 1970)? First, think about what the null and alterntive models are, then fit and compare them

Hint

The models are

```
model.yield.H0 <- lm(yield ~ After1970 + Treatment, data=Yields)
model.yield.H1 <- lm(yield ~ After1970 * Treatment, data=Yields)
```

and you can use `anova()` to compare them.

Answer

If you want to watch the video, it's below, or you can click here

First, fit the models and compare with `anova()`:

```
model.yield.H0 <- lm(yield ~ After1970 + Treatment, data=Yields)
model.yield.H1 <- lm(yield ~ After1970 * Treatment, data=Yields)
anova(model.yield.H0, model.yield.H1)
```

```
## Analysis of Variance Table
##
## Model 1: yield ~ After1970 + Treatment
## Model 2: yield ~ After1970 * Treatment
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     67 34.578
## 2     64 14.767  3    19.811 28.621 7.403e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that the F-ratio is 28.62. If the null model was true, we would expect this to be about 1. So it's bigger, but would it still likely if the null hypothesis (i.e. of no interaction) was true? Well, the p-value is 7.4×10^{-12}, which suggests that the results (i.e. the estimates of the interaction parameters) would be very unlikely if there were no interaction. So we can surmise that there is some interaction, i.e. the relative effects of some of the treatments changed after 1970.

[1] this means the data have to be balanced, so each combination of levels of factors has the same number of observations

We can also use `anova()` on just the interaction model:

```
model.yield.H1 <- lm(yield ~ After1970 * Treatment, data=Yields)
anova(model.yield.H1)
```

```
## Analysis of Variance Table
##
## Response: yield
##                      Df Sum Sq Mean Sq F value    Pr(>F)
## After1970             1 13.062 13.0622  56.613 2.225e-10 ***
## Treatment             3 93.871 31.2905 135.617 < 2.2e-16 ***
## After1970:Treatment   3 19.811  6.6037  28.621 7.403e-12 ***
## Residuals            64 14.767  0.2307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Each row of the `anova()` table shows the effect of adding that effect to a model that has the terms in the rows above it. i.e.

1. The first row of the table compares a model with just a constant (i.e. `yield ~ 1`) with one where `After1970` is an effect (i.e. `yield ~ After1970`).
2. The second row of the table compares a model with `After1970` as an effect (i.e. `yield ~ After1970` to one with `After1970` and `Treatment` as effects (i.e. `yield ~ After1970 + Treatment`).
3. The third row of the table compares the model with `After1970` and `Treatment` (the "main effects") and adds the interaction: `yield ~ After1970 + Treatment + After1970:Treatment`.

So, we just want the third row, and we can see that the statistics are the same as in the output from the `anova(model.yield.H0, model.yield.H1)` output.

Next

If you want to go back to why we select models you can. Or move forward to find out about exploratory model selection.