Linear modelling with R

George Savva

23/07/2020

## Introduction

Here we will use R to answer some research questions based on the walking speed data.

The data are from a randomised experiment of a rehabilition intervention (compared to control) aimed at improving the walking speed of stroke survivors.

We have recorded the age and sex of each participant, the treatment allocation, the hospital department from which they were recruited and their walking speed.

Our questions are:

* What are the mean and standard deviation of walking speed for treated and untreated participants?
* Does the treatment improve walking speed compared to controls?
* By how much, and how certain are we of this?
* Does age affect walking speed?
* Does sex affect walking speed?
* Does sex affect the success of the treatment?
* Was there any difference in treatment effect by department?

## Loading and exploring the data

First we’ll load the data from the excel sheet, and explore using the summary functions and graphics:

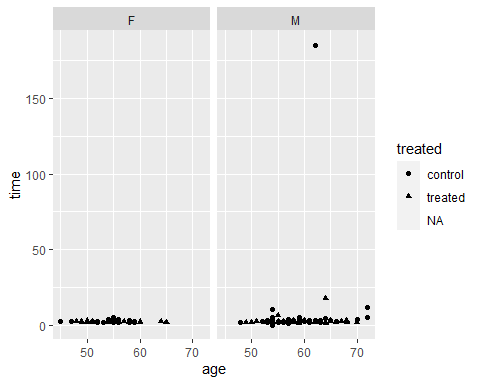
library(readxl)  
library(ggplot2)  
  
alldata <- read\_excel("walkingspeed.xlsx", sheet="combined", range="A1:F139")  
head(alldata)

## # A tibble: 6 x 6  
## patid time sex age department treated  
## <dbl> <dbl> <chr> <dbl> <dbl> <chr>   
## 1 1 1.90 M 53 3 treated  
## 2 2 3.54 M 61 3 control  
## 3 3 2.93 M 65 1 treated  
## 4 4 1.82 M 48 2 control  
## 5 5 2.20 M 62 2 treated  
## 6 6 3.04 M 62 4 control

summary(alldata)

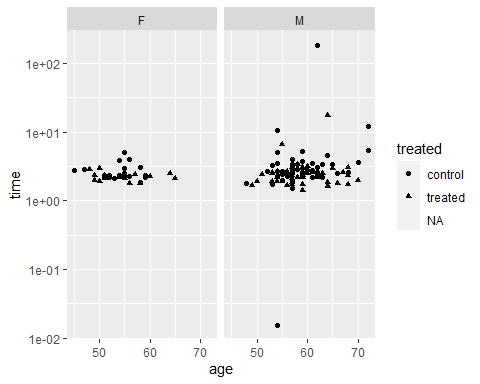
## patid time sex age   
## Min. : 1.00 Min. : 0.0154 Length:138 Min. :45.00   
## 1st Qu.: 34.75 1st Qu.: 2.1675 Class :character 1st Qu.:54.00   
## Median : 68.50 Median : 2.4274 Mode :character Median :57.00   
## Mean : 68.52 Mean : 4.1788 Mean :57.43   
## 3rd Qu.:102.25 3rd Qu.: 2.9442 3rd Qu.:60.00   
## Max. :138.00 Max. :185.3620 Max. :72.00   
## NA's :2 NA's :2   
## department treated   
## Min. :1.000 Length:138   
## 1st Qu.:2.000 Class :character   
## Median :3.000 Mode :character   
## Mean :2.601   
## 3rd Qu.:3.750   
## Max. :4.000   
##

ggplot(alldata, aes(x=age, y=time, shape=treated)) + geom\_point() + facet\_wrap(~sex)



We’ll put the graph onto a log scale:

ggplot(alldata, aes(x=age, y=time, shape=treated)) +   
 geom\_point() +   
 facet\_wrap(~sex) +   
 scale\_y\_continuous(trans="log10")



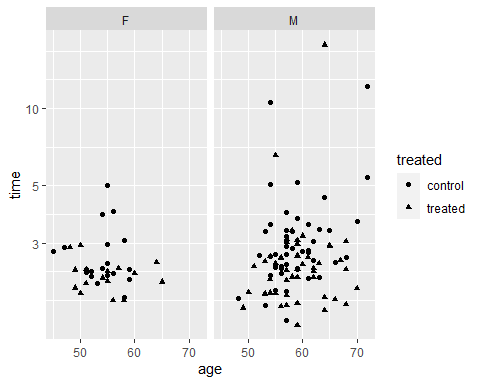
Now we might decide to remove those outlying points, because they are likely to be highly influential in our modelling and they are likely to be wrong.

The subset() function returns the subset of a dataframe that meets the criteria in its second argument.

alldata2 <- subset(alldata, time>1 & time<100)

Now we can plot the dataset without outliers:

ggplot(alldata2, aes(x=age, y=time, shape=treated)) +   
 geom\_point() +   
 facet\_wrap(~sex) +   
 scale\_y\_continuous(trans="log10")



What can you see from the graph?

## Estimating our model

Our first question concerned descriptive statistics around walking time amongst men and women. We saw in the last session that R does not have a good built in way to make nice descriptive tables. In the last session we saw the ‘table1’ package but now we can use the new tbl\_summary() function from the gtsummary package to get these.

library(gtsummary)  
  
tbl\_summary(alldata)

## Table printed with `knitr::kable()`, not {gt}. Learn why at  
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

|  |  |
| --- | --- |
| **Characteristic** | **N = 138** |
| patid | 68 (35, 102) |
| Unknown | 2 |
| time | 2.43 (2.17, 2.94) |
| Unknown | 2 |
| sex |  |
| F | 38 (28%) |
| M | 100 (72%) |
| age | 57.0 (54.0, 60.0) |
| department |  |
| 1 | 29 (21%) |
| 2 | 32 (23%) |
| 3 | 42 (30%) |
| 4 | 35 (25%) |
| treated |  |
| control | 69 (51%) |
| treated | 67 (49%) |
| Unknown | 2 |

This is really nice!. Something to note: tbl\_summary has detected that ‘department’ has only four values so has treated it as a categorical variable. This is fine but in general R functions will not do this (as we will see later) so be careful.

We want our data stratified by treatment group, so we can use:

tbl\_summary(alldata, by=treated)

## 2 observations missing `treated` have been removed. To include these observations, use `forcats::fct\_explicit\_na()` on `treated` column before passing to `tbl\_summary()`.

## Table printed with `knitr::kable()`, not {gt}. Learn why at  
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

|  |  |  |
| --- | --- | --- |
| **Characteristic** | **control**, N = 69 | **treated**, N = 67 |
| patid | 70 (36, 104) | 67 (34, 100) |
| time | 2.69 (2.32, 3.34) | 2.32 (1.92, 2.62) |
| sex |  |  |
| F | 19 (28%) | 17 (25%) |
| M | 50 (72%) | 50 (75%) |
| age | 57.0 (54.0, 59.0) | 57.0 (54.0, 61.5) |
| department |  |  |
| 1 | 16 (23%) | 13 (19%) |
| 2 | 16 (23%) | 15 (22%) |
| 3 | 17 (25%) | 25 (37%) |
| 4 | 20 (29%) | 14 (21%) |

We won’t spend a lot of time on this table, but lets change the statistics displayed, and some of the row names, and drop the rows we don’t want to include. For more customisations see the tbl\_summary vignette.

Take some time to study the tbl\_summary command below, the tbl\_summary help files and vignettes to see how these are specified:

tbl1 <- tbl\_summary(alldata2[,c("treated","time","age", "sex")],   
 by=treated,  
 label=list(time ~ "Time (s)", age ~ "Age (yrs)", sex~"Sex"),  
 statistic=list(time~"{mean} ({sd})"))  
add\_overall(tbl1)

## Table printed with `knitr::kable()`, not {gt}. Learn why at  
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic** | **Overall**, N = 134 | **control**, N = 67 | **treated**, N = 67 |
| Time (s) | 2.86 (1.85) | 3.12 (1.68) | 2.60 (1.99) |
| Age (yrs) | 57.0 (54.0, 60.0) | 57.0 (54.0, 59.0) | 57.0 (54.0, 61.5) |
| Sex |  |  |  |
| F | 36 (27%) | 19 (28%) | 17 (25%) |
| M | 98 (73%) | 48 (72%) | 50 (75%) |

## Regression modelling

First we will test the treatment effect on walking speed. We will use a linear regression model for this. Make sure you understand the commands below:

model1 <- lm( data = alldata2 , time ~ treated)  
summary(model1)

##   
## Call:  
## lm(formula = time ~ treated, data = alldata2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.6223 -0.7225 -0.3901 0.0652 15.0421   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.1162 0.2250 13.851 <2e-16 \*\*\*  
## treatedtreated -0.5169 0.3182 -1.625 0.107   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.841 on 132 degrees of freedom  
## Multiple R-squared: 0.0196, Adjusted R-squared: 0.01218   
## F-statistic: 2.639 on 1 and 132 DF, p-value: 0.1066

How do you interept this model output?

We could also satisfy ourselves that the linear regression here is the same as an unpaired t-test with the equal variances assumption.

t.test(data=alldata2, time~ treated, var.equal=TRUE)

##   
## Two Sample t-test  
##   
## data: time by treated  
## t = 1.6246, df = 132, p-value = 0.1066  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.1124579 1.1462563  
## sample estimates:  
## mean in group control mean in group treated   
## 3.116183 2.599283

So if these outcomes are the same you might wonder why we prefer the linear model function? We should prefer the linear regression because it offers us a lot more flexibility later on.

## Checking the validity of the model

We should always check that the assumption underlying a linear model are met. The assumptions are:

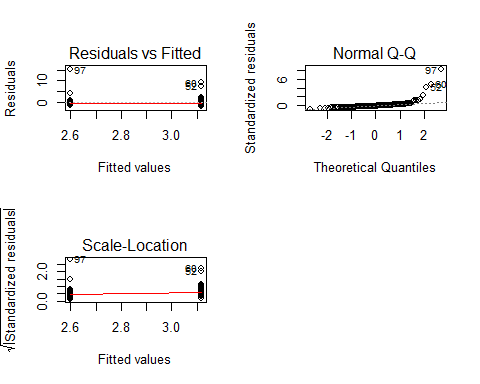
* The residuals (differences between observations and ‘predicted’ values) are identically normally distributed
* The observations are independent of each other

There is no statistical test for these assumptions, we need to use graphical methods to judge visually whether the first is likely to be reasonable, and our knowledge of the experimental design to know whether the second is true.

When you ‘plot’ a linear model object the plot() function makes graphs to help you check the distribution of residuals from the model:

par(mfrow=c(2,2))  
plot(model1)

## hat values (leverages) are all = 0.01492537  
## and there are no factor predictors; no plot no. 5



The second graph shows a normal qqplot of residuals from the model. If the times were normally distributed aroud their predicted values this would follow the straight dotted line. As it is we can see a significant deviation; there are a lot of residuals that are a lot bigger than the model thinks they should be.

The first and third graphs are less useful for this regression (because there are only two possible ‘predicted’ values) but they still illustrate that although the residuals are not normally distributed they do at least seem to be similarly distriuted across groups.

## Transformations and linear models

In the last section we considered two different transformations of the data for the sake of plotting. We looked at the log transformation and the inverse transformation. We could try to model log(time) or 1/time as a function of treatment, to see if these meet the assumptions of the regression model better.

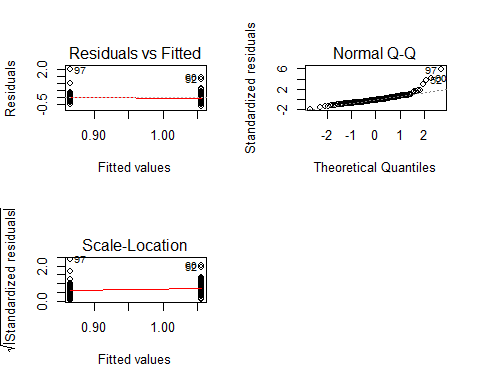
We could create a new variable with the transformed values, or we can add the transformation to our model. First we’ll look at the log-transformation:

model2 <- lm( data=alldata2 , log(time) ~ treated)  
summary(model2)

##   
## Call:  
## lm(formula = log(time) ~ treated, data = alldata2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.65480 -0.21234 -0.05851 0.11180 2.00596   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.05620 0.04267 24.75 < 2e-16 \*\*\*  
## treatedtreated -0.19192 0.06035 -3.18 0.00183 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3493 on 132 degrees of freedom  
## Multiple R-squared: 0.07117, Adjusted R-squared: 0.06413   
## F-statistic: 10.11 on 1 and 132 DF, p-value: 0.001834

par(mfrow=c(2,2))  
plot(model2)

## hat values (leverages) are all = 0.01492537  
## and there are no factor predictors; no plot no. 5

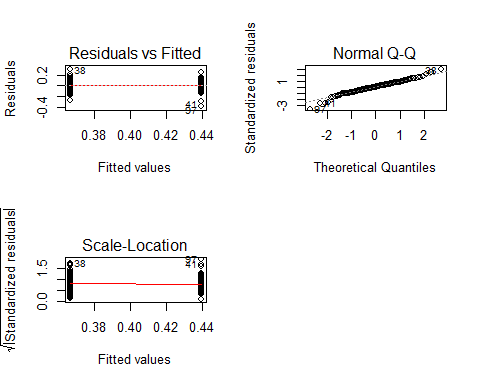


model3 <- lm( data=alldata2 , 1/time ~ treated)  
summary(model3)

##   
## Call:  
## lm(formula = 1/time ~ treated, data = alldata2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.38247 -0.06237 0.00192 0.06765 0.30257   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.36681 0.01285 28.54 < 2e-16 \*\*\*  
## treatedtreated 0.07235 0.01818 3.98 0.000113 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1052 on 132 degrees of freedom  
## Multiple R-squared: 0.1072, Adjusted R-squared: 0.1004   
## F-statistic: 15.84 on 1 and 132 DF, p-value: 0.000113

par(mfrow=c(2,2))  
plot(model3)

## hat values (leverages) are all = 0.01492537  
## and there are no factor predictors; no plot no. 5



The final model looks the best. It seems that if we model the inverse of time (speed) instead of time itself then the distribution of the residuals is close to normal.

How should we interpret the final model?

## Presenting the results

An advantage of regression models is that we get an estimate and confidence interval for our effect as well as a p-value. This is a major disadvantge of analysis or reporting just be placing p-values on plots; by restricting ourselves to this we never get to discuss how much of a difference the treatment makes, and our certainty around that estimate of effect.

The plain text summary of the model gives us all of the information we need, but there are other packages to organise regression model output in a more comprehensible and publication-ready form:

library(sjPlot)  
tab\_model(model1, model2, model3)

time

log(time)

1/time

Predictors

Estimates

CI

p

Estimates

CI

p

Estimates

CI

p

(Intercept)

3.12

2.68 – 3.56

<0.001

1.06

0.97 – 1.14

<0.001

0.37

0.34 – 0.39

<0.001

treatedtreated

-0.52

-1.14 – 0.11

0.107

-0.19

-0.31 – -0.07

0.002

0.07

0.04 – 0.11

<0.001

Observations

134

134

134

R2 / R2 adjusted

0.020 / 0.012

0.071 / 0.064

0.107 / 0.100

tbl\_regression(model1, intercept = TRUE, )

## Table printed with `knitr::kable()`, not {gt}. Learn why at  
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic** | **Beta** | **95% CI** | **p-value** |
| (Intercept) | 3.1 | 2.7, 3.6 | <0.001 |
| treated |  |  |  |
| control |  |  |  |
| treated | -0.52 | -1.1, 0.11 | 0.11 |

## A continuous predictor

We can add the effect of age into our model, by changing the model formula in the lm() call:

model4 <- lm(data=alldata2, 1/time ~ treated + age)  
tbl\_regression(model4)

## Table printed with `knitr::kable()`, not {gt}. Learn why at  
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic** | **Beta** | **95% CI** | **p-value** |
| treated |  |  |  |
| control |  |  |  |
| treated | 0.07 | 0.04, 0.11 | <0.001 |
| age | 0.00 | -0.01, 0.00 | 0.012 |

tab\_model(model4)

1/time

Predictors

Estimates

CI

p

(Intercept)

0.61

0.42 – 0.81

<0.001

treatedtreated

0.07

0.04 – 0.11

<0.001

age

-0.00

-0.01 – -0.00

0.012

Observations

134

R2 / R2 adjusted

0.149 / 0.136

It looks like the effect of age is 0! But it is statistically significant, so the low effect size this is probably just a rounding error. We’ll have to change the level of precision being reported in the tabular output (and tweak another couple of options):

tab\_model(model4, digits = 4)

1/time

Predictors

Estimates

CI

p

(Intercept)

0.6139

0.4212 – 0.8066

<0.001

treatedtreated

0.0743

0.0394 – 0.1093

<0.001

age

-0.0043

-0.0076 – -0.0010

0.012

Observations

134

R2 / R2 adjusted

0.149 / 0.136

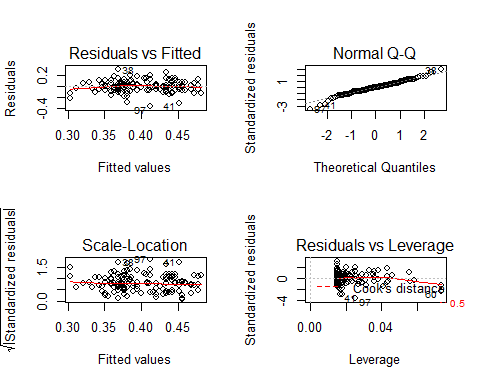
tbl\_regression(model4,   
 show\_single\_row="treated",  
 intercept=TRUE,  
 estimate\_fun = function(x) style\_ratio(x, digits = 4))

## Table printed with `knitr::kable()`, not {gt}. Learn why at  
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic** | **Beta** | **95% CI** | **p-value** |
| (Intercept) | 0.6139 | 0.4194, 0.8084 | <0.001 |
| treated | 0.0743 | 0.0391, 0.1096 | <0.001 |
| age | -0.0043 | -0.0077, -0.0009 | 0.012 |

We should continue to check that the model assumptions are still met.

par(mfrow=c(2,2))  
plot(model4)



## Plotting the model equations

To plot the fitted values from our model we need to get convert the model object into a dataframe that can be plotted.

Before I do this I will make a new variable ‘speed’ corresponding to our transformed outcome. We can use this instead of doing the transformation in the model equation.

predict() creates predictions from a model object, and can calculate confidence or prediction intervals. I will use cbind to stick the predictions and the existing data together.

alldata2$speed <- 6 / alldata2$time  
head(alldata2)

## # A tibble: 6 x 7  
## patid time sex age department treated speed  
## <dbl> <dbl> <chr> <dbl> <dbl> <chr> <dbl>  
## 1 1 1.90 M 53 3 treated 3.16  
## 2 2 3.54 M 61 3 control 1.70  
## 3 3 2.93 M 65 1 treated 2.05  
## 4 4 1.82 M 48 2 control 3.30  
## 5 5 2.20 M 62 2 treated 2.72  
## 6 6 3.04 M 62 4 control 1.97

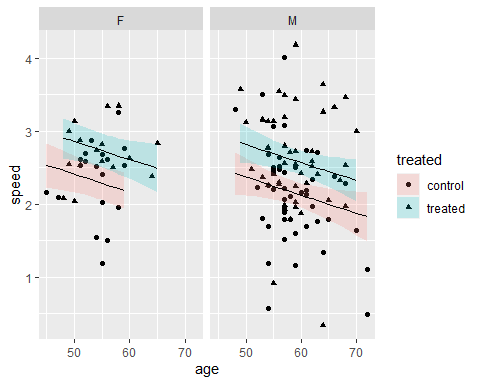
model5 <- lm(data=alldata2, speed ~ age + treated + sex)  
summary(model5)

##   
## Call:  
## lm(formula = speed ~ age + treated + sex, data = alldata2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.13173 -0.37395 0.02741 0.35612 1.81955   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.63522 0.61105 5.949 2.35e-08 \*\*\*  
## age -0.02451 0.01110 -2.207 0.0291 \*   
## treatedtreated 0.44667 0.10738 4.160 5.75e-05 \*\*\*  
## sexM -0.04167 0.13125 -0.317 0.7514   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6208 on 130 degrees of freedom  
## Multiple R-squared: 0.1496, Adjusted R-squared: 0.1299   
## F-statistic: 7.62 on 3 and 130 DF, p-value: 9.85e-05

model5predictions <- predict(model5, interval = "confidence", newdata = alldata2 ) # its helpful to specify 'newdata' in predict.  
alldata3 <- cbind(alldata2,model5predictions)  
  
head(alldata3)

## patid time sex age department treated speed fit lwr upr  
## 1 1 1.897512 M 53 3 treated 3.162035 2.741402 2.533538 2.949265  
## 2 2 3.537158 M 61 3 control 1.696277 2.098683 1.925814 2.271552  
## 3 3 2.927432 M 65 1 treated 2.049578 2.447330 2.237672 2.656988  
## 4 4 1.819787 M 48 2 control 3.297089 2.417260 2.131872 2.702649  
## 5 5 2.204258 M 62 2 treated 2.722004 2.520848 2.345526 2.696170  
## 6 6 3.038065 M 62 4 control 1.974941 2.074177 1.893417 2.254937

ggplot(alldata3, aes(x=age, y=speed, shape=treated)) +   
 geom\_point() +   
 facet\_wrap(~sex) +   
 geom\_ribbon(aes(ymin=lwr, ymax=upr, fill=treated), alpha=0.2) +   
 geom\_line(aes(y=fit))



## Testing interactions

Our existing model does not allow the effect of treatment on walking speed to vary with sex. But we might be interested in whether the effect is the same in men or women (a so called ‘interaction’ effect).

Note it is not valid to do this by comparing models estimated in men and women separately. We should instead estimate a model that includes the interaction between sex and treatment on walking speed.

model6 <- lm(data=alldata2, speed ~ age + treated\*sex)  
summary(model6)

##   
## Call:  
## lm(formula = speed ~ age + treated \* sex, data = alldata2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.14398 -0.38039 0.04062 0.36341 1.83212   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.66397 0.61785 5.930 2.61e-08 \*\*\*  
## age -0.02445 0.01114 -2.195 0.0300 \*   
## treatedtreated 0.37986 0.20802 1.826 0.0701 .   
## sexM -0.08594 0.17675 -0.486 0.6276   
## treatedtreated:sexM 0.09127 0.24308 0.375 0.7079   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6228 on 129 degrees of freedom  
## Multiple R-squared: 0.1505, Adjusted R-squared: 0.1241   
## F-statistic: 5.713 on 4 and 129 DF, p-value: 0.0002892

tab\_model(model6)

speed

Predictors

Estimates

CI

p

(Intercept)

3.66

2.45 – 4.87

<0.001

age

-0.02

-0.05 – -0.00

0.030

treatedtreated

0.38

-0.03 – 0.79

0.070

sexM

-0.09

-0.43 – 0.26

0.628

treatedtreated:sexM

0.09

-0.39 – 0.57

0.708

Observations

134

R2 / R2 adjusted

0.150 / 0.124

Look at the plot of model6 compared to model5:

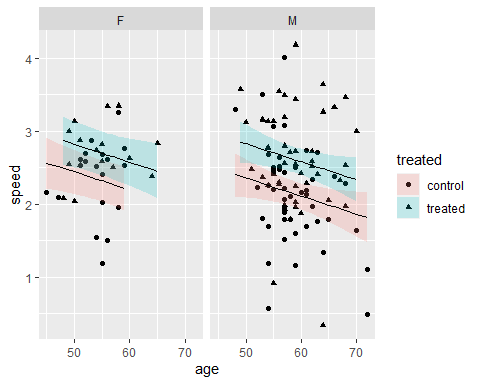
# its helpful to specify 'newdata' in predict.  
model6predictions <- predict(model6, interval = "confidence", newdata = alldata2 )   
head(model6predictions)

## fit lwr upr  
## 1 2.753082 2.535621 2.970542  
## 2 2.086320 1.901036 2.271605  
## 3 2.459629 2.239504 2.679755  
## 4 2.404227 2.109750 2.698704  
## 5 2.532993 2.345798 2.720187  
## 6 2.061866 1.869240 2.254492

alldata3 <- cbind(alldata2,model6predictions)  
  
head(alldata3)

## patid time sex age department treated speed fit lwr upr  
## 1 1 1.897512 M 53 3 treated 3.162035 2.753082 2.535621 2.970542  
## 2 2 3.537158 M 61 3 control 1.696277 2.086320 1.901036 2.271605  
## 3 3 2.927432 M 65 1 treated 2.049578 2.459629 2.239504 2.679755  
## 4 4 1.819787 M 48 2 control 3.297089 2.404227 2.109750 2.698704  
## 5 5 2.204258 M 62 2 treated 2.722004 2.532993 2.345798 2.720187  
## 6 6 3.038065 M 62 4 control 1.974941 2.061866 1.869240 2.254492

ggplot(alldata3, aes(x=age, y=speed, shape=treated)) +   
 geom\_point() +   
 facet\_wrap(~sex) +   
 geom\_ribbon(aes(ymin=lwr, ymax=upr, fill=treated), alpha=0.2) +   
 geom\_line(aes(y=fit))



Finally, we can test whether model6 fits the data better than model5. Anova can be used to compare the fit of two models, and give a p-value for whether the more complex model provides a significantly better fit than the simpler one.

anova(model5, model6)

## Analysis of Variance Table  
##   
## Model 1: speed ~ age + treated + sex  
## Model 2: speed ~ age + treated \* sex  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 130 50.098   
## 2 129 50.043 1 0.054693 0.141 0.7079

## A model with a categorial predictor

We might be interested in whether walking speed varies by department. We could add the department variable to our regression model as follows:

model7 <- lm(data=alldata2, speed ~ age + sex + treated + department)  
summary(model7)

##   
## Call:  
## lm(formula = speed ~ age + sex + treated + department, data = alldata2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.06761 -0.29631 0.05598 0.29456 1.78223   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.41710 0.61447 5.561 1.48e-07 \*\*\*  
## age -0.02483 0.01099 -2.261 0.0255 \*   
## sexM -0.05833 0.13010 -0.448 0.6547   
## treatedtreated 0.44443 0.10621 4.184 5.25e-05 \*\*\*  
## department 0.09691 0.04922 1.969 0.0511 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.614 on 129 degrees of freedom  
## Multiple R-squared: 0.1744, Adjusted R-squared: 0.1488   
## F-statistic: 6.811 on 4 and 129 DF, p-value: 5.259e-05

But notice that R has not recognised that the ‘department’ variable should be treated as a categorical variable. To make sure that ‘department’ is treated as categorical we should make a new variable in our data frame:

alldata2$department\_category <- factor(alldata2$department)  
summary(alldata2)

## patid time sex age   
## Min. : 1.00 Min. : 1.435 Length:134 Min. :45.00   
## 1st Qu.: 36.25 1st Qu.: 2.169 Class :character 1st Qu.:54.00   
## Median : 69.50 Median : 2.427 Mode :character Median :57.00   
## Mean : 69.16 Mean : 2.858 Mean :57.54   
## 3rd Qu.:102.75 3rd Qu.: 2.936 3rd Qu.:60.00   
## Max. :138.00 Max. :17.641 Max. :72.00   
## department treated speed department\_category  
## Min. :1.000 Length:134 Min. :0.3401 1:29   
## 1st Qu.:2.000 Class :character 1st Qu.:2.0434 2:31   
## Median :3.000 Mode :character Median :2.4718 3:41   
## Mean :2.582 Mean :2.4179 4:33   
## 3rd Qu.:3.000 3rd Qu.:2.7668   
## Max. :4.000 Max. :4.1815

Note that in the summary of our dataframe, ‘department\_factor’ is now treated appropriately. Lets see how the regression output changes:

model7 <- lm(data=alldata2, speed ~ age + sex + treated + department\_category)  
summary(model7)

##   
## Call:  
## lm(formula = speed ~ age + sex + treated + department\_category,   
## data = alldata2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.93483 -0.29481 -0.00985 0.34455 1.77996   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.57937 0.60975 5.870 3.57e-08 \*\*\*  
## age -0.02532 0.01116 -2.268 0.025032 \*   
## sexM -0.03937 0.13133 -0.300 0.764829   
## treatedtreated 0.42714 0.10715 3.986 0.000113 \*\*\*  
## department\_category2 -0.07199 0.15953 -0.451 0.652592   
## department\_category3 0.25240 0.15139 1.667 0.097921 .   
## department\_category4 0.20285 0.15666 1.295 0.197736   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6123 on 127 degrees of freedom  
## Multiple R-squared: 0.1916, Adjusted R-squared: 0.1534   
## F-statistic: 5.018 on 6 and 127 DF, p-value: 0.0001201

tab\_model(model7)

speed

Predictors

Estimates

CI

p

(Intercept)

3.58

2.38 – 4.77

<0.001

age

-0.03

-0.05 – -0.00

0.025

sexM

-0.04

-0.30 – 0.22

0.765

treatedtreated

0.43

0.22 – 0.64

<0.001

department category 2

-0.07

-0.38 – 0.24

0.653

department category 3

0.25

-0.04 – 0.55

0.098

department category 4

0.20

-0.10 – 0.51

0.198

Observations

134

R2 / R2 adjusted

0.192 / 0.153

R has given us an estimate of the effect of each department, compared to department 1, with a confidence interval and p-value. It is probably a better question to ask whether the addition of ‘department’ led to a better model, that is, ask for an omnibus test of effect of ‘department’.

anova(model5, model7)

## Analysis of Variance Table  
##   
## Model 1: speed ~ age + treated + sex  
## Model 2: speed ~ age + sex + treated + department\_category  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 130 50.098   
## 2 127 47.619 3 2.4782 2.2031 0.09092 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Finally, we might also be interested in comparisons between each pair of levels. The best way to get this is via the ‘emmeans’ package as follows:

library(emmeans)  
  
pairs(emmeans(model7, ~department\_category))

## contrast estimate SE df t.ratio p.value  
## 1 - 2 0.0720 0.160 127 0.451 0.9693   
## 1 - 3 -0.2524 0.151 127 -1.667 0.3453   
## 1 - 4 -0.2028 0.157 127 -1.295 0.5679   
## 2 - 3 -0.3244 0.147 127 -2.212 0.1254   
## 2 - 4 -0.2748 0.155 127 -1.777 0.2890   
## 3 - 4 0.0496 0.147 127 0.337 0.9868   
##   
## Results are averaged over the levels of: sex, treated   
## P value adjustment: tukey method for comparing a family of 4 estimates

The output from emmeans includes a ‘marginal’ estimate for the walking speed in each department, plus an estimate and statistical test for each department compared to every other, with a suitable p-value correction for multiple testing.

## Exericse:

Can you use lm to test whether walking speed varies with department, and whether the effect of treatment on walking speed varies with department?

## Extensions to other models

Almost every experiment you do can be analysed with this paradigm, that is an outcome variable depending on one or more predictors. And so data from almost every experiment can be analysed and reported with lm() or a related function.

In practice the modelling functions I find useful for most analyses are:

* lm() - regression models and ANOVA
* glm() - generalised linear models (count data and binary outcomes)
* lmer() and glmer() - from the lme4 package for mixed effects models (when the assumption of independence is not met, analogous to repeated measures ANOVA)
* nlme() for non-linear models

## Further reading on analysis with R

More detailed linear modelling tutorial. <http://tutorials.iq.harvard.edu/R/Rstatistics/Rstatistics.html>

Understanding the linear regression diagnostic plots: <http://www.sthda.com/english/articles/39-regression-model-diagnostics/161-linear-regression-assumptions-and-diagnostics-in-r-essentials/>

Using emmeans to get contrasts and margins <https://aosmith.rbind.io/2019/03/25/getting-started-with-emmeans/>