Regression and Mixed-effects Modelling in R : Session 1

Fang Jackson-Yang

2024-04-29

1 Preparation

1.1 Install packages

We will need the following packages for this course:

"tidyverse" : used for cleaning and sorting out data.

"lme4" : used for fitting linear mixed-effects models (LMMs).

"effects" : used for creating tables and graphics that illustrate effects in linear models.

"sjPlot" : used for plotting models.

"interactions": used for plotting interaction effects.

Note that in R markdonw, packages should be installed in the console (the bottom-left panel) and libraries in the source panel (the top-left panel) > install.packages(c("tidyverse", "lme4", "effects", "sjPlot", "interactions")).

1.2 load data

We are going to use a simulated dataset called "vocabtrainingdata". The data are simulated but are based on a real case from UK's education section.

```
vocabdata <- read_csv("vocabtrainingdata.csv")
```

A secondary school in Glasgow were considering to invest in a new online course for vocabulary teaching in foreign language classes (French). Before the school could make a decision, they wanted to look into the effectiveness of the online course. They were particularly interested in its effectiveness in improving the vocabulary competence of pupils in S4 (i.e.,the 4th yr in Scottish secondary schools).

40 students from two proficiency groups took part, half with high proficiency level of French (high proficiency group) and the other half with intermediate proficiency (intermediate proficiency group).

Students participated in this course over 10 weeks. In the first week (Week 0), students were introduced to the course and did a pre-course vocabulary test. In the following nine weeks (week 1-9), each week they received one training session online in their own time followed by a vocabulary test one day after the training session. Most students completed all training sessions and took all tests, however, some students missed a few sessions or dropped out at some point during the course.

Our aim is to build regression models and make inferences about the effectiveness of the online training course on improving students' vocabulary competence, in order to inform the school about whether it would be a good investment.

2 Data wrangling

2.1 Check data

```
summary(vocabdata)
```

```
##  participant        proficiency             week     vocab_test_score
##  Length:400         Length:400         Min.   :0.0   Min.   :39.00   
##  Class :character   Class :character   1st Qu.:2.0   1st Qu.:58.00   
##  Mode  :character   Mode  :character   Median :4.5   Median :72.00   
##                                        Mean   :4.5   Mean   :68.89   
##                                        3rd Qu.:7.0   3rd Qu.:79.00   
##                                        Max.   :9.0   Max.   :93.00   
##                                                      NA's   :31
```

```
head(vocabdata)
```

```
## # A tibble: 6 x 4
##   participant proficiency   week vocab_test_score
##   <chr>       <chr>        <dbl>            <dbl>
## 1 M01         intermediate     0               50
## 2 M01         intermediate     1               52
## 3 M01         intermediate     2               50
## 4 M01         intermediate     3               64
## 5 M01         intermediate     4               71
## 6 M01         intermediate     5               83
```

```
str(vocabdata)
```

```
## spc_tbl_ [400 x 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ participant     : chr [1:400] "M01" "M01" "M01" "M01" ...
##  $ proficiency     : chr [1:400] "intermediate" "intermediate" "intermediate" "intermediate" ...
##  $ week            : num [1:400] 0 1 2 3 4 5 6 7 8 9 ...
##  $ vocab_test_score: num [1:400] 50 52 50 64 71 83 76 58 86 93 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   participant = col_character(),
##   ..   proficiency = col_character(),
##   ..   week = col_double(),
##   ..   vocab_test_score = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
table(is.na(vocabdata))
```

```
## 
## FALSE  TRUE 
##  1569    31
```

```
vocabdata$proficiency <- as.factor(vocabdata$proficiency)# code categorical variables factors
levels(vocabdata$proficiency)# check levels
```

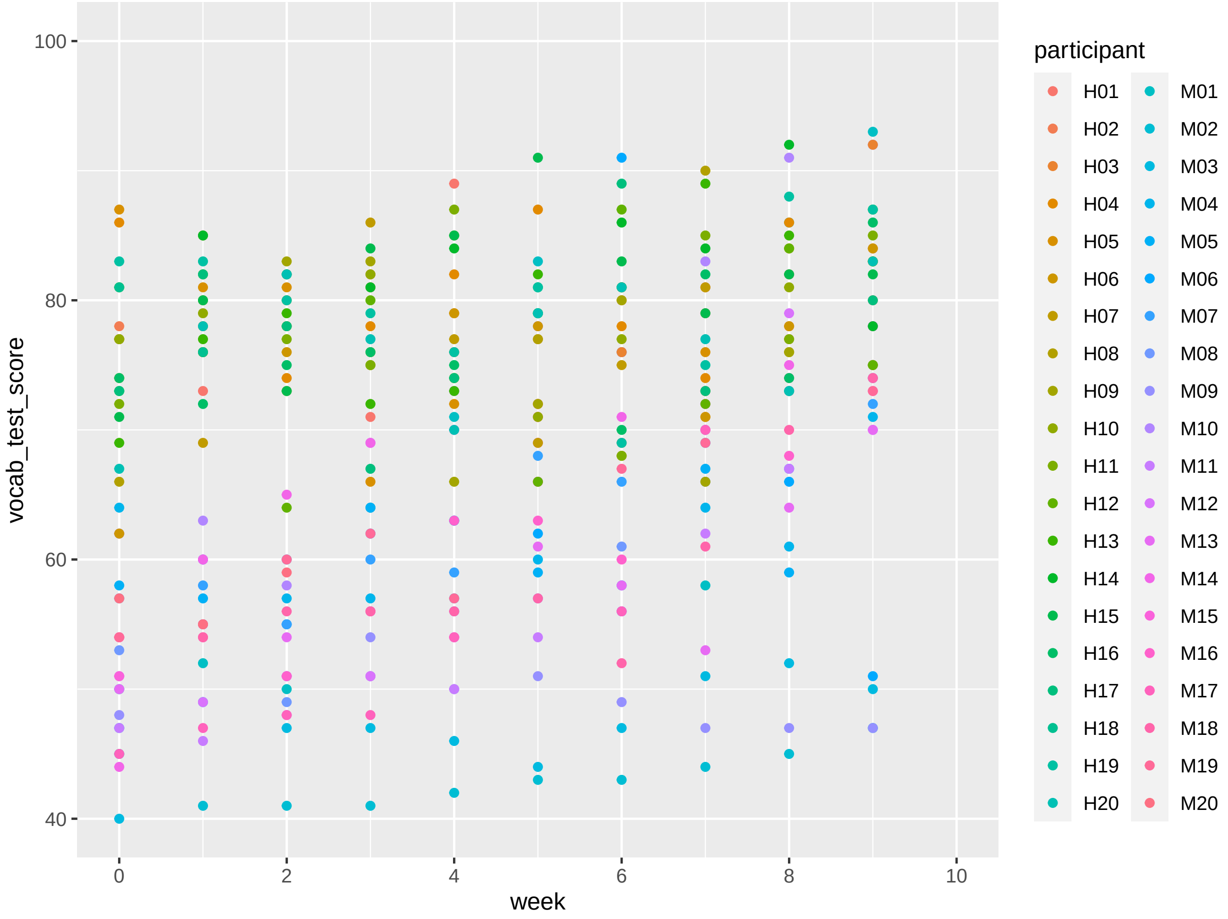
```
## [1] "high"         "intermediate"
```

Remember to properly code your variables before performing analyses. For categorical variables, check the reference level. Relevel it if you want by using the fct_relevel() function, for example: vocabdata$proficiency < -fct_relevel(vocabdata$proficiency, “intermediate”).

You may also want to contrast code your variable. There are many different ways of contrast coding. Here is an excellent source if you want to learn mmore about contrast coding:

https://stats.oarc.ucla.edu/r/library/r-library-contrast-coding-systems-for-categorical-variables/

2.2 Visualise data



3 Simple regression analysis

3.1 Simple regression with one predictor

Recall that our aim is to look into the effectiveness of the online training course on participants’ vocabulary competence - measured in their vocab test scores over the weeks.

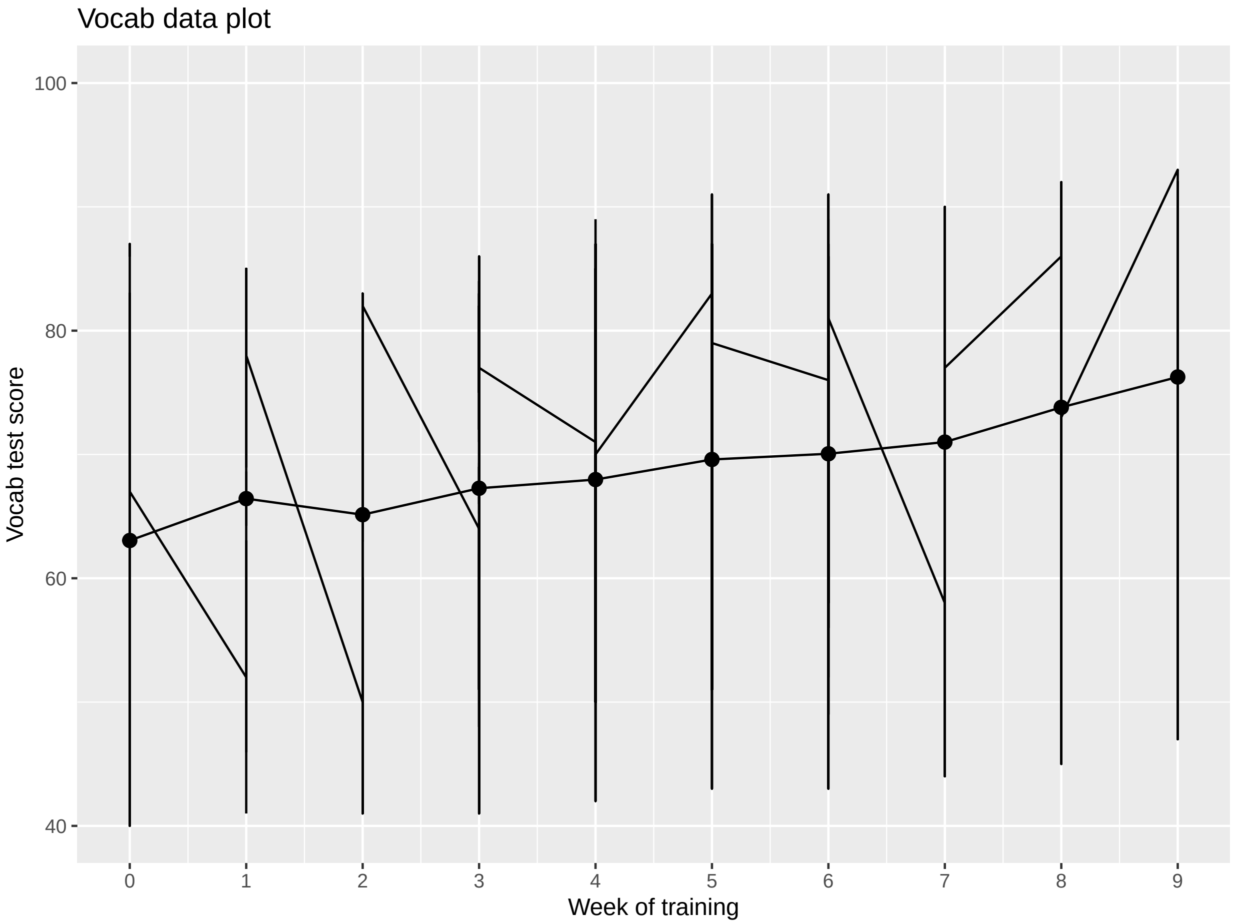
In other words, we need to build a model to inform us about how vocab test scores can be explained by their progress in the online course (week1, week2 and so forth).

we can call their progress in the online course or "week" a PREDICTOR variable (AKA independent variable), and their vocab test score an OUTCOME variable (AKA dependent variable).

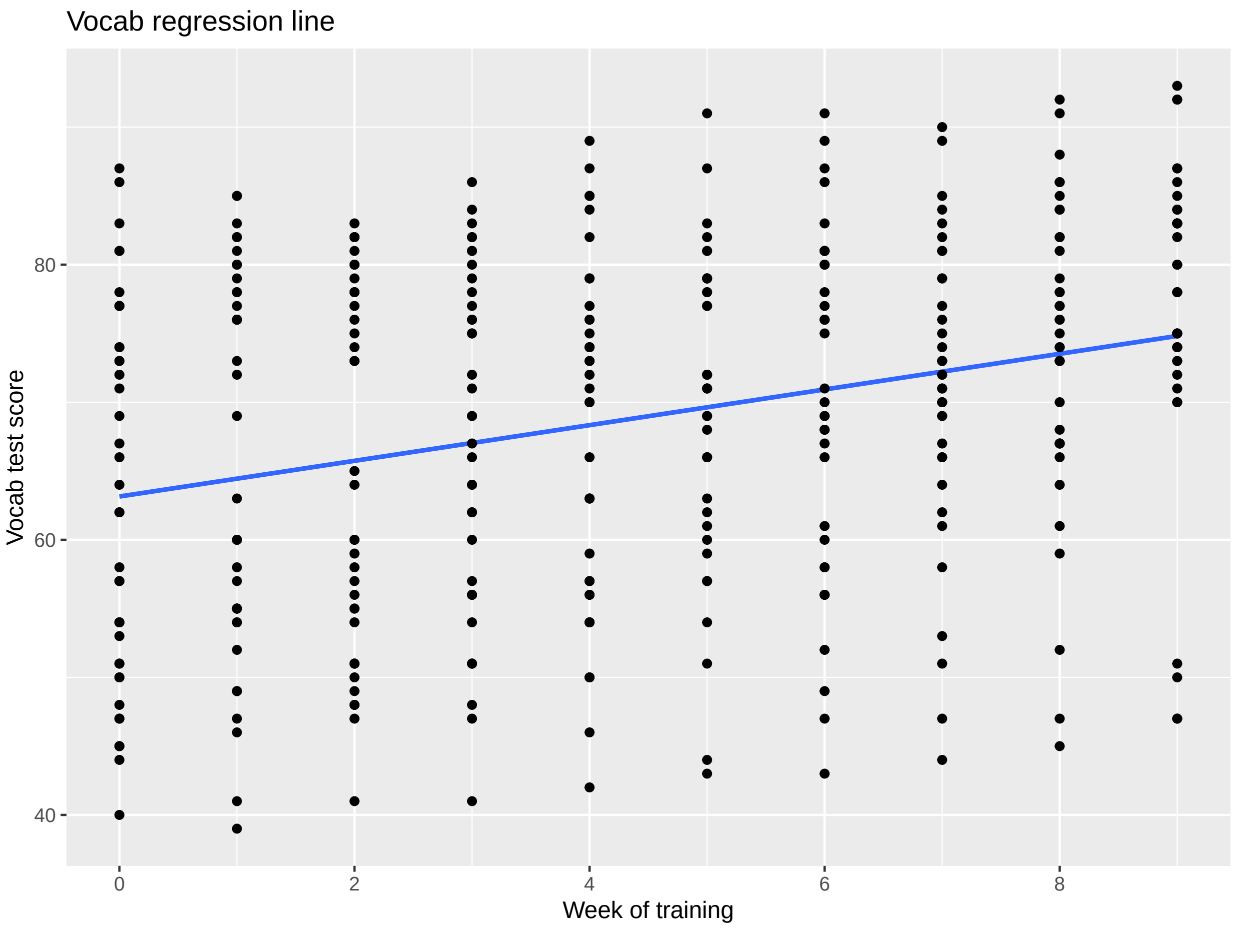
Essentially, we want to investigate how "vocab_test_score" is explained by "week" in a regression model.

3.1.1 Visualise the data

We first plot the data without specifying any relation between week and test score.



Next we plot the data by specifying a linear relation between week and test score - note that we have not built any models yet, we are just smoothing the line/trend by adding this specification.



The relationship does look linear. Next we will try to fit a simple regression model to investigate this further.

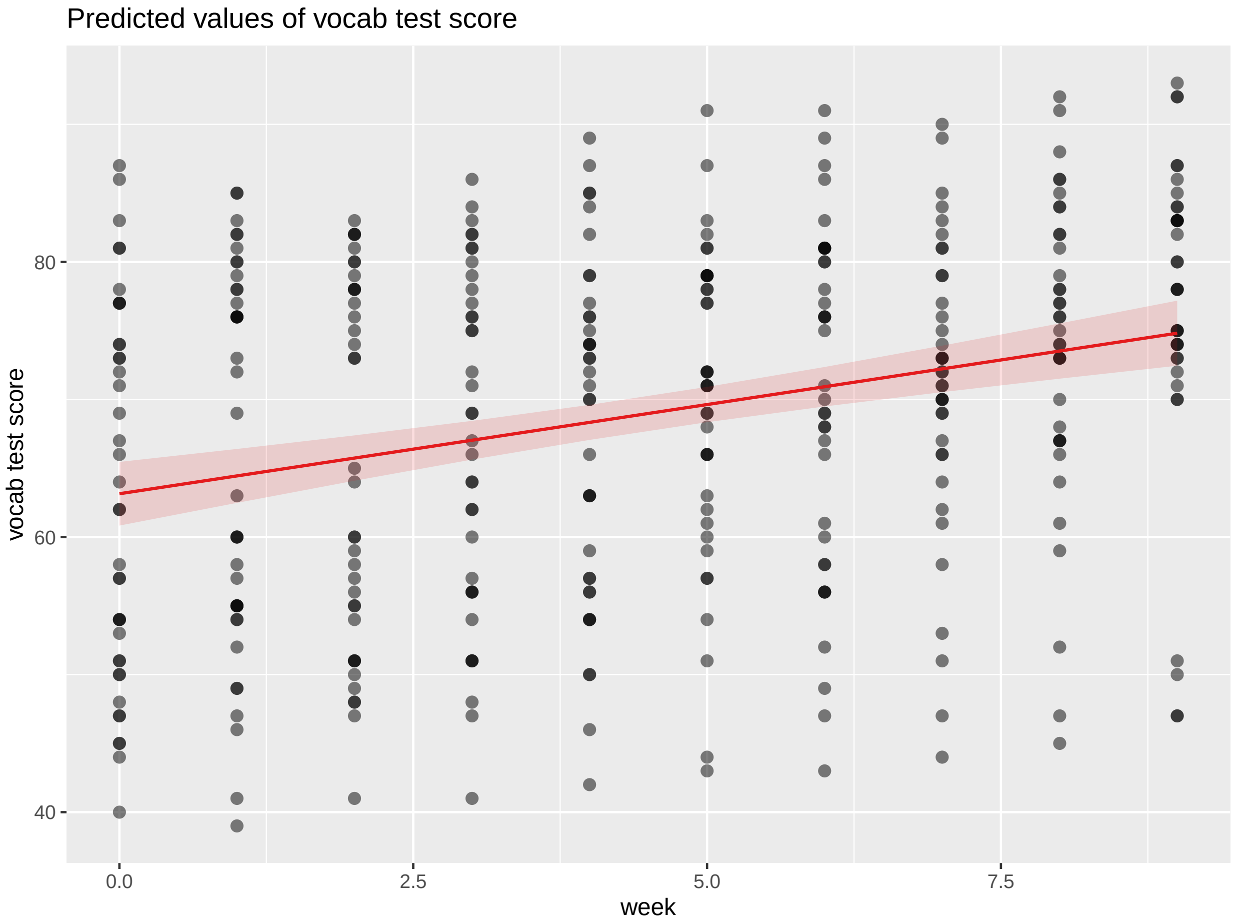
3.1.2 Fit a simple model

We fit a simple regression model by taking information from all observations without considering individual differences. In other words, this model assumes the vocab online training course has the same effect on every student.

```
m1 <- lm(vocab_test_score ~ week, data = vocabdata)
summary(m1)
```

```
##
## Call:
## lm(formula = vocab_test_score ~ week, data = vocabdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -28.5202  -9.4438   0.7764  10.0731  23.8528
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  63.1472     1.1765  53.673  < 2e-16 ***
## week          1.2966     0.2229   5.818  1.3e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.3 on 367 degrees of freedom
##   (31 observations deleted due to missingness)
## Multiple R-squared:  0.08445,    Adjusted R-squared:  0.08195
## F-statistic: 33.85 on 1 and 367 DF,  p-value: 1.297e-08
```

3.1.3 visualise the model (predicted effects)



3.1.4 Interpret model output

Important Concepts

Intercept: the expected value of the outcome variable when the predictor variable is at 0. In this case, overall students are expected to have a score of 63.15 when they start the training course in week0). Represented in the plot, it is the point where the regression line crosses y when x = 0.

Slope: the number of units by which the outcome variable increases, on average, for a unit increase in the predictor variable.In this case, students' test scores are expected to improve by 1.30 point on average, as they proceed into a new week in the course. Represented in the plot, it is the "slope" of the line - hence the name; it shows the rate of change.

Residuals (AKA errors): the difference between the expected value that the model predicts and each actual data point. Residuals measure how good a model is (in terms of fitting or predicting the data).

R-squared: how much the model explains the actually observed data, in this case about 8%.

F-statistics/F-ratio: the significance of the overall model compared to a null hypothesis (which assumes the model explains nothing). The p-values indicates how likely you would get the observed data given the null hypothesis - a smaller p-value indicates a smaller likelihood. In this case, P < .001 (0.1%), it's very very unlikely to get the observed data if the null hypothesis was true. We can say that the results reject the null hypothesis but support our hypothesis that the predictor "week" does help explain the data.

BREAK

WELL DONE on fitting your first regression model (for this dataset)!

Hopefully you now have clearer understanding about the basics of linear regression models and important concepts.

Let's take a break.

3.2 Simple regression with two predictors and their interaction

In the analysis above, We ignored students' proficiency level. Does it help explain the data if we additionally include proficiency level as a predictor?

3.2.1 Additive model (week + proficiency)

```
m2a <- lm(vocab_test_score ~ week + proficiency, data = vocabdata)
summary(m2a)
```

```
##
## Call:
## lm(formula = vocab_test_score ~ week + proficiency, data = vocabdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.2192  -5.4079  -0.1221   5.0211  29.3524
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              72.4073     0.8835  81.951   <2e-16 ***
## week                      1.2858     0.1469   8.752   <2e-16 ***
## proficiencyintermediate -18.4745     0.8445 -21.876   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.111 on 366 degrees of freedom
##   (31 observations deleted due to missingness)
## Multiple R-squared:  0.6032, Adjusted R-squared:  0.6011
## F-statistic: 278.2 on 2 and 366 DF,  p-value: < 2.2e-16
```

Can you interpret the results? What do these numbers mean?

Intercept: 72.41

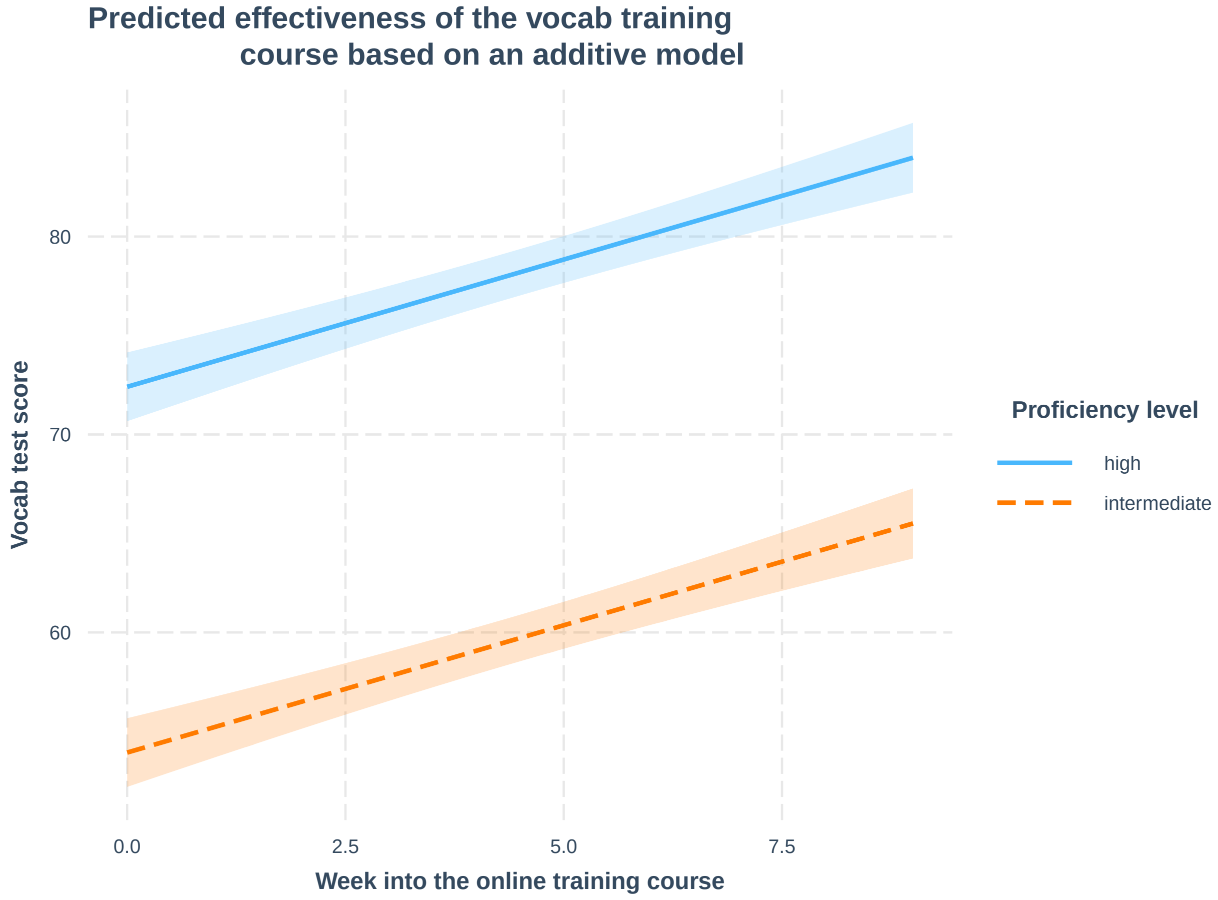
Slope1(week): 1.29

Slope2(proficiencyintermediate): -18.47

Residuals (Adjusted R-squared): 0.60

F-statistics: F(2, 366) = 278.2, p < .001

```
## SIMPLE SLOPES ANALYSIS
##
## Slope of week when proficiency = intermediate:
##
##   Est.   S.E.   t val.      p
## ------ ------ -------- ------
##   1.29   0.15     8.75   0.00
##
## Slope of week when proficiency = high:
##
##   Est.   S.E.   t val.      p
## ------ ------ -------- ------
##   1.29   0.15     8.75   0.00
```



3.2.2 Think point:

Is this additive two-predictor model better than the one-predictor model? In other words, does knowing students' proficiency level help us better understand the effectiveness of the training course?

We can use the anova() function to compare the two models.

```
anova(m1, m2a)
```

```
## Analysis of Variance Table
##
## Model 1: vocab_test_score ~ week
## Model 2: vocab_test_score ~ week + proficiency
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    367 55563
## 2    366 24079  1     31485 478.58 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpret the results of the model comparison:

Results show yes, the two-predictor model (m2a) significantly improved model fit compared to the one-redictor model (m1); F(1) = 478.58, p < 0.001.

Check again the adjusted R-squared of the two models that we got above:

For the one-predictor model "m1", Adjusted R-squared: 0.082, F(1, 367) = 33.85, p <.001 (rejecting null hypothesis)

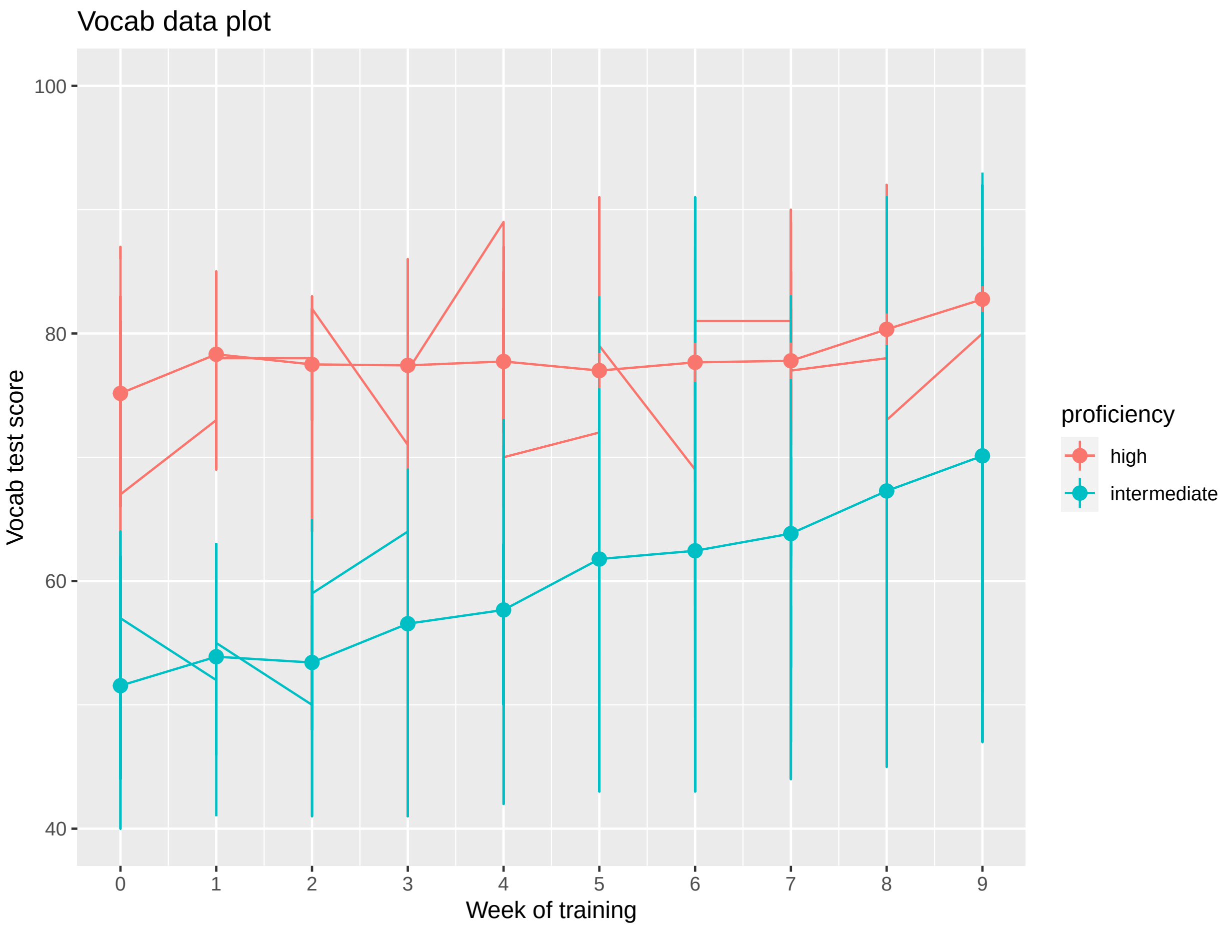
for the two-predictor model "m2a", Adjusted R-squared: 0.60, F(2, 366) = 278.2, p <.001 (rejecting null hypothesis)

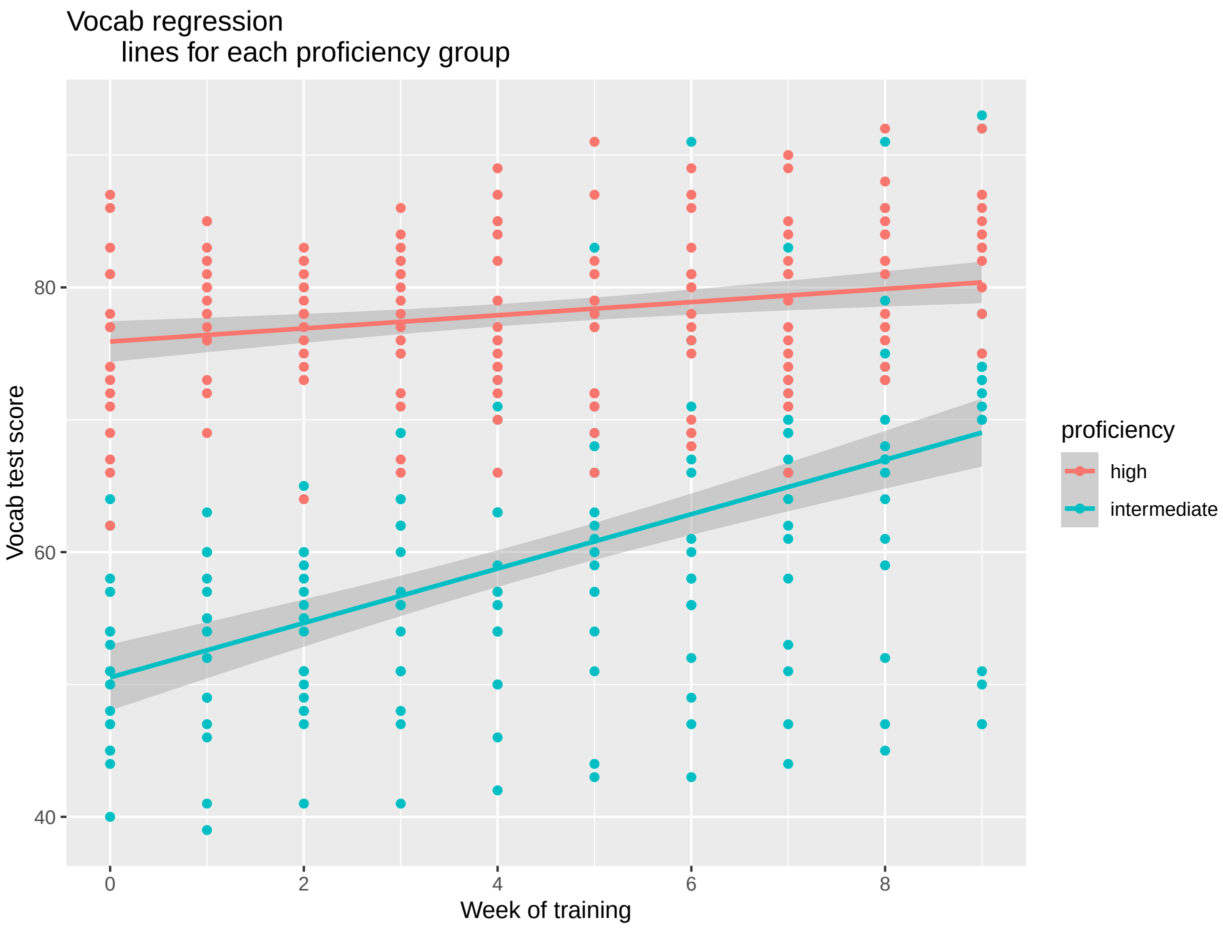
3.2.3 Interactive model (week * proficiency)

How do we know whether the effectiveness of the training course is dependent on students' proficiency level or not? Did students with different proficiency levels show similar or different progressive trend as they took the course?

In other words, is the training course as effective or ineffective for the high-proficient students compared to the intermediate-proficient students?

3.2.3.1 Visualise the data We can use ggplot() for this.


Vocab data plot
100
80
60
40
Vocab test score
0
1
2
3
4
5
6
7
8
9
Week of training
proficiency
high
intermediate



3.2.3.2 Fit an interactive model The following two model structures are identical.

X1*X2 is equal to (X1 + X2 + X1:X2)

Version 2 is just a simplified way of specifying both main effects (e.g., X1, X2) and interaction effects (e.g., X1:X2) in R.

From now on, we will use the shortened version (e.g., X1*X2) in the rest of the course.

```
# version 1
m2bv1 <- lm(
  vocab_test_score ~ week + proficiency + week:proficiency, data = vocabdata
  )
summary(m2bv1)
```

```
##
## Call:
## lm(formula = vocab_test_score ~ week + proficiency + week:proficiency,
##     data = vocabdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.0330  -4.6421   0.5941   4.6204  28.1345
##
```

```
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   75.9092     1.0610  71.545  < 2e-16 ***
## week                           0.4967     0.2011   2.470    0.014 *
## proficiencyintermediate      -25.3788     1.4924 -17.005  < 2e-16 ***
## week:proficiencyintermediate   1.5591     0.2827   5.515 6.61e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 7.803 on 365 degrees of freedom
##   (31 observations deleted due to missingness)
## Multiple R-squared:  0.6338, Adjusted R-squared:  0.6308
## F-statistic: 210.5 on 3 and 365 DF,  p-value: < 2.2e-16
```

```r
# version 2
m2bv2 <- lm(vocab_test_score ~ week * proficiency, data = vocabdata)
summary(m2bv2)
```

```
##
## Call:
## lm(formula = vocab_test_score ~ week * proficiency, data = vocabdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.0330  -4.6421   0.5941   4.6204  28.1345
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   75.9092     1.0610  71.545  < 2e-16 ***
## week                           0.4967     0.2011   2.470    0.014 *
## proficiencyintermediate      -25.3788     1.4924 -17.005  < 2e-16 ***
## week:proficiencyintermediate   1.5591     0.2827   5.515 6.61e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 7.803 on 365 degrees of freedom
##   (31 observations deleted due to missingness)
## Multiple R-squared:  0.6338, Adjusted R-squared:  0.6308
## F-statistic: 210.5 on 3 and 365 DF,  p-value: < 2.2e-16
```

Can you interpret the results? What do these number mean?

Intercept: 75.91 On the plot: The point at which the regression line of the reference level (i.e., high proficiency) cuts the y-axis.

Slope1(week): 0.50 On the plot: The slope of the regression line of the reference level (i.e., high proficiency) (increase of score by 1 week in the reference group).

Slope2(proficiencyintermediate): -25.38 On the plot: the distance between the two regression lines on the y-axis when $x = 0$. i.e., difference of the test scores between teh two groups at week0.

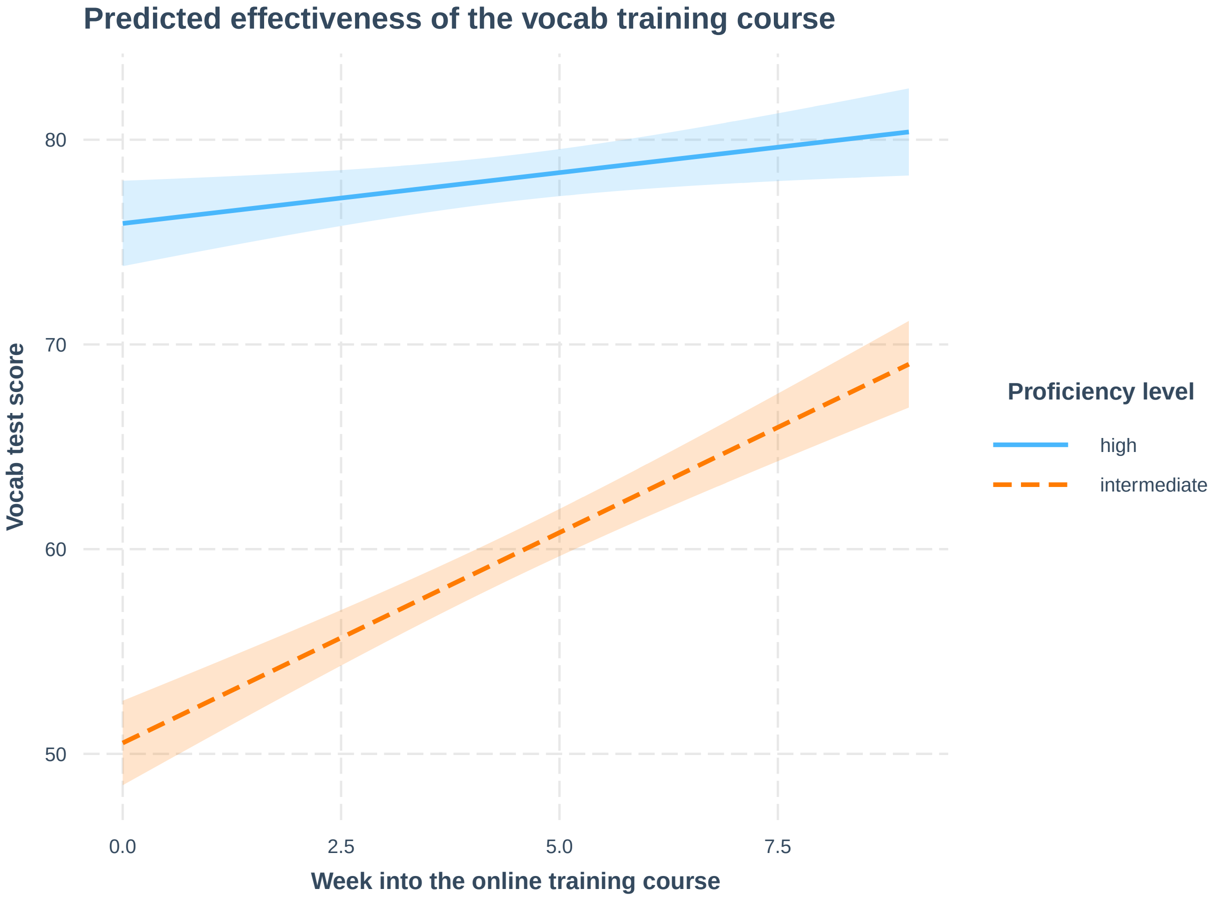
Slope3(week:proficiencyintermediate): 1.56 On the plot: change of the steepness from the regression line of the reference group (high-proficiency, blue line) to the regression line of the other proficiency group (intermediate-proficiency, orange line).

Residuals(errors) Adjusted R-squared: 0.63

F-statistics/F-ratio: $F(3, 365) = 210.5$, $p< .001$

3.2.3.3 Visulise the interactive model (predicted effects)

```
## SIMPLE SLOPES ANALYSIS
##
## Slope of week when proficiency = intermediate:
##
##   Est.   S.E.   t val.      p
## ------ ------ -------- ------
##   2.06   0.20    10.35   0.00
##
## Slope of week when proficiency = high:
##
##   Est.   S.E.   t val.      p
## ------ ------ -------- ------
##   0.50   0.20     2.47   0.01
```



3.2.4 Model comparison

We have built an additive model and an interactive model and we know either of them has improved the model fit compared to null model.

Is the interactive model better than teh additive model? In other words, does the interaction between week and proficiency improve the quality of the model and explain the data better?

As before, we use the anova(model1, model2) function to test this statistically.

```
anova(m2a, m2bv1)
```

```
## Analysis of Variance Table
##
## Model 1: vocab_test_score ~ week + proficiency
## Model 2: vocab_test_score ~ week + proficiency + week:proficiency
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    366 24079
## 2    365 22226  1    1852.3 30.418 6.605e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results suggest that the interactive model significantly improved model fit, F(1)=30.42, P < .001.

Check again the adjusted R-squared, we can see the additive model explained 60% of the data whereas the interactive model explained 63%.

BREAK

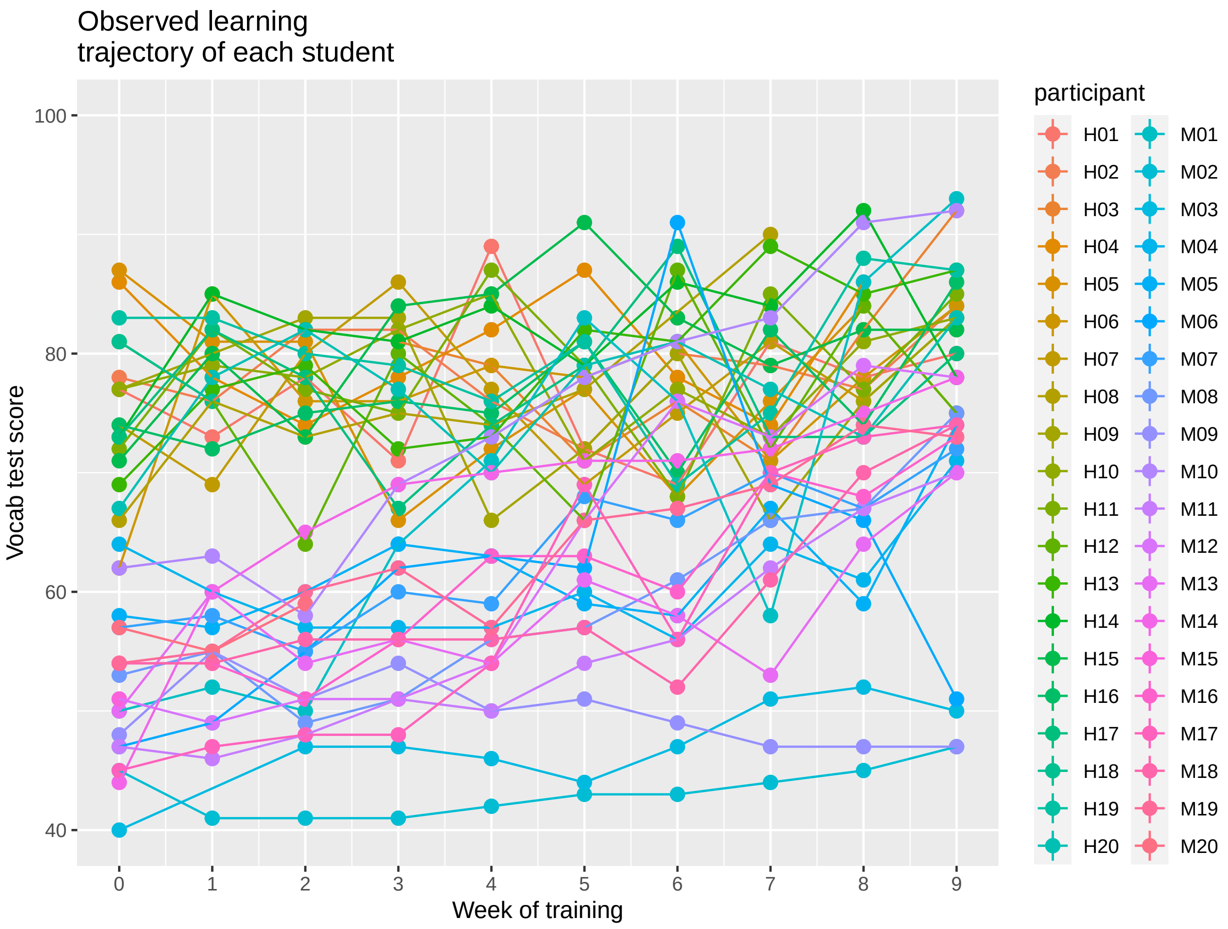
WELL DONE on completing Section 2!

Let's take a break.

4 Individual differences and linear mixed-effects models (LMMs)

Our interactive model can explain 63% of the data, which is not bad. However, the model took information from all observations without considering individual differences. In other words, it assumed the vocab online training course had the same effect on every student. Was this the case?

Let's visualise it and see.



We can clearly see that each student showed a different trend. We should not ignore this information when modelling. The question is how do we deal with such individual differences?

4.1 First appraoch

You might wonder whether we could just add participants as a predictor variable - just as how we did it when adding a second predictor "proficiency" to the one-predictor model (including only one predictor "week"). Theoretically we can do this but whether it is a good approach needs some thinking. Let's try it out first.

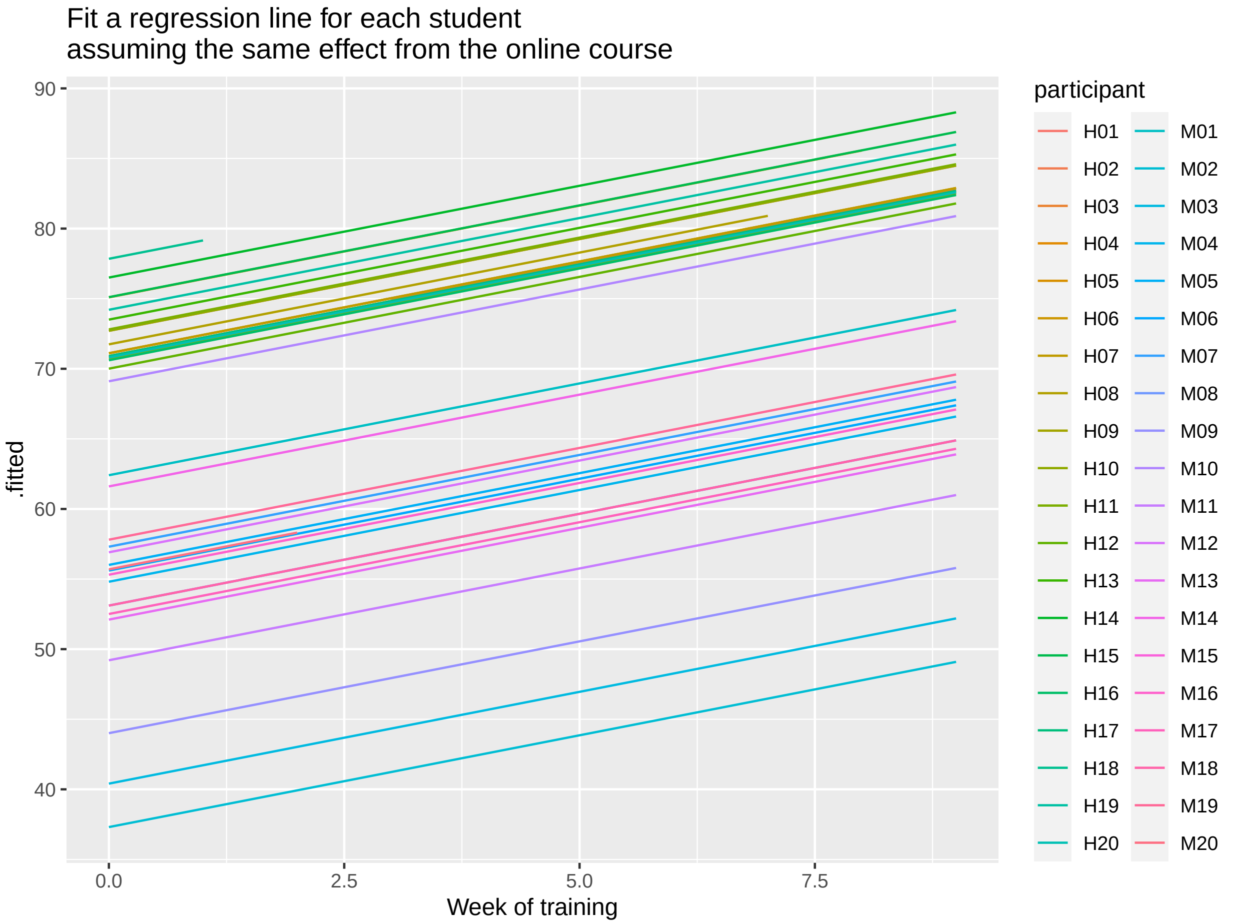
Here our aim is to understand how to best deal with individual difference (rather than build an optimal model that can best explain the data). For this purpose, we want to keep things simple, therefore we just focus on the effect of "week" for now (leaving proficiency aside).

We will build a model including "week" and "participant" as two predictor variables, this way we fit a separate regression line for each participant. As before, We will fit two models, one is additive (including two predictors, i.e., week and participant) and the other is interactive (including three predictors, i.e., week, participant, and the interaction between week and participant).

4.1.1 (a). Additive model accounting for individual difference

4.1.1.1 Fit an additive model This additive model assumes "week" has the same effect on each participant.

```
m3a <- lm(vocab_test_score ~ week + participant, data = vocabdata)
summary(m3a)
```



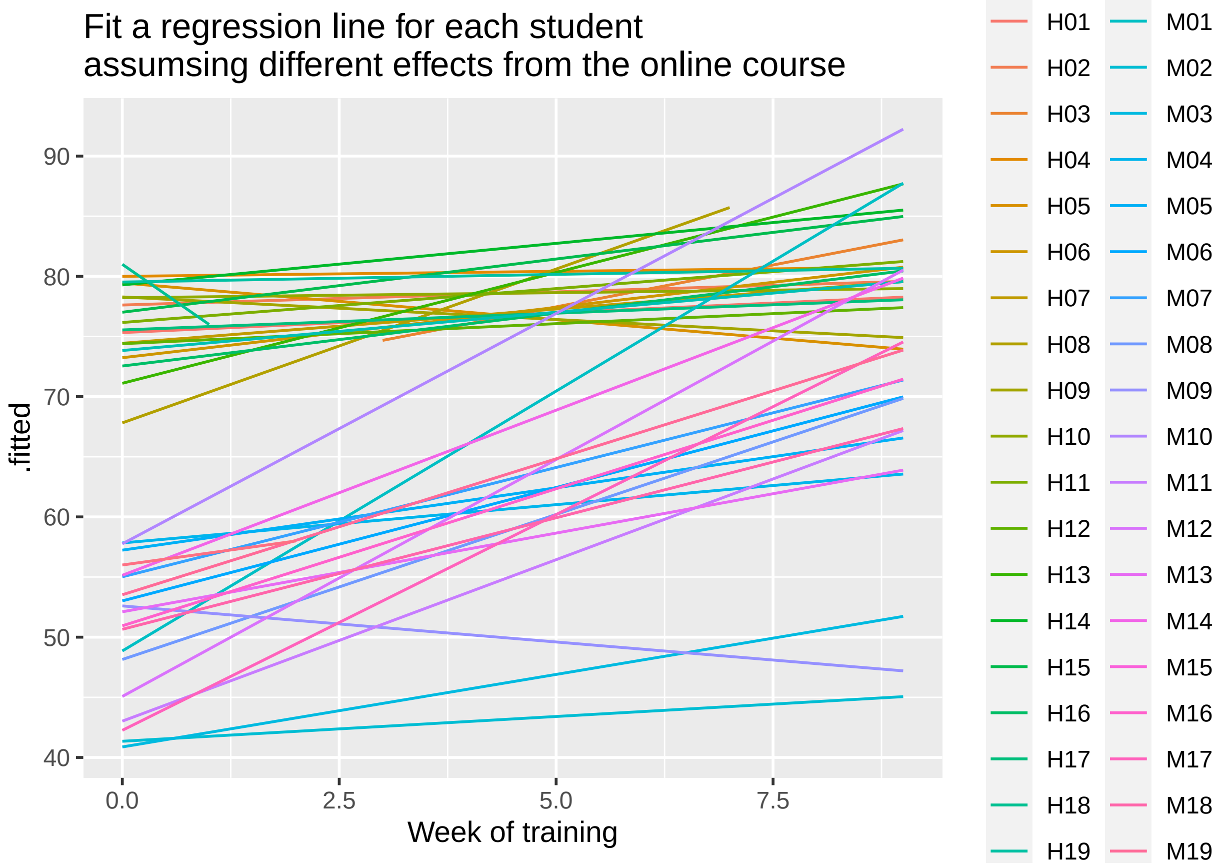
4.1.2 (b). Interactive model accounting for individual difference

In the interactive model, we assume that the effect of training/week on test scores is dependent of participants (i.e., students had different starting point - different intercepts, but also showed different learning patterns as they took the training - different slopes as well).

4.1.2.1 Fit an interactive model The interactive model assumes "week" has different effect on each participant.

```
m3b <- lm(vocab_test_score ~ week * participant, data = vocabdata)
summary(m3b)
```

```
broom::augment(m3b) %>%
 ggplot(.,aes(x=week, y=.fitted, color=participant))+
 geom_line()+
  labs(x="Week of training",  title = "Fit a regression line for each student
assumsing different effects from the online course")
```



4.2 A better approach: Treat is as a random effect in a linear MIXED-effects model

In the first approach above, we included participants as a predictor variable (independent variable). This can be done as you see in the model results, however, this is not very helpful for us.

First of all, the regression results gave us a lot of coefficients (check how many rows the output tables have), making it difficult to make inferences or generalise the general tendency - here we only had 40 participants but imagine your research deals with a larger sample size with hundreds or thousands of participants.

Importantly, the individual differences do not influence the general tendency in a structured manner; they are rather random. For example, a student could be particularly fast or slow in progressing in their learning, or they could have a more or less complex trajectory, these are all random. But in quantitative research, we aim to generalise; we want to see whether there is the main effect of the predictor that we are interested in (here, the online training course) on students' improvement, after accounting for such individual differences.

Therefore, it is better to treat individual differences as a RANDOM EFFECT (rather than a main predictor) in a regression model. We can then call our predictor variables (e.g., week, proficiency) FIXED EFFECTS, which influence the outcome variable in a structured manner.

This leads us to linear MIXED-effects regression models; we mix the FIXED effects and the RANDOM effects. In doing so, we can control for individual differences while modelling the influences of our predictor variables on the outcome variable that we are interested in.

4.2.1 Think point:

We are not going to look into mixed models in detail in today's session, but I want to show you what the model structure looks like for a mixed-effects model.

Recall the structure of our simple regression model (the interactive model):

(1)m2bv2 <- lm(vocab_test_score ~ week * proficiency, data = vocabdata)

Compare it with the two mixed-effects models below:

(2)mixedm1 <- lmer(vocab_test_score ~ week * proficiency + (1 + week | participant), data = vocabdata)

(3)mixedm2 <- lmer(vocab_test_score ~ week * proficiency + (1 | participant), data = vocabdata)

4.2.1.1 Questions for you:

1. How do the two linear mixed-effects models, i.e., (2) & (3), differ from the simple regression model (1)?
2. How does (2) differ from (3)?

You can very quickly run the mixed-effect models (2) and (3) below to get a bit of the taste.

4.2.2 Fit a mixed-effects model

A note on the "lmerTest" package. Here we use the Satterthwaite method to add a column of p-values to the results. Unlike in simple regression models lm(), in mixed models we do not automatically get p values. This is because in mixed models, we have residuals at multiple levels thus we don't know what kind of distributions the ratios of sums of squares are. In contrast, in simple regressions we know better about the distributions (F or t distribution).

```r
mixedm1 <- lmer(vocab_test_score ~ week*proficiency + (1 + week | participant),
                data = vocabdata)
summary(mixedm1)
```

```r
mixedm2 <- lmer(vocab_test_score ~ week * proficiency + (1 | participant),
                data = vocabdata)
summary(mixedm2)
```