Exercise 13 - Repeated measures analysis with mixed models

Zoltan Kekecs

19 november 2020

Table of Contents

Abstract

This exercise is focused on the use of linear mixed models in case of repeated measures designes, and how to re-structure data from a 'wide format' to a 'long format' to be useful for linear mixed model analysis

Data management and descriptive statistics

Loading packages

You will need the following packages for this exercise.

```
library(psych) # for describe
library(tidyverse) # for tidy code and ggplot
library(lme4) # for lmer() mixed models
library(lmerTest)  # for significance test on lmer() mixed models
library(cAIC4) # for cAIC
library(r2glmm) # for r2beta
library(MuMIn) # for r.squaredGLMM
```

Custom functions

This is a function to extract standardized beta coefficients from linear mixed models.

This function was adapted from: https://stackoverflow.com/questions/25142901/standardized-coefficients-for-lmer-model

```
stdCoef.merMod <- function(object) {
  sdy <- sd(getME(object,"y"))
  sdx <- apply(getME(object,"X"), 2, sd)
  sc <- fixef(object)*sdx/sdy
  se.fixef <- coef(summary(object))[,"Std. Error"]
  se <- se.fixef*sdx/sdy
  return(data.frame(stdcoef=sc, stdse=se))
}
```

Load wound healing data

In this exercise we will work with simulated data about wound healing over time after a surgical procedure. We know that psychological factors, especailly stress, can influence recovery after surgery, and the rate of wound healing. Let's say that we have a theory that wound it is important for hospitalized patients to have a connection with the outside world. So we may think that patients who have a window close to their hospital beds may have a better mood and thus, would show a faster recovery after surgery. This hypothesis is tested in a simple study looking at whether the distance of the patients' bed from the closest window would predict rate of wound healing. Distance is measured in meters, and wound healing is measured by rating the wound using a standardized wound rating measure taking into account the size of the wound, its inflammation and scarring. A physician rates the wound each day for seven days in the afternoon at the same time of the day. We will use this variable as our outcome measure.

Let's say that our hypothesis extends to the role of sunlight in this context, where we suppose that the more sunlight a patient gets the better their recovery would be. To test this hypothesis, our model will take into account whether the bed of the patient is in the north wing or the south wing of the hospital (since the hospital is in the northern hemisphere, we can assume that patients in the south wing would get more sunlight overall during their hospital stay).

There are the following variables in the dataset:

- ID: participant ID
- day_1, day_2, ..., day_7: Wound rating scores collected on different days of the study. The would was assessed by an expert first 1 day post surgery, and it was assessed once every day after that until 7 days after the operation. The higher the score the larger the wound size or the worse the wound condition (e.g. due to inflammation). The smaller the score the closer the wound is to being healed.
- distance_window: The distance of the patient's bed from the closest window in meters.
- location: The hospital wing in which the patient's bed is located can take value "north wing" or "sounth wing". This can be important if the amount of sunlight matters, since the south wing hospital rooms get more sunlight in this hospital.

```
data_wound = read_csv("https://raw.githubusercontent.com/kekecsz/PSYP14-
Advanced-Scientific-
Methods/main/Exercise_13_Repeated_measures_analysis_with_mixed_models/data_wo
undhealing_repeated.csv")

##
## -- Column specification ---------------------------------------------------
------
## cols(
##   ID = col_character(),
##   day_1 = col_double(),
##   day_2 = col_double(),
##   day_3 = col_double(),
##   day_4 = col_double(),
##   day_5 = col_double(),
##   day_6 = col_double(),
##   day_7 = col_double(),
##   distance_window = col_double(),
##   location = col_character()
## )

# asign ID and location as factors
data_wound = data_wound %>%
  mutate(ID = factor(ID),
         location = factor(location))
```

Lets inspect the layout of this data frame using the View() function. Notice that each row contains all the data collected from the same participant, and the wound rating data for each day are stored in variables 'day_1', 'day_2', ..., 'day_7' respectively. You can also explore your data using the describe and table functions.

```
View(data_wound)

# descriptives
describe(data_wound)
table(data_wound$location)
```

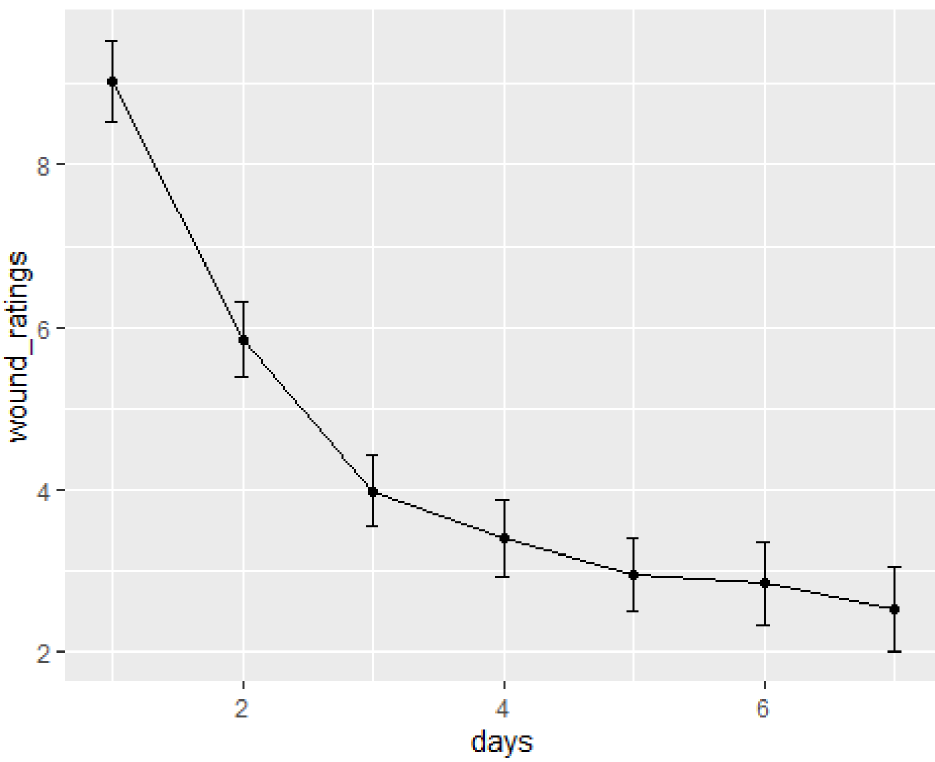
Check the dataset

In the following section we **visualize the change of average wound rating over time** (using geom_point), with the confidence interval of the mean estimate included (using geom_errorbar). (In the code below we first calculate the means and confidence intervals before plugging them in to ggplot. The CI is calculated here manually as the mean +/- 1.96*standard_error (SE)).

```
# designate which are the repeated varibales
repeated_variables = c("day_1", "day_2", "day_3",   "day_4", "day_5",
"day_6",    "day_7")

# explore change over time
wound_ratings = describe(data_wound[,repeated_variables])$mean
repeated_CIs = describe(data_wound[,repeated_variables])$se*1.96
days = as.numeric(as.factor(repeated_variables))
days = as.data.frame(days)
data_for_plot = cbind(days, wound_ratings, repeated_CIs)

data_for_plot %>%
  ggplot() +
  aes(x = days, y = wound_ratings) +
  geom_errorbar(aes(ymin=wound_ratings-repeated_CIs,
ymax=wound_ratings+repeated_CIs), width=.1) +
  geom_point() +
  geom_line()
```



Repeated measures analysis using linear mixed models

Exploring clustering in the data

Now lets look at the relationhsip of the repeated measures of wound healing. (First, we save the variable names of the repeated measures into an object called **repeated_variables** so that we can easily refer to these variable names later in our functions.) We can explore the correlation between the variables with the cor() function. Notice that the repeated measures data points are **highly correlated**. This shows that the different observations of wound rating are **not independent** from each other. This is normal, since the wound rating and the initial size of the incision and the wound healing rate depends on the patient. So this is clustered data. Just like the data in the previous exercise was clustered in classes, here, the data is clustered within participants.

```
# correlation of repeated variables

data_wound %>%
  select(repeated_variables) %>%
  cor()

## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(repeated_variables)` instead of `repeated_variables` to
silence this message.
```

```
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
##           day_1     day_2     day_3     day_4     day_5     day_6
day_7
## day_1 1.0000000 0.8505812 0.6565436 0.5469131 0.4647985 0.3849832
0.2708845
## day_2 0.8505812 1.0000000 0.8360082 0.7686539 0.5932054 0.4679627
0.3461029
## day_3 0.6565436 0.8360082 1.0000000 0.8618242 0.7075322 0.6016984
0.4406317
## day_4 0.5469131 0.7686539 0.8618242 1.0000000 0.8542087 0.7178904
0.6015019
## day_5 0.4647985 0.5932054 0.7075322 0.8542087 1.0000000 0.8898712
0.7978323
## day_6 0.3849832 0.4679627 0.6016984 0.7178904 0.8898712 1.0000000
0.9043339
## day_7 0.2708845 0.3461029 0.4406317 0.6015019 0.7978323 0.9043339
1.0000000
```

Reshape dataframe

Because of this clustering, we can basically treat this data similarly to the bullying dataset. However, first we need to **re-structure** the dataset to a format that will be interpretable to the linear mixed effect regression (lmer()) function.

At this point, the dataframe contains 7 observations of wound rating from the same participant in one row (one for each day of the week while data was collected). This format is called the **wide format**.

For lmer() to be able to interpret the data correctly, we will have to restructure the dataframe so that each row contains a single observation. This would mean that each participant would have 7 rows instead of 1. Data in the variables ID, distance_window, and location would be duplicated in each of the rows of the same participant, and there would be only a single column for wound rating in this new dataframe. This format of the data is usually referred to as the **long format**.

We can do this using the **gather()** function from the tidyr package. In this function we specify the variable name in which will index the repeated obervations in the **"key =" parameter** (in our case we **will call this "days"**), and the variable name in which we want to gather all the observations made on the same variable in the **"value =" parameter** (in our case we will call this "wound rating"). Finally, we will have to specify the variable names in which **the data is currently located in the wide format**. (By saying: day_1:day_7 we say that the variables are the column names between the day_1 and day_7 columns). Using the arrange() function we sort the data table based on the column "ID". This is not important, but it is easier to see the logic behind a long format file if we look at this sorted dataset.

(As always, we create a new object where we will store the data with the new shape, and leave the raw data unchanged. The new object is called data_wound_long.)

```
data_wound_long = data_wound %>%
  gather(key = days, value = wound_rating,  day_1:day_7) %>%
  arrange(ID)

data_wound_long

## # A tibble: 210 x 5
##    ID    distance_window location   days  wound_rating
##    <fct>           <dbl> <fct>      <chr>        <dbl>
##  1 ID_01            6.18 north_wing day_1        10.3
##  2 ID_01            6.18 north_wing day_2         7.44
##  3 ID_01            6.18 north_wing day_3         5.03
##  4 ID_01            6.18 north_wing day_4         5.37
##  5 ID_01            6.18 north_wing day_5         5.37
##  6 ID_01            6.18 north_wing day_6         6.3
##  7 ID_01            6.18 north_wing day_7         6.52
##  8 ID_02            7.21 north_wing day_1         8.88
##  9 ID_02            7.21 north_wing day_2         5.24
## 10 ID_02            7.21 north_wing day_3         3.96
## # ... with 200 more rows
```

We can also clarify the new dataframe a little bit by **ordering** the rows so that each observation from the same participant follow each other using the **arrange()** function.

Also, notice that our 'days' variable now contains the names of the repeated measures variables from the wide format ('day_1', 'day_2' etc.). We will simplify this by simply using numbers 1-7 here to make them a numerical variable which is easier to deal with statistically. The easiest to do this is by using the mutate() and recode() functions.

```
# change the days variable to a numerical vector
data_wound_long = data_wound_long %>%
  mutate(days = recode(days,
                       "day_1" = 1,
                       "day_2" = 2,
                       "day_3" = 3,
                       "day_4" = 4,
                       "day_5" = 5,
                       "day_6" = 6,
                       "day_7" = 7
                       ))
```

Let's explore how this new dataframe looks like.

```
View(data_wound_long)
```

Building the linear mixed model

Now that we have our dataframe in an approriate formate, we can build our prediction model. The outcome will be wound rating, and the fixed effect predictors will be day after surgery, distance of the bed from the window, and south or north location (these information are stored in the variables days, distance_window, and location).

Since our outcome is clustered within participants, **the random effect predictor will be participant ID**. As in the previous exercise, we will fit two models, one will be a random intercept model, the other a random slope model.

Note that the **random intercept model** means that we suspect that the each participant is different in their overall wound rating (or baseline wound rating), but that the effect of the fixed effect predictors (days, distance from window, and location) is the same for each participant. On the other hand, the **random slope model** not only baseline wound rating will be different across participants, but also that the fixed effect(s) will be different from participant to participant as well.

Note that here we have 3 different fixed effect predictors, so we can specify in the random slope model, which of these predictors we suspect that will be influenced by the random effect (participant). By specifying the random effect term as + (days|ID) we allow for the effect of days to be different across participants (basically saying that the rate of wound healing can be different from person to person), but restrict the model to predict the same effect for the other two fixed predictors: distance_window and location.

We use the Nelder_Mead optimizer in the random slope model, because the default bobyqa optimizer does not achieve converegence in this model.

(We could allow for the random slope of distance_window and location as well by adding + (days|ID) + (distance_window|ID) + (location|ID) if you want the random effects to be uncorrelated, or + (days + distance_window + location|ID) if you want all random effects to be correlated). Now let's stick to a simpler model where we only allow for the random slope of days.

```
# random intercept model

mod_rep_int = lmer(wound_rating ~ days + distance_window + location + (1|ID),
data = data_wound_long)

# random slope model

mod_rep_slope = lmer(wound_rating ~ days + distance_window + location +
(days|ID), data = data_wound_long)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
control$checkConv, :
## Model failed to converge with max|grad| = 0.00253673 (tol = 0.002,
component 1)
```

```
# random slope model with Nelder_Mead optimizer to achieve convergence
mod_rep_slope_opt = lmer(wound_rating ~ days + distance_window + location +
(days|ID), control = lmerControl(optimizer = "Nelder_Mead"), data =
data_wound_long)
```

Comparing models

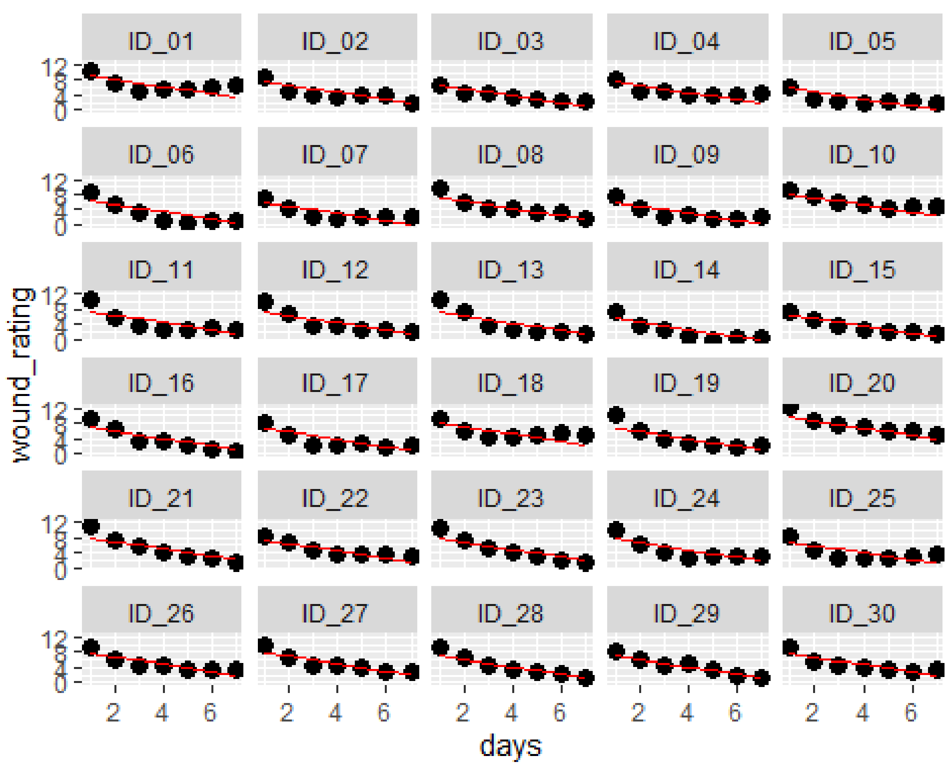
Now let's compare the model predictions of the different random effect models to see which one fits the data better.

First, let's **visualize the predictions**. For this we will have to save the predicted values into new variables, then, we can visualize the predicted values and the actual observations for each participant separately for both the random intercept and the random slope model.

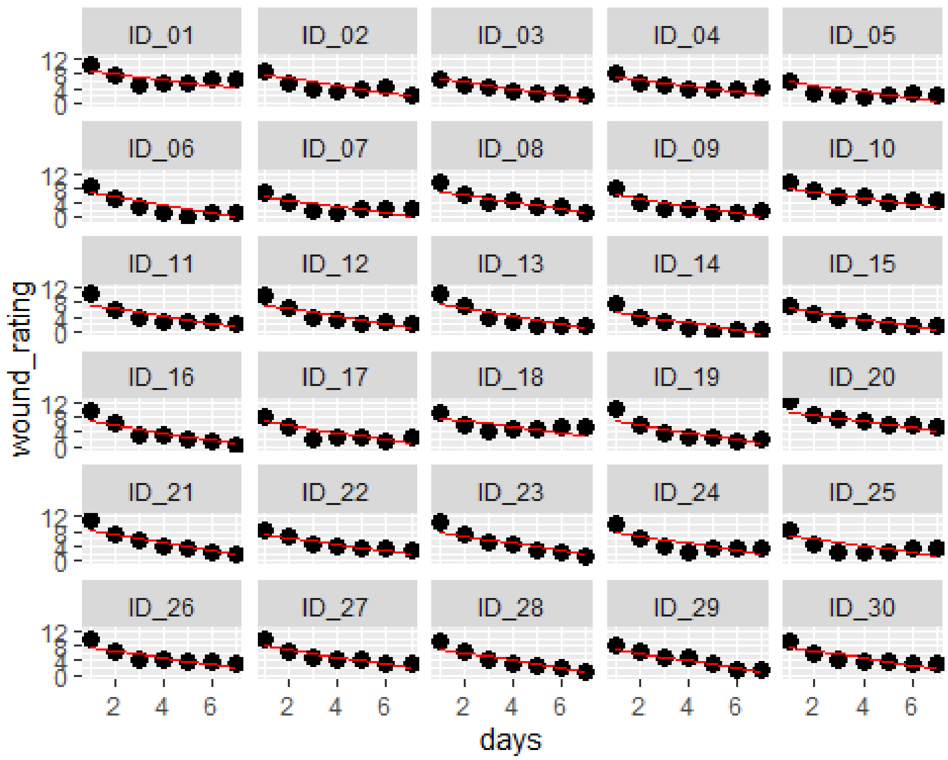
(We create a new copy of the data object so that our long format data can remain unharmed.)

```
data_wound_long_withpreds = data_wound_long
data_wound_long_withpreds$pred_int = predict(mod_rep_int)
data_wound_long_withpreds$pred_slope = predict(mod_rep_slope_opt)

# random intercept model
ggplot(data_wound_long_withpreds, aes(y = wound_rating, x = days, group =
ID))+
  geom_point(size = 3)+
  geom_line(color='red', aes(y=pred_int, x=days))+
  facet_wrap( ~ ID, ncol = 5)
```



```r
# random slope and intercept model
ggplot(data_wound_long_withpreds, aes(y = wound_rating, x = days, group =
ID))+
  geom_point(size = 3)+
  geom_line(color='red', aes(y=pred_slope, x=days))+
  facet_wrap( ~ ID, ncol = 5)
```



The difference between the predictions of the two models is unremarkable.

Furthermore, we can compare the cAIC of the two models and use the likelihood ratio test with the anova() function to get further information about the model fit of the two models in comparison to wach other.

```r
cAIC(mod_rep_int)$caic

## [1] 757.0182

cAIC(mod_rep_slope_opt)$caic

## [1] 757.3522

anova(mod_rep_int, mod_rep_slope_opt)

## refitting model(s) with ML (instead of REML)

## Data: data_wound_long
## Models:
## mod_rep_int: wound_rating ~ days + distance_window + location + (1 | ID)
```

```
## mod_rep_slope_opt: wound_rating ~ days + distance_window + location +
(days | ID)
##                   npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
## mod_rep_int          6 773.40 793.49 -380.70   761.40
## mod_rep_slope_opt    8 774.82 801.60 -379.41   758.82 2.583  2     0.2749
```

None of these methods indicate a significant difference between the prediction efficiency of the models. So in this particular sample thre is not too much benefit for assuming a different slope of days for each participant. But this does not necesseraly mean that there is no point of using it in another sample. Previous studies and theory needs to be evaluated as well.

For now, without any prior knowledge from the literature, we can continue using the random intercept model.

Adding the quadratic term of days to the model

While exploring the plots we might notice that there is a non-linear relationship between days and wound rating. It seems that wounds improve fast in the first few days, and the healing is slower in the days after that.

Let's add the quadratic term of days to the model random intercept model tp account for this non-linear relationship.

```r
mod_rep_int_quad = lmer(wound_rating ~ days + I(days^2) + distance_window +
location + (1|ID), data = data_wound_long)
```

And add the predictions to the new dataframe containing the other predicted values as well.

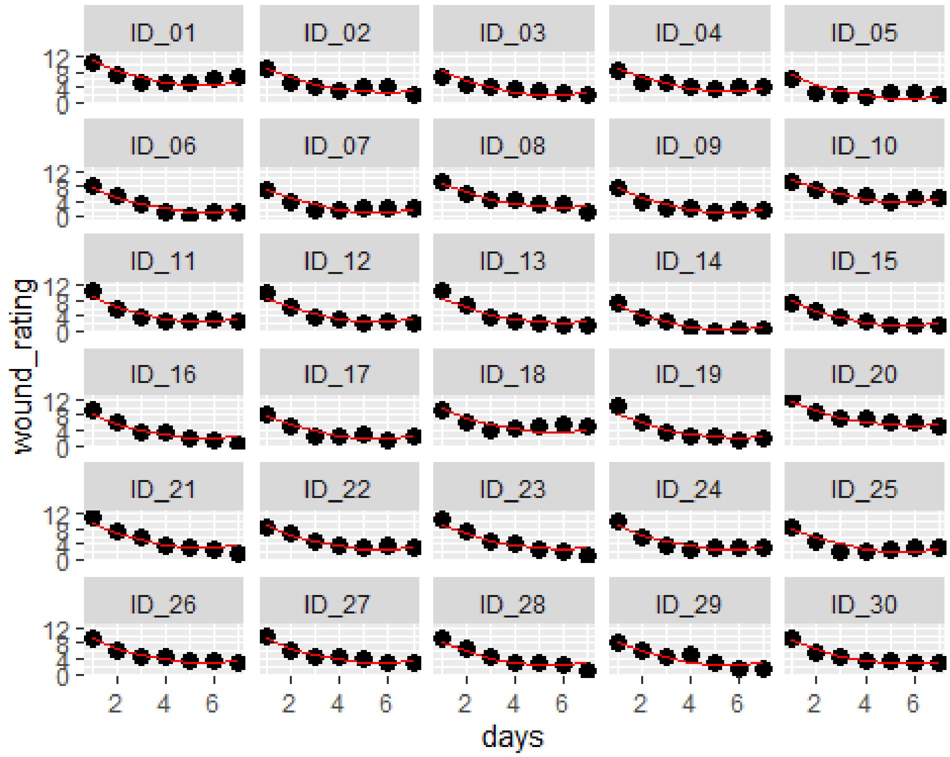
```r
data_wound_long_withpreds$pred_int_quad = predict(mod_rep_int_quad)
```

Now we can compare the model fit of the random intercept model containing the quadratic effect of days with the random intercept model without it. As usual, we use visualization, cAIC and the likelihood ratio test.

```r
data_wound_long_withpreds$pred_int_quad = predict(mod_rep_int_quad)

plot_quad = ggplot(data_wound_long_withpreds, aes(y = wound_rating, x = days,
group = ID))+
  geom_point(size = 3)+
  geom_line(color='red', aes(y=pred_int_quad, x=days))+
  facet_wrap( ~ ID, ncol = 5)

plot_quad
```




```
cAIC(mod_rep_int)$caic

## [1] 757.0182

cAIC(mod_rep_int_quad)$caic

## [1] 583.2994

anova(mod_rep_int, mod_rep_int_quad)

## refitting model(s) with ML (instead of REML)

## Data: data_wound_long
## Models:
## mod_rep_int: wound_rating ~ days + distance_window + location + (1 | ID)
## mod_rep_int_quad: wound_rating ~ days + I(days^2) + distance_window +
location +
## mod_rep_int_quad:     (1 | ID)
##                  npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod_rep_int         6 773.40 793.49 -380.70   761.40
## mod_rep_int_quad    7 621.08 644.51 -303.54   607.08 154.32  1  < 2.2e-16
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results indicate that a model taking into account the nonlinear relationship of days and wound rating produces a significantly better fit to the observed data than a model only allowing for a linear trend of days and wound healing.

The fit seems reasonable, so we stop here and decide that this will be our final model.

Since we entered days's quadratic term into the model, we can expect problems with multicollinearity. As seen in the exercise on model diagnostics, we can avoid this problem by centering the variable days, this way removing the correlation of days and days^2.

Let's do this now and refit our model with the centered days and its quadratic term as predictors.

```
data_wound_long = data_wound_long %>%
  mutate(days_centered = days - mean(days))

mod_rep_int_quad = lmer(wound_rating ~ days_centered + I(days_centered^2) +
distance_window + location + (1|ID), data = data_wound_long)
```

Now we can request the reportable results the same way we did in the previous exercise.

```
# Marginal R squared
r2beta(mod_rep_int_quad, method = "nsj", data = data_wound_long)

##                Effect   Rsq upper.CL lower.CL
## 1               Model 0.763    0.805    0.717
## 2       days_centered 0.700    0.752    0.643
## 3 I(days_centered^2) 0.377    0.470    0.284
## 4     distance_window 0.130    0.220    0.058
## 5 locationsouth_wing 0.103    0.189    0.039

# marginal and conditional R squared values
r.squaredGLMM(mod_rep_int_quad)

## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help
page.

##            R2m       R2c
## [1,] 0.7626648 0.8756216

# Conditional AIC
cAIC(mod_rep_int_quad)$caic

## [1] 583.2994

# Model coefficients
summary(mod_rep_int_quad)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: wound_rating ~ days_centered + I(days_centered^2) +
distance_window +
##     location + (1 | ID)
##    Data: data_wound_long
##
## REML criterion at convergence: 625.7
```

```
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.34290 -0.61819  0.02384  0.61744  2.40352
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ID       (Intercept) 0.7380   0.8591
##  Residual             0.8126   0.9015
## Number of obs: 210, groups:  ID, 30
##
## Fixed effects:
##                     Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)          4.95676    0.50887  28.10724   9.741 1.65e-10 ***
## days_centered       -0.94826    0.03110 178.00000 -30.487  < 2e-16 ***
## I(days_centered^2)   0.27908    0.01796 178.00000  15.541  < 2e-16 ***
## distance_window     -0.09016    0.03174  27.00000  -2.840  0.00846 **
## locationsouth_wing  -0.84297    0.33787  27.00000  -2.495  0.01901 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) dys_cn I(_^2) dstnc_
## days_centrd  0.000
## I(dys_cn^2) -0.141  0.000
## distnc_wndw -0.872  0.000  0.000
## lctnsth_wng -0.288  0.000  0.000 -0.049
```

```
# Confidence intervals for the coefficients
confint(mod_rep_int_quad)
```

```
## Computing profile confidence intervals ...
```

```
##                         2.5 %      97.5 %
## .sig01              0.6048206  1.10514198
## .sigma              0.8112449  0.99766032
## (Intercept)         3.9797744  5.93373628
## days_centered      -1.0092087 -0.88731515
## I(days_centered^2)  0.2438917  0.31426699
## distance_window    -0.1511234 -0.02919934
## locationsouth_wing -1.4918557 -0.19408605
```

```
# standardized Betas
stdCoef.merMod(mod_rep_int_quad)
```

```
##                       stdcoef      stdse
## (Intercept)         0.0000000 0.00000000
## days_centered      -0.7486918 0.02455740
## I(days_centered^2)  0.3816481 0.02455740
## distance_window    -0.1894277 0.06669033
## locationsouth_wing -0.1663901 0.06669033
```

As always, you will need to run model diagnostics before reporting your final results. The next exercise will conver this topic.

____________Practice_____________

Load the surgical pain dataset.

This dataset contains information about pain after surgery, and a number of variables which may be connected to surgical pain.

Variables:

- ID: participant ID
- pain1, pain2, pain3, pain4: In this dataset pain was measured on four consecutive days after surgery using a 0-10 visual analog scale.
- sex: sex reported by the participant
- STAI_trait: score on the State Trait Anxiety Inventroy - t form, which reflects trait anxiety
- pain_cat: pain catastrophising
- cortisol_serum; cortisol_saliva: cortisol is a stress hormone. This was measured on the day of surgery (after surgery) at the same time when the first pain rating was taken. Cortisol was measured from blood and also from the saliva.
- mindfulness: trait mindfulness of the participant measured by a mindfulness questionnaire
- weight: weight in kiligramms
- IQ: IQ score on an IQ test one week before surgery
- household_income: household income in USD

Practice tasks:

1. Load the dataset (a .csv file) from this link: "https://tinyurl.com/data-pain1".
2. Convert the data to long format (you can use the gather() function or the melt() function to do this) so that each observation/repeated measure is in a different row.
3. Build a linear mixed model to predict as much of the variance in postoperative pain as possible. (You can use any variable as a fix predictor that makes sense for you to predict postoperative pain.) Since data is clustered within participants, include the random effect of participant ID in the model.
4. Experiment with both random intercept and random slope models, and compare them using cAIC.
5. Build a random intercept and a random slope model where the only fixed predictor is time (days after sugery). Visualize the regression lines prodcued by these two models for each participant separately, and compare their fit the the observations. Is there a significant gain in allowing for random slope of time?
6. Answer the same question using cAIC.
7. What is the marginal R^2 for the random intercept model? Based on the confidence interval, is this model better than the null model at predicting pain?