Kevert modellek - Ismetelt mereses elemzesek

Zoltan Kekecs

03 december 2019

Contents

1 Absztrakt

Ez a gyakorlat az ismetelt mereses elemzessekkel foglalkozik. Amikor az elemzesunkben ugyan attol a vizsgalati szemelytol tobb adat is szerepel ugyan abbol a valtozobol, ismetelt mereses elemzest vegzunk (pl. a szemely kezeles elotti es utani depresszio szintje).

2 Adatmenedzsment es leiro statisztikak

2.1 Package-ek betoltese

A kovetkezo package-ekre lesz szukseg a gyakorlathoz:

```
library(psych)  # for describe
library(tidyverse)  # for tidy code and ggplot
library(cAIC4)  # for cAIC
library(r2glmm)  # for r2beta
```

2.2 Sajat funkcio

Ezzel a funkcioval kinyerhetjuk a standardizalt Beta egyutthatot a kevert modellekbol. Ez a funkcio innen lett atemelve: 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))
}
```

2.3 Sebgyogyulas adat betoltese

A gyakorlat soran a sebgyogyulas adatbazissal fogunk dolgozni. Ez egy szimulat adatbazis, ami a mutet soran ejtett bemetszesek gyogyulasat vizsgaljuk annak fuggvenyeben hogy a paciensek agya milyen kozel van az ablakhoz, es hogy mennyi napfeny eri oket a felepules idoszak alatt. Mondjuk hogy az az elmeletunk hogy a korhazi betegeknek szukseguk van a kulvilaggal valo kapcsolatra ahhoz hogy gyorsan felepuljenek. Egy ablak ami a szabadba nyilik megteremtheti ezt a kapcsolatot a kulvilaggal, ezert a kutatasunk azt vizsgalja, hogy befolyasolja-e a sebgyogyulas merteket az, hogy a szemelynek milyen kozel van az agya a legkozelebbi ablakhoz. Az elmelet egy valtozata azt allitja, hogy az ablak nem csak a kulvilaggal valo szorosabb kapcsolat megteremtesen keresztul vezet gyorsabb gyogyulashoz, hanem azon keresztul is hogy tobb napfenyt enged a szobaba, es az elmelet szerint a napfeny is jotekony hatassal van a gyogyulasra.

Valtozok az adatbazisban:

- ID: azonosito kod
- day_1, day_2, . . . , day_7: A mutet utani 1-7. napon egy orvos megvizsgalta a pedeg bemetszesi sebeit, es ertekelte azokat egy standardizalt seb-allapot ertekkel. Minel nagyobb ez az ertek, annal nagyobb vagy rosszabb allapotu a seb (pl. gyulladt). Minden szemelynek mind a 7 naphoz kulon seb-allapot ertek tartozik.
- distance_window: A szemely agyahoz legkozelebbi ablak tavolsaga az agytol meterben.
- location: A korhazi szarny, ahol a paciens agya van. Ket allasu faktor valtozo: szintjei “north wing” es “south wing” (a “south wing”-ben tobb napfeny eri a pacienseket, ez a valtozo azert fontos).

```
data_wound = read_csv("https://raw.githubusercontent.com/kekecsz/PSYP13_Data_analysis_class-2018/master/
```

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

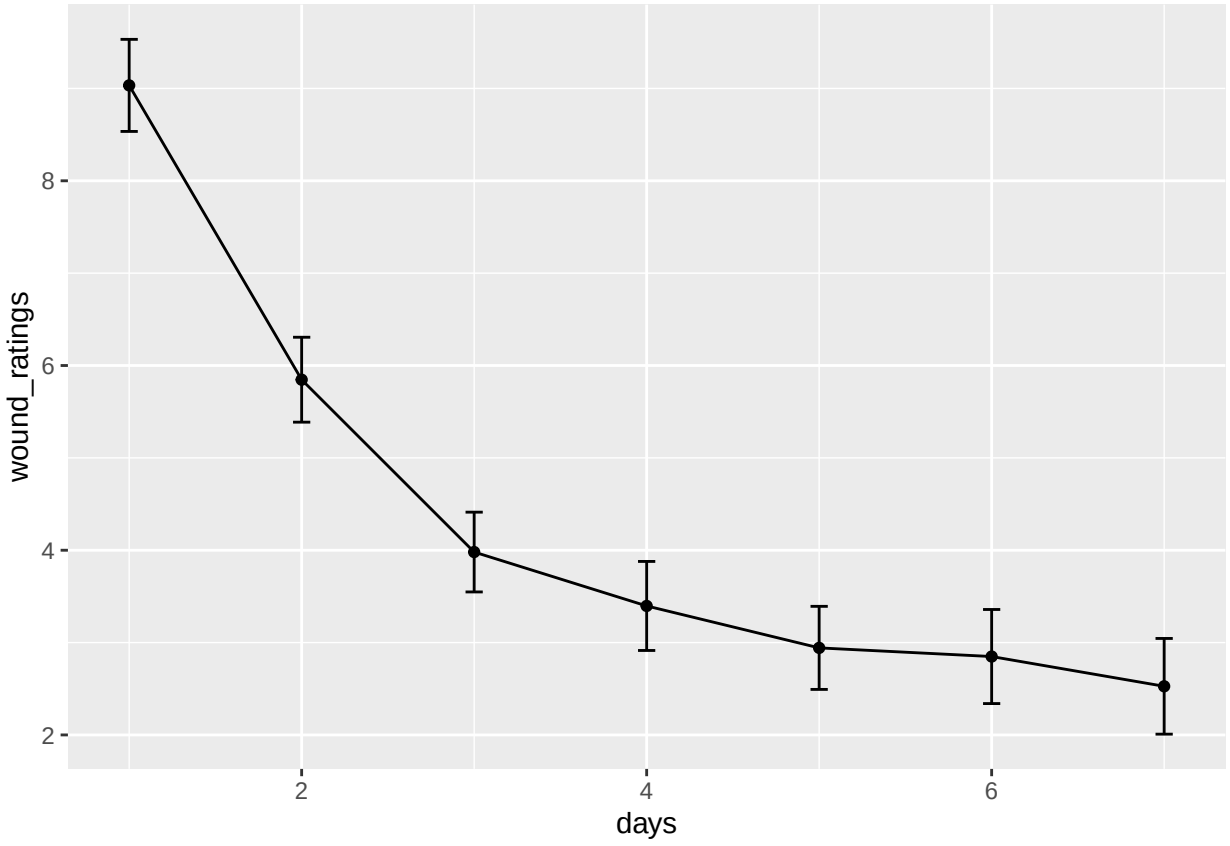
2.4 Adatellenorzes

Vizsgaljuk meg az adattablat a View(), describe(), es table() fukciok segitsegevel. Fontos, hogy az adattablaban jelenleg minden adata mit egy adott szemelytol gyujtottek egy sorban talalhato. Vizualizalhatjuk is az adatokat.

```r
View(data_wound)

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

Az alabbi abra elkeszitesehez eloszor a day1-day7 valtozokban kulon-kulon kiszamoljuk az atlagokat es a standard hibat, majd a standard hibat megszorozva 1.96-al megkapjuk a konfidencia intervallumot. Vegul mindezt egy uj adat objektumba tesszuk, es a geom_errorbar, geom_point, es geom_line segitsegevel viualizaljuk. Lathato hogy a seb allapot ertek egyre csokken ahogy telnek a napok.

```r
# 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 = 0.1) + geom_point() + geom_line()
```



3 Ismételt mérések eredményinek vizsgálata kevert lineáris modellekkel

3.1 Klaszteres szerkezet keresése az adatokban

Vizsgáljuk meg a továbbiakban a sebgyógyulásra vonatkozó ismételt mérések eredményeit! Ehhez eloször mentsük el az adatokhoz tartozó változóneveket egy repeated_variables elnevezésu objektumba, hogy később könnyen hivatkozhassunk rájuk az álltalunk írt függvényekben! A változók közötti korrelációt a cor() függvénnyel tudjuk megvizsgálni. Figyeljük meg, hogy az ismételt mérések adatponjai között eros korreláció fedezheto fel, azaz az egyes seb-állapot értekekre vonatkozó megfigyelések nem függetlenek egymástól. Ez várható is, hiszen a seb-állapot érték, az eredeti bemetszés mérete és a seb gyógyulásának üteme mind függenek a vizsgált betegtol. Adataink tehát klaszteres szerkezetet mutatnak, hasonlóan a korábbi példánkhoz. Azonban mígo ott osztály szerinti klaszterek fordultak elo, itt most a klaszterek résztvevo szerintiek.

```
# correlation of repeated variables

cor(data_wound[, repeated_variables])
```

```
##           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
```

3.2 Dataframe átrendezése

A klaszteres szerkezetbol kifolyólag hasonlóan kezelhetjük adatainkat mint ahogy azt a bántalmazással kapcsolatos adatsornál tettük. Ehhez azonban eloször át kell rendeznünk az adatainkat, hogy használhassuk a lineáris kevert hatás regressziós (lmer()) függvényt.

Jelenleg a dataframe-ünk minden sora egy adott pácienshez tartozó seb-állapot értékre vonatkozó 7 (vagyis az adatgyüjtés alatt napi egy) megfigyelésbol áll. Erre az elrendezésre **wide format**-ként (széles formátum) is szokás hivatkozni.

Az lmer() függvény megelelo muködéséhez a bemenet minden sorához csak egyetlen megfigyelés tartozhat. Jelen esetben ez azt jelentené, hogy az egyes résztvevokhöz tartozó sorok száma 1 helyett 7 kell legyen. Így az ID, distance_window, és location változók az adott pácienshez tartozó sorokban egyeznének, és csak az egyes seb-állapot értekek különböznének, melyekhez minden sorban mindössze egyetlen oszlop tartozna így. Erre az elrendezésre álltalában **long format**-ként (hosszú formátum) szokás hivatkozni.

A fenti átalakítás elvégzésének egy egyszeru módja, ha a gather() függvényt alkalmazzuk a tidyr csomagból. Ennek használatához eloször meg kell határoznunk az ismételt megfigyelések indexét tartalmazó változó nevét, vagyis itt azt hogy melyik nap végezték az adott megfigyelést. Ez nálunk a "days" változót jelenti. Az indexeket tartalmazó változón kívül a vizsgált mennyiséget tartalmazó változót is meg kell határoznunk. Ez nálunk a "wound_rating", vagyis a seb-állapot érték. Végül meg kell még határoznunk azt is, hogy a jelenleg használt széles formátumban mely nevek jelölik azokat a változókat amellyekben az adatainkat tároljuk. A day_1:day_7 kifejezés a day_1 és day_7 közötti oszlopok neveit jelöli. Az arrange() függvény használatával az adatok rendezhetoek a hozzájuk tartozó azonosító ("ID") alapján. Bár az adott feladat elvégzéséhez nem sükséges rendezni az adatainkat, de mégis segít a hosszú formátum átláthatóbbá tételében.

Az eredeti adatokat változatlanul hagyva most is új objektumot hozunk létre adatainknak, a már megszokott módon. Az új objektum neve data_wound_long lesz.

```
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
```

Tovább növelhetjük az adataink átláthatóságát, ha az adott beteghez tartozó megfigyelések egymás után következnek.

A fontos megjegyezni, hogy a 'days' változó jelenleg a széles formátumból származó változó neveket tartalmazza ('day_1', 'day_2' stb.). Az egyszerubb kezelhetoség érdekében ezeket egyszeruen az egyes napokat jelölo számokra (1-7) cseréljük. Ezt legkönyebben a mutate() és recode() fügvényekkel valósíthatjuk meg.

```
# 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))
```

Tekintsük most meg, hogyan néz ki az új dataframe-ünk!

```
View(data_wound_long)
```

3.3 Kevert lineáris modell kialakítása

Most, hogy megfelelo alakba hoztuk adatainkat, eloállíthatjuk az elorejelzésekhez szükséges modellt. Ezzel a modellel a mutét utáni nap (days), az ablaktól való távolság ('distance_window') és északi vagy déli elhelyezés ('location') alapján megbecsülheto lesz a seb-állapot érték ('wound_rating').

Mivel az elorejelzésünk kimenete a résztvevok szerinti klaszteres szerkezetet mutat, ezért a véletlen hatás prediktora a résztvevo azonosítója ('ID') lesz. Az korábbi gyakorlathoz hasonlóan, most is két modellt fogunk illeszteni, a random intercept és a random slope modelleket.

Említést érdemel, hogy a **random intercept model** esetében azt feltételezzük, hogy minden résztvevo eltér a teljes vagy baseline seb-állapot értékeit tekintve, de a fix hatás elorejelzok ('days', 'distance_window', és 'location') azonosak az egyes résztvevok esetében. Ezzel szemben, a **random slope model** esetében nem csak a baseline seb-állapot érték, de a fix hatás elorejelzok is résztvevonként változóak.

Mivel 3 különbözo fix hatás elorejelzo is rendelkezésünkre áll, ezért alkalmazhatjuk a random slope modellt, ami az elorejelzok mellett a résztvevokbol származó véletlen hatástól is függeni fog. A véletlen hatás kifejezését (random effect term) + (days|ID) alakban definiálva elérhetjük, hogy az ido múlása más mértékben hasson az egyes résztvevokre.

További lehetoségként felmerül, hogyha a másik két elorejelzo szerinti véletlen meredekséget is szeretnénk bevezetni a modellbe, akkor azt a + (days|ID) + (distance_window|ID) + (location|ID) kifejezéssel érhetjük el, ha azt szeretnénk hogy ne legyen köztük korreláció, és a + (days + distance_window + location|ID) kifejezéssel, ha azt szeretnénk hogy korreláljanak. Most maradjunk egyelore a korábban leírt, egyszerubb + (days|ID) modellnél.

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

3.4 Az eltéro modellek összehasonlítása

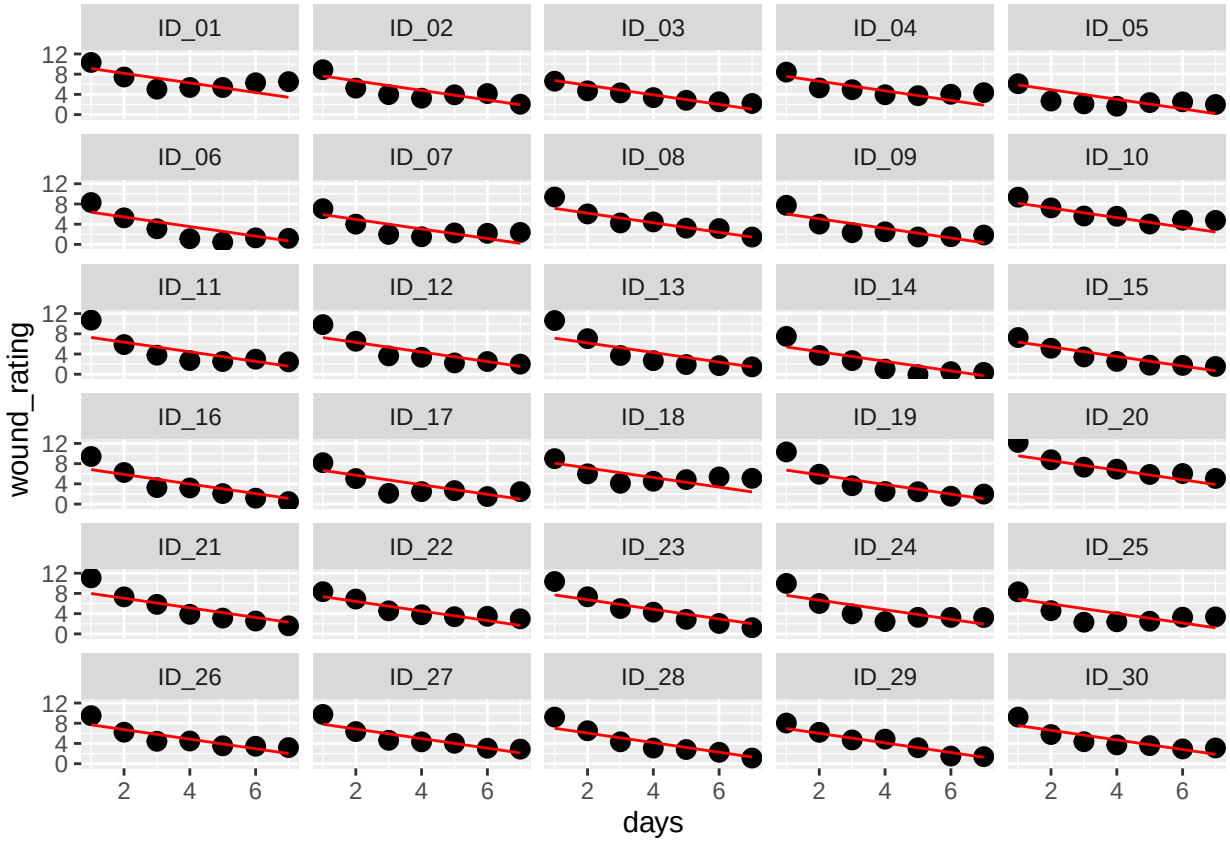
Hasonlítsuk most össze a különbozo modellek alapján alkotott elorejelzéseket!

A könnyebb összehasonlíthatóság kedvéért, vizualizáljuk adatainkat! Ehhez eloször tároljuk el predikciónk eredményeit egy új változóban, majd ábrázolhatjuk az egyes elorejelzett értékeket a valódi értékek függvényében, az egyes (random intercept és random slope) modellekre vonatkozó külön-külön ábrákon.

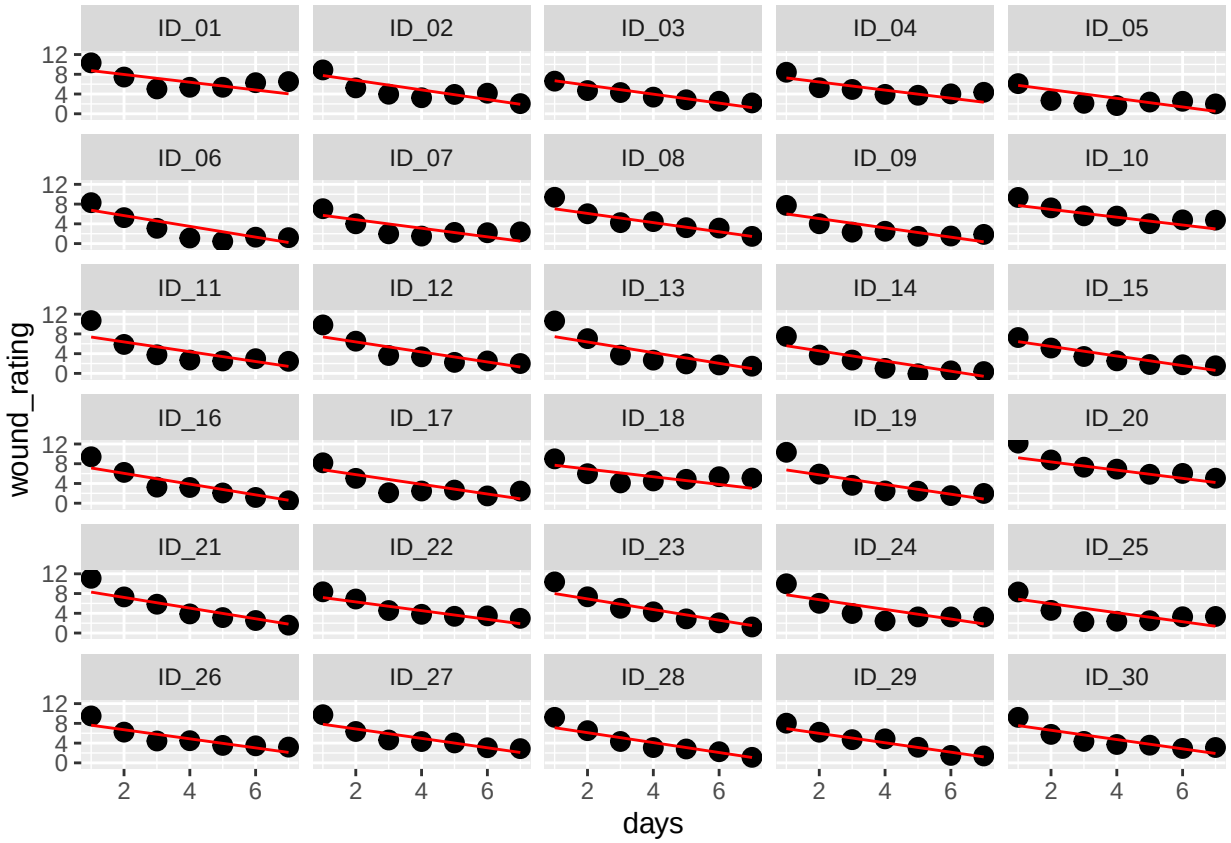
(Az alábbiakban létrehoztunk egy másolatot az adatokat tartalmazó objektumról, hogy az eredeti adatok változatlanul megmaradhassanak.)

```
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)

# 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)
```




```
# 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)
```



Látható, hogy az eltéro modellek alapján kapott eredmények között nincs számottevo eltérés.

A cAIC() és anova() függvények segítségével további megállapításokat tehetünk az egyes modellek illeszkedéséről, ami egy újabb lehetséges szempont lehet a modellek összehasonlításánál.

```
cAIC(mod_rep_int)$caic
```

```
## [1] 757.0182
```

```
cAIC(mod_rep_slope)$caic
```

```
## [1] 757.3564
```

```
anova(mod_rep_int, mod_rep_slope)
```

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

A fenti módszerek egyikével se találunk jelentos eltérést a két modell használata között, így a jelenlegi minta esetén semmiféle elonnyel sem jár a random slope módszer. Ez persze magában még nem elegendo ahhoz, hogy feltételezhessük, hogy más mintánál is hasonló lenne a helyzet. Látható tehát, hogy az adatelemzés során fontos tisztában lennünk a korábbi kutatások eredményeivel, és a vizsgált kérdéskörre vonatkozó elméletekkel.

Jelenleg,-híjján bármiféle korábbi ismeretnek,- folytassuk a random intercept modell használatával.

3.5 A modell kiegészítése a napokból származó négyzetes járulékkal

Az egyes ábrákat vizsgálva megfigyelhetjük, hogy a napok és a seb-állapot értékek közötti összefüggés nem lineáris. A sebek látszólag gyorsabban gyógyulnak az elso néhány napban, mint később.

A nem lineáris viselkedés figyelembevétele érdekében, adjuk hozzá a napokból származó négyzetes járulékot a modellünkhöz!

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

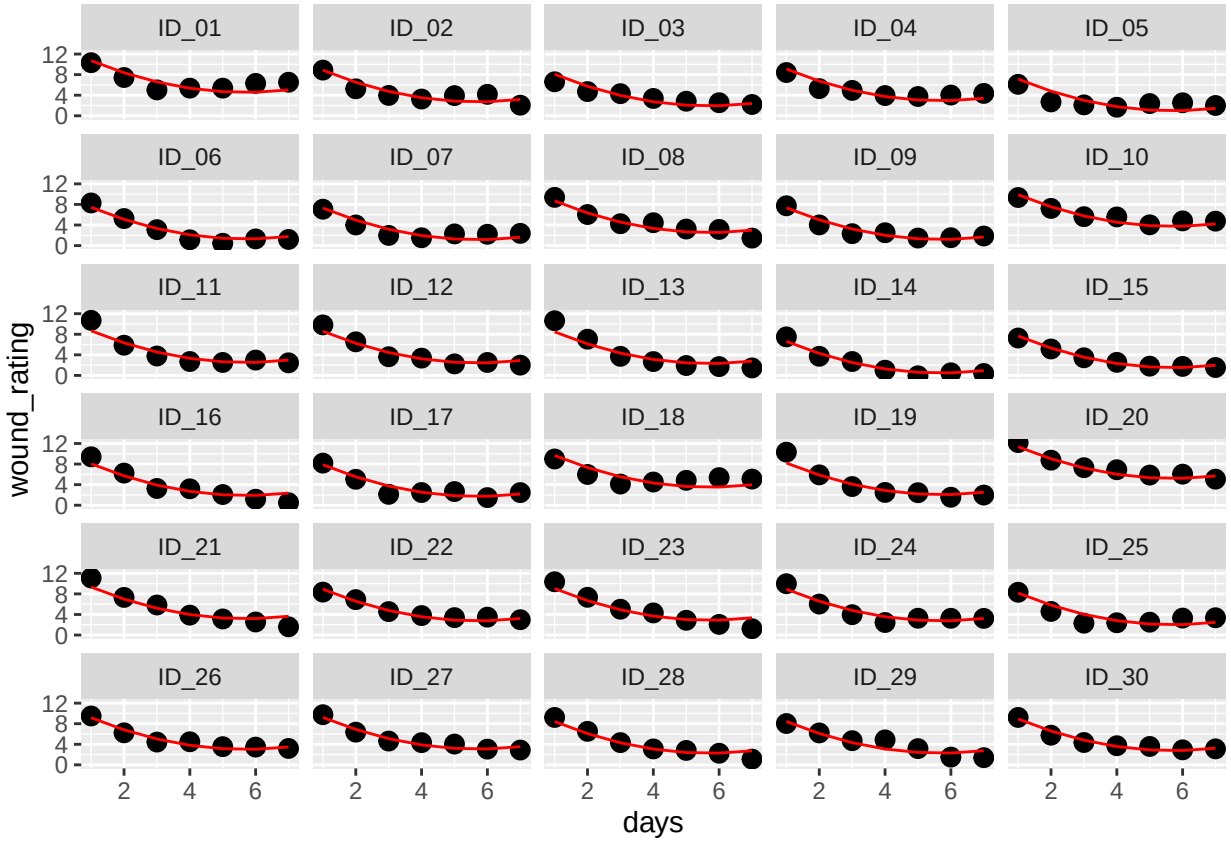
Mentsük elorejelzéseinket egy új dataframe-be, amely a korábbi elorejelzéseket is tartalmazza!

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

Most pedig hasonlítsuk össze a négyzetes tagokkal bovített, és az eredeti modellt a modellek összehasonlításánál korábban tárgyalt módon!

```
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)
##                  Df    AIC    BIC  logLik deviance  Chisq Chi 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
```

Az összehasonlítás alapján úgy tunik, hogy a négyzetes tagokat is megengedo modell elorejelzései lényegesen pontosabbak mint a csak lineáris tagokat használóé.

Mivel modellünk látszólag jól illeszkedik az adatokra, nem bovítjük tovább tagokkal azt.

A négzetes elemek felhasználásából következoen várható, hogy problémák fognak jelentkezni a collinearitás tekintetében. A 'days' változó centrálásával ez a probléma a model diagnosztika c. gyakorlatban tárgyalt módon kiküszöbölheto, hiszen megszünteti a 'days' és 'days^2' közötti korrelációt.

Végezzük el a centrálást, és illesszük újra modellünket az így kapott prediktorokat használva.

```
data_wound_long_centered_days = data_wound_long
data_wound_long_centered_days$days_centered = data_wound_long_centered_days$days -
    mean(data_wound_long_centered_days$days)


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

Az elozo gyakorlathoz hasonlóan kérjük eredményeink bemutatását!

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

```
##                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
```

```
# Conditional AIC
cAIC(mod_rep_int_quad)$caic
```

```
## [1] 583.2994
```

```r
# Model coefficients
summary(mod_rep_int_quad)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: wound_rating ~ days_centered + I(days_centered^2) + distance_window +
##     location + (1 | ID)
##    Data: data_wound_long_centered_days
##
## 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 t value
## (Intercept)          4.95676    0.50887   9.741
## days_centered       -0.94826    0.03110 -30.487
## I(days_centered^2)   0.27908    0.01796  15.541
## distance_window     -0.09016    0.03174  -2.840
## locationsouth_wing -0.84297    0.33787  -2.495
##
## 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
```

```r
# 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
```

```r
# 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
```

Mielott elfogadnánk eredményeinket véglegesnek, mindig futassunk modell diagnosztikát is. Ennek módjára a következo gyakorlatban fogunk kitérni.

_____Gyakorlás_____

Olvassuk be a mutéti fájdalom adatsort!

Ez az adatsor a mutét utáni fájdalom mérékérol, és az ezzel feltételezhetoen összefüggo néhány egyéb értékekrol tartallmaz információkat.

Változóink:

- ID: résztvevo azonosítója
- pain1, pain2, pain3, pain4: A használt adatsorban a fájdalom a mutét utáni négy egymást követo napon volt mérve egy 0tól-10ig terjedo folytonos vizuális skálán.
- sex: a résztvevo bejelentett neme
- STAI_trait: A résztvevo State Trait Anxiety Inventroy-n elért pontszáma
- pain_cat: fájdalom katasztrofizálása
- cortisol_serum; cortisol_saliva: A kortizol egy a stress hatására eloállított hormon. A kortizol szintet vérbol és nyálból, közvetlenül a mutét után határozták meg.
- mindfulness: A Mindfulness kérdoív alapján a résztvevore jellemzo Mindfulness érték
- weight: résztvevo tömege kg-ban.
- IQ: Résztvevo IQ-ja a mutét elott egy héttel felvett IQ teszt alapján
- household_income: résztvevo háztartásának bevétele USD-ben

Gyakorló feladatok:

1. Olvassuk be az adatokat (egy .csv kiterjesztésu file-ból). Az adatokat az alábbi linkrol tölthetjük le: “https://tinyurl.com/data-pain1”.
2. Alakítsuk adatainkat hosszú formátumúvá (célszeru a gather() fvagy a melt() függvények valamelyikét használni erre a célra), hogy az egyes megfigyelések külön sorba kerüljenek.
3. Állítsunk össze egy kevert lineáris modellt, hogy amivel képesek vagyunk a mutét utáni fájdalom varianciájának leheto legszélesebb köru lefedésére. (A mutét utáni fájdalom meghatározásához tetszoleges fix elorejelzot választhatunk, amennyiben annak feltehetoen van valami köze a fájdalom mértékéhez.) Mivel adataink a résztvevok szerinti klaszteres szerkezetet mutatnak, modellünkben vegyük figyelembe a résztvevok azonosítója szerinti véletlen hatást.
4. Kísérletezzünk mind a random intercept, mind pedig a random slope modellekkel, majd hasonlítsuk össze oket a cAIC() függvény felhasználásával.
5. Alkossunk olyan random intercept és random slope modelleket, ahol az egyetlen prediktor az ido (mutét óta eltellt napok száma). Vizualizáljuk a modelljeink alapján kapott regressziós vonalakat, minden résztvevore külön-külön, és hasonlítsuk össze hogyan illeszkednek a megfigyeléseinkre. Van bármi elonye ha az idot külön változó hatásként vizsgáljuk a random slope modellben?
6. Hasonlítsuk össze az 5. pont modelljeit a cAIC() függvény eredményei alapján is!
7. Mi a határ R^2 érték a random intercept modell esetében? Pontosabb-e a konfidencia intervallum alapján a fájdalom elorejelzésében ez a modell, mint a null modell?
