

Kevert modellek és ismételt mérési modellek

Zoltan Kekecs

09 November, 2021

Contents

1	Absztrakt	2
2	Adatmenedzsment és leíró statisztikák	2
2.1	Package-ek betöltése	2
2.2	Saját funkció	2
2.3	A Bully-zás adatbázis betöltése	2
2.4	Adatellenőrzés és adattisztítás	3
3	A kevert modellek alapfogalmai	3
3.1	Clustering (csoportosulás) feltarasa	3
3.2	Kevert modellek	5
3.3	A hatások (prediktorok) két típusa	5
3.4	Kevert modellek felépítése az R-ben	7
3.5	Melyik modell reprezentálja legjobban a valóságot?	8
3.6	Mit kell közölni az elemzésről	12
4	Ismételt mérési modellek	14
5	Adatmenedzsment	14
5.1	Szógyógyulás adat betöltése	14
5.2	Adatellenőrzés és leíró statisztikák	15
6	Ismételt mérések eredményének vizsgálata kevert lineáris modellekkel	16
6.1	Klaszteres szerkezet keresése az adatokban	16
6.2	Az adattábla átformázása szélesből hosszú formátumba	17
6.3	Kevert lineáris modell kialakítása	18
6.4	Az eltérő modellek összehasonlítása	19
6.5	A modell kiegészítése a kvadrátikus hatás hozzáadásával	22

1 Absztrakt

Az eddig tanult lineáris regressziós modellek a csoportokba rendezett adatokat úgy kezelik, hogy prediktorként bevonják azokat a modellbe. Ez remekül működik ha kevés csoport van (a csoportosító változónak kevés szintje van) és minden csoportot van módunk megfigyelni. Pl. kísérleti vs. kontroll csoport. De ezek a modellek nem jól működnek olyan esetekben ha az adataink csoportokba/klaszterekbe rendeződnek egy olyan változó mentén aminek a kutatásunk célpopulációjában **sok csoportszintjét különíthetjük el, de a mi kutatásunkban ennél kevesebb figyelhető meg**. Ilyen eset például ha a vizsgálati személyeink különböző iskolákból érkeznek, és elképzelhető hogy az iskolának hatása van a kimeneti változóra, de néhány iskolából vannak adataink és nem tudunk az ország összes iskolájából mintát venni, így sok lehetséges iskola hiányzik az adatok közül. Ilyen esetekben **kevert modelleket** célszerű használni.

Ebben a gyakorlatban megismerheted a kevert modellekkel kapcsolatos alapfogalmakat, valamint hogy hogyan lehet őket felépíteni.

2 Adatmenedzsment és leíró statisztikák

2.1 Package-ek betöltése

Ebben a gyakorlatban a következő package-ekre lesz szükség:

```
library(psych) # for describe\t
library(tidyverse) # for tidy code and ggplot\t
library(cAIC4) # for cAIC\t
library(r2glmm) # for r2beta\t
library(lme4) # for lmer
library(lmerTest) # to get singificance test in lmer
library(MuMIn) # for r.squaredGLMM
```

2.2 Saját funkció

Ezzel a funkcióval kinyerhetjük a standardizált Beta együtthatót a kevert modellekből. Ez a funkció innen lett átveve: <https://stackoverflow.com/questions/25142901/standardized-coefficients-for-lmer-model>

```
stdCoef.lmerMod <- 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 A Bully-zás adatbázis betöltése

Ebben a gyakorlatban az általános **iskolai bully-zás**ról (magyarul talán “iskolai zaklatás”) teszünk fel kutatási kérdéseket. Ez egy szimulált adatbázis. Egy olyan kutatás adatait szimulálja, melyben az érdekel minket, hogy a **testsúly** hogyan befolyásolja a gyerekek **serulekenységet a bully-zással szemben**. A kutatók azt feltételezik hogy a testsúly összefügg az elvett szendvicsek számával.

Változók:

- **sandwich_taken** - A bullyzással kapcsolatos serulekenység mérszama. A kutatásban megkerdezték a vizsgáltai személyeket (általános iskolai gyerekek) hogy az elmúlt hónapban hányszor kenyészerítették ki maguk a bully-k az ébredre hozott szendvicseiket
- **weight** - testsúly

- **class** - faktor változó ami azt mutatja melyik iskolai osztályba jár a vizsgalati személy. Faktorszintek: class_1, class_2, class_3, class_4.

Két adatfajlt is betöltünk. Mindkét adatfajl úgy lett legenerálva, hogy a diákok különböznek abban, hogy mennyi szendvicset vesznek el tőlük attól függetlenül, hogy milyen a testsúlyuk és attól függetlenül is, hogy melyik iskolai osztályba járnak. Vagyis mind a testsúlynak, mind az osztálynak van hatása az elvett szendvicsek számára.

Vizsgálat a két adatbázis különbözik abban, hogy az, hogy a diák melyik osztályba jár, befolyasolja-e, hogy a testsúlynak mekkora hatása van az elvett szendvicsek számára. A **data_bully_int.csv** adatfajlban *a testsúly hatása ugyanakkor minden osztályban* (függetlenül az osztálytól), míg a **data_bully_slope.csv** adatfajlban *a testsúly hatása különbözik osztályonként* (néhány osztályban a testsúly hatása nagyobb, mint másokban).

```
# load data\t
data_bully_int = read_csv("https://raw.githubusercontent.com/kekecsz/PSYP13_Data_analysis_class-2018/master/data_bully_int.csv")

## Rows: 80 Columns: 3

## -- Column specification -----
## Delimiter: ","
## chr (1): class
## dbl (2): sandwich_taken, weight

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

# assign class as a grouping factor\t
data_bully_int = data_bully_int %>%
  mutate(class = factor(class))

data_bully_slope = read_csv("https://raw.githubusercontent.com/kekecsz/PSYP13_Data_analysis_class-2018/master/data_bully_slope.csv")

## Rows: 80 Columns: 3

## -- Column specification -----
## Delimiter: ","
## chr (1): class
## dbl (2): sandwich_taken, weight

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

data_bully_slope = data_bully_slope %>%
  mutate(class = factor(class))
```

2.4 Adatellenorzes es adattisztitas

Ahogy mindig, eloszor kezdj az adatok ellenorzesével és az esetleges adattisztitással. Ehhez használhatod a View(), summary(), és describe() funkciókat, és a ggplot() funkciót vizualizáláshoz.

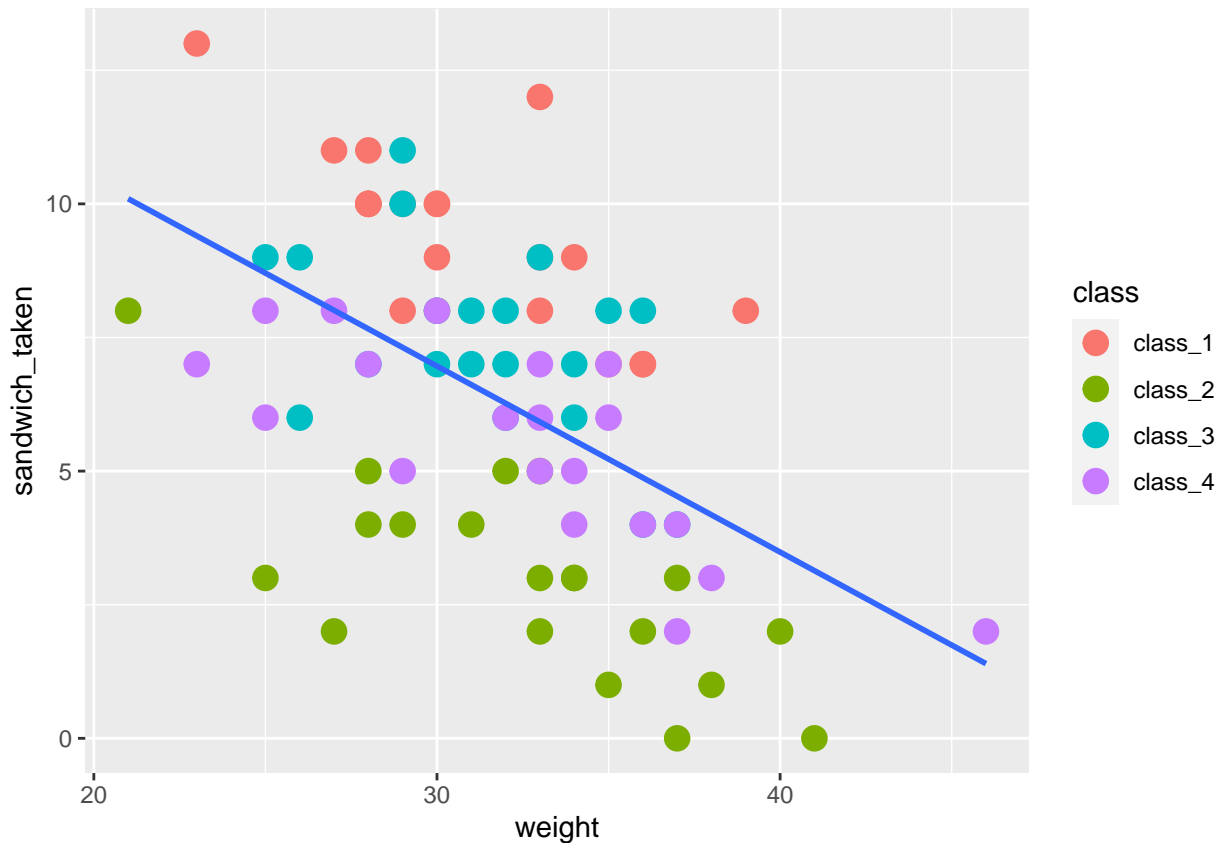
3 A kevert modellek alapfogalmai

3.1 Clustering (csoportosulas) feltarasa

Vizualizáljuk a sandwich_taken és weight változók összefüggését egy pontdiagram (scatterplot) segítségével. Az adatok egyértelmű negatív összefüggést mutatnak a sandwich_taken és weight változók között, de az

adatok variabilitása nagyon nagy.

```
data_bully_int %>%  
  ggplot() + aes(y = sandwich_taken, x = weight) + geom_point(aes(color = class),  
    size = 4) + geom_smooth(method = "lm", se = F, formula = "y ~ x")
```



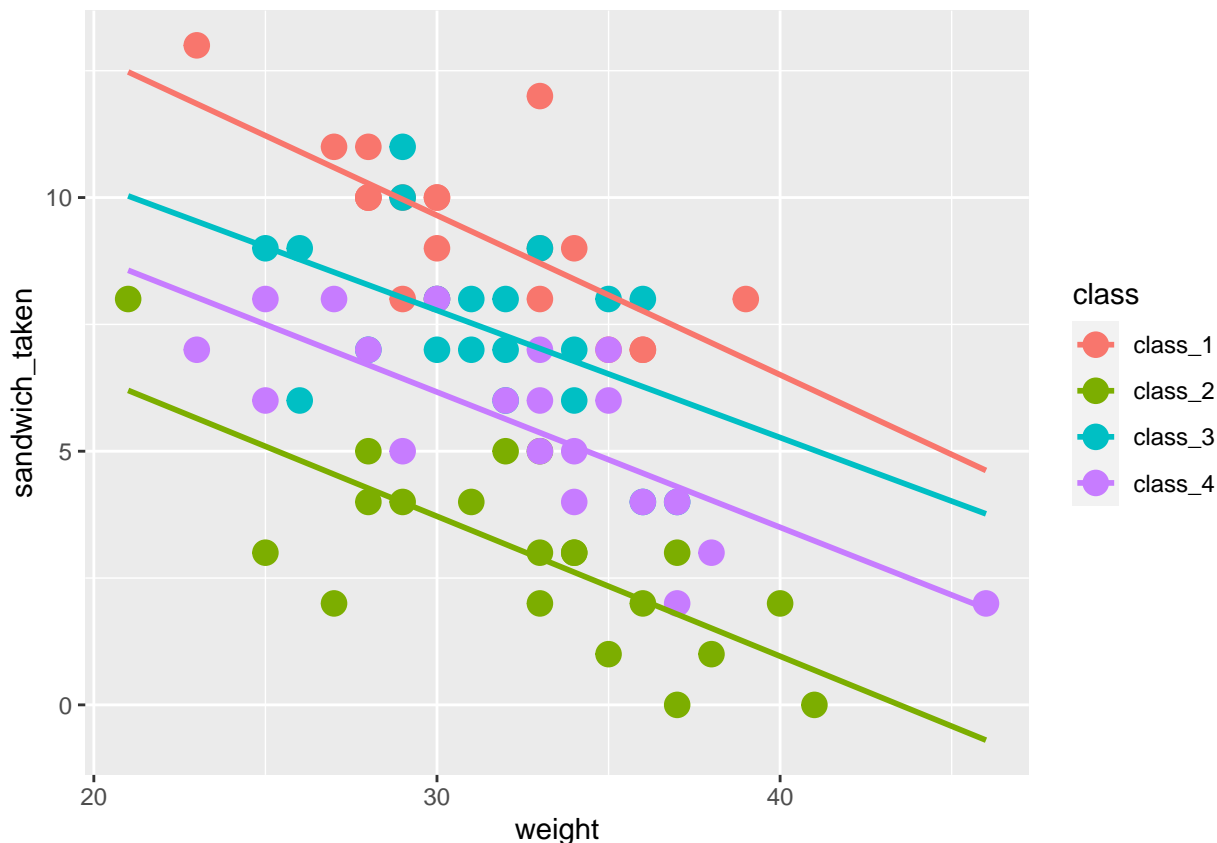
A pontok színe az ábrán azt mutatja, a diák **melyik iskolai osztályba** jár (class_1, class_2, class_3, vagy class_4). Ha jobban megnézzük, úgy tűnik, hogy az azonos színű pontok egymáshoz közel helyezkednek el az ábrán, nem pedig random módon elszórva, ami arra utal, hogy az adatpontok nem teljesen függetlenek egymástól, hanem csoportosulnak (klasztreket alkotnak).

Nezzük meg, hogy az iskolai osztály meg tudja-e magyarázni a variabilitás egy részét. Például felrajzolhatjuk a **regressziós egyenesteket csoportonként**. Ez úgy tűnik, hogy megmagyarázza a variabilitás egy részét, hiszen a regressziós vonalak közelebb kerülnek a valós megfigyelésekhez. Szóval úgy tűnik, hogy érdemes lenne a class változót is figyelembe venni a modellünk megegyezésénél.

(Alább az ábrát elmentjük egy int_plot nevű objektumba, hogy később ugyan ezt az ábrát könnyen előhívhassuk.)

```
int_plot = data_bully_int %>%  
  ggplot() + aes(y = sandwich_taken, x = weight, color = class) +  
  geom_point(size = 4) + geom_smooth(method = "lm", se = F,  
    fullrange = TRUE, formula = "y ~ x")
```

int_plot



3.2 Kevert modellek

Akkor használjuk a kevert modelleket amikor olyan prediktor változónk van, aminek sok szintje/lehetséges értéke van a valóságban, de nekünk ebből a sok lehetséges értékből csak kevésről van információnk a mintánkban.

Jelen kutatásban csak az érdekel minket hogy a testsúly befolyasolja-e az elvett szendvicsek számát, és ha igen, mennyire. Az iskolai osztályok hatása nem része a fő kutatási kérdésnek, és még ha az is lenne, az információt amit ezekről az iskolai osztályokról szerzünk **nem tudnánk általánosítani más iskolákban**. Nem lenne sok értelme megtudni, hogy mi a hatása annak ha valaki a “class 1”-be jár, amikor a regressziós modellünket új mintán szeretnénk bejósolni használni egy új iskolában, hiszen a többi iskolában más osztályok vannak, amiknek vélhetően mások a karakterisztikái. Szóval az iskolai osztály ebben az esetben egy “zavaró tényező” (**nuisance variable**). Vagyis ezt a class változót nem szeretnénk figyelembe venni a regressziós egyenletben, hiszen akkor más iskolákban nem tudnánk felhasználni az egyenletet.

A **kevert modellek** segítségével **figyelembe vehetjük az adatok ilyen fajta csoportosulását anélkül hogy a regressziós egyenletünkbe be kellene tennünk ezeket a zavaró tényezőket**.

3.3 A hatások (prediktorok) két típusa

Itt fontos megkülönböztetnünk a **hatások két típusát**.

Fix hatások (Fixed effects) - Azokat a hatásokat amikkel eddig a lineáris modellekben foglalkoztunk, “fix hatásoknak” (fixed effects) nevezzük. Ezek azok a hatások/prediktorok, amikre regressziós együtthatókat számítunk ki. Ezek a prediktorok a regressziós egyenletünk részei, amiket a későbbiekben is felhasznalunk majd a bejósoláshoz.

Random hatások (Random effects) - Azokat a hatásokat, amiket a “zavaró tényezőknek” tulajdonítunk,

modellezhetjük random hatásként. A random hatásokat úgy modellezzük, hogy bár a megfigyelések különböznek a klaszterek (csoportok) mentén, de az egyes csoportok (mint például itt az osztályok) **hatása nem szisztematikus**, hanem egyfajta véletlenszerű különbségből fakad a csoportok között. A csoportok közötti ilyen véletlenszerű különbség felismerése segít abban, hogy pontosabban kiszámítsuk a fix hatások regressziós együtthatóival kapcsolatos bizonytalanságot (konfidencia intervallumot). Ezek a random hatások **nem kerülnek bele a regressziós egyenletünkbe**, és nem kapunk velük kapcsolatban regressziós együtthatókat.

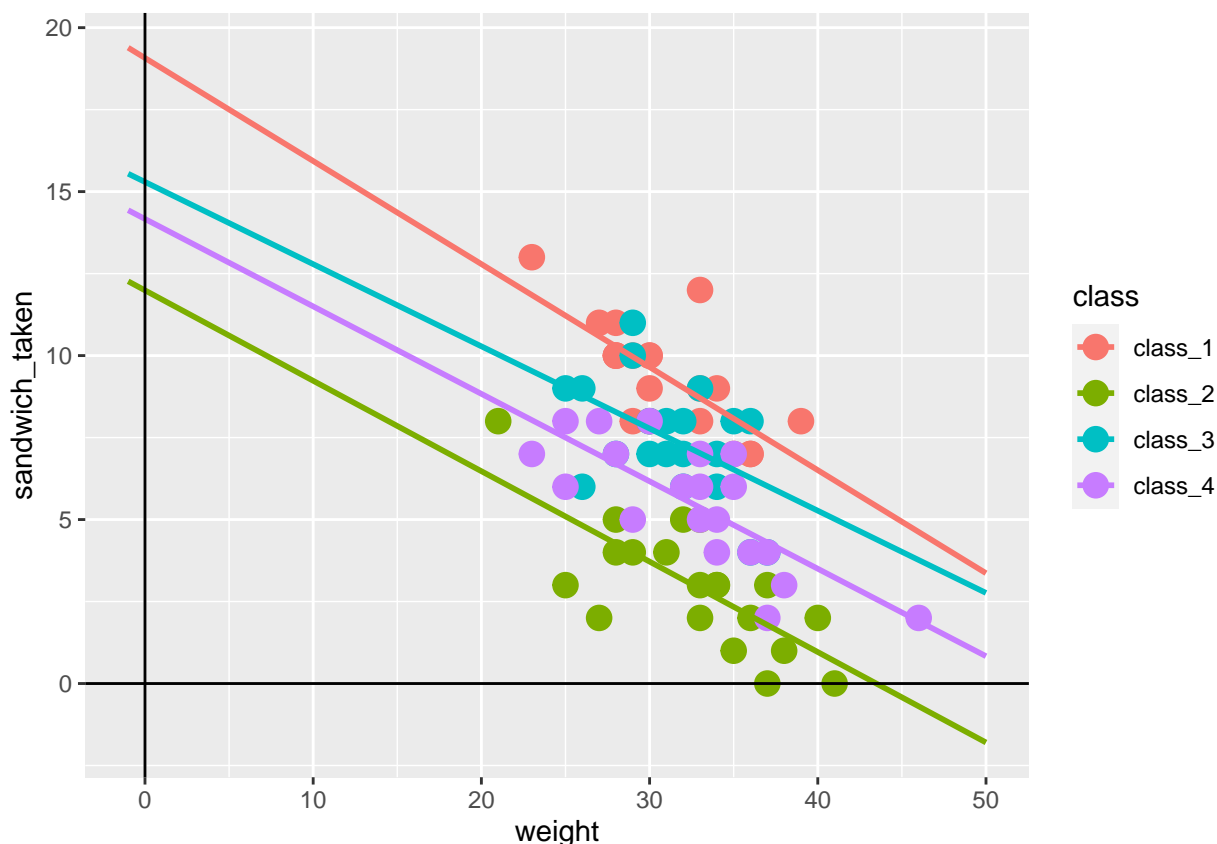
Ezért azokat a modelleket, amik mind fix, mind random hatásokat tartalmaznak, **kevert modelleknek (mixed models)** nevezzük.

3.3.1 A random hatások előfordulási fajtái

Általánosságban a random hatások két módon lehetnek hatással a **kimeneti változóra**. Az egyik hogy **direkt hatást fejtenek ki rá (random intercept)**, a másik hogy a **fix hatások mértékét és irányát befolyásolják (random slope)**.

random intercept, random slope nélkül: Lehetőséges hogy a csoportok (klaszterek) csak abban különböznek egymástól, hogy a **kimeneti változon átlagosan milyen értéket vesznek fel**, de a **fix hatások azonosak** a klaszterek között. Ez igaz a `data_bully_int` adatbázisra. Megfigyelhetjük az ábrán, hogy a regressziós egyenesek meredeksége (slope) nem különbözik az osztályok között, ami arra utal, hogy a testsúly hatása ugyanakkor az egyes osztályokban. Az osztályok csak abban különböznek, hogy milyen „magasan” vannak a regressziós egyenesek, vagyis abban, hogy a regressziós egyenesek milyen értékkel metszik az Y tengelyt. Ez látható az alábbi ábrán is.

```
int_plot + xlim(-1, 50) + geom_hline(yintercept = 0) + geom_vline(xintercept = 0)
```



random intercept, es random slope: A fentiekben csak a `data_bully_int` adatbázist használtuk. Most

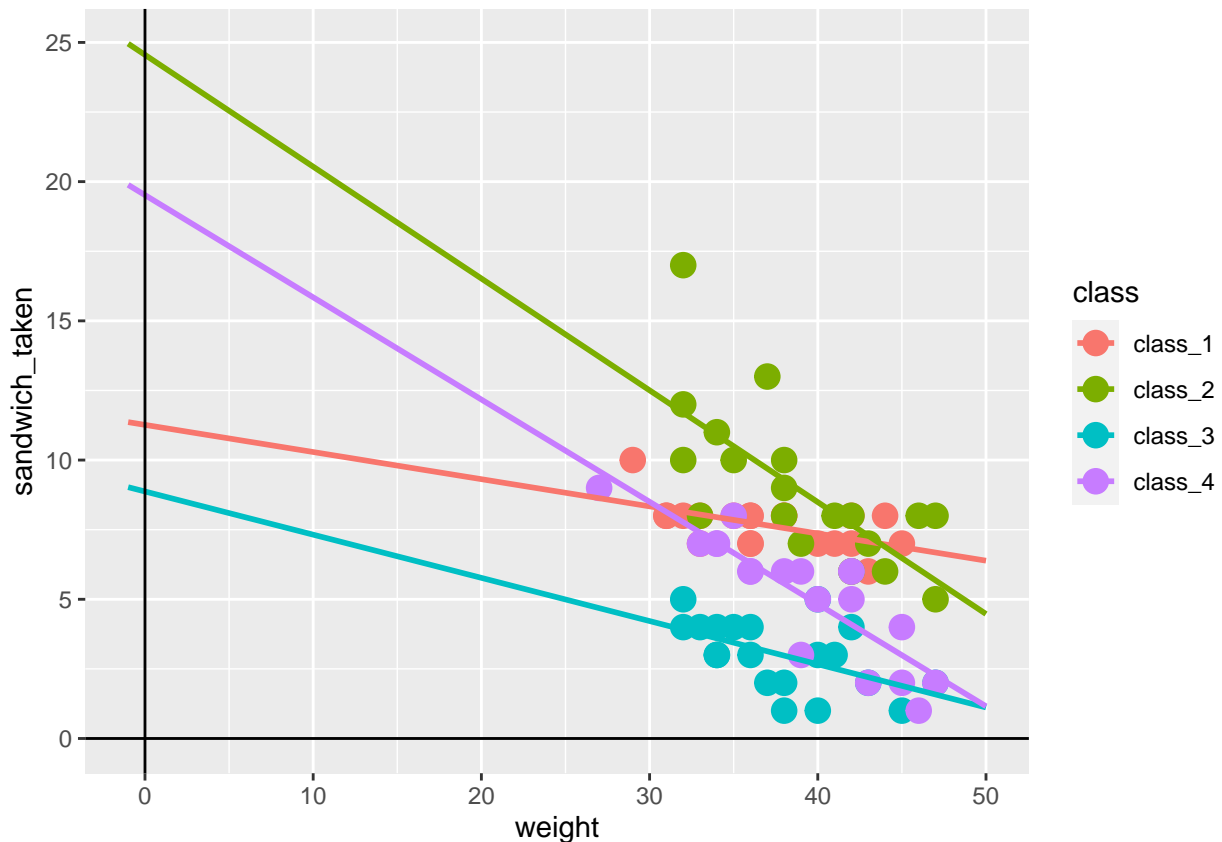
vizsgáljuk meg a másik adatbázist (`data_bully_slope`). Ahogy fent említettük, ebben az adatbázisban azt szimuláltuk, hogy az osztálynak nem csak az elvett szendvicsek számára van hatása, hanem **a testsúly hatása is különbözik az osztályok között**.

Az ábrán jól látszik, az osztályok nem csak abban különböznek, hogy a hozzájuk tartozó regressziós egyenes hol metszi az Y tengelyt, de **a regressziós egyenesek meredeksege is különbözik**.

Peldául a `class_1`-ben a testsúly hatása elhanyagolhatónak tűnik abból a szempontból, hogy kitől mennyi szendvicset vesznek el, míg a `class_2`-ben és `class_4`-ben a testsúly hatása számottevő.

```
slope_plot = data_bully_slope %>%
  ggplot() + aes(y = sandwich_taken, x = weight, color = class) +
  geom_point(size = 4) + geom_smooth(method = "lm", se = F,
  fullrange = TRUE) + xlim(-1, 50) + geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0)
slope_plot

## `geom_smooth()` using formula 'y ~ x'
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
## Warning: Removed 1 rows containing missing values (geom_point).
```



3.4 Kévert modellek felepítése az R-ben

Az alábbi példa bemutatja, hogy hogyan lehet a random hatásokat beépíteni a modellekbe.

Harom modellt fogunk építeni. Először egy szimpla fix hatásokat tartalmazó modellt, majd egy **random intercept modellt**, és egy **random slope modellt**.

A fenti ábra alapján arra lehet következtetni hogy a **data_bully_slope** adatbázisban az iskolai osztály egy olyan random hatás, ami mind a regressziós egyenes intercept-jét, mind a meredkséget (slope) befolyasolja. Ezért normalis esetben csak a random slope modellt illesztünk. A többi modell csak demonstrációs célból építjük, hogy összehasonlítsuk azok formuláit és bejósoló erejét a random slope modellel.

Eloszor építünk egy egyszerű regressziós modellt, melyben egyetlen fix hatás prediktor van: weight. Ezt a modellt a `mod_fixed` objektumba mnetjük.

egyszerű regressziós modell (csak fix hatás)

```
mod_fixed = lm(sandwich_taken ~ weight, data = data_bully_slope)
```

random intercept modell (a random intercept megengedett, de a random slope nem)

A kevert modellek formulája nagyon hasonló a csak fix hatást tartalmazó modellekehez, de az `lm()` függő helyett az `lmer()` függőt használjuk.

A random intercept random hatást a “+ (1|class)” **hozzadádasával** tehetjük a modellbe.

Ez gyakorlatilag azt jelenti, hogy megengedjük a modellnek hogy **külön regressziós egyenest illesszen minden klaszterre** (a mi esetünkben minden iskolai osztályra), de azt meghatározzuk, hogy **minden regressziós egyenesnek ugyan olyan legyen a meredeksége**.

Ezt normalis esetben akkor tennénk, ha azt gyanítanánk hogy az osztályok között nincs lényegi eltérés a fix hatásokban, csak a kimeneti változó átlagos szintjében. Ez a modell jól illeszkedne a `data_bully_int` adatbázisra, de a fenti ábra alapján azt várjuk hogy a `data_bully_slope` adatbázisra kevesbé jól illeszkedik majd.

```
mod_rnd_int = lmer(sandwich_taken ~ weight + (1 | class), data = data_bully_slope)
```

random slope modell (mind a random intercept, mind a random slope megengedett):

Ennek a modellnek a formulája szinte teljesen megegyezik a random intercept modellel, egyedül abban különbözik, hogy a random hatásról szóló részben “+ (1|class)” helyett “+ (weight|class)” szerepel. Ez arra utal, hogy a class random hatás nem csak az interceptre, hanem a weight prediktor hatására is kiterjed.

Ezzel megengedjük a modellünknek, hogy **külön regressziós egyenest illesszen minden klaszterre**, és hogy azoknak **mind** az Y tengellyel való metszéspontja (**intercept**), **mind a meredeksége (slope) különbözhet**.

```
mod_rnd_slope = lmer(sandwich_taken ~ weight + (weight | class),  
  data = data_bully_slope)
```

3.5 Melyik modell reprezentálja legjobban a valóságot?

Hogyan döntjük el hogy **melyik modellt használjuk**? Ahogy korábban is láthattuk, a modellválasztásnál mindig **az elméletileg leginkább megalapozott** modellt érdemes választani. Ha van okunk feltételezni hogy egy hatás különböző lesz a különböző klaszterekben, akkor használjuk a random slope modellt. Ha elméleti alapon inkább úgy ítéljük, hogy a fix hatások valószínűleg állandóak a csoportok között, illesszünk random intercept modellt.

Ennek ellenére van olyan eset, **amikor elméletileg mindket eshetőség elképzelhető**. Ilyen esetben hagyatkozhatunk a **vizualizációra** és a **modellilleszkedési mutatókra**, hogy eldöntsük, melyik modellt érdemesebb használni.

3.5.1 Random hatások vizualizációja

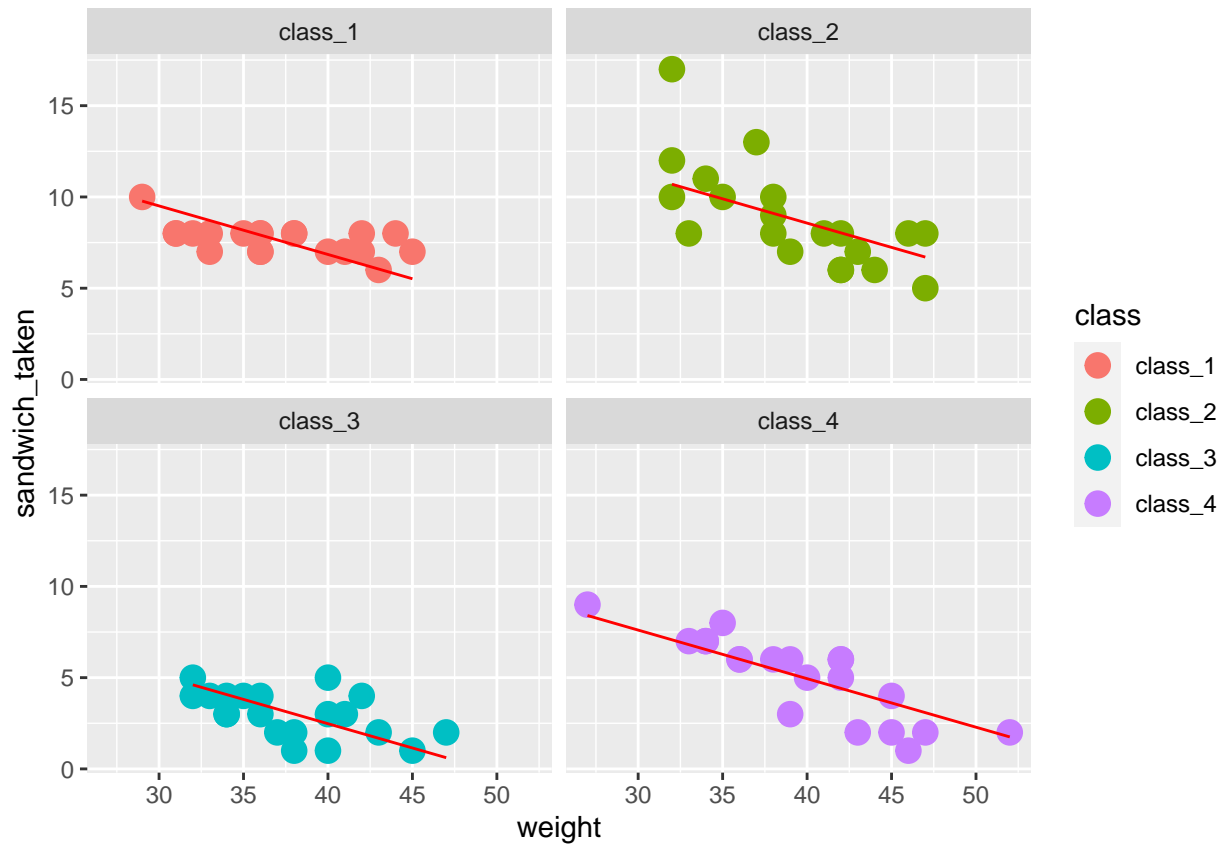
A random hatások explorációja esetén a vizualizáció kulcsszerepet tölt be.

Eloszor érdemes elmentenünk az intercept és a slope modellek által **bejosolt értékeket új változókba** (alább a `pred_int` és `pred_slope` változókba mentjük ezeket). Az eredeti adatbázisból származó predikciókat a `predict()` funkcióval nyerhetjük ki.

```
data_bully_slope = data_bully_slope %>%
  mutate(pred_int = predict(mod_rnd_int), pred_slope = predict(mod_rnd_slope))
```

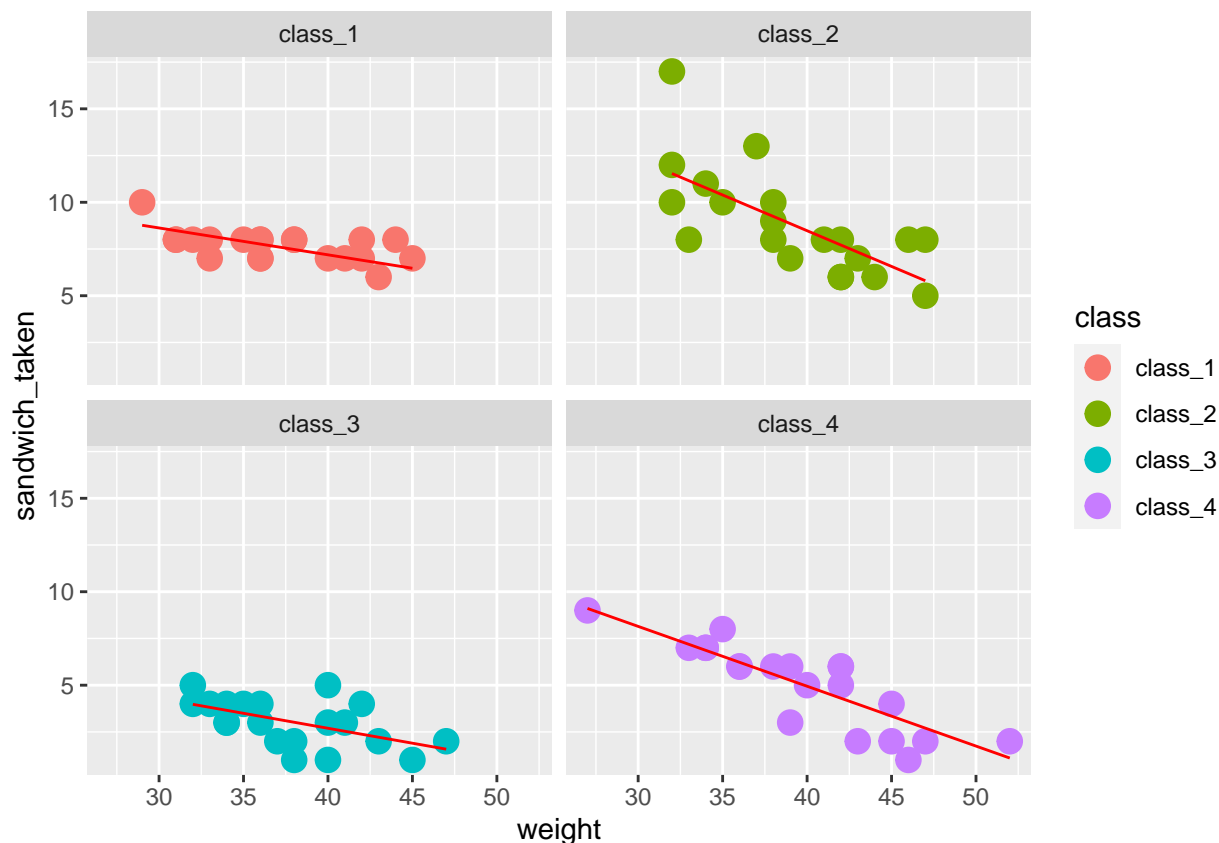
Igy vizualizáljuk a random **intercept modell** predikcióit:

```
data_bully_slope %>%
  ggplot() + aes(y = sandwich_taken, x = weight, group = class) +
  geom_point(aes(color = class), size = 4) + geom_line(color = "red",
  aes(y = pred_int, x = weight)) + facet_wrap(~class, ncol = 2)
```



Igy pedig a random **slope modell** predikcióit:

```
data_bully_slope %>%
  ggplot() + aes(y = sandwich_taken, x = weight, group = class) +
  geom_point(aes(color = class), size = 4) + geom_line(color = "red",
  aes(y = pred_slope, x = weight)) + facet_wrap(~class, ncol = 2)
```



Az ábrákon azt kell megvizsgálnunk, hogy a pontok mintázata kelle-e különbözni a csoportok között, hogy arra engedjen következtetni, hogy csoportonként különbözik a fix hatás.

Mivel a modellek által generált regressziós egyeneseket is tartalmazzák az ábrák, megtehetjük, hogy megvizsgáljuk, mi a hatása annak, hogy megengedjük a regressziós egyenes meredekségének változását a random intercept modellhez képest, ahol ez nincs megengedve. Az illesztés (szinte) mindig jobb lesz a random slope modellben. Ez szükséges, hiszen a modell flexibilisebb, több szabadsága van, ezért közelebb tud helyezkedni a pontokhoz minden csoportban. De **ha az illesztéskésbéli különbözet a két modell között nem számottevő, biztonságosabb a random intercept modellnel maradni, hogy elkerüljük a túlillesztést**, ha csak az elmélet nem támogatja egyértelműen a random slope modellt.

3.5.2 Reziduais hiba összehasonlítása (ezt nem használjuk a gyakorlatban)

Talán első ránézésre csabítónak tűnhet, hogy egyszerűen arra hagyatkozunk a modellválasztás során, hogy melyik modell produkálta a legkevesebb reziduais hibát.

Ha összehasonlítjuk a három modell **reziduais hibáját** (residual sum of squares - RSS), láthatjuk, hogy a csak fix hatást tartalmazó modell használatakor marad a legtöbb hiba, a random intercept modell a második, és a **legkevesebb hibát a random slope modell esetén találjuk**.

```
sum(residuals(mod_fixed)^2)
```

```
## [1] 581.6364
```

```
sum(residuals(mod_rnd_int)^2)
```

```
## [1] 159.5818
```

```
sum(residuals(mod_rnd_slope)^2)
```

```
## [1] 132.228
```

De ez nem igazan meglepo, hiszen a modell komplexitasa es ezzel **flexibilitasa egyre nott**, es errol tudjuk, hogy csokkenti a hibát azon az adatbazison amin a modellt epítettük, de a flexibilitas novelese miatt ez **tulilleszteshez** vezethet, ami új adatokon rosszabb bejoslasi hatekonysaghoz vezet. Ezert a nyers rezidualis hiba osszehasonlitas helyett olyan modell-illeszkedesi mutathoz kell fordulnunk, amik korrigalva vannak a flexibilitasara (ezt ugy is mondhatjuk hogy a modell parameterek szamara.)

3.5.3 conditional AIC

Az egyik olyan modell-illeszkedesi mutato, amely korrigal a modell parameterek szamara az AIC. A kevert modellekhez egy specialis AIC mutato-t szamitunk ki, a **cAIC** mutatot (ami a conditional AIC roviditese). A cAIC-t megkaphatjuk peldaul a cAIC4 package cAIC() funkcioja segitsegevel.

```
AIC(mod_fixed)
```

```
## [1] 391.7357
```

```
cAIC(mod_rnd_int)$caic
```

```
## [1] 294.4023
```

```
cAIC(mod_rnd_slope)$caic
```

```
## [1] 285.6435
```

Ahogy korabban is lattuk az AIC eseten, ha az egyik modell cAIC mutatoja legalabb 2-vel alacsonyabb mint a masik modellhez tartozo cAIC, akkor azt mondhatjuk az alacsonyabb cAIC mutatoval biro modell szignifikansan jobban illeszkedik az adatokhoz.

3.5.4 likelihood ratio test

A masik bevett mod a modellek osszehasonlitasara a likelihood ratio test. Ez a modszer kevesbe elfogadott manapsag, de meg mindig sokan hasznaljak a szakirodalomban.

Ezt csakugy mint a nem kevert modelleknel, a kevert modelleknel is az anova() funkcioval vegezgetjuk el. Fontos, hogy ezt a likelihood ratio test-et csak a beagyazott modellek (nested models) eseten hasznalhatjuk (lasd a modell-osszehasonlitas gyakorlatot).

Az anova() funkcio hasznalatkor egy figyelmeztetést kapunk: 'refitting model(s) with ML (instead of REML)'. ez aztert van mert a likelihood ratio test csak a Maximum likelihood (ML) becslessel dolgozo modellek osszehasonlitasara alkalmas. Viszont a kevert modelleket alapertelmezett modon a Restricted maximum likelihood (REML) becslessel dolgozunk, mert ez kevert modelleknel jobb becsleshez vezet. Ennek ellenere a REML es az ML becslesek hasznalo modellek altalaban nagyon hasonloak egymashoz, ezert ezt a figyelmeztetést legtobbszor figyelmen kívül hagyható.

```
anova(mod_rnd_int, mod_rnd_slope)
```

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

```
## Data: data_bully_slope
```

```
## Models:
```

```
## mod_rnd_int: sandwich_taken ~ weight + (1 | class)
```

```
## mod_rnd_slope: sandwich_taken ~ weight + (weight | class)
```

```
##           npar      AIC      BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod_rnd_int      4 310.00 319.53 -151.00   302.00
## mod_rnd_slope    6 307.94 322.23 -147.97   295.94 6.0595  2    0.04833 *
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.6 Mit kell kozolni az elemzesrol

A kozlendo informaciok nagyon hasonloak ahhoz, amit a fix hatas modellek eseten kozoltunk.

3.6.1 A statisztikai modszer leirasa:

“Ahhoz hogy a bullyzassal szembeni serulekenyseget meghatározzuk, egy **kevert linearis modellt illesztet-tunk**. A kevert modellben az elvett szendvicsek szamat mint **kimeneti valtozot** a testsullyal mint **fix hatasu prediktorral** jósoltuk be. A modellben ezen felul az iskolai osztaly **random hatasat** modelleztuk. Epitettunk mind egy **random slope** es egy **random intercept modellt**. Ahogy ezt a kutatasi tervunkben meghatároztuk, a ket modellt **összehasonlitottuk a cAIC** modellilleszkedeis mutatojuk alapjan, es ez alapjan határoztuk meg, melyik lesz a vegso bejoslo modellunk.”

A kovetkezo funkciókkal kapnank meg a kutatasi jelenteshoz szukseges eredmenyeket:

cAIC:

```
cAIC(mod_rnd_int)$caic
```

```
## [1] 294.4023
```

```
cAIC(mod_rnd_slope)$caic
```

```
## [1] 285.6435
```

anova:

```
anova(mod_rnd_int, mod_rnd_slope)
```

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

```
## Data: data_bully_slope
```

```
## Models:
```

```
## mod_rnd_int: sandwich_taken ~ weight + (1 | class)
```

```
## mod_rnd_slope: sandwich_taken ~ weight + (weight | class)
```

```
##          npar      AIC      BIC logLik deviance  Chisq Df Pr(>Chisq)
```

```
## mod_rnd_int      4 310.00 319.53 -151.00   302.00
```

```
## mod_rnd_slope     6 307.94 322.23 -147.97   295.94 6.0595  2    0.04833 *
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.6.2 A teljes modell illeszkedesenek jellemzese

Az `r2beta()` funkcio kiszamitja a “marginalis R^2 ” mutatot Nakagawa, Johnson & Schielzeth (2017) cikkenek ajanlasi alapjan. Ez az R^2 mutato specialis fajtaja, ami azt mutatja meg, hogy mekkora a modell fix hatasu prediktorai által megmagyarazott varianciaarany. Ezt az R^2 mutatot erdemes hasznalni a modell bejoslo hatekonysaganak megadasara, hiszen a random hatasu prediktorokat uj adatokon nem tudjuk majd hasznalni bejoslasra.

Nincs egy klasszikus F-test aminek az eredmenyet fel lehetne hasznalni annak ertekelesere, hogy a teljes modell szignifikansen jobb bejoslast eredmenyez-e a null-modellnel, de az `r2beta` megadja a 95%-s konfidencia intervallumot, amit felhasznalhatunk szignifikanciatesztelesre. Ahogy korabban is, ha a konfidencia intervallim tartalmazza a 0-t, akkor a modell nem szignifikansen kulonbozik a null modelltol bejoslo hatekonysag tekinteteben.

Ezen felul mind a marginalis mind a kondicionalis R^2 ertekeket megkaphatjuk az `r.squaredGLMM()` funkcio hasznalataval a MuMIn package-bol. Ez a funkcio szinten a Nakagawa, Johnson & Schielzeth (2017) által publikalt formulat hasznalja ezen ertekek kiszamitasahoz.

Hivatkozás: Nakagawa, S., Johnson, P.C.D., Schielzeth, H. (2017) The coefficient of determination R^2 and intraclass correlation coefficient from generalized linear mixed-effects models revisited and expanded. J. R. Soc. Interface 14: 20170213.

```
# marginal R squared with confidence intervals
r2beta(mod_rnd_slope, method = "nsj", data = data_bully_slope)
```

```
## Effect Rsq upper.CL lower.CL
## 1 Model 0.147 0.303 0.036
## 2 weight 0.147 0.303 0.036
```

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

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

```
## R2m R2c
## [1,] 0.1469687 0.833361
```

Az eredmények reszben így írhatjuk le az eredményeket:

“A random slope modell jobb modell-illeszkedéshez vezetett mint a random intercept modell mind a likelihood ratio test ($X^2 = 6.06$, $df =$, $p = .048$) mind a cAIC alapján (cAIC intercept = 294.4, cAIC slope = 285.64). Ezért az alábbiakban a random slope modell eredményeit közöljük.

A kevert lineáris modell szignifikánsan jobb volt mint a null modell. A modellben a fix hatasu prediktorok az elvett szendvicsek varianciajának 14.7%-at magyaráztak meg ($R^2 = 0.15$ [95% CI = 0.04, 0.3]).”

3.6.3 Regressziós egyutthatok közlése

Ezen felül a prediktorokhoz tartozó regressziós egyutthatokról is közölnünk kell az eredményeket. Ezt a korábbiakhoz hasonlóan egy táblázatban szoktuk megtenni, ami minden prediktorra külön sorban közli az adatokat. (Itt csak egy fix hatasu prediktor van, szóval csak két sor lesz a táblázatban, egy az intercept-nek és egy a testsúly prediktornak.)

A végső táblázat valahogy így néz majd ki:

```
## b 95%CI lb 95%CI ub Std.Beta p-value
## (Intercept) 15.89 8.02 23.72 0 .022
## weight -0.25 -0.41 -0.09 -0.42 .041
```

A táblázat egyes elemeit itt találhatod meg:

Regressziós egyutthatok és a hozzájuk tartozó p-értékek a `summary()` függvénnyel kaphatók meg (csak akkor fog p-értéket kiadni a `summary` függvény a kevert modellekre, ha az `lmerTest` package be van töltve)

```
summary(mod_rnd_slope)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: sandwich_taken ~ weight + (weight | class)
## Data: data_bully_slope
##
## REML criterion at convergence: 297.4
##
## Scaled residuals:
## Min 1Q Median 3Q Max
## -2.3408 -0.5631 -0.0075 0.3895 4.0509
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
```

```
## class      (Intercept) 44.77549 6.6914
##           weight      0.01699 0.1303   -0.94
## Residual              1.81776 1.3482
## Number of obs: 80, groups: class, 4
##
## Fixed effects:
##           Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) 15.88863    3.55844   2.94555   4.465   0.0217 *
## weight      -0.25154    0.07234   2.96840  -3.477   0.0408 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## weight -0.940
```

A regressziós egyutthatokhoz tartozó konfidencia intervallumok: (ez a funkció sokaig fut, mert sok iterációt végez a kiszámításhoz)

```
confint(mod_rnd_slope)
```

A standardizált beta értékeket pedig a `stdCoef.merMod()` saját funkcióval lehet kinyerni:

```
stdCoef.merMod(mod_rnd_slope)
```

```
##           stdcoef    stdse
## (Intercept) 0.0000000 0.000000
## weight      -0.4221314 0.121393
```

4 Ismételt méréses modellek

A gyakorlat következő részében az ismételt méréses elemzésekkel foglalkozik. Amikor az elemzésünkben ugyan attól a vizsgálati személytől több adat is szerepel ugyan abból a változóból, ismételt méréses elemzést végzünk (pl. a személy kezelése előtti és utáni depresszió szintje).

5 Adatmenedzsment

5.1 Sebgyógyulás adat betöltése

A gyakorlat során a sebgyógyulás adatbázissal fogunk dolgozni. Ez egy szimulált adatbázis, ami a műtét során ejtett bemetszések gyógyulását vizsgáljuk annak függvényében, hogy a betegek ágya milyen közel van az ablakhoz, és hogy mennyi napfény éri őket a felépülés időszak alatt. Ez a kutatás azt az elméletet teszteli, hogy a kórházi betegeknek szükségük van a külvilággal való kapcsolatra ahhoz, hogy gyorsan felépüljenek. Egy ablak, ami a szabadba nyílik megteremtheti ezt a kapcsolatot a külvilággal, ezért a kutatásunk azt vizsgálja, hogy befolyásolja-e a sebgyógyulás mértékét az, hogy a személynek milyen közel van az ágya a legközelebbi ablakhoz. Az elmélet egy változata azt állítja, hogy az ablak nem csak a külvilággal való szorosabb kapcsolat megteremtésén keresztül vezet gyorsabb gyógyuláshoz, hanem azon keresztül is, hogy több napfényt enged a szobába, és az elmélet szerint a napfény is jótékony hatással van a gyógyulásra.

Változók az adatbázisban:

- ID: azonosító kód
- day_1, day_2, ..., day_7: A műtét utáni 1-7. napon egy orvos megvizsgálta a pedeg bemetszési sebeit, és értékelte azokat egy standardizált seb-allapot értékekkel. Minél nagyobb ez az érték, annál nagyobb vagy rosszabb állapotú a seb (pl. gyulladt). Minden személynek mind a 7 naphoz külön seb-allapot érték tartozik.

- distance_window: A személy agyához legközelebbi ablak távolsága az agytól meterben.
- location: A korhási szarny, ahol a páciens agya van. Ket állasu faktor változo: szintjei “north wing” és “south wing” (a “south wing”-ben több napfény éri a pácienseket, ez a változo azért fontos).

```
data_wound = read_csv("https://raw.githubusercontent.com/kekecsz/PSYP13_Data_analysis_class-2018/master/
# assign ID and location as factors
data_wound = data_wound %>%
  mutate(ID = factor(ID), location = factor(location))
```

5.2 Adatellenorzes és leíró statisztikák

Vizsgáljuk meg az adattáblát a View(), describe(), és table() funkciók segítségével. Fontos, hogy az adattáblában jelenleg minden adat amit egy adott személytől gyűjtöttek **egy sorban található**.

```
View(data_wound)

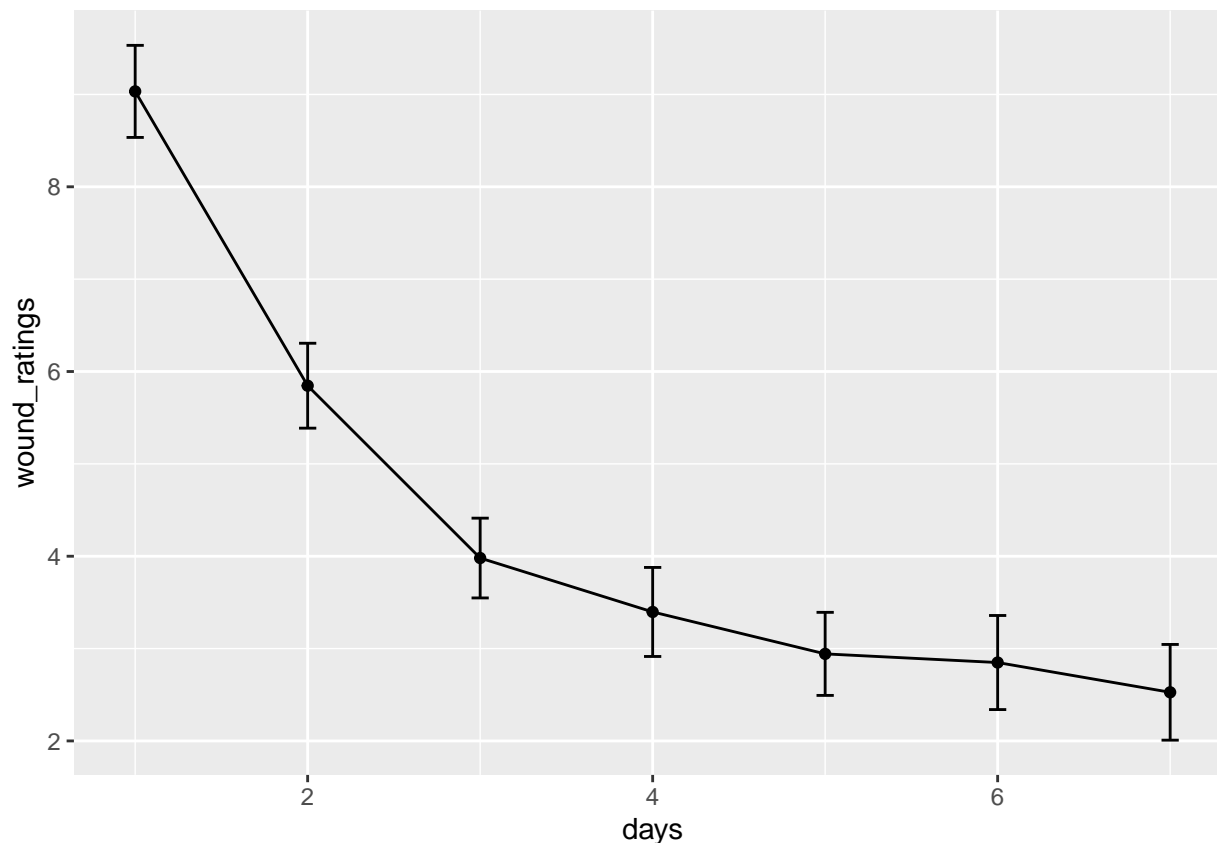
# descriptives
describe(data_wound)
table(data_wound$location)
```

Vizualizálhatjuk is az adatokat. (Az alábbi ábra elkészítéséhez először a day1-day7 változókban külön-külön kiszámoljuk az átlagokat és a standard hibát, majd a standard hibát megszorozva 1.96-al megkapjuk a konfidencia intervallumot. Végül mindezt egy új adat objektumba tesszük, és a geom_errorbar, geom_point, és geom_line segítségével vizualizáljuk. Látható hogy a seb állapot értéke egyre csökken ahogy telnek a napok.)

```
# designate which are the repeated variables
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()
```



6 Ismetelt merések eredményének vizsgálata kevert lineáris modellekkel

6.1 Klaszteres szerkezet keresése az adatokban

Vizsgáljuk meg a továbbiakban a sebgyógyulásra vonatkozó ismetelt merések eredményeit! Ehhez először mentünk el az adatokhoz tartozó **változónéveket egy repeated_variables** elnevezésű objektumba, hogy később könnyen hivatkozhatunk rájuk az alátalunk írt függvényekben!

A változók közötti korrelációt a `cor()` függvénnyel tudjuk megvizsgálni. Figyeljük meg, hogy az ismetelt merések adatpontjai között **eros korreláció fedezhető fel**, azaz az egyes seb-allapot értékekre vonatkozó megfigyelések **nem függetlenek egymástól**. Ez várható is, hiszen a seb-allapot érték, az eredeti bemetszés mérete és a seb gyógyulásának üteme mind függenek a vizsgált betegtől. Adataink tehát csoportokba tomorulnak (klaszterekbe), hasonlóan a korábbi példánkhoz. Azonban míg ott osztály szerinti klaszterek fordultak elő, itt most a klaszterek maguk a résztvevők.

```
# 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
```

6.2 Az adattábla atformazása szélesbol hosszú formatumba

A klaszteres szerkezetből kifolyólag hasonlóan kevert modellekkel elemezhetjük adatainkat mint ahogy azt a bántalmazással kapcsolatos adatsornál tettük. Ehhez azonban először **át kell rendeznünk az adatainkat**, hogy használhassuk a lineáris kevert modell (`lmer()`) függvényt.

Jelenleg a dataframe-ünk minden sora egy adott pácienshez tartozó seb-állapot értékre vonatkozó 7 megfigyelésből áll (vagyis az adatgyűjtés alatt napi egy). Ezt az elrendezést **wide format**-nak nevezik (széles formátum).

Az `lmer()` függvény megfelelő működéséhez az adattábla minden sorához csak egyetlen megfigyelés tartozhat. Jelen esetben ez azt jelentené, hogy az egyes résztvevőkhöz tartozó sorok száma 1 helyett 7 kell legyen. Így az ID, `distance_window`, és `location` változók az adott pácienshez tartozó sorokban megegyeznek majd, és csak az egyes seb-állapot értékek különböznek majd, melyekhez minden sorban mindössze egyetlen oszlop tartozna így. Ezt az elrendezést általában **long format**-nak hívjuk (hosszú formátum).

A fenti átalakítás elvégzésének egy egyszerű módja, ha a `gather()` függvényt alkalmazzuk a `tidyr` csomagból.

1. meghatározunk egy változónevet, amiben az **ismételt megfigyelések indexét** tároljuk majd az új formátumú adatbázisban. Ez nálunk az alábbi példában “days”-nek neveztük el (**key = days**).
2. meghatározunk egy változónevet, amiben az **ismételten megfigyelt adatok** kerülnek majd. Mivel nekünk a megfigyelt adatunk a seb állapota, ezért ezt “wound_rating”-nek neveztük el (**value = wound_rating**).
3. meghatározzuk, hogy a jelenleg használt széles formátumban **mely oszlopok tartalmazzák az ismételten megfigyelt adatot**. Ez a széles adatbázisunkban a `day_1`, `day_2` ... `day_7` oszlopok. Ezt a `tidyverse`-ben könnyen lerövidíthetjük, a **day_1:day_7** kifejezés a `day_1` és `day_7` közötti oszlopok neveit jelöli.

Az `arrange()` függvény használatával az adatok rendezhetőek a hozzájuk tartozó azonosító (“ID”) alapján. Bár az adott feladat elvégzéséhez nem szükséges 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
```

A fontos megjegyezni, hogy a ‘days’ változó jelenleg a széles formátumból származó változó neveket tartalmazza (‘day_1’, ‘day_2’ stb.). Az egyszerűbb kezelhetőség érdekében ezeket egyszerűen az egyes napokat jelölő számokra (1-7) cseréljük. Ezt legkönnyebben a `mutate()` és `recode()` függvényekkel 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)
```

6.3 Kevert lineáris modell kialakítása

Most, hogy megfelelő alakba hoztuk adatainkat, előállíthatjuk az előrejelzésekhez szükséges modellt. Ezzel a modellel a műtét utáni nap (days), az ablaktól való távolság (‘distance_window’) és északi vagy déli elhelyezés (‘location’) alapján megbecsülhető lesz a seb-állapot érték (‘wound_rating’).

Mivel az előrejelzésünk kimenete a résztvevők szerinti klaszteres szerkezetet mutat, ezért a **random effect prediktor** a résztvevő **azonosítója** (‘ID’) lesz. Az korábbi gyakorlathoz hasonlóan, most is két modellel fogunk illeszteni, a random intercept és a random slope modelleket.

Említést érdemel, hogy a **random intercept model** esetében azt feltételezzük, hogy minden résztvevő eltér az átlagos seb-állapot értékeit tekintve, de a fix hatás prediktorok (‘days’, ‘distance_window’, és ‘location’) azonosak az egyes résztvevők esetében. Ezzel szemben, a **random slope model** esetében nem csak a baseline seb-állapot érték, de a fix hatás prediktorok hatásmértéke is résztvevőnként változóak.

Mivel 3 különböző fix hatás prediktor is van a modellben, ezért a random slope modellben ezek közül bármelyiket vagy akár mindegyiket specifikálhatjuk mint aminek hatása a résztvevőkből származó random hatástól is függeni fog. A days prediktor random slope-ját például a **“+ (days|ID)”** alakban tehetjük a **modellhez**. Ez a kifejezés azt jelenti hogy a modellünkben megengedjük hogy idő múlásának hatása más és más legyen résztvevőnként a seb gyógyulására. Másnéven **az emberek különbözhetnek abban, egy nap alatt mennyit gyógyul a sebük**.

További lehetőségként felmerül, hogyha a másik két prediktor random slope-ját 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 egyelőre a korábban leírt, egyszerűbb + (days|ID) modellnél.

A random slope modell amit mod_rep_slope-nek neveztünk el, lefuttatáskor figyelmeztetést (Warning Message) ad: “Model failed to converge with max|grad| = ...”. Ez azt jelenti hogy az elemzés végeredménye kevésbé megbízható mint szeretnénk. Ez gyakori jelenség amikor random slope modellt építünk, mert ezek bonyolult modellek és nagy elemszámra van szükség hogy helyesen tudjuk illeszteni ezeket a modelleket. Ilyen esetben néha segíthet ha az alapértelmezett bobyqa optimizer helyett egy másik optimizer-t használunk. Alább a Nelder_Mead optimizer-t adjuk a modellhez, amit mod_rep_slope_opt-nak nevezzük el, és ez már nem ad konvergencia-hibát.

```
# 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)

```

6.4 Az eltérő modellek összehasonlítása

Hasonlítsuk most össze a különböző modellek által kapott bejósolt értékeket (predikciókat)!

A könnyebb összehasonlíthatóság kedvéért, vizualizáljuk adatainkat! Ehhez először mentjük el a modellek predikcióit egy-egy új változóban, majd ábrázolhatjuk az egyes bejósolt értékeket a valódi értékek függvényében, az egyes (random intercept és random slope) modellekre vonatkozó ábrákon külön-külön.

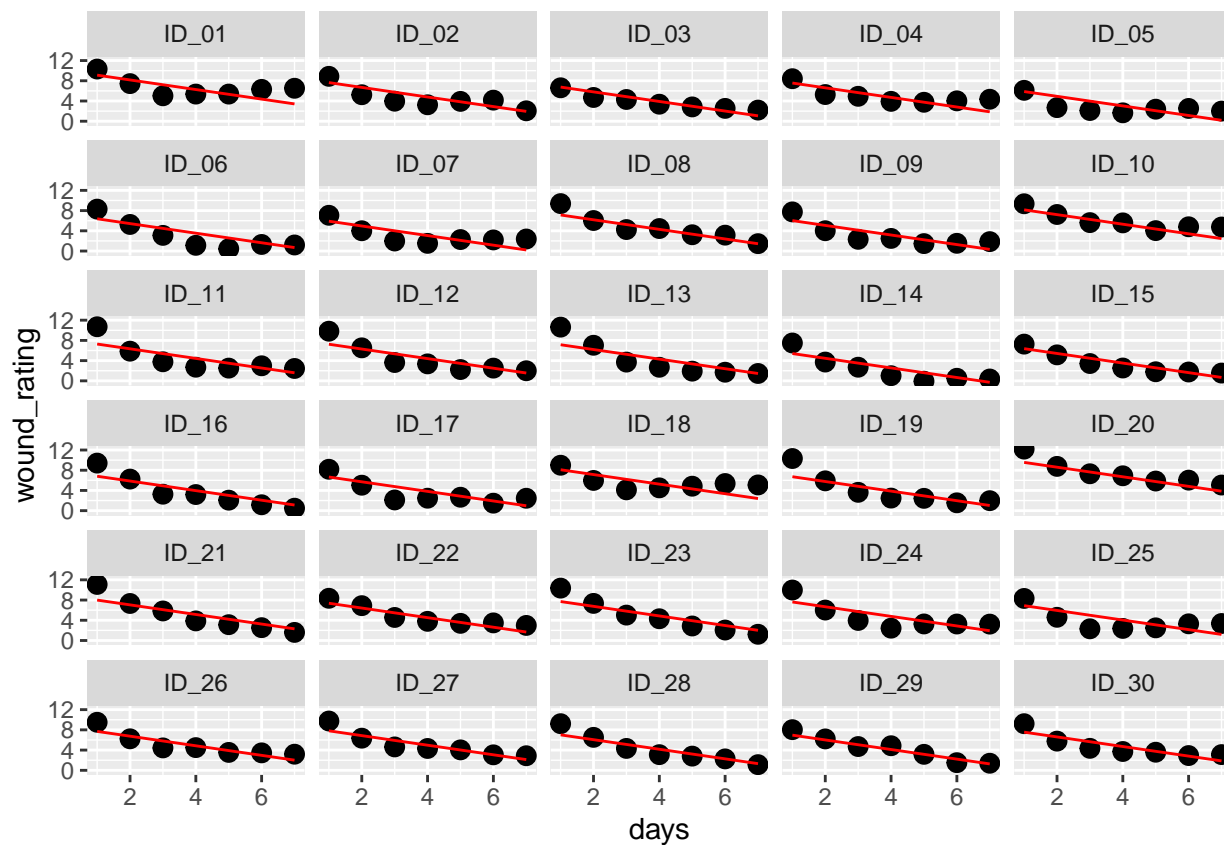
(Az alábbiakban létrehoztunk egy másolatot az adatokat tartalmazó objektumról, hogy az eredeti adatok változatlanul megmaradhassanak.)

```

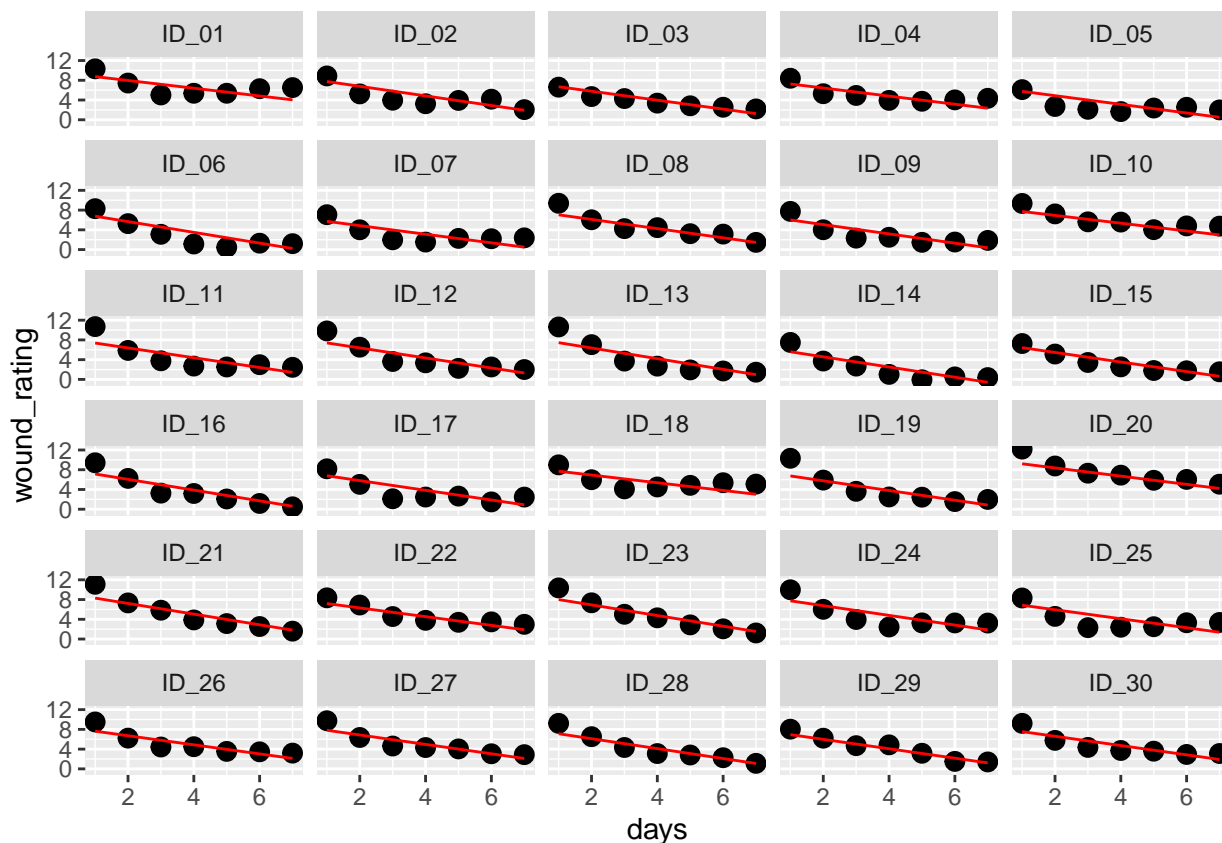
data_wound_long_withpreds = data_wound_long
data_wound_long_withpreds$pred_int = predict(mod_rep_int)
data_wound_long_withpreds$pred_slope = predict(mod_rep_slope_opt)

# 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)

```



```
# 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 eltérő modellek alapján kapott eredmények között nincs számottevő 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_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
```

A fenti módszerek egyikével sem találunk jelentős eltérést a két modell használata között, így a jelenlegi minta esetén semmiféle előnnyel sem jár a random slope módszer. Ez persze magában még nem elegendő 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.

6.5 A modell kiegészítése a quadratikus hatás hozzáadásával

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

A nem lineáris viselkedés figyelembevétele érdekében, adjuk hozzá a days prediktor quadratikus (négyzetes) hatását a modellhez, hogy modellezzük az U alakú összefüggést!

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

Mentsük a modell által bejósolt értékeket egy új dataframe-be, amely tartalmazza a korábbi bejósolt értékeket is!

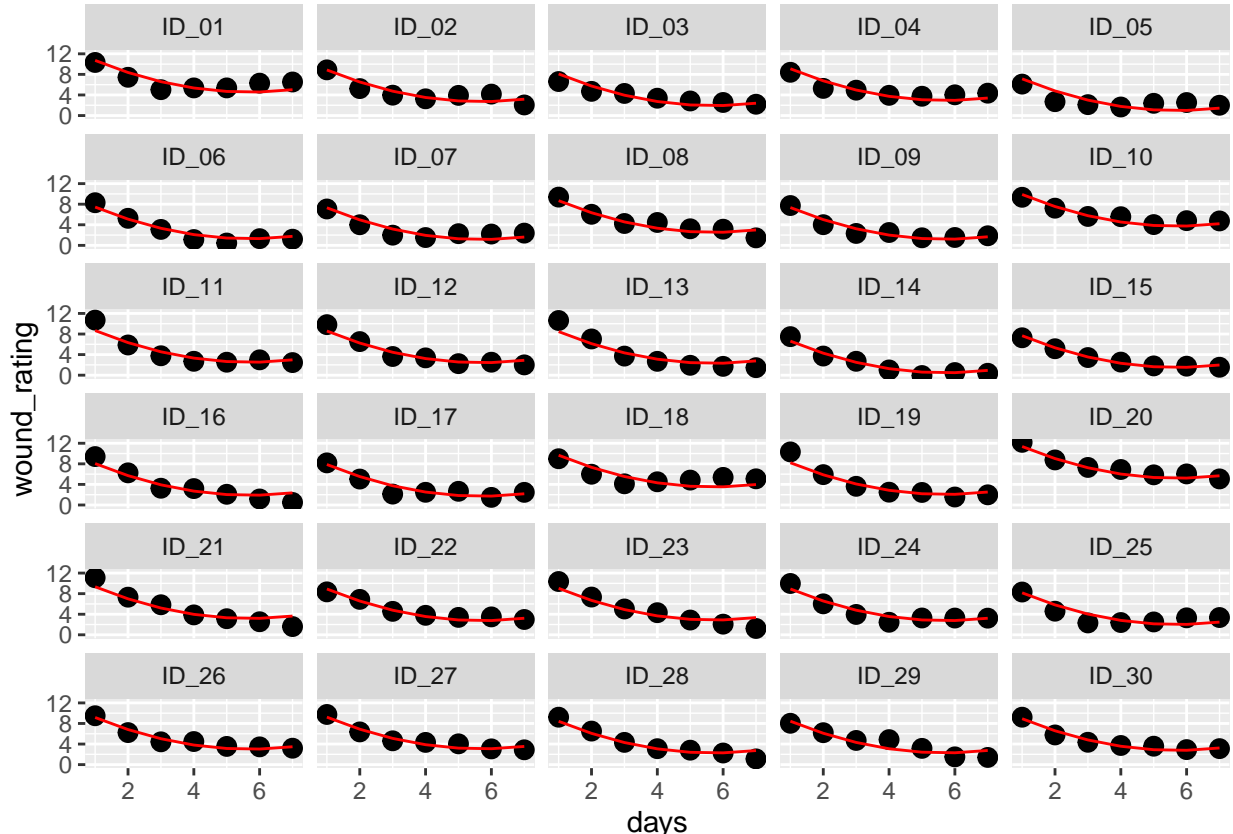
```
data_wound_long_withpreds$pred_int_quad = predict(mod_rep_int_quad)
```

Most pedig hasonlítsuk össze a négyzetes tagokkal bővített, 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 + (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
```

Az összehasonlítás alapján úgy tűnik, hogy a quadratikus hatást is megengedő modell előrejelzései lényegesen pontosabbak mint a csak lineáris tagokat használóé.

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

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

A modelldiagnosztika gyakorlatban tanultak szerint végezzük el a centrálást, és illesszük újra modellünket az így kapott prediktorokat használva.

```
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)
```

Az előző gyakorlathoz hasonlóan kérjük eredményeink bemutatását!

```
# 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)
```

```
##           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 0.0000000
## 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
```

Gyakorlás

Olvassuk be a műtési fájdalom adatsort!

Ez az adatsor a műtét utáni fájdalom mértékéről, és az ezzel feltételezhetően összefüggő néhány egyéb értékekről tartalmaz információkat.

Változóink:

- ID: résztvevő azonosítója
- pain1, pain2, pain3, pain4: A használt adatsorban a fájdalom a műtét utáni négy egymást követő napon volt mérve egy 0-tól-10-ig terjedő folytonos vizuális skálán.
- sex: a résztvevő bejelentett neme
- STAI_trait: A résztvevő State Trait Anxiety Inventory-n elért pontszáma
- pain_cat: fájdalom katasztrofizálása
- cortisol_serum; cortisol_saliva: A kortizol egy a stress hatására előállított hormon. A kortizol szintet véréből és nyálból, közvetlenül a műtét után határozták meg.
- mindfulness: A Mindfulness kérdőív alapján a résztvevőre jellemző Mindfulness érték
- weight: résztvevő tömege kg-ban.
- IQ: Résztvevő IQ-ja a műtét előtt egy héttel felvett IQ teszt alapján
- household_income: résztvevő háztartásának bevétele USD-ben

Gyakorló feladatok:

1. Olvassuk be az adatokat (egy .csv kiterjesztésű file-ból). Az adatokat az alábbi linkről tölthetjük le: ["https://tinyurl.com/data-pain1"](https://tinyurl.com/data-pain1).
 2. Alakítsuk adatainkat hosszú formátumúvá (célszerű a gather() vagy 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 műtét utáni fájdalom varianciájának lehető legszélesebb körű lefedésére. (A műtét utáni fájdalom meghatározásához tetszőleges fix prediktort választhatunk, amennyiben annak feltehetően van valami köze a fájdalom mértékéhez.) Mivel adataink a résztvevők szerinti klaszteres szerkezetet mutatnak, modellünkben vegyük figyelembe a résztvevők azonosítója szerinti véletlen hatást.
 4. Kísérletezzünk mind a random intercept, mind pedig a random slope modellekkel, majd hasonlítsuk össze őket a cAIC() függvény felhasználásával.
 5. Alkossunk olyan random intercept és random slope modelleket, ahol az egyetlen prediktor az idő (műtét óta eltelt napok száma). Vizualizáljuk a modelljeink alapján kapott regressziós vonalakat, minden résztvevőre külön-külön, és hasonlítsuk össze hogyan illeszkednek a megfigyeléseinkre. Van bármi előnye ha az időt 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 előrejelzésében ez a modell, mint a null modell?
-