Kevert modellek es imsetelt mereses modellek

Zoltan Kekecs

09 November, 2021

Contents

1 Absztrakt

Az eddig tanult linearis regresszios modellek a csoportokba rendezodott adatokat ugy kezelik, hogy prediktorkent bevonjak azokat a modellbe. Ez remekul mukodik ha keves csoport van (a csoportosito valtozonak keves szintje van) es minden csoportot van modunk megfigyelni. Pl. kiserleti vs. kontroll csoport. De ezek a modellek nem jol mukodnek olyan esetekben ha az adataink csoportokba/klaszterekbe rendezodnek egy olyan valtozo menten aminek a kutatasunk celpopulaciojaban **sok csoportszintjet kulonithetjuk el, de a mi kutatasunkban ennel kevesebb figyelheto 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 esetkben **kevert modelleket** celszeru hasznalni.

Ebben a gyakorlatban megismerheted a kevert modellekkel kapcsolatos alapfogalmakat, valamint hogy hogyan lehet oket felepiteni.

2 Adatmenedzsment es leiro statisztikak

2.1 Package-ek betoltese

Ebben a gyakorlatban a kovetkezo package-ekre lesz szukseg:

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

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 A Bully-zas adatbazis betoltese

Ebben a gyakorlatban az altalanos **iskolai bully**-zasrol (magyarul talan "iskolai zaklatas") teszunk fel kutatasi kerdeseket. Ez egy szimulalt adatbazis. Egy olyan kutatas adatait szimulalja, melyben az erdekel minket, hogy a **testsuly** hogyan befolyasolja a gyerekek **serulekenyseget a bully-zassal szemben**. A kutatok azt feltetelezik hogy a testsuly osszefugg az elvett szendvicsek szamaval.

Valtozok:

- **sandwich_taken** - A bullyzassal kapcsolatos serulekenyseg meroszama. A kutatasban megkerdeztek a vizgaltai szemelyeket (altalanos iskolai gyerekek) hogy az elmult honapban hanyszor kenyszeritettek ki toluk a bully-k az ebedre hozott szendvicsuket
- **weight** - testsuly
- **class** - faktor valtozo ami azt mutatja melyik iskolai osztalyba jar a viszgalati szemely. Faktorszintek: class_1, class_2, class_3, class_4.

Ket adatfajlt is betoltunk. Mindket adafajl ugy lett legeneralva, hogy a diakok kulonboznek abban, hogy mennyi szendvicset vesznek el toluk attol fuggoen hogy milyen a testsulyuk es attol fuggoen is hogy melyik iskolai osztalyba jarnak. Vagyis mind a testsulynak, mind az osztalynak van hatasa az elvett szendvicsek szamara.

Viszont a ket adatbazis kulonbozik abban, hogy az, hogy a diak melyik osztalyba jar, befolyasolja-e hogy a testsulynak mekkora hatasa van az elvett szendvicsek szamara. A **data_bully_int.csv** adatfajlban *a testsuly hatasa ugyan akkora minden osztalyban* (fuggetlen az osztalytol), mig a **data_bully_slope.csv** adatfajlban *a testsuly hatasa kulonbozik osztalyonkent* (nehany osztalyban a testsuly hatasa nagyobb mint masokban).

```
# load data
data_bully_int = read.csv("https://raw.githubusercontent.com/kekecsz/PSYP13_Data_analysis_class-2018/ma

# asign class as a grouping factor
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/

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

2.4 Adatellenorzes es adattisztitas

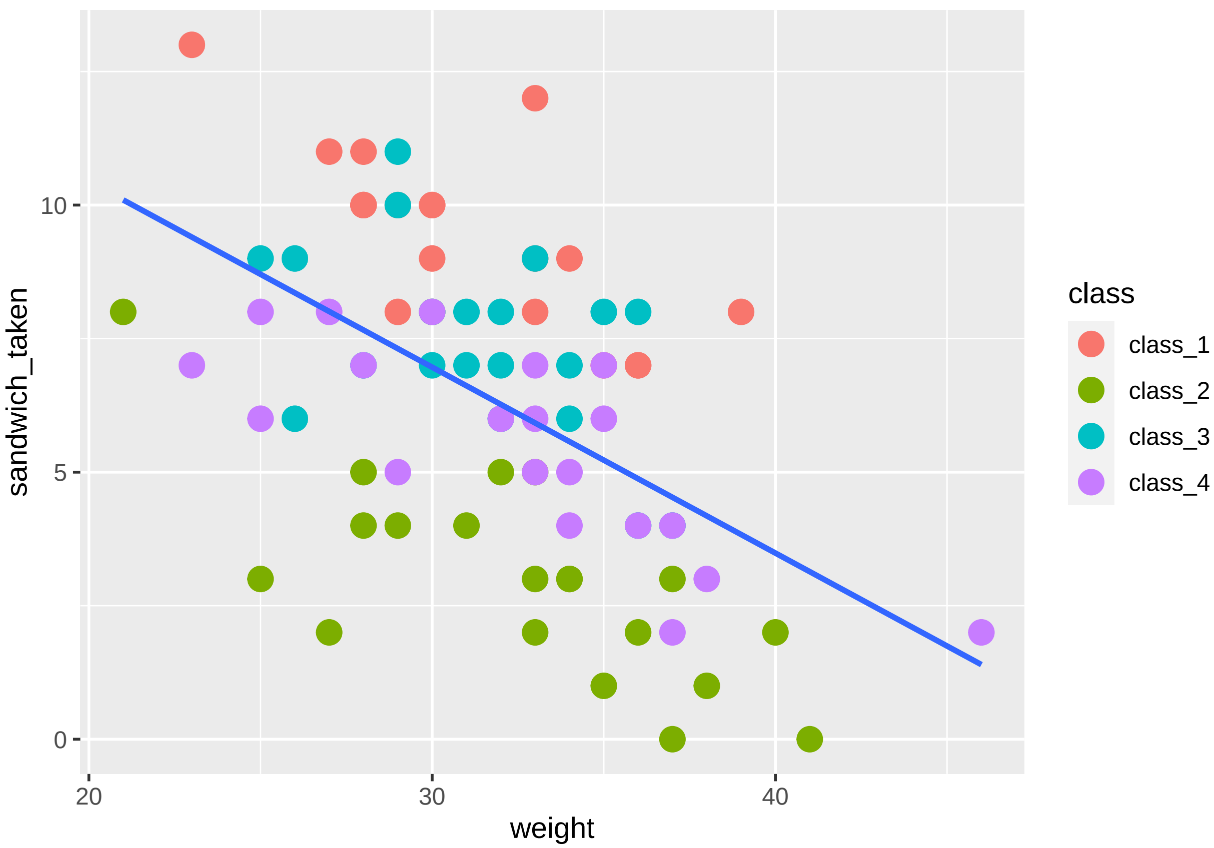
Ahogy mindig, eloszor kezdd az adatok ellenorzesevel es az esetleges adattisztitassal. Ehhez hasznalhatod a View(), summary(), es describe() funkciokat, es a ggplot() funkciot vizualizalashoz.

3 A kevert modellek alapfogalmai

3.1 Clustering (csoportosulas) feltarasa

Vizualizaljuk a sandwitch_taken es weight valtozok osszefuggeset egy pontdiagram (scatterplot) segitsegevel. Az adatok egy egyertelmu negativ osszefuggest mutatnak a sandwitch_taken es weight valtozok kozott, de az adatok variabilitasa 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')
```

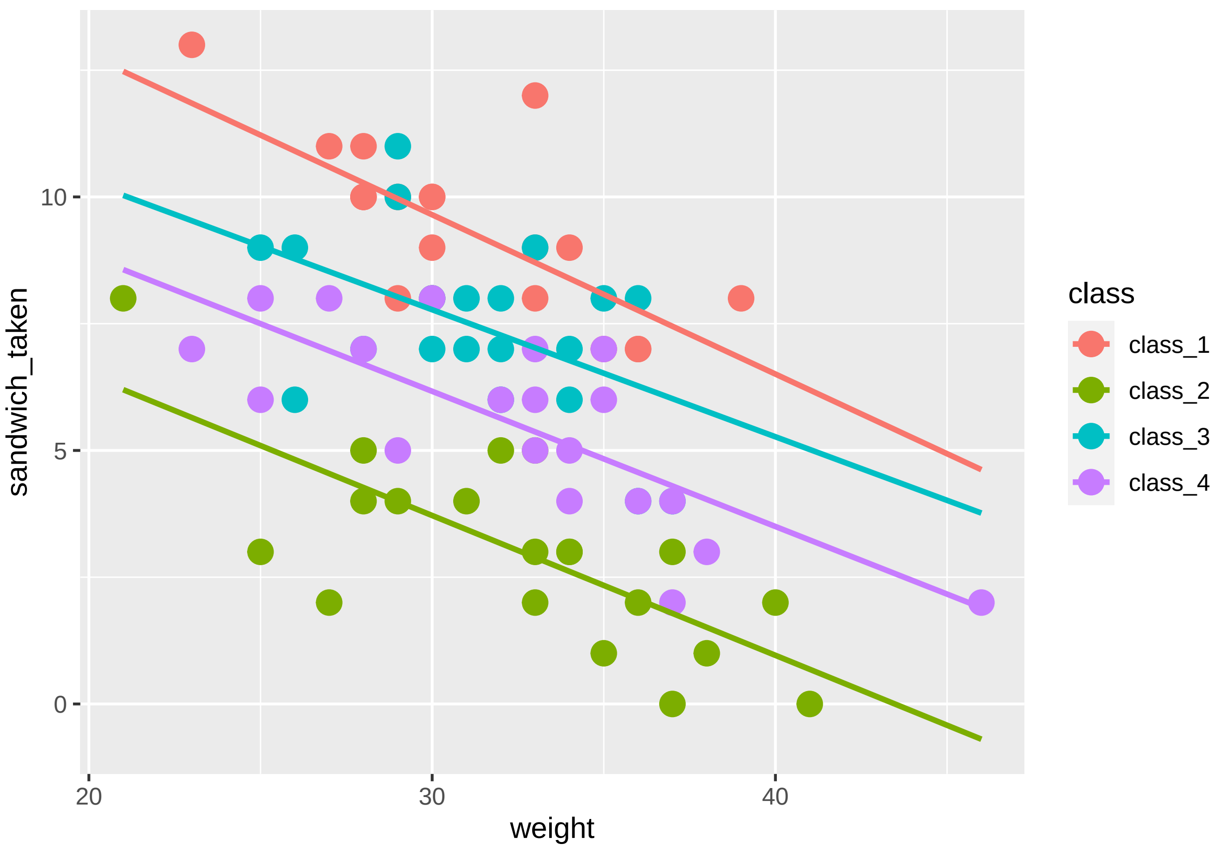


A pontok szine az abran azt mutatja, a diak **melyik iskolai osztalyba** jar (class_1, class_2, class_3, vagy calss_4). Ha jobban megnezzuk, ugy tunik, hogy az azonos szinu pontok egymashoz kozel helyezkednek el az abran, nem pedig random modon elszorva, ami arra utal, hogy az adatpontok nem teljesen fuggetlenek egymastol, hanem csoportosulnak (klasztreket alkotnak).

Nezzuk meg, hogy az iskolai osztaly meg tudja-e magyarazni a variabilitas egy reszet. Peldaul felrajzolhatjuk a **regresszios egyeneseket csoportonkent**. Ez ugy tunik hogy megmagyarazza a variabilitas egy reszet, hiszen a regresszios vonalak kozelebb kerulnek a valos megfigyelesekhez. Szoval ugy tunik hogy erdemes lenne a class valtozot is figyelembe venni a modellunk megepitesenel.

(Alabb az abrat elmentjuk egy int_plot nevu objektumba, hohgy kesobb ugyan ezt az abrat konnyen elohivhassuk.)

```
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 kutasban csak az erdekel minket hogy a testsuly befolyasolja-e az elvett szendvicsek szamat, es ha igen, mennyire. Az iskolai osztalyok hatása nem része a fő kutatási kérdésnek, és még ha az is lenne, az informaciot amit ezekrol az iskolai osztalyokrol szerzunk **nem tudnank altalanositani mas iskolakban**. Nem lenne sok értelme megtudni, hogy mi a hatása annak ha valaki a "class 1"-be jár, amikor a regressziós modellünket új mintán szeretnenk bejoslasra hasznalni egy uj iskolaban, hiszen a tobbi iskolaban mas osztalyok vannak, amiknek velhetoen masok a karakterisztikai. Szoval az iskolai osztaly ebben az esetben egy "zavaro tenyezo" (**nuisance variable**). Vagyis ezt a class valtozot nem szeretnenk figyelembe venni a regresszios egyenletben, hiszen akkor mas iskolakban nem tudnak felhasznalni az egyenletet.

A **kevert modellek** segitsegevel **figyelembe vehetjuk az adatok ilyen fajta csoportosulasat anelkul hogy a regresszios egyenletunkbe be kellene tennunk** ezeket a zavaro tenyezoket.

3.3 A hatasok (prediktorok) ket tipusa

Itt fontos megkulonboztetnonk a **hatasok ket tipusat**.

Fix hatasok (Fixed effects) - Azokat a hatasokat amikkel eddig a linearis modellekben foglalkoztunk, "fix hatasoknak" (fixed effects) nevezzuk. Ezek azok a hatasok/prediktorok, amikre regresszios egyutthatokat szamitunk ki. Ezek a predikcios a regressziós egyenletünk reszei, amiket a kesobbieknben is felhasznalunk majd a bejoslashoz.

Random hatasok (Random effects) - Azokat a hatasokat, amiket a "zavaro tenyezoknek" tulajdonitunk, modellezhetjuk random hataskent. A random hatasokat ugy modellezzuk, hogy bar a megfigyelesek kulonboznek a klaszterek (csoportok) menten, de az egyes csoportok (mint peldaul itt az osztalyok) **hatasa nem szisztematikus**, hanem egyfajta veletlenszeru kulonbsegbol fakad a csoportok kozott. A csoportok kozotti ilyen veletlenszeru kulonbozoseg felismerese segit abban, hogy pontosabban kiszámítsuk a fix hatasok regresszios egyutthatoival kapcsolatos bizonytalansagot (konfidencia intervallumot). Ezek a random hatasok **nem kerulnek bele a regresszios egyeletunkbe**, es nem kapunk veluk kapcsolatban regresszios egyutthatokat.

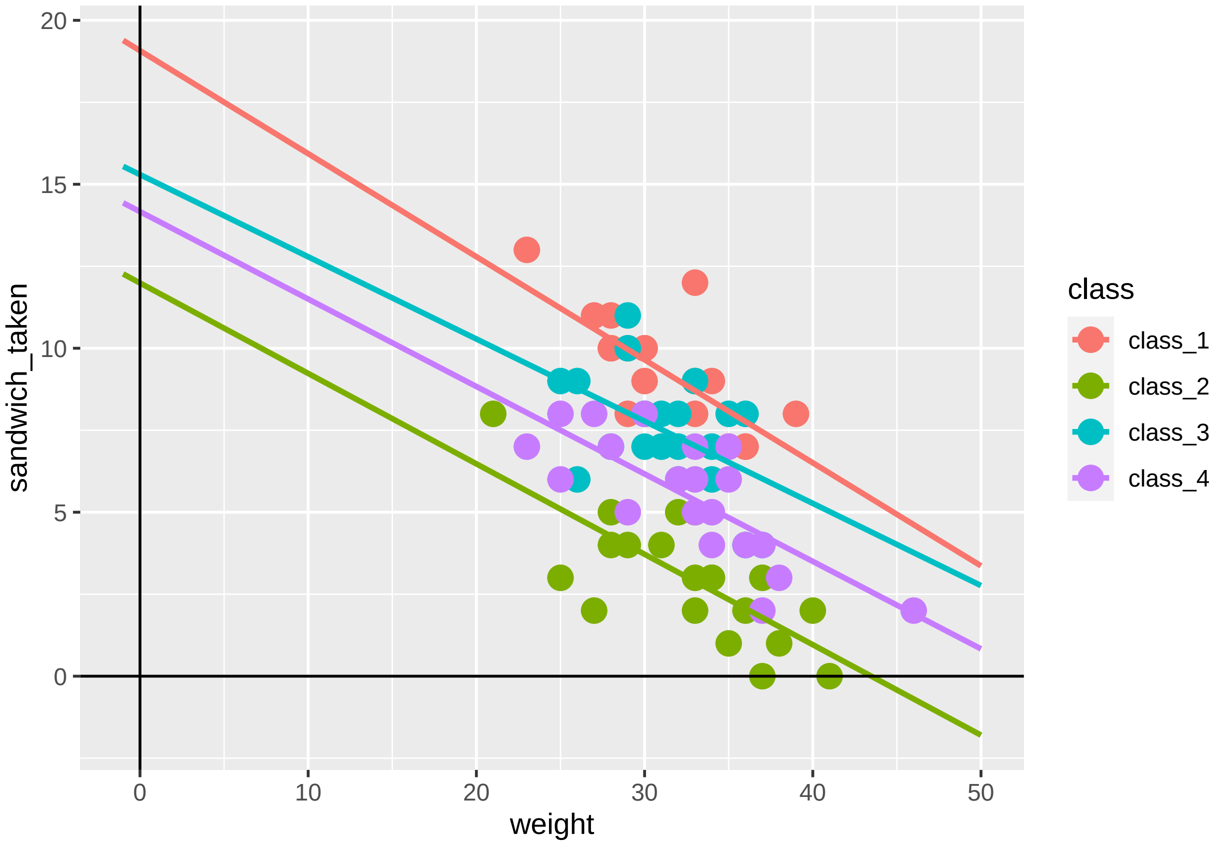
Ezert azokat a modelleket, amik mind fix, mind random hatasokat tartalmaznak, **kevert modellekenek (mixed models)** nevezzuk.

3.3.1 A random hatasok elofordulasi fajtai

Altalanossagban a random hatasok ket modon lehetnek hatassal a **kimeneti valtozora**. Az egyik hogy **direk hatast fejtenek ki ra (random intercept)**, a masik hogy a **fix hatasok merteket es iranyat befolyasoljak (random slope)**.

random intercept, random slope nelkul: Lehetseges hogy a csoportok **(klaszterek) csak abban kulonboznek egymastol, hogy a kimeneti valtozon atlagosan milyen erteket vesznek fel, de a fix hatasok azonosak** a klaszterek kozott. Ez igaz a data_bully_int adatbazisra. Megfigyelhetjuk az abran, hogy a regresszios egyenbesek meredeksege (slope) nem kulonbozik az osztalyok kozott, ami arra utal hogy a testsuly hatasa ugyan akkora az egyes osztalyokban. Az osztalyok csak abban kulonboznek, hogy milyen "magasan" vannak a regresszios egyenesek, vagyis abban, hogy a regresszios egyenesek milyen erteknel metszik az Y tengelyt. Ez lathato az alabbi abran 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 vizsgaljuk meg a masik adatbazist (**data_bully_slope**). Ahogy fent emlitettuk, ebben az adatbazisban azt szimulaltuk, hogy az osztalynak nem csak az elvett szendvicsek szamara van hatasa, hanem **a testsuly hatasa is kulonbozik az osztalyok kozott**.

Az abran jol latszik, az osztalyok nem csak abban kulonboznek, hogy a hozzajuk tartozo regresszios egyenes hol metszi az Y tengelyt, de **a regresszios egyenesek meredeksege** is kulonbozo.

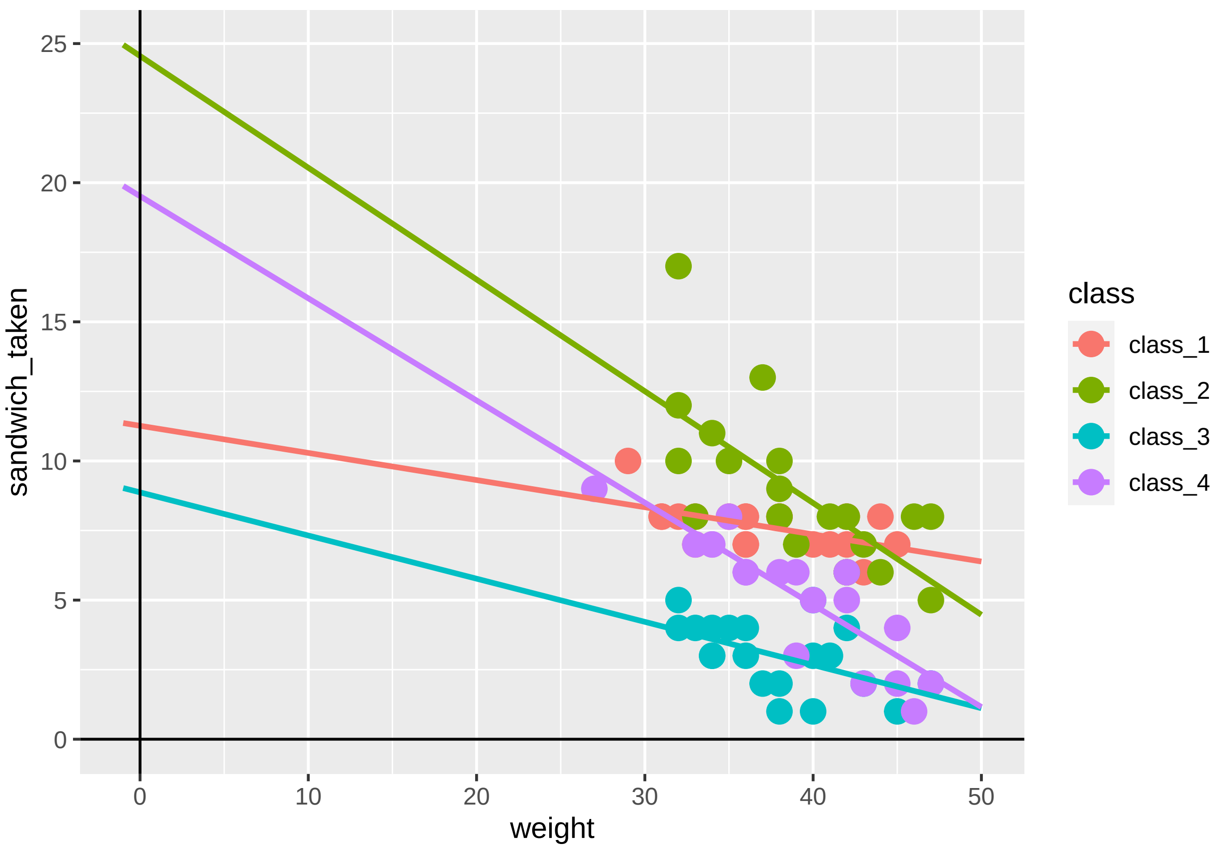
Peldaul a class_1-ben a testsuly hatas elhanyagolhatonak tunik abbol a szempontbol hogy kitol mennyi szendvicset vesznek el, mig a class_2-ben es class_4-ben a testsuly hatasa szamottevo.

```
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 Kevert modellek felepitese az R-ben

Az alabbi pelda bemutatja, hogy hogyan lehet a random hatasokat beepiteni a modellekbe.

Harom modellt fogunk epiteni. Eloszor egy szimpla fix hatasokat tartalmazo modellt, majd egy **random intercept modelt**, es egy **random slope modelt**.

A fenti abra alapjan arra lehet kovetkeztetni hogy a **data_bully_slope** adatbázisban az iskolai osztaly egy olyan random hatas, ami mind a regresszios egyenes intercept-jet, mind a meredekseget (slope) befolyasolja. Ezert normalis esetben csak a random slope modellt illesztenenk. A tobbi modell csak demonstracios celbol epitjuk, hogy osszehasonlitsuk azok formulait es bejoslo erejet a random slope modellevel.

Eloszor epitunk egy egyszeru regresszios modellt, melyben egyetlen fix hatas prediktor van: weight. Ezt a modellt a mod_fixed objektumba mnetjuk.

egyszeru regresszios model (csak fix hatas)

```
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 formulaja nagyon hasonlo a csak fix hatast tartalmazo modellekehez, de az lm() funcio helyett az lmer() funkciot hasznaljuk.

A random intercept random hatast **a "+ (1|class)" hozzadadasaval** tehetjuk a modellbe.

Ez gyakorlatilag azt jelenti, hogy megengedjuk a modellnek hogy **kulon regresszios egyenest illesszen minden klaszterre** (a mi esetunkben minden iskolai osztalyra), de azt meghatarozzuk, hogy **minden regresszios egyenesnek ugyan olyan legyen a meredeksege**.

Ezt normalis esetben akkor tennenk, ha azt gyanitanank hogy az osztalyok kozott nincs lenyegi elteres a fix hatasokban, csak a kimeneti valtozo atlagos szintjeben. Ez a modell jol illeszkedne a data_bully_int adatbazisra, de a fenti abra alapjan azt varjuk hogy a data_bully_slope adatbazisra kevesbe jol 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 modelnek a formulaja szinte teljesen megegyezik a random intercept modellevel, egyedul abban kulonbozik, hogy a random hatasrol szolo reszben "+ (1|class)" helyett "+ (weight|class)" szerepel. Ez arra utal, hogy a class random hatas nem csak az interceptre, hanem a weight prediktor hatasara is kiterjed.

Ezzel megengedjuk a modellunknek, hogy **kulon regresszios egyenest illesszen minden klaszterre**, es hogy azoknak **mind** az Y tengellyel valo metszespontja **(intercept), mind a meredeksege (slope) kulonbozhet**.

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

3.5 Melyik modell reprezentalja legjobban a valosagot?

Hogyan dontjuk el hogy **melyik modellt hasznaljuk?** Ahogy korabban is lathattuk, a modellvalasztasnal mindig **az elmeletileg leginkabb megalapozott** modellt erdemes valasztani. Ha van okunk feltetelezni hogy egy hatas kulonbozo lesz a kulonbozo klaszterekben, akkor hasznaljuk a random slope modellt. Ha elmeleti alapon inkabb ugy iteljuk, hogy a fix hatasok valoszinuleg allandoak a csoportok kozott, illesszunk random intercept modellt.

Ennek ellenere van olyan eset, **amikor elmeletileg mindket eshetoseg elkepzelheto**. Ilyen esetben hagyatkozhatunk a **vizualizációra** es a **modellilleszkedesi mutatokra**, hogy eldontsuk, melyik modellt erdemesebb hasznalni.

3.5.1 Random hatasok vizualizacioja

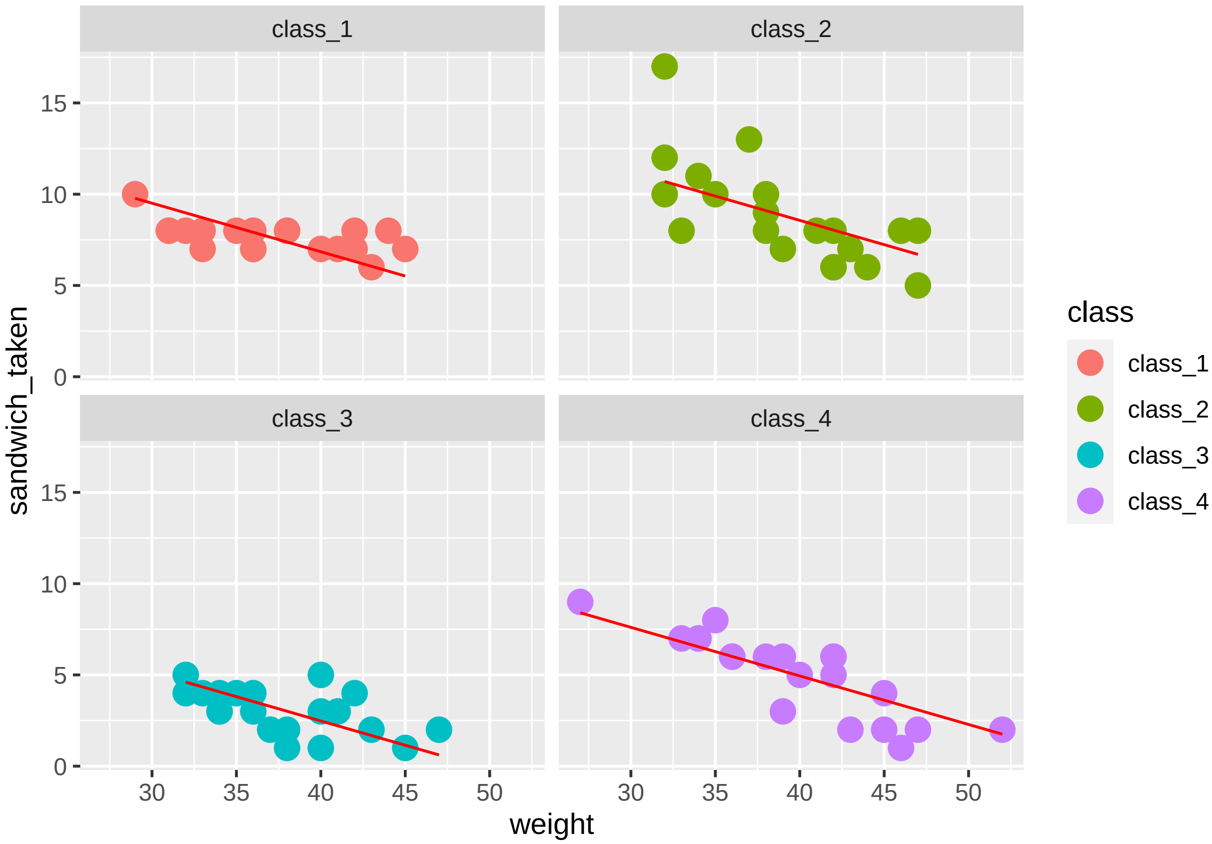
A random hatasok exploracioja eseten a vizualizacio kulcsszerept tolt be.

Eloszor erdemes elmentenunk az intercept es a slope modellek altal **bejosolt ertekeket uj valtozokba** (alabb a pred_int es pred_slope valtozokba mentjuk ezeket). Az eredeti adatbazisbol szarmazo predikciokat a predict() funkcioval nyerhetjuk ki.

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

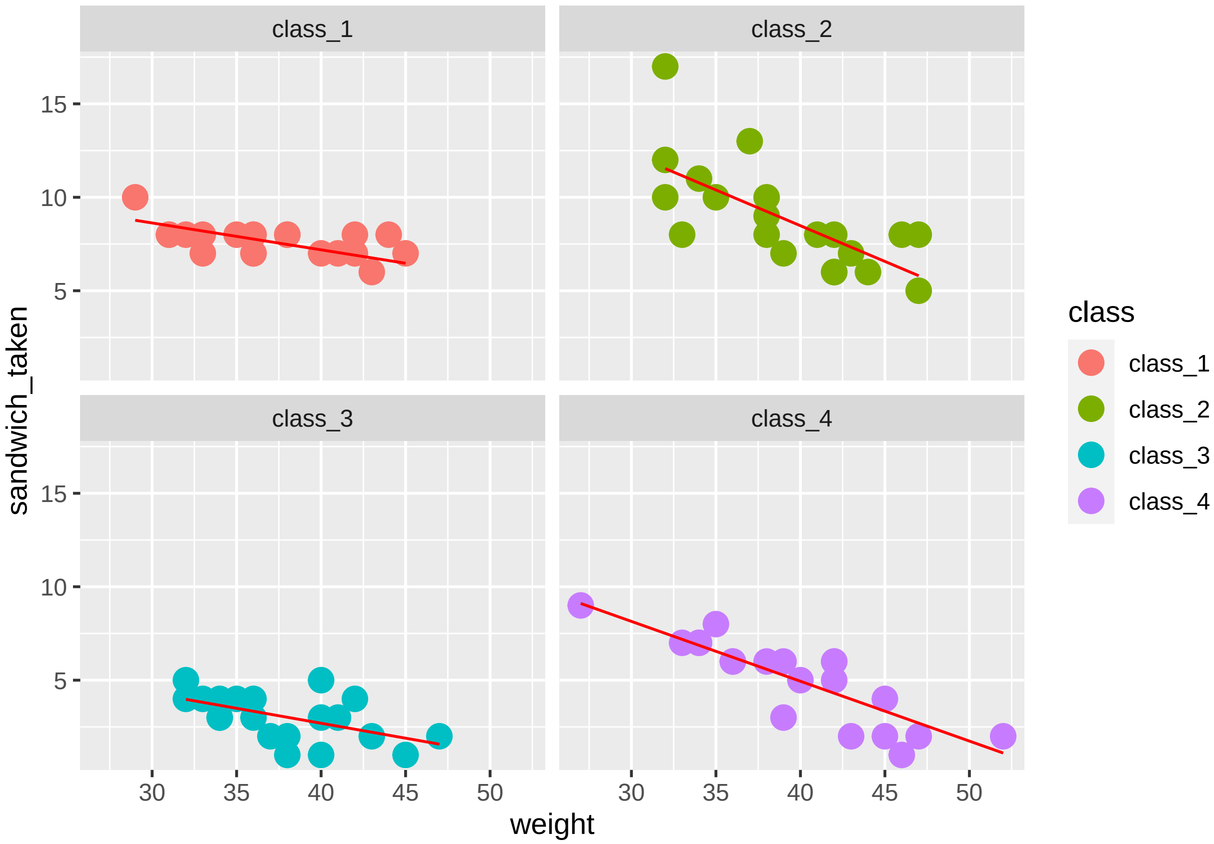
Igy vizualizaljuk a random **intercept modell** predikciojat:

```
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** predikciojat:

```
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 abrakon azt kell megneznunk, hogy a pontok mintazata kelloen kulonbozo-e a csoportok kozott hogy arra engedjen kovetkeztetni hogy csoportonkent kulonbozik a fix hatas.

Mivel a modellek altal generalt regresszios egyeneseket is tartalmazzak az abrak, megtehetjuk hogy megvizsgaljuk mi a hatasa annak, hogy megengedtuk a random slope modellben a regresszios egyenes meredeksegene valtozasat a random intercept modellhez kepest, ahol ez nincs megengedve. Az illeszkedes (szinte) mindig jobb lesz a random slope modellben. Ez szuksegszeru, hiszen a modell flexibilisebb, tobb szabadsaga van, ezert kozelebb tud helyezkedni a pontokhoz minden csoportban. De **ha az illeszkedesbeli kulonbseg a ket modell kozott nem szamottevo, biztonsagosabb a random intercept modellnel maradni, hogy elkeruljuk a tulillesztest**, hacsak az elmelet nem tamogatja egyertelmuen a random slope modellt.

3.5.2 Rezidualis hiba osszehasonlitasa (ezt nem hasznaljuk a gyakorlatban)

Talan elso ranezesre csabitonak tunhet, hogy egyszeruen arra hagyatkozzunk a modellvalasztas soran, hogy melyik modell produkalta a legkevesebb rezidualis hibat.

Ha osszehasonlitjuk a harom modell **rezidualis hibajat** (residual sum of squares - RSS), lathatjuk hogy a csak fix hatast tartalmazo modell hasznalatakor marad a legtobb hiba, a random intercept modell a masodik, es **a legkevesebb hibat a random slope modell eseten talaljuk**.

```
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 koplexitasa es ezzel **flexibilitasa egyre nott**, es errol tudjuk, hogy csokkenti a hibat azon az adatbazison amin a modellt epitettuk, de a flexibilitas novelese miatt ez **tulilleszteshez** vezethet, ami uj adatokon rosszabb bejoslasi hatekonysaghoz vezet. Ezert a nyers rezidualis hiba osszehasonlitas helyett olyan modell-illeszkedesi mutatohoz kell fordulnunk, amik korrigalva vannak a flexibilitasara (ezt ugy is mondhatjuk hogy a modell parameterek szamara.)

3.5.3 conditional AIC

Az egyik olyen 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 figyelmeztetest kapunk: 'refitting model(s) with ML (instead of REML)'. ez azert 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 becsleseket hasznalo modellek altalaban nagyon hasolnoak egymashoz, ezert ezt a figyelmeztetest legtobbszor figyelmen kivul hagyhato.

```
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 meghatarozzuk, egy **kevert linearis modellt illesztettunk**. A kevert modellben az elvett szendvicsek szamat mint **kimeneti valtozot** a testsullyal mint **fix hatasu prediktorral** josoltuk 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 meghataroztuk, a ket modellt **osszehasonlitottuk a cAIC** modellilleszkedeis mutatojuk alapjan, es ez alapjan hataroztuk meg, melyik lesz a vegso bejoslo modellunk."

A kovetkezo funkciokkal kapnank meg a kutatasi jelenteshez 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 ajanlasa alapjan. Ez az R^2 mutato specialis fajtaja, ami azt mutatja meg, hogy mekkora a modell fix hatasu prediktorai altal 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 szignifikansan 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 szignifikansan kulonbozik a null modelltol bejoslo hatekonysag tekinteteben.

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

Hivatkozas: Nakagawa, S., Johnson, P.C.D., Schielzeth, H. (2017) The coefficient of determination R2 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 eredmenyek reszben igy irhatjuk le az eredmenyeket:

"A random slope modell jobb modell-illeszkedeshez vezetett mint a random intercept modell mind a likelihood ratio test (X2 = 6.06, df = , p = .048) mind a cAIC alapjan (cAIC intercept = 294.4, cAIC slope = 285.64). Ezert az alabbiakban a random slope modell eredmenyeit kozoljuk.

A kevert linearis modell szignifikansan jobb volt mint a null modell. A modellben a fix hatasu prediktorok az elvett szendvicsek varianciajanak 14.7%-at magyaraztak meg (R^2 = 0.15 [95% CI = 0.04, 0.3])."

3.6.3 Regresszios egyutthatok kozlese

Ezen felul a prediktorokhoz tartozo regresszios egyutthatokrol is kozolnunk kell az eredmenyeket. Ezt a korabbiakhoz hasonloan egy tablazatban szoktuk megtenni, ami minden prediktorra kulon sorban kozli az adatokat. (Itt csak egy fix hatasu perdiktor van, szoval csak ket sor lesz a tablazatban, egy az intercept-nek es egy a testsuly prediktornak.)

A vegso tablazat valahogy igy nez 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 tablazat egyes elemeit itt talalhatod meg:

Regresszios egyutthatok es a hozzajuk tartozo p-ertekek a summary() fuggvennyel kaphatok meg (csak akkor fog p-erteket kiadni a summary fuggveny a kevert modellekre, ha az lmerTest package be van toltve)

```
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 regresszios egyutthatokhoz tartozo konfidencia intervallumok: (ez a funkcio sokaig fut, mert sok iteraciot vegez a kiszamitashoz)

```
confint(mod_rnd_slope)
```

A standardizalt beta ertekeket pedig a kot elejen levo stdCoef.merMod() sajat funkcioval 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 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).

5 Adatmenedzsment

5.1 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 ágya milyen kozel van az ablakhoz, es hogy mennyi napfeny eri oket a felepules idoszak alatt. Ez a kutatas azt az elmeletet teszteli, 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 ágya 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/

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

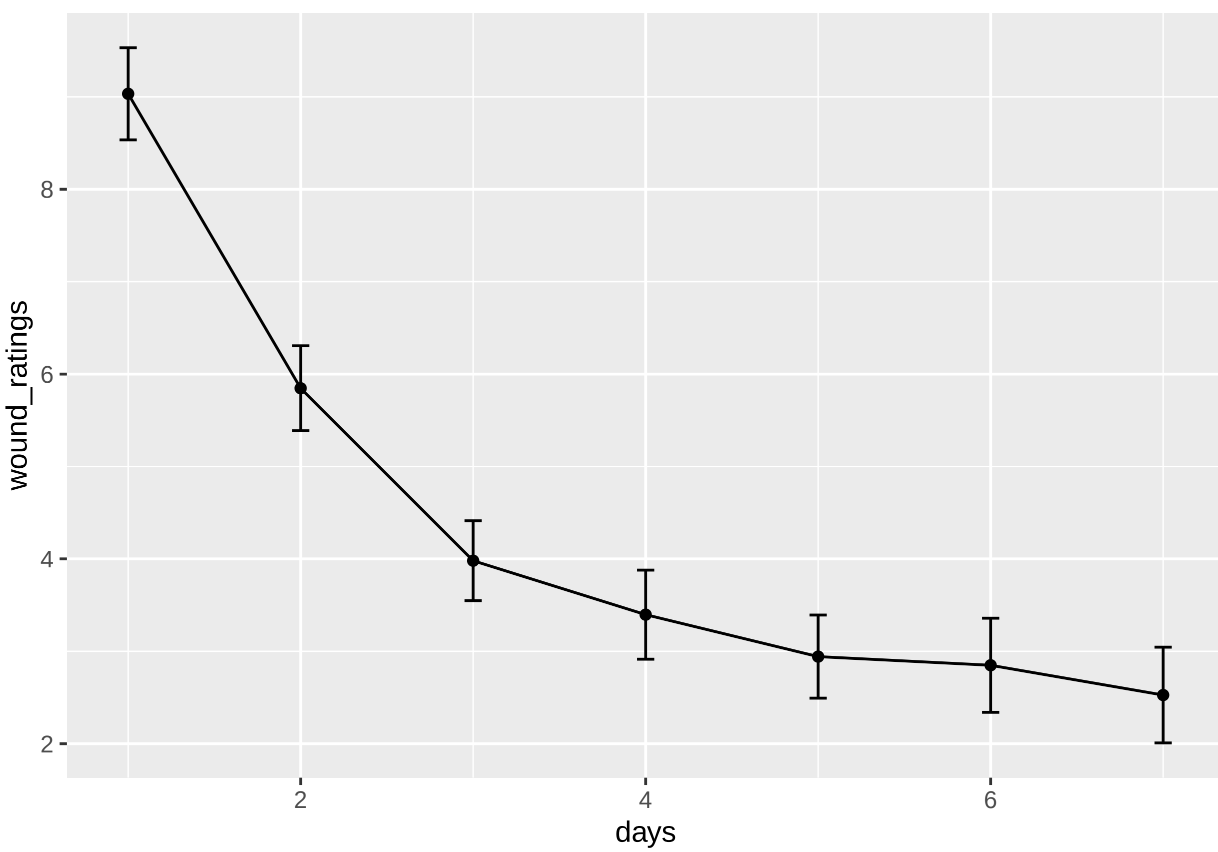
5.2 Adatellenorzes és leíró statisztikák

Vizsgaljuk meg az adattablat a View(), describe(), es table() fukciok segitsegevel. Fontos, hogy az adattablaban jelenleg minden adat amit egy adott szemelytol gyujtottek **egy sorban talalhato**.

```
View(data_wound)

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

Vizualizalhatjuk is az adatokat. (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 vizualizaljuk. Lathato hogy a seb allapot ertek egyre csokken ahogy telnek a napok.)

```
# 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()
```



6 Ismetelt meresek eredmenyenek vizsgalata kevert linearis modellekkel

6.1 Klaszteres szerkezet keresese az adatokban

Vizsgaljuk meg a tovabbiakban a sebgyogyulasra vonatkozo ismetelt meresek eredmenyeit! Ehhez eloszor mentsuk el az adatokhoz tartozo **valtozoneveket egy repeated_variables** elnevezesu objektumba, hogy kesobb konnyen hivatkozhassunk rajuk az alltalunk irt fuggvenyekben!

A valtozok kozotti korrelaciot a cor() fuggvennyel tudjuk megvizsgalni. Figyeljuk meg, hogy az ismetelt meresek adatponjai kozott **eros korrelacio fedezheto fel**, azaz az egyes seb-allapot ertekekre vonatkozo megfigyelesek **nem fuggetlenek egymastol**. Ez varhato is, hiszen a seb-allapot ertek, az eredeti bemetszes merete es a seb gyogyulasanak uteme mind fuggenek a vizsgalt betegtol. Adataink tehat csoportokba tomorulnek (klaszterekbe), hasonloan a korabbi peldankhoz. Azonban mig ott osztaly szerinti klaszterek fordultak elo, itt most a klaszterek maguk a resztvevok.

```
# 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 adattabla atformazasa szelesbol hosszu 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 adatgyüjtés 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 értekek 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 adabázisban. 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 lerövidíthetjuk, 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 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
```

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 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)
```

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 modellt 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 Messaget) 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 nevezünk 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_lo
```

```
## 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 = lmerCo
```

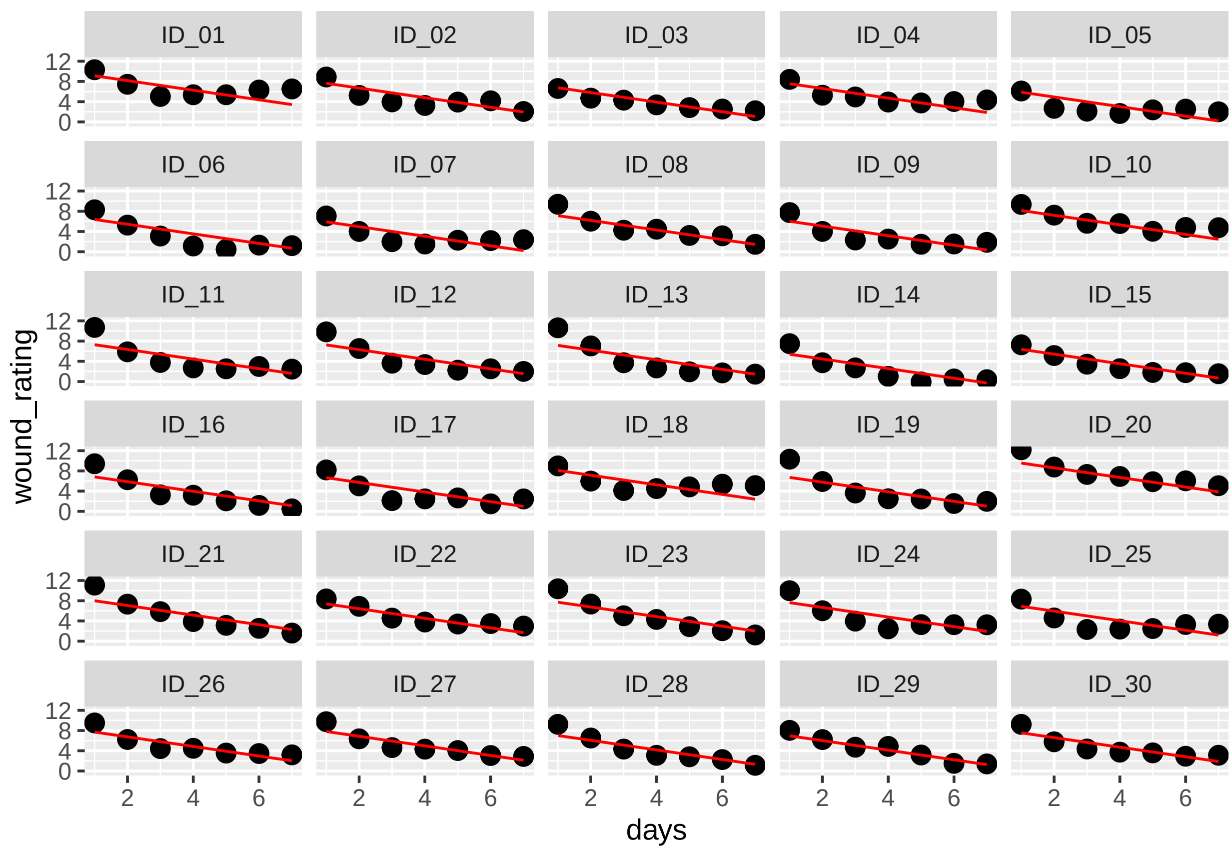
6.4 Az eltérő modellek összehasonlítása

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

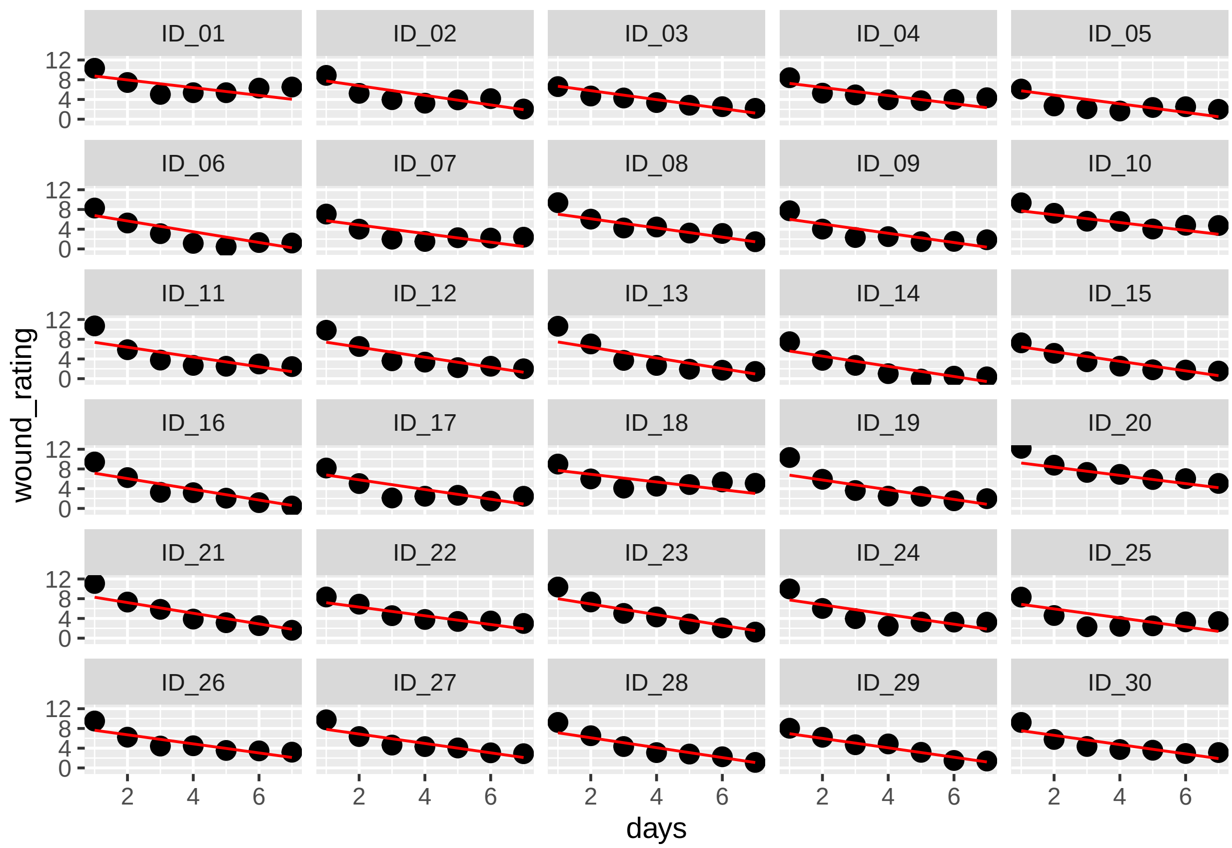
A könnyebb összehasonlíthatóság kedvéért, vizualizáljuk adatainkat! Ehhez először mentsük el a modellek predikcióit egy-egy új változóban, majd ábrázolhatjuk az egyes bejósolt értékeket a valódi értékek függvényében, az egyes (random intercept és random slope) modellekre vonatkozó ábrákon külön-külön.

(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 = da
```

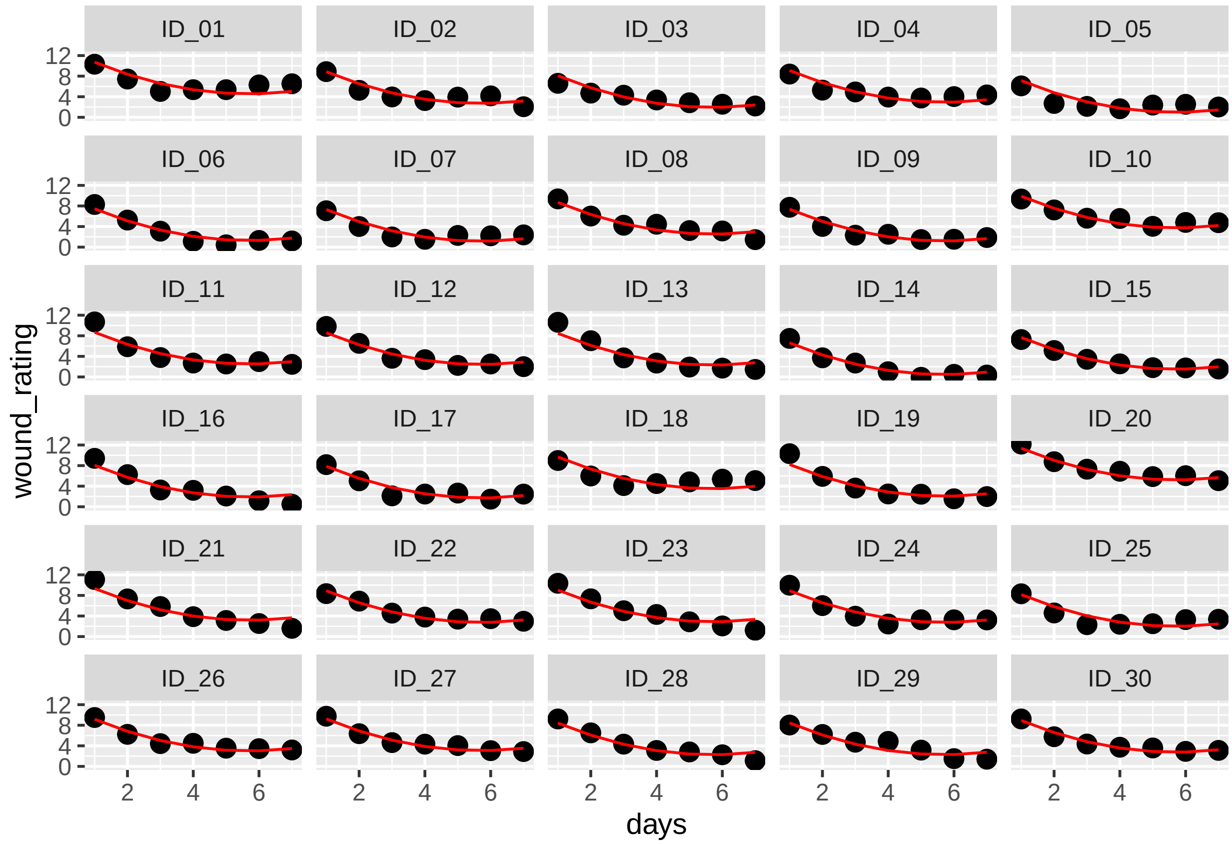
Mentsük a modell által bejósolt értékeket egy új dataframe-be, amely tartalmazza a korábbi bejósolt értékeket is!

```
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 modelldiagnoszika gykorlatban 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 +
```

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
```

```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
```

Gyakorlás

Olvassuk be a műtéti 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 0tól-10ig terjedő folytonos vizuális skálán.
- sex: a résztvevő bejelentett neme
- STAI_trait: A résztvevő State Trait Anxiety Inventroy-n elért pontszáma
- pain_cat: fájdalom katasztrofizálása
- cortisol_serum; cortisol_saliva: A kortizol egy a stress hatására előállított hormon. A kortizol szintet vérből 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://raw.githubusercontent.com/kekecsz/PSZB17-210-Data-analysis-seminar/master/seminar_09/home_sample_5.csv”.
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 eltellt 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?
