Kevert modellek - alapok

Zoltan Kekecs

17 November 2020

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\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 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 "zaklatas"?) teszunk fel kutatasi kerdeseket. Ez egy szimulalt adatbazis, vagyis nem valodi adatokat tartalmaz, de kepzeljuk el, hogy az adatok a kovetkezo kutatasbol szarmazank: Ebben a kutatasban 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* (néhány osztályban a testsúly hatása nagyobb mint másokban).

```
# load data
data_bully_int = read_csv("https://raw.githubusercontent.com/kekecsz/PSYP13_Data_analysis_class-2018/mas
```

```
## Parsed with column specification:
## cols(
##   sandwich_taken = col_double(),
##   weight = col_double(),
##   class = col_character()
## )
```

```
# asign class as a grouping factor
data_bully_int %>% mutate(class = factor(class))
```

```
## # A tibble: 80 x 3
##    sandwich_taken weight class
##             <dbl>  <dbl> <fct>
##  1              8     30 class_1
##  2             10     28 class_1
##  3             11     27 class_1
##  4             12     33 class_1
##  5              7     36 class_1
##  6             10     30 class_1
##  7              9     30 class_1
##  8             13     23 class_1
##  9              8     33 class_1
## 10              7     36 class_1
## # ... with 70 more rows
```

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

```
## Parsed with column specification:
## cols(
##   sandwich_taken = col_double(),
##   weight = col_double(),
##   class = col_character()
## )
```

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

```
## # A tibble: 80 x 3
##    sandwich_taken weight class
##             <dbl>  <dbl> <fct>
##  1              8     44 class_1
```

```
## 2               8     38 class_1
## 3               7     40 class_1
## 4               8     42 class_1
## 5               8     36 class_1
## 6              10     29 class_1
## 7               6     43 class_1
## 8               8     35 class_1
## 9               8     31 class_1
## 10              7     36 class_1
## # ... with 70 more rows
```

2.4 Adatellenorzes es adattisztitas

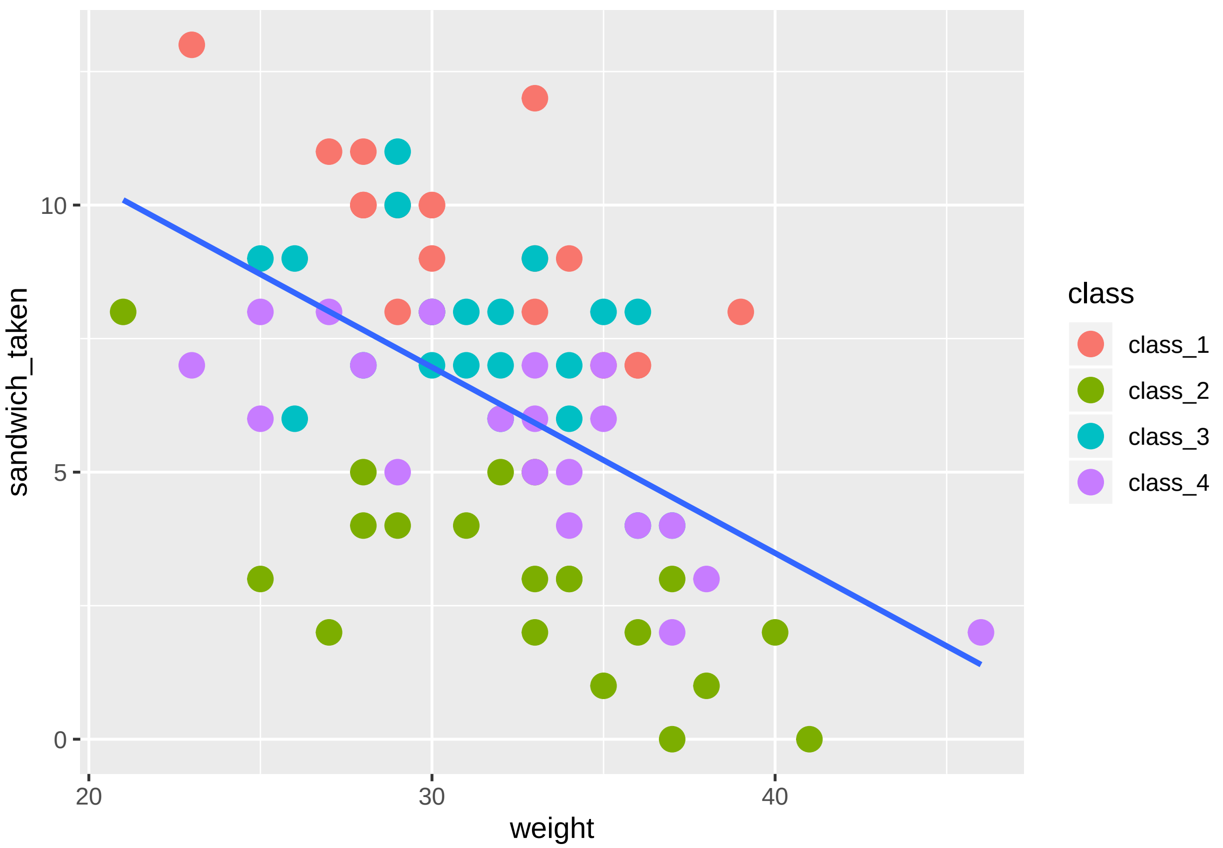
Ahogy mindig, eloszor kezdd az adatok ellenorzesevel es az esetleges adattisztitassal. Ehhez hasznalhatod a View(), summary(), es describe() funkciokat, es a ggplot() funkciot vizualizalashoz.

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")
```



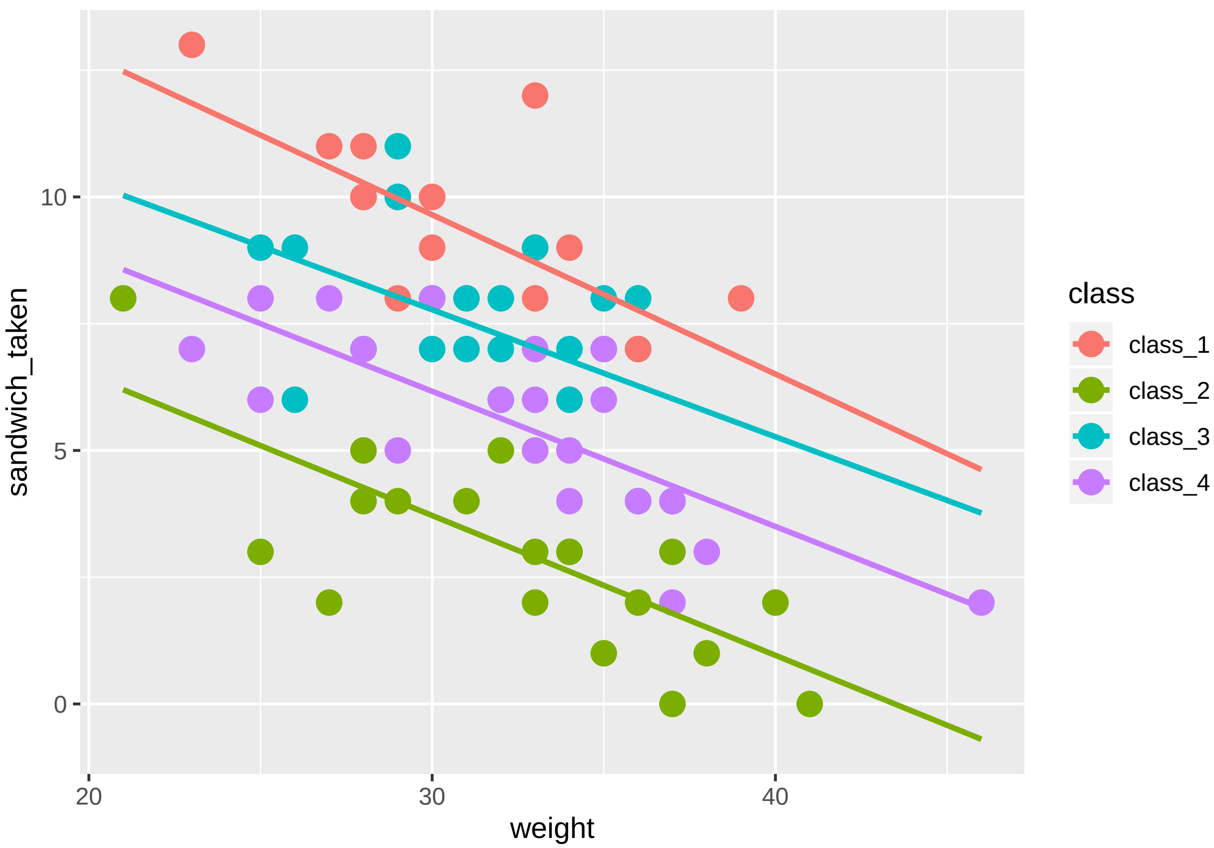
A pontok szine az abran azt mutatja, a diak **melyik iskolai osztalyba** jar (class_1, class_2, class_3, vagy calss_4). Ha jobban megnezzuk, ugy tunik, hogy az azonos szinu pontok egymashoz kozel helyezkednek el az abran, nem pedig random modon elszorva, ami arra utal, hogy az adatpontok nem teljesen 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 kapunk 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 szeretnénk bejóslásra használni egy új iskolában, 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 egyeletben, hiszen akkor mas iskolakban nem tudnak felhasznalni az egyeletet.

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.

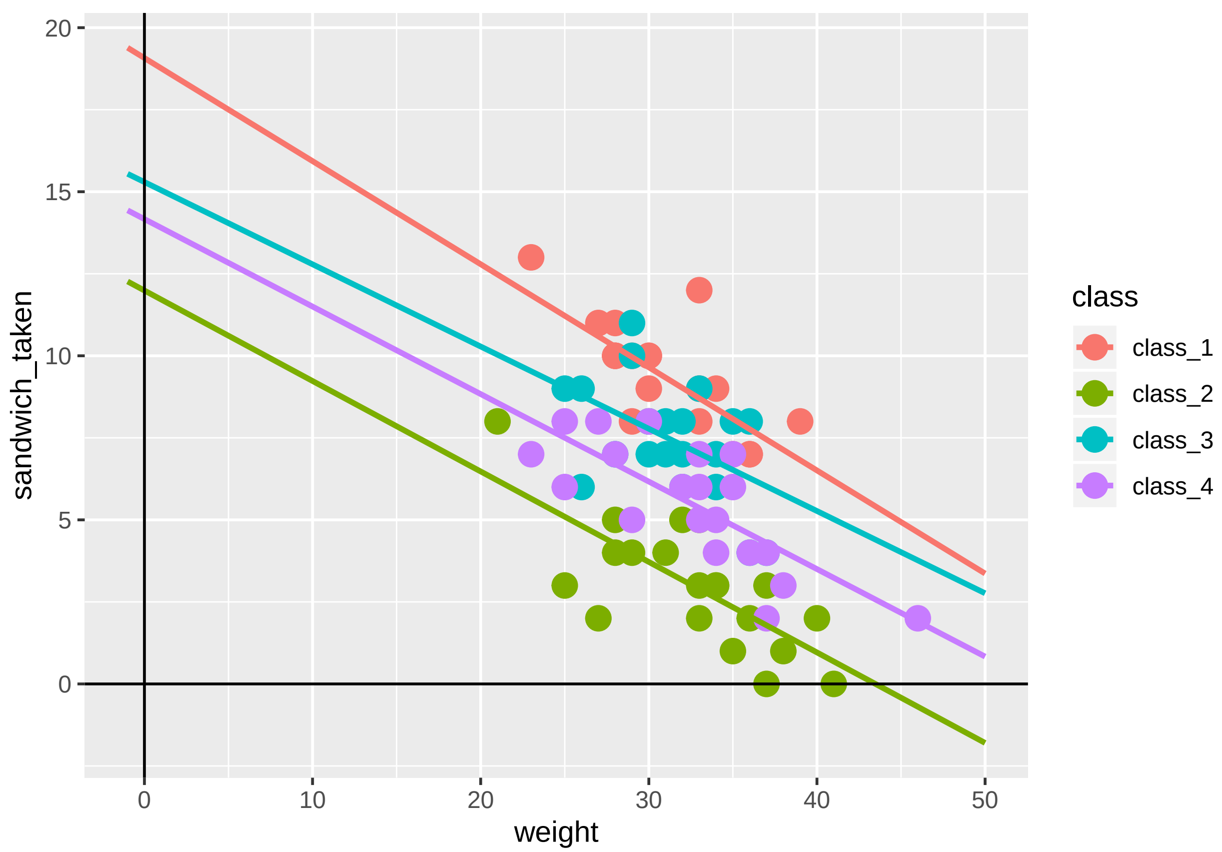
Ezert azokat a modelleket, amik mind fix, mind random hatasokat tartalmaznak, **kevert modellekenek (mixed models)** nevezzuk.

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**.

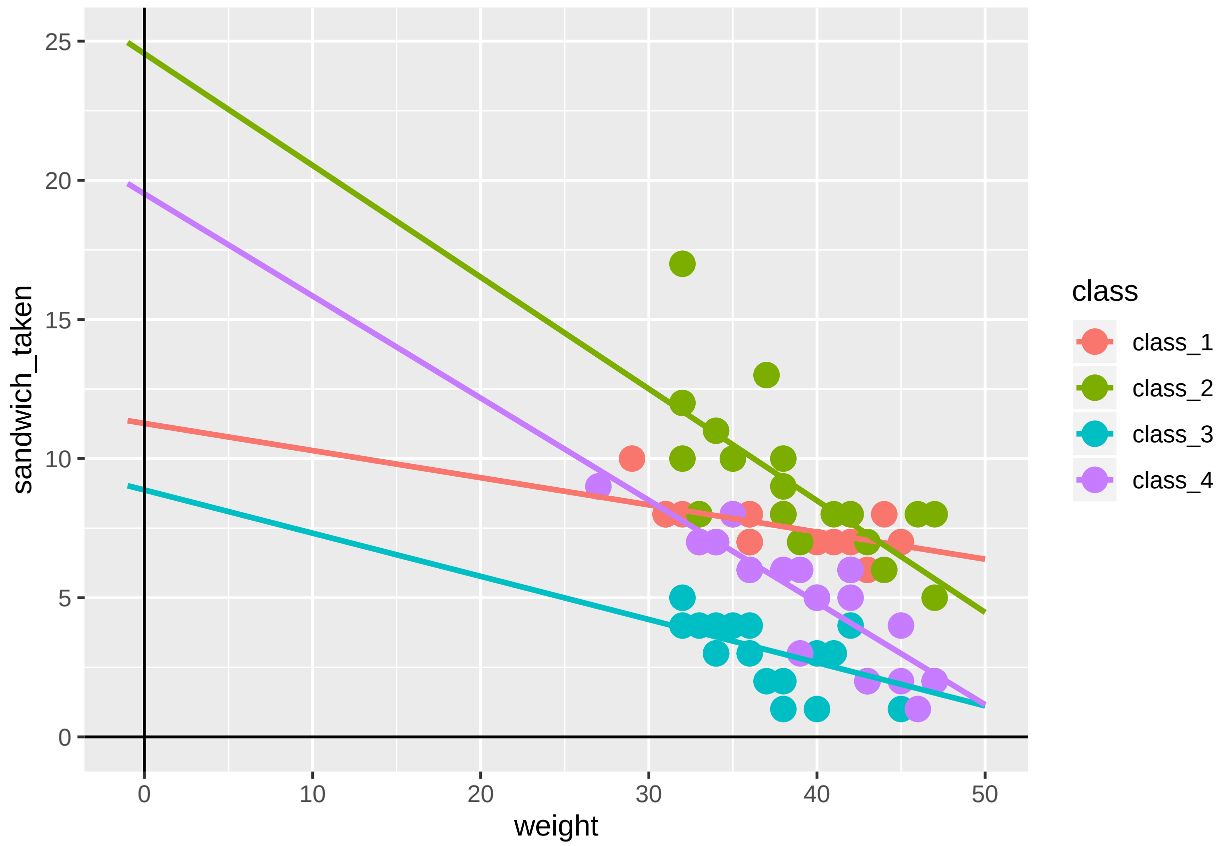
Az abran jol latszik, az osztalyok nem csak abban kulonboznek, hogy a hozzajuk tartozo regresszios egyenes hol metszi az Y tengelyt, de **a regresszios egyenesek meredeksege** is kulonbozo.

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
```

```
## 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 meredkseget (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)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00317298 (tol = 0.002, component 1)
```

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

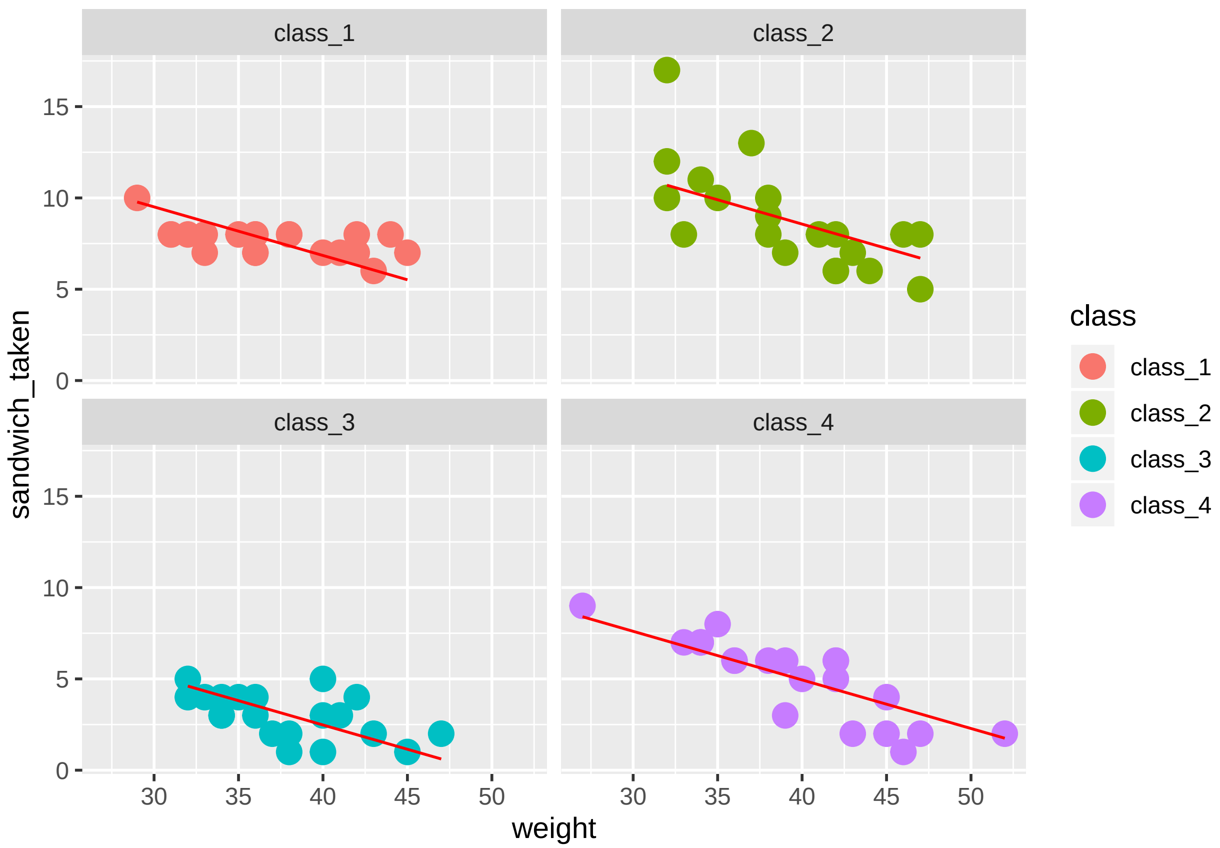
A random hatasok exploracioja eseten a vizualizacio kulcsszerept tolt be.

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))
```

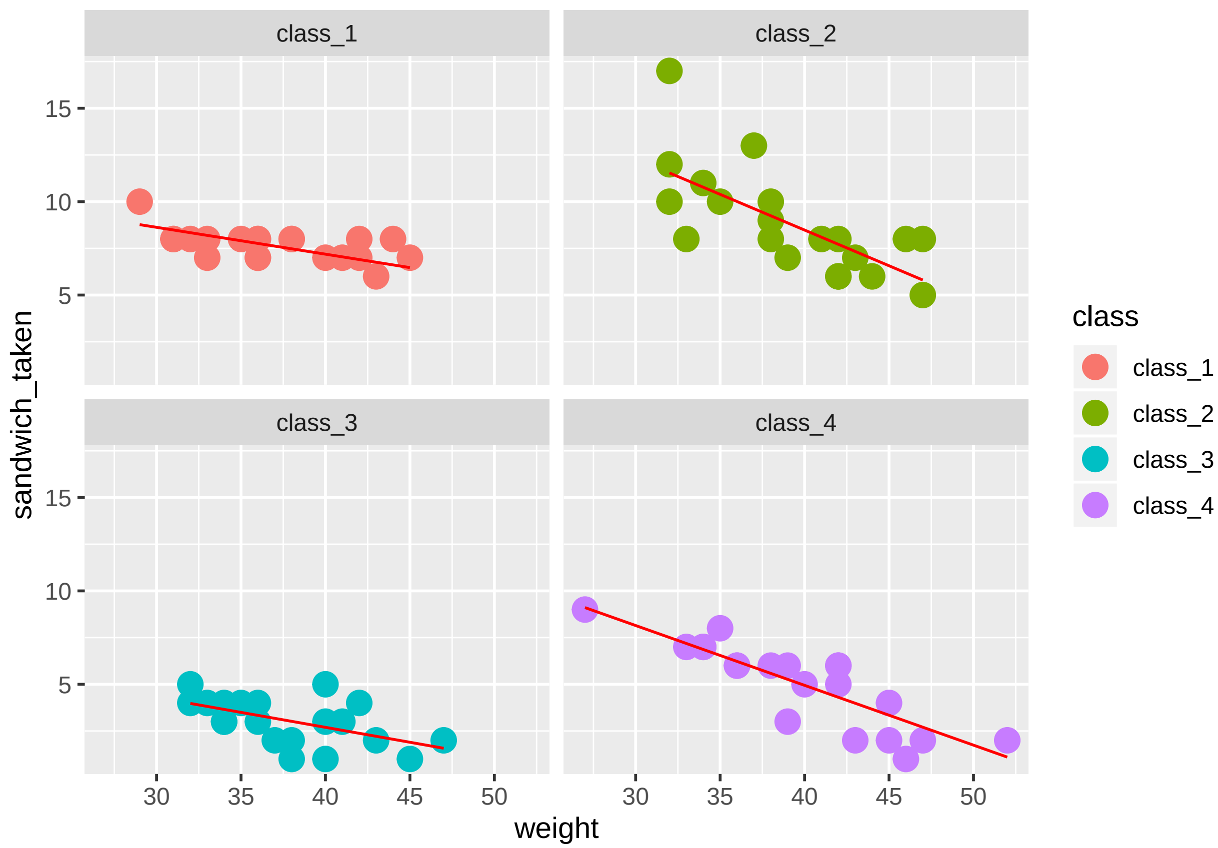
Igy vizualizaljuk a random **intercept modell** predikciojat:

```
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 egszeruen 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
```

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

```
## [1] 132.2322
```

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.

```r
AIC(mod_fixed)
```

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

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

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

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

```
## [1] 285.6445
```

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.

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

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

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

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

```
## [1] 285.6445
```

anova:

```r
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)
##               Df    AIC    BIC  logLik deviance  Chisq Chi 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    0.304    0.036
## 2 weight 0.147    0.304    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,] 0.147134 0.8331627
```

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    .021
## weight      -0.25    -0.41    -0.09    -0.42     .04
```

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
## -2.3403 -0.5630 -0.0076  0.3896  4.0511
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
```

```
##  class    (Intercept) 44.61736 6.6796
##           weight       0.01693 0.1301   -0.94
##  Residual              1.81799 1.3483
## Number of obs: 80, groups:  class, 4
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept) 15.88887    3.55290  2.95835   4.472   0.0215 *
## weight      -0.25155    0.07224  2.98048  -3.482   0.0404 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##        (Intr)
## weight -0.940
## convergence code: 0
## Model failed to converge with max|grad| = 0.00317298 (tol = 0.002, component 1)
```

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

```r
confint(mod_rnd_slope)
```

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

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

```
##               stdcoef     stdse
## (Intercept)  0.000000 0.0000000
## weight      -0.422144 0.1212255
```