Eötvös Loránd Tudományegyetem
Természettudományi Kar

Varga Eszter

Két-fázisú regressziós modell és számítógépes használata

Szakdolgozat

Matematika BSc, elemző szakirány

Témavezető

Prőhle Tamás

Valószínűségelméleti és Statisztika Tanszék

Budapest, 2020.

NYILATKOZAT

Név: Varga Eszter

ELTE Természettudományi Kar, szak: Matematika BSc

NEPTUN azonosító: MH2ZJV

Szakdolgozat címe:
Két-fázisú regressziós modell és számítógépes használata

A **szakdolgozat** szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés nélkül nem használtam fel.

Budapest, 2020.12.10.

Varga Eszter

a hallgató aláírása

Köszönetnyilvánítás

Elsősorban szeretnék köszönetet mondani konzulensemnek, Prőhle Tamásnak, aki a téma kiválasztásától kezdve egészen a végső dolgozat megalkotásáig segítségemre állt, és tudásának és munkájának köszönhetően jobban megszerettem a matematika ezen ágát.

Még szeretnék köszönetet mondani családomnak, legfőképpen a szüleimnek és nővéremnek, Varga Ágnesnek, illetve a legjobb barátnőimnek, Revóczi Orsolyának és Répás Ritának, akik az egyetemi éveim alatt lelkesen támogattak, illetve ebben a nehéz időszakban is végig mellettem álltak.

Tartalomjegyzék

1. Bevezetés

Az elmúlt évek tendenciája azt mutatja, hogy az érdeklődés folyamatosan növekszik olyan statisztikai módszer iránt, amely a vizsgált intervallumot szakaszaira bontja és a különböző szakaszokon vizsgálja az adatokat.

Ez a módszer az úgynevezett *többfázisú regresszió*. Bár ennek a módszernek a *spline* regresszió egy igen speciális esete, mi mégis csak azon oldalról vizsgálódunk, amikor a regressziós függvény valamilyen ismeretlen csomópontban szakaszonként lineáris függvény.

Az alapmodell nem feltétlen folytonos esetben vizsgálódik, mi mégis megköveteljük ennek a létét. Elképzelésünk szerint a regressziós problémának a megoldásához a regressziós függvényt zárt paraméteres formában kell felírni.

Nézzünk meg egyszerű egy példát. Bontsuk fel a függvényt mindösszesen csak két szakaszra. Legyen a töréspont ψ-ben illetve a függvény értéke 0-ban B_0. Legyen a függvény meredeksége ψ-től balra B_1 jobbra pedig B_2.

Ekkor a keresett $f(u) = f_{B_0,B_1,B_2}(u)$ függvény a

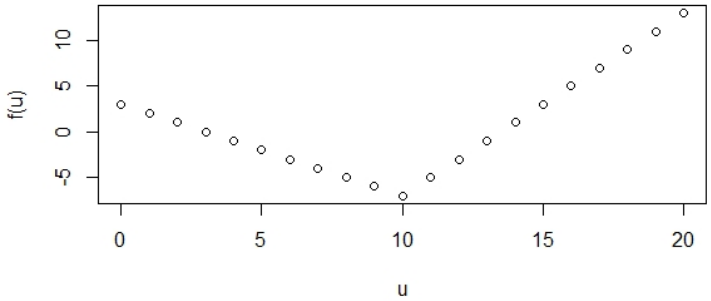
$$\begin{aligned} \beta_0 &= B_0 \\ \beta_1 &= B_1 \\ \beta_2 &= B_2 - B_1 \end{aligned}$$

paraméterekkel a

$$f(z) = \beta_0 + \beta_1 \cdot z + \beta_2 \cdot (z - \psi) \cdot \chi_\psi(z) \qquad (a)$$

formába írható, ahol a $\chi_c(x)$ a $\chi_c(x) = Ind(x > c)$ függvényt jelöli, melynek 1 az értéke, ha $x > c$ és 0, ha nem.

```
## konstans és a két meredegség
B0  <-  3
B1  <- -1
B2  <-  2
psi <- 10  ## törési pont

chi <- function(x) ifelse(x>psi,1,0)

b0 <- B0
b1 <- B1
b2 <- B2-B1

f <- function(x) b0+b1*x+b2*(x-psi)*chi(x)

u <- 0:20
y_alt <- b0+(b1+b2*chi(u))*z-chi(u)*psi*b2

sum((f(u)-y_alt)^2) # ugyanaz

f(0) ## B0=3, azaz átmetszés jó
diff(f(u)) ## B1,...B1,B2,...,B2 a meredekségek jók

c(B1,B2)
plot(u,f(u))
```



I. ÁBRA. ***Töréspont ábrázolása***

1.1. Illesztési algoritmus

Vizsgáljuk meg az előző példa esetét. [1]

Az illesztés módja iteratív. Megengedjük, hogy a felírt (a) függvény esetében a ψ_k ne az optimális töréspont legyen.

Ekkor ezt az úgynevezett hibát igyekszünk korrigálni a törésponttól jobbra, egy nem nulla konstans tag hozzáadásával.

Az aktuális ψ_k, és a β_1, β_2, γ paraméterek becsült értéke alapján, javíthatjuk a ψ_k értékét:

$$\psi_{k+1} = \psi_k - \widehat{\gamma} \Big/ \widehat{\beta}_2$$

Ha az eljárás konvergál, akkor a $\widehat{\gamma} \approx 0$. [2]

1.2. Többfázisú regresszió

Tegyük fel, hogy létezik valamilyen regressziós kapcsolat az x és az y valószínűségi változók között. Felírhatjuk a feltételes várhatóérték függvényét [3]

$$f(x;\theta,\tau) = \begin{cases} f_1(x;\theta_1), & x \leq \tau_1 \\ f_2(x;\theta_2), & \tau_1 < x \leq \tau_2 \\ \vdots & \vdots \\ f_D(x;\theta_D), & \tau_{D-1} < x \end{cases} \tag{1}$$

Ahol θ_i tartalmazza a j-edik fázishoz tartozó együtthatókat, $j = 1, \ldots, D$ és $\tau = (\tau_1, ..., \tau_{D-1})$. A D egészszám adja meg, hogy hány fázisú regresszióról van szó. Az egyes fázisokban a rendre $\theta_1, ..., \theta_n$ paraméterű $f_1, ..., f_D$ függvények érvényesek. A fázisoknak (az értelmezés rész intervallumainak) τ osztópontjai ismeretlenek. Ezeket a pontokat nevezzük **váltópontoknak** vagy **csatlakozási pontoknak**.

Modellt olyan esetekben érdemes használni, amikor:

a) Kevés fázisra bontható (azaz a D értéke relatív kicsi)

b) Ezeken az intervallumon a $\mathbb{E}[y|x]$ függvény viselkedése egyszerű - például egy lineáris vagy kvadratikus - paraméteres függvénnyel írható le

c) Az egyes fázisokon érvényes függvények közt viszonylag nagy a különbség.

Mint ahogy említettük (1) formula használható spline regresszió esetén is. Esetünkben a modell alkotása a sima regressziós kapcsolaton alapul, spline esetében viszont az egyes fázismodellek polinomiálisak és sokkal szigorúbb megkötések tartoznak az egyes váltópontokra.

1.3. Lineáris regresszió lineáris feltétel mellett

Keressük azt a β^* paramétert melyre:

$$\begin{aligned} R\beta = r \qquad & \text{egyenlőség teljesül és a} \\ (Y - X\beta)^T(Y - X\beta) \qquad & \text{hiba négyzetösszeg minimális.} \end{aligned}$$

Lagrange multiplikációs módszert felhasználva keressük meg a megoldást. A Lagrange függvény a következőképpen írható fel:

$$L(\lambda, \beta) = (Y - X\beta)^T(Y - X\beta) + 2\lambda^T(R\beta - r)$$

A fenti függvény β szerinti deriváltja a megfelelő β^* pontan 0-val egyenlő. Tehát

$$2(\beta^*)^T X^T X - 2Y^T X + 2\lambda^T R = 0$$

Ezt az egyenletet kell megoldani. Megoldás után átrendezzük és vesszük a transzponáltját. Ekkor a következő egyenletet kapjuk:

$$X^T X\beta^* + R^T\lambda = X^T Y$$

Tegyük fel, hogy létezik $X^T X$ inverze. Fejezzük ki β^* ot majd szorozzuk meg $(X^T X)^{-1}$ -el

$$\beta^* = (X^T X)^{-1} X^T Y - (X^T X)^{-1} R^T\lambda = \widehat{\beta} - (X^T X)^{-1} R^T\lambda$$

Mivel $R\beta^* = r$, ha fenti egyenlőséget balról R-rel szorozva

$$r = R\widehat{\beta} - R(X^T X)^{-1} R^T\lambda$$

Ebből az egyenletből rendezzük λ értékre, feltéve, hogy létezik a felhasznált inverz:

$$\lambda = (R(X^TX)^{-1}R^T)^{-1}(R\widehat{\beta} - r)$$

Helyettesítsük be a λ-ra kapott értéket a fentiekben kifejezett β^* egyenletébe

$$\beta^* = \hat{\beta} - (X^TX)^{-1}(R(X^TX)^{-1}R^T)^{-1}(R\hat{\beta} - r) \qquad (\star)$$

A $(\star)$ által kifejezett β^* érték lesz az, melyre az $X\beta^*$ a legkisebb négyzetes hibával közelíti Y vektort, az $R\beta^* = r$ feltétel mellett.

Vegyük észre, hogy a kapott becslés a feltétel nélküli $\widehat{\beta}$ becslés javítása annak az $R\widehat{\beta} - r$ hibának az alapján, amivel az egyenlőségi feltétel csorbul, ha a β értékét $\widehat{\beta}$-nak vesszük. Mint látható, a feltételi egyenlőségnek a – feltétel nélküli regressziós együtthatók mellett adódó – hibáját két szorzótényező módosítja. Ezek közül az első, az $R(X^TX)^{-1}R^T$ inverze, azaz az $R\widehat{\beta}$ varianciájával arányos tényező. A második, a $(X^TX)^{-1}R^T$ tényező pedig a $\widehat{\beta}$ és az $R\widehat{\beta}$ közti kovarianciával arányos. Tehát a $\widehat{\beta} - \beta$ előtti hosszú szorzó a legkisebb négyzetek modellje szerinti 'covariancia/variancia' szorzónak felel meg.

Megmutatjuk, hogyan tesztelhető egy, az együtthatókra megfogalmazott lineáris feltétel. De ehhez előbb elvégzünk néhány részletszámítást.

Írjuk fel a $\beta^* - \beta$ differenciát felhasználva a β^* előbbi képletét és azt, hogy $r = R\beta$.

$$\begin{aligned}
\beta^* - \beta &= \widehat{\beta} - \beta - (X^TX)^{-1}R^T \cdot \big(R(X^TX)R^T\big)^{-1} \cdot (R\widehat{\beta} - R\beta) \\
&= \widehat{\beta} - \beta - (X^TX)^{-1}R^T\big(R(X^TX)R^T\big)^{-1}R \cdot (\widehat{\beta} - \beta)
\end{aligned}$$

Jelölje P_R azt a mátrixot, amelyik a korábban vizsgált szorzó, R-el jobbról megszorozva:

$$P_R = (X^T X)^{-1} R^T \big(R(X^T X)^{-1} R^T\big)^{-1} R$$

A P_R jelölést felhasználva, és a $\beta^* - \beta$ differenciát balról X-el szorozva adódik:

$$X(\beta^* - \beta) = X(I - P_R)(\widehat{\beta} - \beta) = X(I - P_R)(X^T X)^{-1} X^T \varepsilon = (P - P_Q)\varepsilon \quad (2)$$

hogyha P_Q a

$$P_Q = X P_R (X^T X)^{-1} X^T = X(X^T X)^{-1} R^T \big(R(X^T X)^{-1} R^T\big)^{-1} R (X^T X)^{-1} X^T,$$

képlet szerinti, és

$$P = X(X^T X)^{-1} X^T$$

A modell szerint $Y = X\beta + \varepsilon$ és $P\beta = \beta$, ezért:

$$\widehat{\beta} = (X^T X)^{-1} X^T (X\beta + \varepsilon) = \beta + (X^T X)^{-1} X^T \varepsilon$$

Ebből $\widehat{\beta} - \beta = (X^T X)^{-1} X^T \varepsilon$, amit balról X-el megszorozva:

$$X(\widehat{\beta} - \beta) = X(X^T X)^{-1} X^T \varepsilon = P\varepsilon$$

Ezt és a (2) képletet felhasználva számoljuk ki az $X\widehat{\beta} - X\beta^*$ differenciát!

$$\begin{aligned} X\widehat{\beta} - X\beta^* &= (X\widehat{\beta} - X\beta) - (X\beta^* - X\beta) = X(\widehat{\beta} - \beta) - X(\beta^* - \beta) \\ &= P\varepsilon - (P - P_Q)\varepsilon = P_Q \varepsilon \end{aligned}$$

Tehát azt kaptuk, hogy:

$$X(\widehat{\beta} - \beta^*) = P_Q \varepsilon \quad (3)$$

A következő egyenlőségsor eredményét is fel fogjuk használni:

$$Y - X\widehat{\beta} = Y - X(X^TX)^{-1}X^TY = (I-P)Y = (I-P)(X\beta+\varepsilon) = (I-P)\varepsilon$$

Itt az utóbbi egyenlőség azért igaz, mert $PX = X(X^TX)^{-1}X^TX = X$ tehát $(I-P)X = 0$.

Vegyük az $Y - X\beta$ illeszkedési hiba következő, három részre való bontását!

$$Y - X\beta = (Y - X\widehat{\beta}) + (X\widehat{\beta} - X\beta^*) + (X\beta^* - X\beta)$$

Ez a három részre bontás a korábban levezetett 3 képlet szerint:

$$\varepsilon = (I-P)\varepsilon + (P-P_Q)\varepsilon + P_Q\varepsilon$$

Ahol az $I-P$, a $P-P_Q$ és a P_Q három kölcsönösen ortogonális, ortogonális projekció

Mivel mint projekciók idempotensek is, a:

$$\varepsilon^T\varepsilon = \varepsilon^T(I-P)\varepsilon + \varepsilon^T(P-P_Q)\varepsilon + \varepsilon^TP_Q\varepsilon$$

3 tagú összeg a hiba varianciájának 3 független χ^2 eloszlású változó összegére való bontása. Ha a $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ eloszlású, akkor az összeg eloszlása $\sigma^2\chi^2_n$, és három tag rendre $\sigma^2\chi^2_{n-p}$, $\sigma^2\chi^2_{p-q}$ illetve $\sigma^2\chi^2_q$ eloszlású, ahol n az összes megfigyelésszám, p a regresszió magyarázó változóinak a száma (rangja) a q pedig a feltételi egyenletek száma (rangja).

Mivel P a magyarázó változók által kifeszített altérre, a P_Q pedig ennek az altérnek a feltételi egyenletekkel meghatározott alterére való vetítés, a fenti felhontásnak megfelelő három négyzetösszeg alapján tesztelhető, hogy az $Y = X\beta + \varepsilon$ regressziós modellben a $R\beta = r$ lineáris feltétel teljesül-e.

Ugyanis az előzőekből következően

$$\frac{(\widehat{\beta}-\beta^*)^T X^T X(\widehat{\beta}-\beta^*)/(q)}{(Y-X\widehat{\beta})^T(Y-X\widehat{\beta})/(n-p)} \sim F_{q,n-p}. \qquad \text{eloszlású}$$

ha a feltétel nélküli regresszióra is érvényes, hogy $R\beta = r$ volna.

1.3.1. Alkalmazás

Megmutatjuk egy **R** program segitségével, hogy a feltételes regresszió ($\star$) szerinti megoldása helyes.

```
rm(list=ls())
set.seed(123)
b <- runif(4)
X <- cbind(1,matrix(rnorm(15*3),15,3))
e <- rnorm(15)
Y <- X%*%b+e

r <- matrix(runif(2),2)
R <- matrix(runif(2*4),2,4)

# Y a célváltozó értéke
# X a tervmátrix
# a b ismeretlen
# R*b=r a lineáris feltétel

bhat <- solve(t(X)%*%X)%*%t(X)%*%Y

bstar <- bhat - solve(t(X)%*%X)%*%t(R) %*%
        solve(R%*%solve(t(X)%*%X)%*%t(R)) %*% (R%*%bhat-r)

# a megoldás kielégíti a feltételt
cbind(R%*%bstar,r) # 0.3694889 0.9842192
all.equal(R%*%bstar,r) # TRUE

# a megoldás hibája
sum((X%*%bstar-Y)^2) # 13.06401

# bplus az Rb=r feltételt kielégítő más megoldás rosszabb
Rplus<-rbind(R,matrix(runif(2*4),2,4)) # dim(Rp)==c(4,4)
rplus<-rbind(r,matrix(runif(2),2,1)) # dim(rp)==c(4,1)

bplus <- matrix(solve(Rplus,rplus),,1)
all.equal(R%*%bplus,r) # TRUE a feltetelt kielegiti

# a hibája tényleg nagyobb
sum((X%*%bplus-Y)^2) # 44.68366 > 13.06401
sum((X%*%bstar-Y)^2) < sum((X%*%bplus-Y)^2) # TRUE
```

2. Kétfázisú regresszió

Továbbiakban azzal az esettel foglalkozunk, amikor csak két fázisra bontjuk fel a (1) függvényt, azaz azt esetet vizsgáljuk amikor a D szám értéke kettő. Ezt nevezzük *két-fázisú* regressziónak. Ekkor a (1) képlet a következő módon írható fel:

$$\mathbb{E}[y|x] = \begin{cases} \beta_{11} + \beta_{12}x, & x \leq \tau \\ \beta_{21} + \beta_{22}x, & x > \tau \end{cases} \tag{4}$$

Általánosan folytonos esetre van értelmezve. Mivel két minta között feltehetően egy ismeretlen t pontban történik a változás ($\{x_t : t = 1, \dots T\}$) ezért, ha nem használjuk a folytonosságot biztosító kritériumot, akkor a legjobb esetben is csak egy becslést adhatunk t értékre, amely csak annyit ír le, hogy melyik tag után következhet egy váltópont. Ezért mi megköveteljük a folytonos esetet.

2.1. Ismert c osztópont esete

Legyen a célváltozó értéke a modell szerint egy adott $c \in \mathbb{R}$ értékre

$$\begin{aligned} y &= \alpha_1 + \beta_1 x + \varepsilon \text{ ha } x \leq c \\ y &= \alpha_2 + \beta_2 x + \varepsilon \text{ ha } x \geq c \end{aligned}$$

valamely ismeretlen (α_1, β_1) és (α_2, β_2) értékekre.

Tegyük fel, hogy rendelkezésünkre áll n_1 megfigyelés, $(y_{1,i}, x_{1,i})$, $i = 1, ..., n_1$ amelyre $x_{1,i} < c$, és n_2 olyan megfigyelés $(y_{2,i}, x_{2,i})$, $i = 1, ..., n_2$, amelyre

$x_{1,i} > c$. A feladatunk tehát olyan paraméterek találása, amelyre az

$$\alpha_1 + \beta_1 c = \alpha_2 + \beta_2 c \tag{F}$$

feltétel teljesül.

Ekkor a feladat Lagrange függvénye

$$L = \sum_{k=1}^{2} \sum_{i=1}^{n_k} (y_{k,i} - (\alpha_k + \beta_k x_{k,i}))^2 + 2\lambda(\alpha_2 - \alpha_1 + c(\beta_2 - \beta_1))$$

Az optimális $(\alpha_1, \alpha_2, \beta_1, \beta_2)$ paraméter értékek tehát kielégítik az L függvény, rendre α_1, α_2, β_1, β_2 szerint vett parciális deriváltjait:

$$0 = -2 \sum_{i=1}^{n_1} (y_{1,i} - (\alpha_1 + \beta_1 x_{1,i})) - 2\lambda \tag{A}$$

$$0 = -2 \sum_{i=1}^{n_2} (y_{2,i} - (\alpha_2 + \beta_2 x_{2,i})) + 2\lambda \tag{B}$$

$$0 = -2 \sum_{i=1}^{n_1} x_{1,i}(y_{1,i} - (\alpha_1 + \beta_1 x_{1,i})) - 2\lambda c \tag{C}$$

$$0 = -2 \sum_{i=1}^{n_2} x_{2,i}(y_{2,i} - (\alpha_2 + \beta_2 x_{2,i})) + 2\lambda c \tag{D}$$

Az (A) és a (B) egyenlet alapján $k = 1, 2$-re

$$\alpha_k = \overline{y}_{k,.} - \beta_k \overline{x}_{k,.} - (-1)^k \lambda / n_k$$

Ezt az (F) feltételi egyenlőségben felhasználva, azt kapjuk, hogy

$$\lambda = \frac{n_1 n_2}{n_1 + n_2}((\overline{y}_{2,.} - \overline{y}_{1,.}) + \beta_1(\overline{x}_{1,.} - \gamma) - \beta_2(\overline{x}_{2,.} - \gamma))$$

Az így nyert α_1, α_2 és λ értékeket a (C) és (D) egyenletben felhasználva, a (β_1, β_2) paraméterpárra a következő egyenletrendszert nyerjük

$$\begin{pmatrix} c_{1,1} & c_{12} \\ c_{21}, & c_{22} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} c_{1,3} \\ c_{2,3} \end{pmatrix}$$

ahol $k = 1, 2$-re

$$\begin{aligned} c_{k,k} &= \sum_{i=1}^{n_k} (x_{k,i} - \overline{x}_{k,.})^2 + w(\overline{x}_{k,.} - c)^2 \\ c_{1,2} &= c_{2,1} = -w(\overline{x}_{1,.} - c)(\overline{x}_{2,.} - c) \\ c_{k,3} &= \sum_{i=1}^{n_k} (y_{k,i} - \overline{y}_{k,.})(x_{k,i} - \overline{x}_{k,.}) + (-1)^k w(\overline{y}_{2,.} - \overline{y}_{1,.})(\overline{x}_{k,.} - c) \end{aligned}$$

Az egyenletet megoldva megkapjuk a (β_1, β_2) keresett értékét. Ezek segítségével a λ értéke is adóddik a λ előző képlete alapján. Végül az így nyert β_1, β_2 és λ értékek segítségével már az α_1, α_2 értéke is meghatározható.

Vizsgáljuk meg hogy a kapott képletek valóban helyesek. Ehhez a Bevezetésben leírt példát használjuk.

```
x1 <- 0:10
# első szakasz x_i értékeinek az átlaga
atl_x1 <- sum(x1)/length(x1)
x2 <- 11:20
# második szakasz x_i értékeinek az átlaga
atl_x2 <- sum(x2)/length(x2)

y1 <- c(3,2,1,0,-1,-2,-3,-4,-5,-6,-7 )
# első szakasz y_i értékeinek az átlaga
atl_y1 <- sum(y1)/length(y1)
y2 <- c (-1, 5,11,17,23,29,35,41,47, 53)
# második szakasz y_i értékeinek az átlaga
atl_y2 <- sum(y2)/length(y2)

# feladatban leírt konstans értékek
```

```
n1 <- length(x1)
n2 <- length(x2)
w <- n1*n2/(n1+n2)
gam <- 10 #megadott töréspont értéke

# beta1, beta2 értékek kiszámolása
k <- 0:1
for(i in 1: n1){ c11 = (x[i] - atl_x1)^2 + w*(atl_x1 - gam)^2}
for(i in 1: n2){ c22 = (x[i] - atl_x2)^2 + w*(atl_x2 - gam)^2}
c12 <-  - w * (atl_x1 - gam)*(atl_x2 - gam)
c21 <- c12
c <- matrix( c(c11,c12,c21,c22), nrow=2, ncol = 2)
for(i in 1: n1){c13 = (y1[i] - atl_y1) *(x1[i] - atl_x1)+
                    (-1)*  w * (atl_y2 - atl_y1) * (atl_x1 - gam)}
for(i in 1: n2){c23 = (y2[i] - atl_y2)*(x1[i] - atl_x2)  +
                    ((-1)^2)* w * (atl_y2 - atl_y1) * (atl_x2 - gam}
C <- matrix( c(c13, c23), nrow = 2)
beta <- solve(c,C)

# lambda értékének számolása
lamb <- w * ( (atl_y2 - atl_y1) +
        beta[1] * (atl_x1 - gam) - beta[2] *(atl_x2 - gam))

# alpha1, alpha2 kiszámítása
alpha1 <- atl_y1 - beta[1] * atl_x1 + lamb/n1
alpha2 <- atl_y2 - beta[2] * atl_x2 - lamb/n2

# ellenőrzés - az [F] függvény által leírt alapján
gam1 <- (alpha2 - alpha1)/(beta[1] - beta[2])
gam1 #=10 azaz az együtthatók kiszámításához
     #jók a képletek
```

2.2. Ismeretlen c osztópont esete

Legyenek x_i normális eloszlású valószínűségi változók, konstans szórással, melyet σ^2 jelöl. Továbbá jelöljük γ-val az együtthatókból kifejezett töréspontot.

$$\gamma = \frac{\beta_{21} - \beta_{22}}{\beta_{12} - \beta_{11}}$$

A γ érték segítségével írjuk fel az egyenleteket [7]

$$y = \begin{cases} \lambda + \delta_0(x_i - \gamma), & i = 1, \ldots, \tau \\ \lambda + \delta_1(x_i - \gamma), & i = \tau + 1, \ldots \mathrm{T} \end{cases} \tag{5}$$

ahol $x_1 < \ldots < x_\tau \leq \gamma < x_{\tau+1} < \ldots < x_T$. A modell folytonossághoz megkötjük, hogy $x_\tau \leq \gamma < x_{\tau+1}$. λ jelöli a bal oldali meredekséget, δ_i pedig a regressziós együtthatókat.

Az osztópont, γ értékének meghatározása a következő alapján becsülhető meg. Vegyük a maximum likelihood becsléssel előállított függvényt egy tetszőleges u pontban, azt esetet vizsgálva, amikor minden együttható értéke ismeretlen [6].

$$Z(u) = \frac{[C_t - D_t(\widetilde{\gamma}_t + u) + E_t\widetilde{\gamma}_t u]^2}{C_t - 2 \cdot D_t u + E_t u^2} \cdot \frac{(\widetilde{\delta}_{0t} - \widetilde{\delta}_{1t})^2}{S_{x_1,t}} \tag{6}$$

(6) egyenlet esetében felhasználunk egy becslést az együtthatókra és feltesszük, hogy valamelyik $\widehat{\delta_{it}}, \widehat{\gamma}$ becsült érték fogja maximalizálni az adott függvényt, és ezzel a keresett értékeket megadni. Az egyenlet együtthatóit

pedig a következőképpen írhatjuk le:

$$
\begin{aligned}
C_t &= S_{x_1,t} \cdot S_{x_2,t} + w \cdot (\overline{x_{1t}}^2 S_{x_2,t} + \overline{x_{2t}}^2 S_{x_1,t}) \\
D_t &= w \cdot (\overline{x_{1t}} S_{x_2,t} + \overline{x_{2t}} S_{x_1,t}) \\
E_t &= w \cdot (S_{x_1,t} + S_{x_2,t})
\end{aligned}
$$

ahol $\overline{x_{1t}}$ átlag, $S_{x_1,t}$ korrigált négyzetösszeg $\{x_1, \dots x_t\}$ esetén, és $\overline{x_{2t}}$ átlag, $S_{x_2,t}$ korrigált négyzetösszeg $\{x_{t+1} \dots x_T\}$. A $w = \frac{n_1 \cdot n_2}{T}$ jelöli azt hányadost, ahol n_1 az első intervallum hossza, n_2 a második intervallum hossza és $n_1 + n_2 = T$.

A $\widehat{\gamma}$ meghatározása során végig megyünk az összes t értékén egészen $t = 2$ -től $t = T - 2$-ig. Amikor $x_t \leq \widetilde{\gamma}_t \leq x_{t+1}$, akkor a Z függvény $\widetilde{\gamma}_t$ értékét számoljuk ki. Ellenkező esetben mind az x_t mind a x_{t+1} pontban vett Z értékét. Ismert osztópont esetéhez hasonlóan itt is kapunk egy mátrixegyenletet.

$$
A_t(\hat{\gamma}, \hat{\gamma}) \begin{bmatrix} \hat{\delta}_0 \\ \hat{\delta}_1 \end{bmatrix} = A_{\hat{\tau}}(\widetilde{\gamma}_t, \hat{\gamma}) \begin{bmatrix} \widetilde{\delta}_{0t} \\ \widetilde{\delta}_{1t} \end{bmatrix}
$$

Ahol az A mátrix a következőképpen írható fel:

$$
A_t(u, z) = \begin{bmatrix} TS_{x_1,t} + n_1 n_2 (\overline{x_{1t}} - u)(\overline{x_{1t}} - z) & -n_1 n_2 (\overline{x_{2t}} - y)(\overline{x_{1t}} - z) \\ -n_1 n_2 (\overline{x_{1t}} - u)(\overline{x_{2t}} - z) & TS_{x_2,t} + n_1 n_2 (\overline{x_{2t}} - u)(\overline{x_{1t}} - z) \end{bmatrix}
$$

Amikor a $\widehat{\gamma}$ értéke egy j pontban vett $\widetilde{\gamma}$ értékkel megegyezik, akkor a $(\widehat{\delta}_0, \widehat{\delta}_1) = (\widetilde{\delta}_{0j}, \widetilde{\delta}_{1j})$. Ekkor a keresett γ ez az érték lesz.

3. Illesztési kísérletek, a `segmented` program

A problémák megoldására a `segmented` csomagot használjuk, melyet Vito. M. R. Muggeo szerkesztett és tett közzé. A csomag programjai közül a `segmented()` parancsot használjuk, mely vázlatos leírását a [1] cikkben találhatunk. Az elemzés során a pontosabb leírás érdekében a [8], [9] és [10] cikkeket is felhasználjuk. Illetve a programot a forrásprogram vizsgálata alapján is vizsgáljuk.

A szegmentált regressziót általánosan két paramétersorozat együttese írja le: $\alpha_k, \beta_k, k = 1, \ldots, m$, ahol általában ezek értéke ismeretlen. Feltételezzük, hogy a regressziós görbe minden pontjában *folytonos* és, hogy az α_k pontban véget erő k-adik szakaszon az $f_k(x\beta)$ paraméteres függvénnyel egyenlő. Négyféle váltási pontot becsülhetünk meg így.

Kevés megfigyelés esetén a töréspontok megbecsülhetőségét erőteljesen befolyásolja **a)** hogy mekkora a görbe törésszöge az egyes illeszkedési pontokban. Pontosabban, hogy a trendváltás deriváltja szignifikánsan eltér-e nullától, és **b)** hogy a töréspont nem esik-e egybe (szignifikánsan eltér-e) a környezetében lévő megfigyelési pontoktól.

Az ilyen feltételekkel megadott feladat tipikusan elég rosszul oldható csak meg. Ezért gyakorlatban a módszert különböző megszorítások mellett alkalmazzák. Legáltalánosabb feltételként azt használják, hogy a $f_k(x, \beta)$ függvények lineáris függvények legyenek. Ekkor, ha az α osztópontok ismeretlenek, akkor a feladat a korábbiakban ismertetett feltételes regressziós számításra egyszerűsödik. Ám gyakorlatban tipikusan pont az a probléma, hogy a töréspont időpontja ismeretlen, és hogy általában ennek a töréspontnak helyét akarjuk megbecsülni. Viszont, ha az α pontos ismeretlenek akkor a feladat egy nemlineáris feladattá válik.

A `segmented()` parancs eljárása

Az előbbiekben említett akadályok miatt a szegmentált regresszió illesztési eljárása tipikus a rácskeresés. Amikor az ismeretlen paramétert feltételezhetően tartalmazó m dimenziós térben egy lokálisan sűrűsödő rácson keressük meg azt a pontot, amelyikre a modell illeszkedése a legjobb. A sűrűsödés környékét, helyét ez esetben mindig az a pont jelöli ki, ahol az illeszkedés a legjobb. Általában ezek az eljárások, legkisebb négyzet eljárások melyek a hibanégyzetösszegét minimalizálják.

Ennek egy alternatív módszere, amikor a maximum likelohood becslést használjuk, azaz amikor azokat a paraméterétéket keressük meg, amelyekre a minta a „legvalószínűbb". Ezeknek az eljárásoknak a legfőbb nehézsége, hogy a célfüggvény nem konvex. Felmerülő problémák leküzdésére tipikusan véletlen kezdőérték választást használunk. Ennek a megoldásnak a hátránya, hogy az olyan esetekben, mint amilyenek a szegmentált regresszió is mutatkozik, csak a véletlennek köszönhető, ha nem csap be az eljárás. Ugyanis ezeknek a célfüggvényeknek az esetén tipikusak a nagykiterjedésű, kismélységű, kislejtésű lokális horpadások jellemzőek, míg a tényleges minimum egy szűk meredek környezetben helyezkedik el. Ilyenkor a tetszőlegesen választott indulási pont nagy eséllyel egy lokális minimumba vezeti a keresőt.

A megoldás: lineáris közelítés és „ bumped bootstrap"

Legyenek a megfigyelési értékek (y_k, z_k), $k = 1, \ldots, n$. Jelölje a $(.)_+$ a zárójelezett kifejezés pozitív részét, azaz

$$(x)_+ = \begin{cases} 0 & x < 0 \\ x & \text{egyébként} \end{cases}$$

Illetve abban az esetben amikor „." egy logikai kifejezés, akkor

$(HAMIS)_+ = 0$ és $(IGAZ)_x = 1$. A a célváltozó mért értékének

$$\mathbb{E}(y_k) \sim \beta_0 + \beta_1(z_k - \widehat{\psi})_+$$

szerinti közelítését a

$$\mathbb{E}(y_k) \sim \beta_0 + \beta_1(z_k - \widehat{\psi})_+ - \gamma(z_k > \widehat{\psi})_+$$

formában írja fel, ahol $\widehat{\psi}$ az aktuális ψ becslése.

Ez az egyenlet az ismeretlen β_0, β_1, γ paraméterekben lineáris. Az eljárás veszi az így kapott lineáris regresszió megoldását, majd a $\widehat{\psi}$ új becslésének a $\widehat{\psi} - \widehat{\gamma}/\widehat{\beta}$ értékének tekinti. Az iterációt addig folytatja, amíg a kapott γ értéke numerikusan nulla nem lesz.

Egy lehetséges alternatív megoldás a problémák elkerülése érdekében az úgynevezett „simulated annealing", azaz a szimulált kilágyítás adaptálása az aktuális feladatra. Ez az esetünkben azt jelentené, hogy az iterációval nyert, lokálisan optimálisnak tekintett megoldást, javító ciklusonként változóan egy-egy koordinátát véletlenszerűen megválasztva indítjuk újra az optimum kereső eljárás. Sajnálatosan ettől a módszertől sem várható lényegi javulás, hiszen a lokális megoldás koordinátánként megválasztásától nem várható a lokális megoldás konvergencia környezetéből való kilépés. Ezért az alkalmazott eljárás a bootsrap és az annaling bizonyios ötleteit felhasználva keresi az optimumot a következők szerint.

Egy adott lokális megoldást, azaz feltehetően csak lokálisan optimális paraméterbecslést félretéve a megfigyelt adatokból egy bootstrap mintát vesz, azaz a mintából visszatevéses mintavétellel egy, az eredetivel azonos megfigyelésszámű mintát. Ez a minta feltehetően őrzi az eredeti megfigyelés sorozat

globális jellemzőit, azonban a mintavétel miatt a lokális sajátosságai megváltoztak. Erre a bootstrap mintára elindítja a lokális optimum kereső eljárást az aktuális paraméter becsléssel, mint az induló paraméter értékkel, egy új paraméter becslést kap a mintavétellel torzult bootstrap adatok szerint. Ezzel a torzultbecsléssel mint kezdőértékkel az eredeti, a teljes mintán keres optimális megoldást, azaz egy új paraméter becslés áll rendelkezésre, amely az eredeti megfigyeléssorra vonatkozik. Megvizsgálja, majd a jobbat megtartja és azzal folytatja az eljárást addig, míg a két becslés különbsége elhanyagolhatóvá nem válik. Ezt a becslési eljárást nevezik „bumped bootstrap”-nek.

Tekintsük azt a kétfázisú adatot, amely a $(a, 0]$ szakaszon, $a-\alpha$ meredekségű egyenestől, a [0,b) szakaszon pedig a β meredekségű egyenestől egy független nulla várhatóértékű hibával különbözik.

A hiba szórásáról előbb feltesszük, hogy a két szakaszon azonos szórású, illetve, hogy az átlagértékeket az x- tengellyel párhuzamosra forgatva kapunk olyan megfigyelési értékeket, amelyek azonos szórásúak.

A vizsgálat egyszerűsítés kedvéért feltesszük azt is, hogy $a = b$ továbbá, hogy az x-tengely nulla pontjának két oldalára azonos számú megfigyelés esik.

A leírt modell a felsorolt megszorítások mellett vizsgáljuk először szimulációval, majd saját magunk által szerkesztett programmal, hogy különböző megfigyelés számok és szögek milyen pontos, és hogyan változik a fázisváltás pontossága.

3.1. Váltási pont becslése

Váltási pont becslése először a `segmented` programcsomag parancsainak felhasználásával.

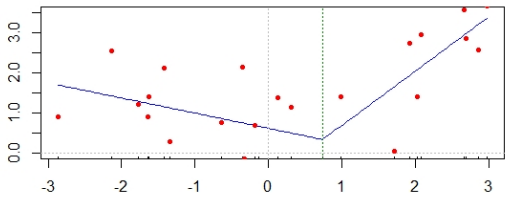
Először is generáljuk a teszt adatokat. Ne adjunk meg váltópont értéket, azaz a váltási pontok száma nulla legyen. Bal oldali meredekség értéke -1, jobb oldalié pedig 1

```
set.seed(123)
n <- 12     # a megfigyelésszám
a <- b <- 3   # az értelmezési tartomány
sa <- sb <- 1 # a hiba szórása
alpha <- beta <- pi/4 # a regresszió egyes szöge

ma <- -tan(alpha) # a meredekség a szög alapján
mb <-  tan(beta)

xa <- runif(n,-a,0)
xb <- runif(n,0,b)
ya <- ma*xa+rnorm(n,sd=sa)
yb <- mb*xb+rnorm(n,sd=sb)

x <- c(xa,xb)
y <- c(ya,yb)
plot(x,y,pch=20,col="red")
```

Miután konstruáltuk a szükséges bemenő paraméterek, a váltási pont becsléséhez felhasználjuk a `segmented::segmented()` paracsot

```
library(segmented)
M <- lm(y~x)
S <- segmented(M)
```

```
plot(S,ylim=c(0,3.5),las=1,ylab="",xlab="",col="blue")
abline(h=0,v=0,lty=3,col="gray")
points(x,y,pch=20,col="red")
abline(v=S$psi[2],lty=3,col="green4")
```

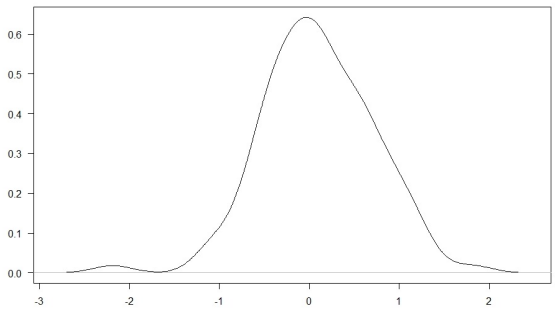


II. ÁBRA. ***Ábrán jól kivehető az illesztett modell. A váltópont becslése pedig*** 0.7383

3.2. Váltási pont becslésének tapasztalati statiszikái

Nézzük meg a várható értékét, szórását, illetve tapasztalati sűrűségfüggvényét is vizsgáljuk meg.

```
set.seed(123)
n <- 12     # a megfigyelésszám
a <- b <- 3   # az értelmezési tartomány
sa <-sb <- 1 # a hiba szórása

alpha <- beta < -pi/4 # a regresszió egyes szöge
ma <- -tan(alpha) # a meredekség a szög alapján
mb <-  tan(beta)
valt <- function()
  {
    xa <- runif(n,-a,0)
    xb <- runif(n,0,b)
    xx <- c(xa,xb)
    yy <- c(ma*xa+rnorm(n,sd=sa),mb*xb+rnorm(n,sd=sb))
    M <- lm(yy~xx)
    S <- segmented(M)
    return(S$psi[2])
  }

rm(.Random.seed);valt()
N <- 100
change <- vector("numeric",N)
for(k in 1:N) {rm(.Random.seed);change[k]<-valt()}
plot(density(change),las=1)
mean(change) # -0.05814304
sd(change) # 0.6039295 elég nagy!
```



III. ÁBRA. ***Becsült töréspont sűrűségfüggvénye***

3.3. Pontosság a szög függvényében

A töréspont becslés pontosságának vizsgálatát írjuk fel függvénybe, majd ezt futtassuk le különböző szögek esetén. Először veszünk egy derékszöget, majd megvizsgálunk egy derékszögnél nagyobb, illetve derékszögnél kisebb szöget

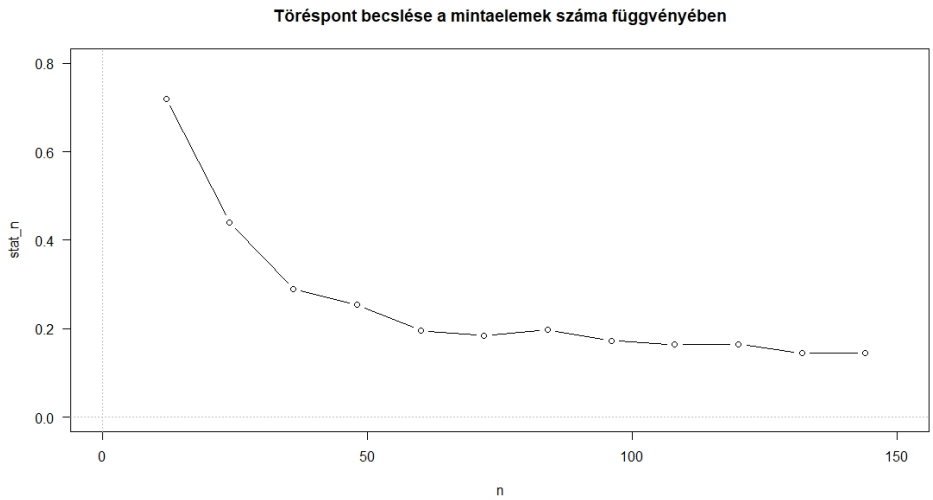
```
rm(list=ls())
a <- b <- 3   # az ertelmezési tartomany
sa <- sb <- 1 # a hiba szórása
alpha <- beta <- pi/4 # a regresszió egyes szöge
valt <- function(alpha=pi/4,beta=alpha,sa=1,sb=sa,n=12)
  {
    xa <- runif(n,-a,0)
    xb <- runif(n,0,b)
    xx <- c(xa,xb)
    yy <- c(-tan(alpha)*xa+rnorm(n,sd=sa),
             tan(beta)*xb+rnorm(n,sd=sb))
    M <- lm(yy~xx)
    S <- segmented(M)
    return(S$psi[2])
  }

# alpha=pi/2 a törés derékszögű
change <- vector("numeric",100)
for(k in 1:100) { rm(.Random.seed);
  change[k] <- valt(alpha=pi/4,sa=.5) }
sd(change) # 0.284965

# alpha=pi/5 a törés derékszögnél nagyobb
change <- vector("numeric",100)
for(k in 1:100) { rm(.Random.seed);
  change[k]<-valt(alpha=pi/5,sa=.5) }
sd(change) # 0.4082123 nagyobb szórás

# alpha=pi/3 a törés derékszögnél kisebb
change <- vector("numeric",100)
for(k in 1:100) { rm(.Random.seed);
  change[k]<-valt(alpha=pi/3,sa=.5) }
sd(change) #  0.1755956 kisebb szórás
```

Tehát kisebb szög esetében kisebb szórást, nagyobb szögnél nagyobb szórást kapunk.

3.4. Pontosság a megfigyelésszám függvényében

Vizsgáljuk meg, hogy miképpen változik a becslés tapasztalati szórása adott törésszög mellett, ha a megfigyelés számát növeljük.

```
n <- seq(12,by=12,length=12) # a megfigyelésszámok
stat_n <- vector("numeric",length(n)) # a becslés
N <- 100
for(j in 1:length(stat_n))
 {
  change <- vector("numeric",N)
  for(k in 1:N)
   { rm(.Random.seed);
     change[k]<-valt(n=n[j]) }
  stat_n[j]<-sd(change)
 }
plot(n,stat_n,las=1,xlim=c(0,150),ylim=c(0,.8),t="b",
    main = "Töréspontbecslés a mintaelemszámok szerint")
abline(h=0,v=0,lty=3,col="gray")
```

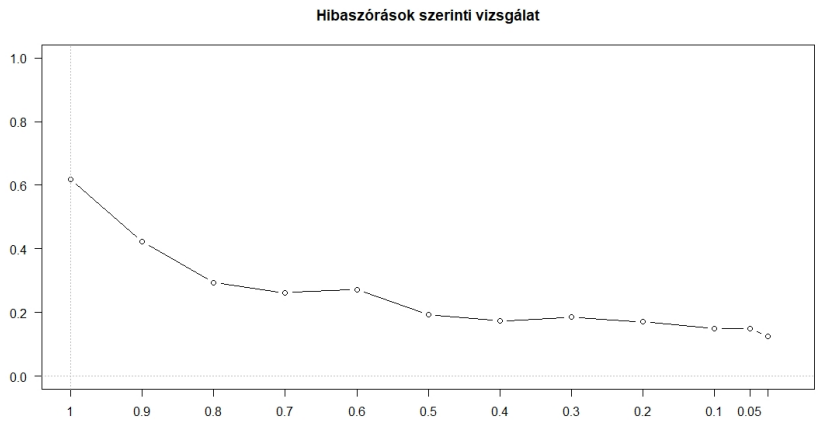


IV. ÁBRA. ***Látható, hogy adott törésszög mellett a megfigyelésszám növekedésével a becslés szórása csökken.***

3.5. Pontosság a hibaszórás függvényében

Végül vizsgáljuk meg a becslést a hibaszórás függvényében. Vegyünk egyre csökkenő szórás értékeket, majd ismételten futtassuk le az eddig használt függvényünket.

```
s <- c(1,.9,.8,.7,.6,.5,.4,.3,.2,.1,.05,.025)
stat_s <- vector("numeric",length(s)) # a becslés
N <- 100
for(j in 1:length(stat_s))
{
  change <- vector("numeric",N)
  for(k in 1:N)
  { rm(.Random.seed);
    change[k]<-valt(n=n[j]) }
  stat_s[j]<-sd(change)
}
plot(-s,stat_s,t="b",
     las=1,xlim=c(-1,0),ylim=c(0,1),
     xlab="",ylab="",
     axes=FALSE,frame=TRUE,
     main = "Hibaszórások szerinti vizsgálat")
axis(2,las=1)
axis(1,at=-s,label=s)
abline(h=0,v=-1,lty=3,col="gray")
```



V. ÁBRA. ***Látható, hogy a becslés pontossága növekszik a szórás értékeinek csökkenésével.***

4. Többfázisú regresszió illesztése

4.1. Kétfázisú regressziós eset

Tudható, hogy a Down szindrómával születés kockázata a növekszik a szülőanya korával, viszont fontos lenne tudni, hogy mekkora kockázat és milyen kortól kezd veszélyessé válni.

A következő kérdésekre igyekszünk választ adni [1]: Tényleg növeli a kockázatot az anyuka kora? Ez a kockázat növekszik, vagy konstans? Ha ténylegesen függ a szülőanya korától, akkor létezik valamilyen küszöbérték?

A probléma megoldásához a `segmented` programcsomagot fogjuk használni

```
library("gtools")
library("segmented")
data(down)
#age: anya életkora
#birth: születések száma
#cases: Donw-szindrómával születettek száma

arany <- logit(down$cases/down$births)
evek <- down$age
plot(evek, arany)
fit.glm <- glm(cases/births~age, weight=births,
             family=binomial, data=down)
fit.seg <- segmented(fit.glm, seg.Z=~age)

plot(fit.seg,link = TRUE, col = "blue",
    ylab = "logit(esetek/születések)",xlab = "évek")
points(evek, arany, pch = 20)
abline(v=fit.seg$psi[2], lty=3, col = "green3")
```

Az illesztés a $\psi \approx 31$ értékre tette a váltási pontot. Ezzel a létezés kérdését meg is válaszoltuk. Vajon ez a pont jól írja le a modellünket és ezáltal véglegesen kijelenthetjük, hogy 31 éves kortól a kockzázatban változást tapasztalunk? Hasonlítsuk össze az eredeti adatsorral a lineáris modellekből:

```
# lineáris modellből jósolt adat
pr.glm <- predict(fit.glm, type = "response")
# szegmentális modellből jósolt adat
pr.seg <- predict(fit.seg, type = "response")
# eredeti adatsor
y  <- down$cases/down$births

sum((pr.glm - y)^2) #=0.003632466
sum((pr.seg-y)^2) #=0.0005020527
```

Látszik, hogy a szegmentált modell jobban írja le az eredeti adatokat.

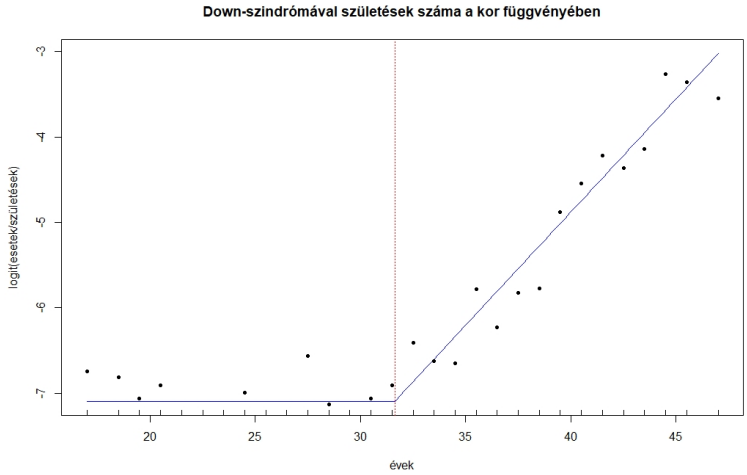
Érdemes megvizsgálni a modell szórását is, mely ebben az esetben elég nagy érték. Ennek javítására a bal oldali null meredekséggel frissítjük az illesztést.

```
fit.glm <- update(fit.glm,.~.-age)
fit.seg1 <- update(fit.seg)
```

Ekkor a két modell illesztése által kapott töréspont becslése és a hozzátartozó szórása a következő:

```
> fit.seg$psi
         Initial     Est.    St.Err
psi1.age      32 31.08115 0.7242074
> fit.seg1$psi
         Initial     Est.    St.Err
psi1.age      32 31.63761 0.5781952
```

Látható, amíg a becsült töréspont $\psi \approx 31$ nem változott sokat addig a szórás jelentősen csökkent.



VI. ÁBRA. ***Kockázat kezdetben konstans, majd egy időpillanattól kezdve növekszik.***

Az illesztésből látszik, hogy a kor előrehaladtával a kockázati tényező is növekszik, illetve létezik olyan érték, ahonnan növekedni kezd. Arra a kérdésünkre is választ kaptunk, hogy konstans kockázat van egészen töréspontig.

4.2. Kitenkintés több fázisú illesztésre

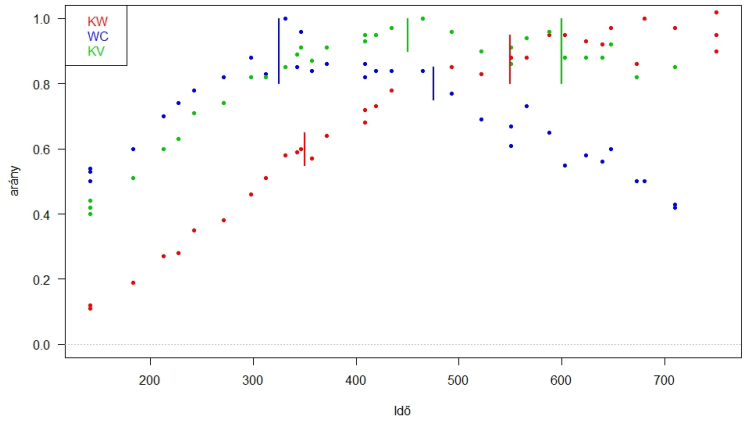
Ebben a példában a `Plant` adatsort használjuk.[1] Az y paraméter jelöli a növények növekedését az idő függvényében, *time* az időt jelöli, *group* pedig három különböző csoportot tartalmaz a levél típusok alapján. Implementáljuk a szukséges csomogokat illetve a feladatban lévő változókat hozzuk létre

```
library(segmented)
library("gtools")
library(ggplot2)

# adatok
rm(list=ls())
data(plant)
y <- plant$y
time <- plant$time
group <- plant$group

# csoportosításhoz szükséges változók
gr <- as.numeric(factor(group))
gr <- gr-1
gr[gr==0] <- 3
table(gr,group)

# a "time" adatok három részre osztása a "group" szerint
time.KW <- time.WC <- time.KV <- time

# csak az 1.csoport megfigyelesi idopontjai
time.KW[group!="RKW"] <- 0
# csak a 2.csoport megfigyelesi idopontjai
time.WC[group!="RWC"] <- 0
# csak a 3.csoport megfigyelesi idopontjai
time.KV[group!="RKV"] <- 0

csoport <- c("KW"   , "WC"   , "KV")
szin    <- c("red"  , "blue" , "green3")
```

A `segmented` parancs használatával, több magyarázó változó esetében kezdeti töréspontokat kell megadnunk, ami lehet egy konkrét pont vagy egy

intervallum, melyen feltételezhetően töréspont található. Ezt az adatok ábrázolása után könnyen választhatunk kezdeti értéket. Jelen esetben „KW, „WC”, „KV” nevezetű csoportokhoz rendre 450 és 600, 300 és 450, 300 és 450 a kezdetben becsült töréspontok értéke.

```
# adatok rajza, toresek kezdoertekei

plot(time,y,pch=20,col=szin[gr],las=1,ylim=c(0,1),
    xlab = "Idő", ylab = "arány")
abline(h=0,col="gray",lty=3)
segments(KW.i,c(.55,.8),KW.i,c(.65,.95),col="red"  ,lwd=2)
segments(WC.i,c(.8,.75),WC.i,c(1,.85),col="blue",lwd=2)
segments(KV.i,c(.9,.8),KV.i,c(1,1),col="green3"   ,lwd=2)
legend("topleft",paste(cs,"   "),text.col=szin)
```



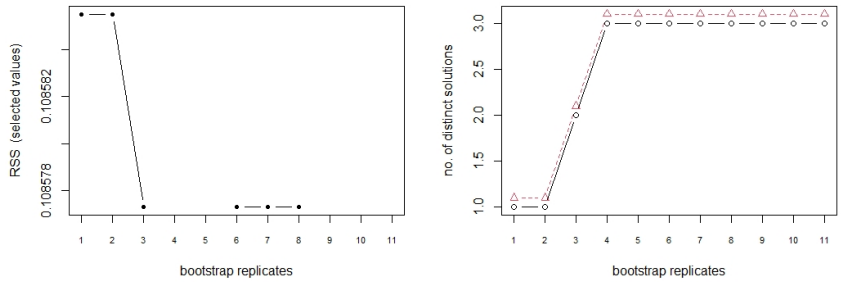
VII. ÁBRA. ***Ábrán jelzett függőleges vonalak jelzik töréspontok kezdeti értékeit***

Az előző fejezetekben ismertetett módszer szerint illesztjük a modellt a megfelelő adatokra.

```
# modell illesztés
olm <- lm(y~0+group+ time.KW + time.WC+ time.KV)
os <- segmented(olm, seg.Z= ~ time.KV + time.KW + time.WC,
              psi=list(time.KW=c(450,600),
                       time.WC=c(300,425),
                       time.KV=c(300,450)))
```

Ha ábrázoljuk a kapott értékeket a modell kifejezetten rosszul illeszkedik. Érdemes megvizsgálnunk az egyes iterációkban becsült töréspontokat.

```
draw.history(os,  term="time.KW")
draw.history(os,  term="time.KV")
draw.history(os,  term="time.WC")
# az ábrákon látszik, hogy 3 iteráció után
# már a bootstrap algoritmus megtalálja a töréspontokat
```



VIII. ÁBRA. `Boostrap` ***algoritmus töréspont becslése az iterációk függvényében***

A fenti képen látható, hogy az algoritmus már a harmadik iterációs lépés után megbecsüli a töréspontokat. Bár a kép csak egy csoport esetében mutatja meg a `boostrap` iterációs lépéseit, a többi csoport esetén is hasonló eredményekre jutunk. Ezért érdemes már magában az illesztésben megszabni az iterációk számát

```
os <- update(os, control=seg.control(h=.3))
# illesztések vizsgálata
plot(time, y, pch = 20,  xlab = "Idő", ylab = "arány")
plot(os, "time.KV", col=3, coef(os)["groupRKW"])
plot(os, "time.KW", col=2, coef(os)["groupRKW"])
plot(os, "time.WC", col=4, coef(os)["groupRKW"])
# az illesztés még mindig nem "szép"
```

Feltételezhetően az egyik csoport esetében a jobb oldali null meredekség „engedélyezett”, melyet a program nem kezel. Viszont pár változtatással kiküszöbölhető ez a probléma.

```
# jobb oldali null meredekséggel kiegészítés
neg.time.KW <- -time.KW
# modell frissités
olm1 <- lm(y~0+group+time.KV+time.WC+neg.time.KW)
os1 <- segmented(olm1, seg.Z=~ time.KV + time.WC+neg.time.KW,
               psi=list(time.KV=c(300,450), time.WC=c(300,450),
                neg.time.KW=c(-600,-450)), rev.sgn = TRUE)
```

Mivel használunk egy „negatív” változót az illesztés során, ezért a többi csoporthoz tartozó együtthatókat manuálisan kell beállítítanunk a negatív változóhoz tartozó becsült együtthatók alapján az adott csoportra vonatkozóan.

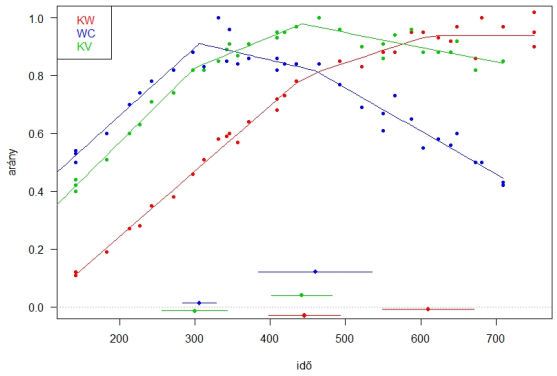
```
## az együtthatók definiálása
# ,,mínusz'' érték, miatt a többi csoport együtthatóját
# külön meg kell adni
const.KW <- coef(os1)["groupRKW"]
const.KV <- coef(os1)["groupRKV"]-
  + coef(os1)["U1.neg.time.KW"]*
  + os1$psi["psi1.neg.time.KW","Est."]-
  + coef(os1)["U2.neg.time.KW"]*
  + os1$psi["psi2.neg.time.KW","Est."]
const.WC <- coef(os1)["groupRWC"]-
  + coef(os1)["U1.neg.time.KW"]*
  + os1$psi["psi1.neg.time.KW","Est."]-
  + coef(os1)["U2.neg.time.KW"]*
  + os1$psi["psi2.neg.time.KW","Est."]

c(const.KW=const.KW,const.WC=const.WC,const.KV=const.KV)
# const.KW.groupRKW const.WC.groupRWC const.KV.groupRKV
#        0.93700000        0.18996378        0.05199698
```

Végül egy ábrán megmutatható, hogy a parancs által becsült pontok hogyan írják le az adatokat. [1]

```
# illesztés ábrázolása
plot(time, y,xlab="idő",ylab="arány",pch=20,col=szin[gr],
    las=1,ylim=c(0,1))
```

[1] Az ábrázolás során a cikkben leírt "minusz" értékek miatt használt `rev.sgn` parancs hibát eredményezett, (feltehetőleg a cikk megírása utáni csomag frissítésnek köszönhetően). A `plot.segmented()` parancs, kifejezetten erre a példára való javítása után már hiba nélkül lefutott

```
legend("topleft",paste(csoport,"   "),text.col=szin)
abline(h=0,col="gray",lty=3)

plot(os1, "neg.time.KW", add=TRUE, col="red",
const=const.KW,rev.sgn=TRUE)
plot(os1, "time.WC", add=TRUE, col="blue"  ,const=const.WC)
plot(os1, "time.KV", add=TRUE, col="green3",const=const.KV)
# a torespontok konfidencia tartomanyai
lines(os1,term="neg.time.KW",col="red",rev.sgn=TRUE)
lines(os1,term="time.WC",col="blue",k=10)
lines(os1,term="time.KV",col="green3",k=20)
```



IX. ÁBRA. ***Látható, hogy a "WC" csoport leveleinek nagysága az első töréspontig növekszik, majd nem növekszik tovább, A "KV" leveleinek nagyságával hasonló a helyzet, viszont ott a második törésponttól kezdve nem tapasztalunk változást, a harmadik csopot esetében folyamatos növekedést tapasztalunk.***

Hivatkozások

[1] Vito M. R. Muggeo: 'segmented': An R Package to Fit Regression Models with Broken-Line Relationships; R-News – The Newsletter of the R Project; Volume 8/1, pp20-25, May 2008.

[2] V. Muggeo: Estimating regression models with unknown break-points; Statistics in Medicine; Vol 22: pp 3055-3071, 2003.

[3] G.A.F. Seber and C.J. Wild, "Nonlinear regression", *John Wiley & Sons, 2003.*

[4] R Core Team; R Foundation for Statistical Computing; R: A Language and Environment for Statistical Computing; Vienna, Austria; 2020; https://www.R-project.org/ .

[5] Xiao-Feng Wang: 'fANCOVA': Nonparametric Analysis of Covariance; 2010; R package version 0.5-1; https://CRAN.R-project.org/package=fANCOVA.

[6] David V. Hinkley, "Inference about the Intersection in Two-Phase Regression", *Biometrika, Vol 56., pp 495-504, 1969.*

[7] David V. Hinkley, "Inference in Two-Phase Regression", *Journal of the American Statistical Association, Vol 66., pp 736-743, 1971.*

[8] DJ. Hudson: Fitting Segmented Curves Whose Join Points Have to be Estimated; JASA, Vol. 61, No. 316 pp. 1097-1129, 1966

[9] SN. Wood: Minimizing Model Fitting Objectives That Contain Spurious Local Minima by Bootstrap Restarting; Biometrics 57, pp 240-244, 2010.

[10] A regression model selection criterion based on bootstrap bumping for use with resistant fitting; CSDA Vol 35, 155–169, 2000.