Statisztikai elemzés az ingyenes R szoftverben

Mády Katalin

MTA Nyelvtudományi Intézet

2015. október 29.

Eredeti programnyelv S, ebből licensz alapú S+, ennek felel meg állítólag az S-, azaz R. Fejlesztők: **R**oss Ihaka és **R**obert Gentleman.

Letöltés: www.r-project.org, onnan elérhető tükrök. Linuxon repository-ból.

Windows és Mac GUI, Linuxon terminálablakban is működik R paranccsal megnyitva, vagy R Commander.

RStudio: Matlabhoz hasonló ablak.
`http://mathesaurus.sourceforge.net/octave-r.html`

Bevezetés az R-be, FAQ, teljes kézikönyv (*help* teljes anyaga pdf-ben).

Baayen, R. H. (2008): Analyzing linguistic data: a practical introduction to statistics using R. Cambridge: University Press.

Peter Dalgaard (2008): Introductory statistics with R. New York: Springer.

Field, Andy, Miles, Jeremy, & Field, Zoë (2012): *Discovering statistics using R.* London: SAGE.

Reiczigel, Harnos & Solymosi (2010): Biostatisztika nem statisztikusoknak. Nagykovácsi: Pars.

fuggvenyem(): függvény neve után kerek zárójelben az argumentumok és az opciók. `function(argument1,argument2,...)`

fuggvenyem(): függvény neve után kerek zárójelben az argumentumok és az opciók. `function(argument1,argument2,...)`

Adatok beírása:

`a = c(1,2,3,4)`

`c()`: concatenate = kösd össze

fuggvenyem(): függvény neve után kerek zárójelben az argumentumok és az opciók. `function(argument1,argument2,...)`

Adatok beírása:
`a = c(1,2,3,4)`
`c()`: concatenate = kösd össze

`b = c(3,5,2,1)`
`a` és `b` vektor egydimenziós, x-edik eleme `a[x]`, `b[x]`. Számozás 1-gyel kezdődik.

fuggvenyem(): függvény neve után kerek zárójelben az argumentumok és az opciók. `function(argument1,argument2,...)`

Adatok beírása:
`a = c(1,2,3,4)`
`c()`: concatenate = kösd össze

`b = c(3,5,2,1)`
`a` és `b` vektor egydimenziós, x-edik eleme `a[x]`, `b[x]`. Számozás 1-gyel kezdődik.

`m = cbind(a,b)` cbind = bind as columns
Mátrix `m`: kétdimenziós, első változó a sor, a második az oszlop.
Egész oszlop: m[,1], egész sor: m[2,], egy adott cella: m[1,2].

fuggvenyem(): függvény neve után kerek zárójelben az argumentumok és az opciók. `function(argument1,argument2,...)`

Adatok beírása:
`a = c(1,2,3,4)`
`c()`: concatenate = kösd össze

`b = c(3,5,2,1)`
`a` és `b` vektor egydimenziós, x-edik eleme `a[x]`, `b[x]`. Számozás 1-gyel kezdődik.

`m = cbind(a,b)` cbind = bind as columns
Mátrix `m`: kétdimenziós, első változó a sor, a második az oszlop.
Egész oszlop: m[,1], egész sor: m[2,], egy adott cella: m[1,2].

String változók idézőjelek között:
`d = c("n","n","f","f")`

fuggvenyem(): függvény neve után kerek zárójelben az argumentumok és az opciók. `function(argument1,argument2,...)`

Adatok beírása:
`a = c(1,2,3,4)`
`c()`: concatenate = kösd össze

`b = c(3,5,2,1)`
`a` és `b` vektor egydimenziós, x-edik eleme `a[x]`, `b[x]`. Számozás 1-gyel kezdődik.

`m = cbind(a,b)` cbind = bind as columns
Mátrix `m`: kétdimenziós, első változó a sor, a második az oszlop.
Egész oszlop: m[,1], egész sor: m[2,], egy adott cella: m[1,2].

String változók idézőjelek között:
`d = c("n","n","f","f")`

Lehet = helyett <- jelet is használni, `c <- cbind(a,b)`, sőt, fordítva is: `cbind(a,b) -> c`. R-specifikus, hardcore felhasználók így adják meg leírásokban.

Honnan tudjuk, milyen változó?

`class(valtozonev)`

pl. ha vektor: „numeric”, „character”, ha mátrix: „matrix”, ha különböző hosszúságú vektorokból áll: „list” stb.

Honnan tudjuk, milyen változó?
`class(valtozonev)`
pl. ha vektor: „numeric”, „character”, ha mátrix: „matrix”, ha különböző hosszúságú vektorokból áll: „list” stb.

Ha bizonyos változókkal bizonyos műveleteket végzünk, az R néha átalakítja őket!
`e = cbind(a,d)`
Ok: mátrix csak egyféle változótípusból állhat, ezért a számból betű lesz.

Honnan tudjuk, milyen változó?
`class(valtozonev)`
pl. ha vektor: „numeric”, „character”, ha mátrix: „matrix”, ha különböző hosszúságú vektorokból áll: „list” stb.

Ha bizonyos változókkal bizonyos műveleteket végzünk, az R néha átalakítja őket!
`e = cbind(a,d)`
Ok: mátrix csak egyféle változótípusból állhat, ezért a számból betű lesz.

`e = data.frame(a,b,d)`
Adatmátrix, az oszlopok lehetnek eltérő típusú értékek, pl. karakter és numerikus. Az oszlopoknak lehet nevet adni names()[hanyadik] függvénnyel.
Oszlop osztályának lekérdezése: class(e[,3]) vagy e$oszlopnev.

Feladat

Dobjunk fel egy dobókockát tizenkétszer, és írjuk le, milyen számokat kaptunk.

Feladat

Dobjunk fel egy dobókockát tizenkétszer, és írjuk le, milyen számokat kaptunk.

Egész számok véletlenszerű generálása: `sample()`

Függvény kötelező és opcionális argumentumainak lekérdezése súgóból: `?sample` vagy `help(sample)`.
Két argumentum megadása kötelező: `x` = miből húzzon, `size` = hányszor.

x lehet `1:6` vagy `seq(1,6)`, sequence.

Feladat

Dobjunk fel egy dobókockát tizenkétszer, és írjuk le, milyen számokat kaptunk.

Egész számok véletlenszerű generálása: `sample()`

Függvény kötelező és opcionális argumentumainak lekérdezése súgóból: `?sample` vagy `help(sample)`.
Két argumentum megadása kötelező: `x` = miből húzzon, `size` = hányszor.

x lehet `1:6` vagy `seq(1,6)`, sequence.

```
kocka = sample(1:6,12)
```

Feladat

Dobjunk fel egy dobókockát tizenkétszer, és írjuk le, milyen számokat kaptunk.

Egész számok véletlenszerű generálása: `sample()`

Függvény kötelező és opcionális argumentumainak lekérdezése súgóból: `?sample` vagy `help(sample)`.
Két argumentum megadása kötelező: `x` = miből húzzon, `size` = hányszor.

x lehet `1:6` vagy `seq(1,6)`, sequence.

```
kocka = sample(1:6,12)
```

$\Rightarrow$ hibajelzés.

Figyelem! Default argumentum szerint egy számot csak egyszer lehet kihúzni: `replace=FALSE`. Megoldás: replace=TRUE. Default értékek is kiderülnek a súgóból.

Feladat

Foglaljuk össze a `kocka` adatmátrixban (data.frame), hányadikra hányat dobtunk. Így nézzen ki:

dobas	pontszam
1	6
2	6
3	5
4	6
.	.
.	.
.	.

Adjuk meg a móduszt, mediánt, átlagot és a dobások összegét.

Hasznos parancsok: `cbind()`, `rbind()`, `table()`, `median()`, `mean()`, `sum()`.

Logikai vektorok

Egy adatmátrixon egy adott változón belüli csoportok definiálása.

Operátorok:

==	azonos
!=	nem azonos
%in%	tartalmazza a vektor valamely elemét
<, >	kisebb, nagyobb
<>	nem egyenlő
|	vagy
&	és

Logikai vektorok definíciója

z = kocka$dobas == 6: hatos dobások
z = kocka$dobas < 1: egyesnél nagyobb dobások

feltételt teljesítő sorok listázása:
`kocka$dobas[z,]`
smallskip
összes elem feltételt teljesítő elemei vektorként:
`kocka$dobas[z]`

Melyik elemekre igaz:
`which(z)`

Összes előfordulás:
`sum(z)`

Adatok beolvasása az R-be

Adatokat többnyire más szoftverrel állítottuk elő (E-Prime, Praat, manuális lejegyzés, Excel stb). Ezek beolvasása:
`read.table()`
`read.table(file, header = FALSE, sep = "", dec = ".")`
`header`: vannak-e oszlopnevek. Ha első sor eggyel kevesebb elemet tartalmaz, R automatikusan HEADER-ként kezeli. Ha nincsenek, az R automatikusan V1, V2 . . . nevet ad az oszlopoknak.

`sep`: szóköz, tab, vessző, pontosvessző stb. Szóköz és tabulátor problémás, ha több üres cella van. Ilyenkor vagy NA-t kell beírni (*not available*), vagy érdemes pontosvesszőt használni.

`dec`: ha közép-európai kódolású szoftvert használunk, a decimális vessző. tehát `dec = ","`

Töltsünk le egy adatfájlt innen:
http://clara.nytud.hu/∼mady/courses/statistics/soc.dat

Fájl beolvasása Linuxban

Adatfájl helye: `/home/user/R/kurzus/soc.dat` (tetszés szerinti könyvtár). Beolvasás:

```
soc=read.table("/home/user/R/kurzus/soc.dat",
header=T,sep=";")
```

Ezzel a `soc` változóba/objektumba írtuk a soc.dat fájl tartalmát.

Idézőjel szerepe: ha nincs, R a munkamemóriában tárolt változót keres.

Linux előnye: R bármelyik könyvtárból megnyitható az R parancs beírásával. Ha soc.dat-ot ide mentettük, elég a `read.table("soc.dat",...)` függvényt beírni.

Gyakorlati haszon: R-fájlokat projekteknek megfelelő könyvtárban tudjuk tárolni.

Grafikus felület (Mac, Windows)

Betöltés nem lehetséges közvetlen elérési útvonallal. Ehelyett:

(1) R-konzolban (ablak) File > Change directory... megkeressük a könyvtárat, ahova soc.dat-ot mentettük.

```
soc=read.table("C:/Users/en/Downloads/soc.dat",
header=T,sep=";")
```

VAGY

(2) aktuális munkamemória: getwd(). Betöltendő fájl helyének megadása: setwd("konyvtar").

Fontos: Windows-ban is / jelet használunk!

Néhány hasznos függvény

`read.csv()`: Excelben vagy más táblázatkezelőben tárolt adatok beolvasása, miután csv-ként mentettük. Előny: default paramétereken nem vagy alig kell állítani.

Néhány hasznos függvény

`read.csv()`: Excelben vagy más táblázatkezelőben tárolt adatok beolvasása, miután csv-ként mentettük. Előny: default paramétereken nem vagy alig kell állítani.

`ls()`: R munkamemóriában tárolt objektumok (változók).

Néhány hasznos függvény

`read.csv()`: Excelben vagy más táblázatkezelőben tárolt adatok beolvasása, miután csv-ként mentettük. Előny: default paramétereken nem vagy alig kell állítani.

`ls()`: R munkamemóriában tárolt objektumok (változók).

`names(soc)`: oszlopban tárolt változók neve.

Néhány hasznos függvény

`read.csv()`: Excelben vagy más táblázatkezelőben tárolt adatok beolvasása, miután csv-ként mentettük. Előny: default paramétereken nem vagy alig kell állítani.

`ls()`: R munkamemóriában tárolt objektumok (változók).

`names(soc)`: oszlopban tárolt változók neve.

`head(soc)`: első hat adatsor, `tail()` utolsó hat.

Néhány hasznos függvény

`read.csv()`: Excelben vagy más táblázatkezelőben tárolt adatok beolvasása, miután csv-ként mentettük. Előny: default paramétereken nem vagy alig kell állítani.

`ls()`: R munkamemóriában tárolt objektumok (változók).

`names(soc)`: oszlopban tárolt változók neve.

`head(soc)`: első hat adatsor, `tail()` utolsó hat.

Az R-ben tárolt objektumokat ki tudjuk írni R-ben olvasható formátumban a save(objektum,file="idements.RData") függvénnyel. Beolvasás: load("idements.RData").

Néhány hasznos függvény

`read.csv()`: Excelben vagy más táblázatkezelőben tárolt adatok beolvasása, miután csv-ként mentettük. Előny: default paramétereken nem vagy alig kell állítani.

`ls()`: R munkamemóriában tárolt objektumok (változók).

`names(soc)`: oszlopban tárolt változók neve.

`head(soc)`: első hat adatsor, `tail()` utolsó hat.

Az R-ben tárolt objektumokat ki tudjuk írni R-ben olvasható formátumban a save(objektum,file="idements.RData") függvénnyel. Beolvasás: load("idements.RData").

rm(): objektum törlése

Néhány hasznos függvény

`read.csv()`: Excelben vagy más táblázatkezelőben tárolt adatok beolvasása, miután csv-ként mentettük. Előny: default paramétereken nem vagy alig kell állítani.

`ls()`: R munkamemóriában tárolt objektumok (változók).

`names(soc)`: oszlopban tárolt változók neve.

`head(soc)`: első hat adatsor, `tail()` utolsó hat.

Az R-ben tárolt objektumokat ki tudjuk írni R-ben olvasható formátumban a save(objektum,file="idements.RData") függvénnyel. Beolvasás: load("idements.RData").

rm(): objektum törlése

`write.table()`: mátrix kiírása szöveges táblázatba

Adatok mentése

Kilépés NEM a GUI (grafikus felület, graphical user interface) bezárásával, hanem a

`q()`

függvénnyel. Save directory? yes/no/cancel

Érdemes menteni, akkor az objektumok megnyitáskor ismét betöltődnek.

Linux: automatikusan abba a könyvtárba ment, ahonnan megnyitottuk az R-t.

Mac és Windows: default: R.exe fájl könyvtára. Módosítható `setwd()` függvénnyel.

.Rhistory: beadott parancsok listája, .RData: tárolt objektumok.

Előfordulási gyakoriságok

- kördiagram,
- oszlopdiagram.

R-funkciók:
`table(dataframe$oszlop)`: előforduló kategóriák gyakorisága.

Ábrázolás: `pie(table(x))`, `barplot(table(x))`.

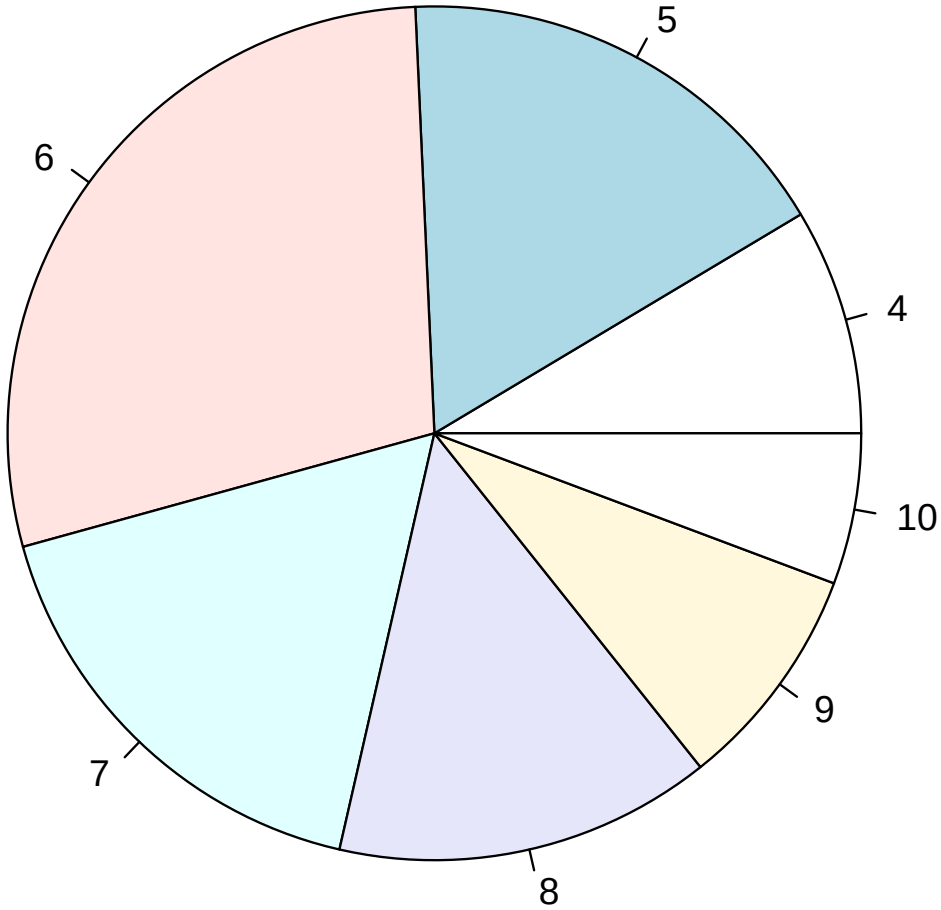
Példa

Növény- és állatnevek hosszúsága betűkben megadva a `szohossz` változóban.

```
> szohossz
nev          típus    betu
medve        allat    5
palma        noveny   5
paradicsom   noveny   10
elefant      allat    7
```

Kördiagram

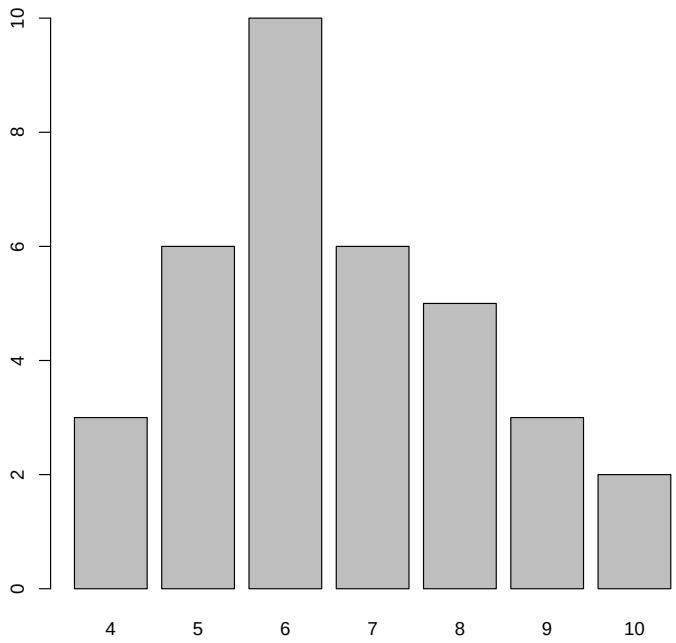
Betűszámok gyakoriságai

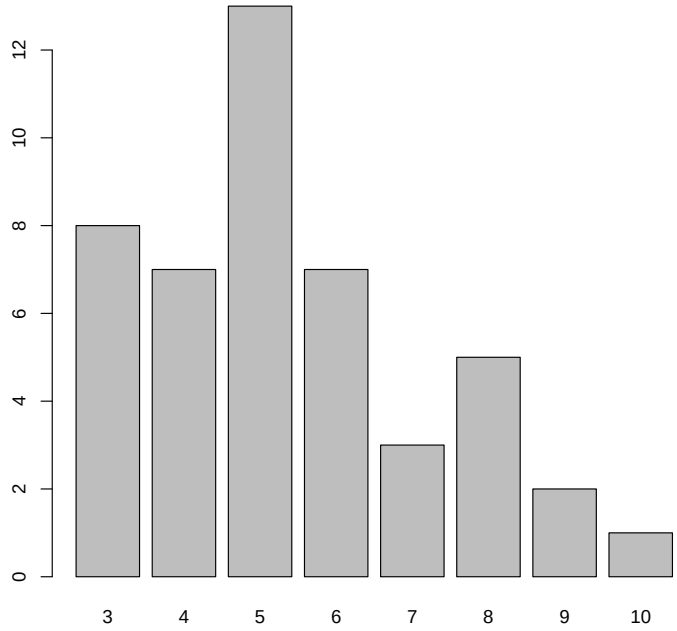



```
pie(table(szohossz$betu))
```

Oszlopdiagram

Növény- (bal), és állatnevek (jobb) betűszámának gyakoriságai:






```
noveny = szohossz$tipus == "noveny"
barplot(table(szohossz$betu)[z])
```

`barplot(table(szohossz$betu)[!z])`: minden, amit a logikai vektor FALSE-ként tárol

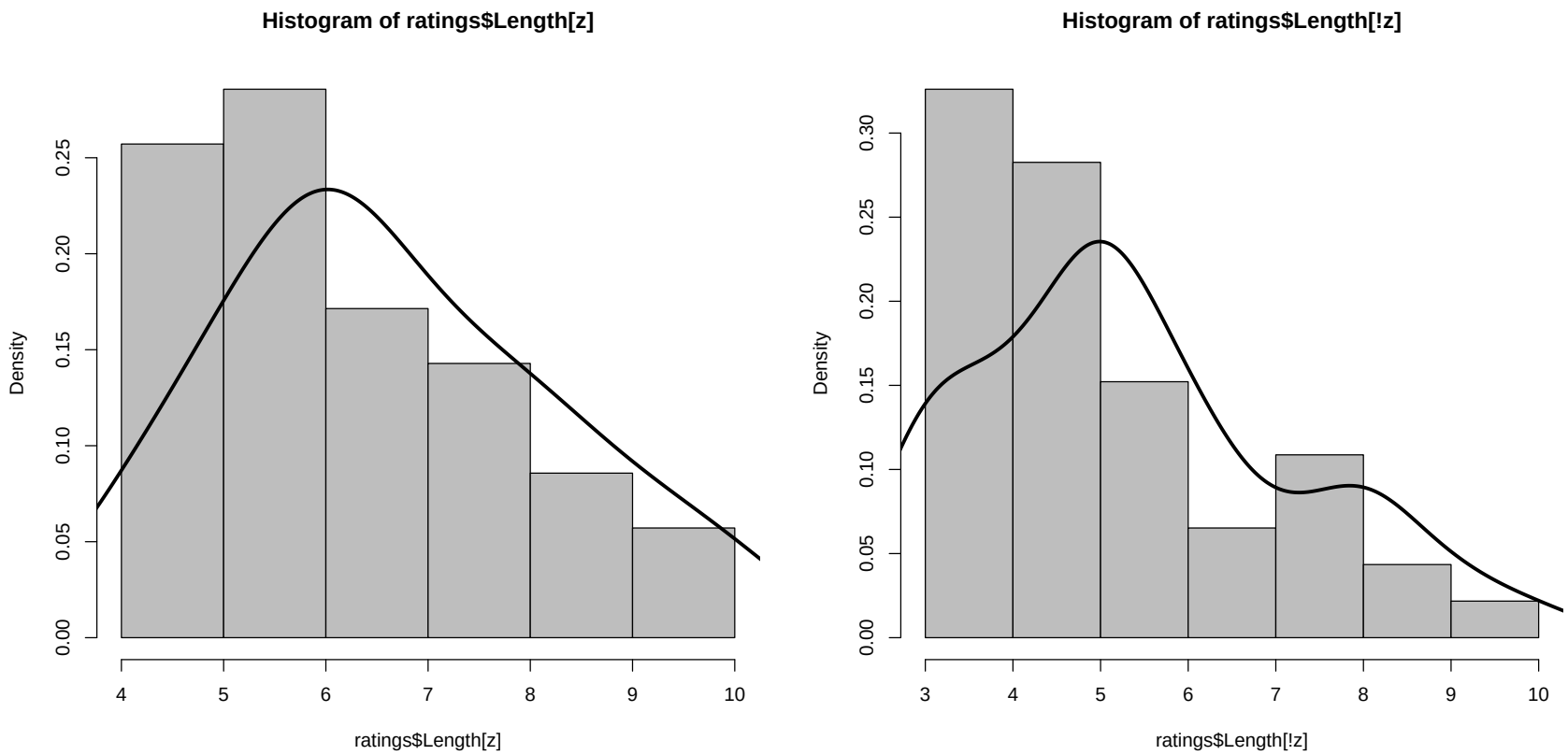
Feladat

http://clara.nytud.hu/~mady/courses/statistics/materials/ratings.RData betöltése.

Class: animal/plant, Length: betűk száma.
Oszlopdiagram készítése a növények és állatok betűhosszaira, de nem abszolút számokkal, hanem százalékos értékekkel.

Hisztogram

Növény- (bal), és állatnevek (jobb) betűszámának sűrűsége:



`hist()`, `plot(density())`, utóbbi sűrűségfüggvény.

Grafikus paraméterek

Rengeteg paraméteren lehet állítani. Hogyan lehet ezekről tudni?

- grafikus parancs opcionális argumentumai. Lekérdezés: `?boxplot`, `?plot`, `?barplot` stb.
- parancsok súgója gyakran utal további hasznos parancsokra (`line()`,`title()`,`abline()` stb.
- `par()`: rengeteg paraméter, pl. tengelyek feliratozása (felirat mérete, elhelyezése, egységek mérete), tengelyek aránya stb.

Első lépés súgó. Felépítés: (1) kötelező és opcionális argumentumok listája, (2) argumentumok rövid magyarázata, (3) részletek: többnyire innen derül ki a releváns infó, ha még nem ismerjük a parancsot, gyakran hivatkozások is, (4) lásd még - hasznos, esetenként hasznosabb, további parancsok, (5) példák - ezek általában túl bonyolultak, ezért nem túl hasznosak.

Néhány hasznos paraméter

- **xlab, ylab:** "x-tengely felirata", "y-tengely felirata".
- **main:** "Ábra címe".
- **xlim, ylim:** Ábrázolt értékek tól-ig. Főleg y-tengelynél fontos, ha összehasonlítható ábrákat akarunk. Pl százalékos ábrázolásnál `ylim = c(0,100)`, azaz 0–100%-ig. Egyenlőségjel előtti szóköz opcionális.
- **col:** színek, vagy névvel, vagy számmal. Pl. `col=2` és `col="red"` azonosak.
- **cex:** `cex.main,cex.axis,cex.names` stb. Default: cex=1, ehhez képest cím, mérőszámok, címkék betűmérete nagyobb (1.3, 1.7) vagy kisebb (0.7).

Ábra mentése

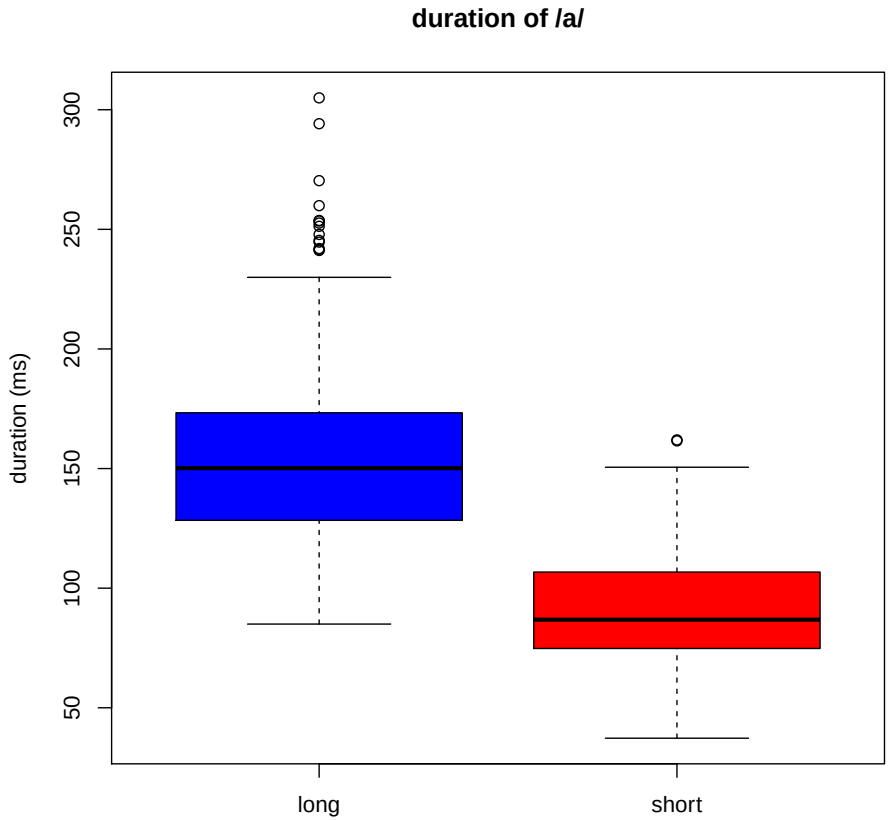
Alapeset: mentés pdf-ként vagy postscript fájléként.
pdf: LaTeX-felhasználóknak hasznos, ha pdflatex-et használnak. eps: Word-ben és LaTeX-ben egyaránt használható.

`dev.print("célfájl",device=pdf|postscript)`

postscript fájlok alapértelmezése: fektetett, horizontal=T $\Rightarrow$ mentés: `horizontal=F` vagy `dev.copy2eps()`.

Ha nem adunk meg elérési útvonalat: mentés aktuális könyvtárba (`getwd()` paranccsal megtudható).

Dobozdiagram




```
boxplot(fuggovaltozo~fuggetlenvaltozo,data=objektumnev
```

Dobozdiagram

Ábrázolás módja: y-tengely: függő változó interkvartilis eloszlása,
x-tengely: csoportok, esetleg további tagolással.

```
boxplot(függő változó~függetlenváltozó,data=objektumnev
```

Ha további csoportosítás:

```
boxplot(függő változó~függetlenváltozó1*függetlenváltozó2,data
```

Például:

```
boxplot(ratings$Frequency~ratings$Class*ratings$Complex)
```

Egyszerű és összetett állat- és növénynevek gyakorisága.

hasznos paraméterek:

```
col=c("red","blue")
names=c("állat","növény")
```

Programcsomagok telepítése az R-ben

Mivel az R nyílt forráskódú szoftver, bárki fejleszthet hozzá csomagokat.
Ellérhető csomagok listája az R mirror oldalakon, Packages menüpont alatt.
Telepítés interneten keresztül:
`install.packages("languageR")`
mirror kiválasztása (minél közelibb, annál kevesebb adatforgalmat generálunk).

Fontos! Ha objektumokat tartalmazó csomagot telepítünk, jegyezzük meg, hova telepíti őket az R! Az objektumok listáját később ott találjuk meg.

Windows 7: a programkönyvtár nem írható, adminisztrátori jogokkal sem. Ezért az R a felhasználó könyvtárában létrehoz egy R könyvtárat, és oda tölti le a languageR csomagot.

Csomag betöltése

betöltés (R megnyitása után minden egyes alkalommal):
`library(languageR)`
Ha $>$ jelet kapunk „válaszként”, akkor a csomag betöltődött az R-be.

Ellenőrzés:
`search()`
aktuálisan betöltött csomagok listája.
Elérhető objektumok listája és rövid leírása: languageR telepítésének könyvtárában, az INDEX fájlban.

Feladatok

1. A ratings objektumban található szógyakorisági adatok (Frequency) ábrázolása dobozdiagrammal a növényekre és az állatokra. Magyar feliratokkal, két különböző színnel, az alapértelmezettnél nagyobb (1.3-szoros) tengelyfeliratokkal.

2. Két dobókockával való 10, 100, 1000 dobás összege, és az összegek ábrázolása sűrűségfüggvénnyel.

A *t*-próba típusai

- Egymintás *t*-próba: egy adatsort hasonlítunk egy adott értékhez, pl. 0 vagy deklarált átlag (pl. IQ-teszt átlaga teljes populációra 100).
- Kétmintás független *t*-próba varianciahomogenitás esetén.
- Welch-próba, ha a varianciák nem azonosak.
- Páros *t*-próba.

```
t.test(x, y = NULL,
alternative = c("two.sided", "less", "greater"),
mu = 0, paired = FALSE, var.equal = FALSE,
conf.level = 0.95, ...)
```

Normális eloszlás és varianciahomogenitás tesztelése

Normális eloszlás: `shapiro.test()`.

Varianciahomogenitás

- ***F*-próba:** mindkét mintában normális eloszlás, független minták. R: `var.test()`.
- **Levene-próba:** közelítő próba, de normális eloszlás hiányában is használható, több mintára is. R: `leveneTest` a `car` könyvtárban.
- **Bartlett-próba:** normális eloszlás, páros mintákra is használható. R: `bartlett.test()`.

Feladat

Töltsük le a trans.RData fájlt innen:
clara.nytud.hu/∼mady/courses/statistics/materials/trans.RData
Letöltés `load("konyvtar/trans.RData")` függvénnyel, NEM `read.table()`.

A mátrixban angol, ill. portugál, kb. 1500 szavas szövegek hossza van megadva, majd a másik nyelre való lefordítás utáni hosszuk.

Ellenőrizzük a normális eloszlást és a varianciahomogenitás meglétét.

```
z = trans$language == "English"
shapiro.test(trans$length[z])/(trans$length[!z])
leveneTest(trans$length∼trans$language
```

Feladat

Töltsük le a trans.RData fájlt innen:
clara.nytud.hu/~mady/courses/statistics/materials/trans.RData
Letöltés `load("konyvtar/trans.RData")` függvénnyel, NEM `read.table()`.

A mátrixban angol, ill. portugál, kb. 1500 szavas szövegek hossza van megadva, majd a másik nyelre való lefordítás utáni hosszuk.

Ellenőrizzük a normális eloszlást és a varianciahomogenitás meglétét.

```
z = trans$language == "English"
shapiro.test(trans$length[z])/(trans$length[!z])
leveneTest(trans$length~trans$language
```

Nem teljesülnek a *t*-próba feltételei, ezért Mann-Whitney-próbát alkalmazunk:
`wilcox.test(trans$length~trans$language)`

További feladatok

`ratings` adatmátrix a languageR könyvtárból:

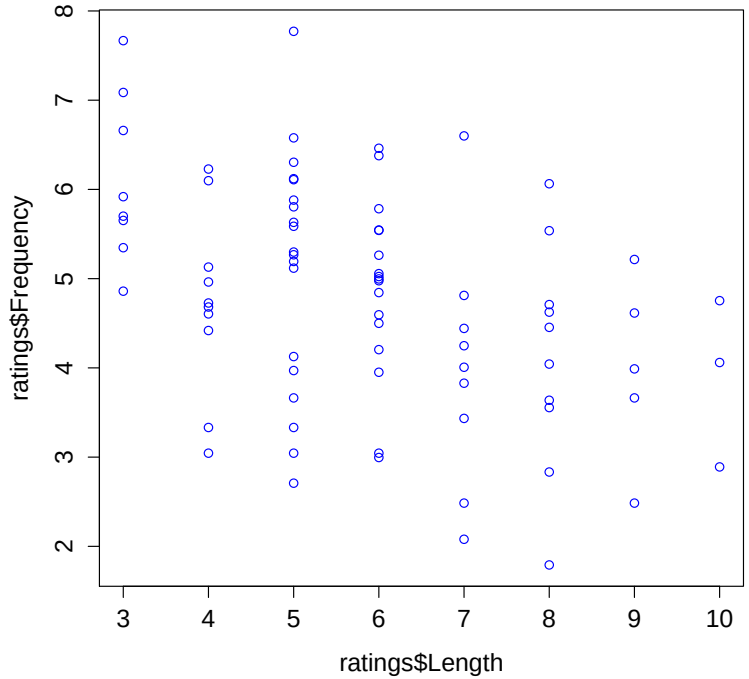
1. Igaz-e, hogy az állatnevek gyakorisága alacsonyabb mértékű, mint a növényeké? És az ismertségük?

2. Különbözik-e a növény- és állatnevek egyes és többes számú gyakoriségi indexe?

Teszteljük minden esetben, hogy az adatok normális eloszlásúak-e, és hogy a varianciák homogének-e.

Lineáris regresszió

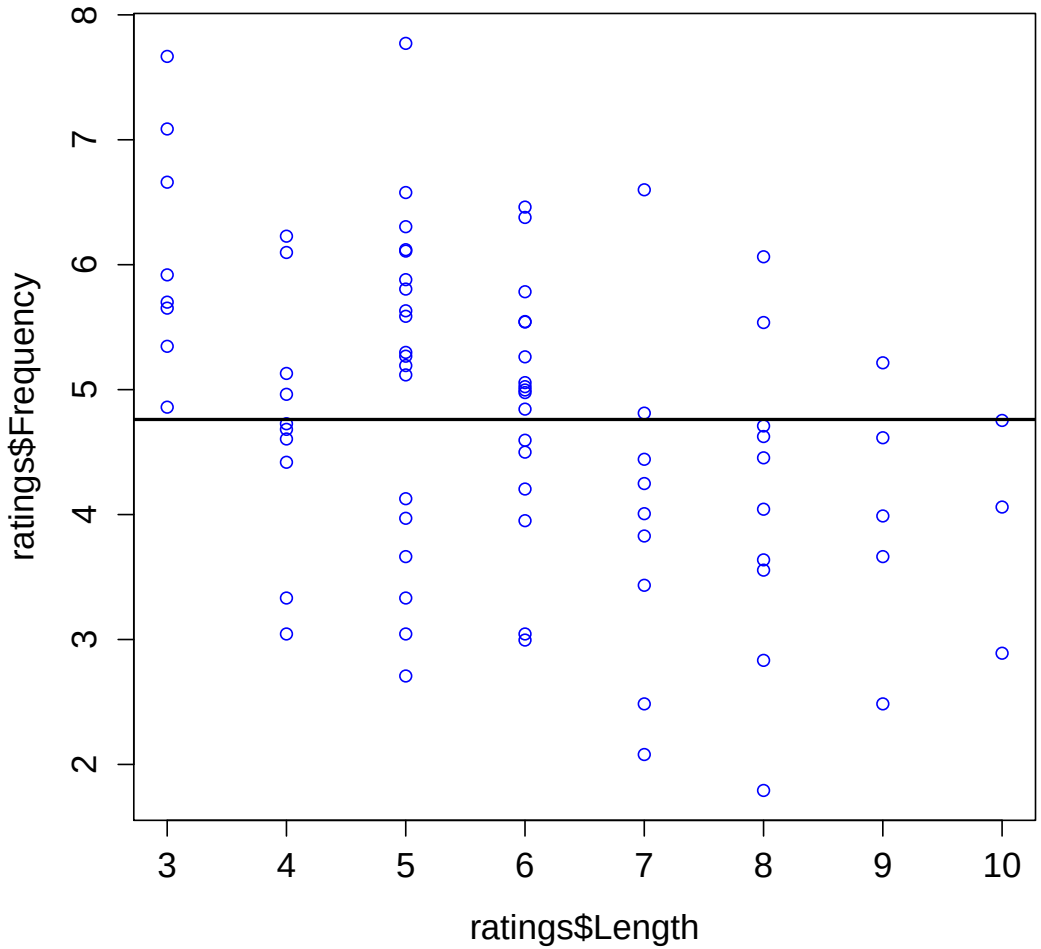
`ratings` adatmátrixban (languageR csomag) található állat- és növénynevek hosszúsága (betűk száma) és nevek gyakorisága közötti összefüggés:



Korrelációs együtthatók: $r = -0,43$, $\rho = -0,43$, $\tau = -0,31$.

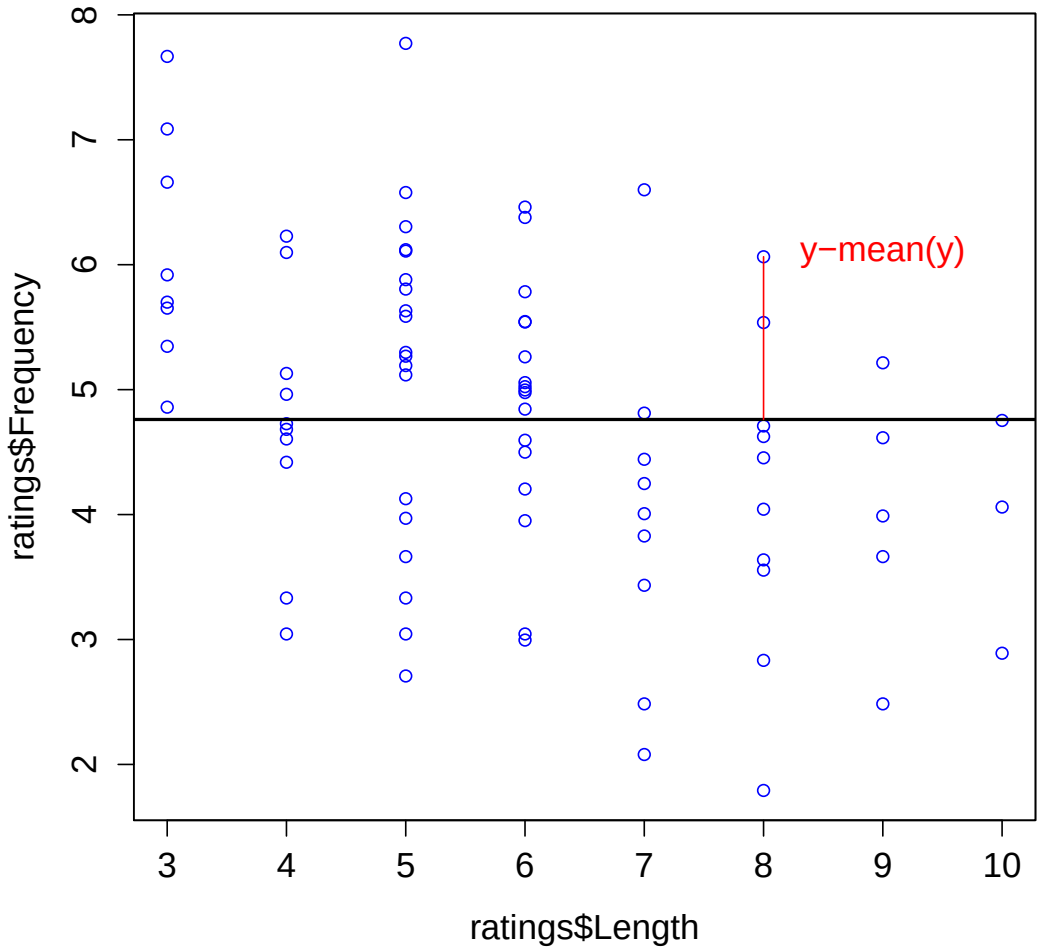
Közönséges legkisebb négyzetek

Kiindulás: (1) kiszámítjuk az y értékek átlagát, (2) minden egyes y érték átlagtól való eltérését (reziduum), és ezek négyzetének összegét.

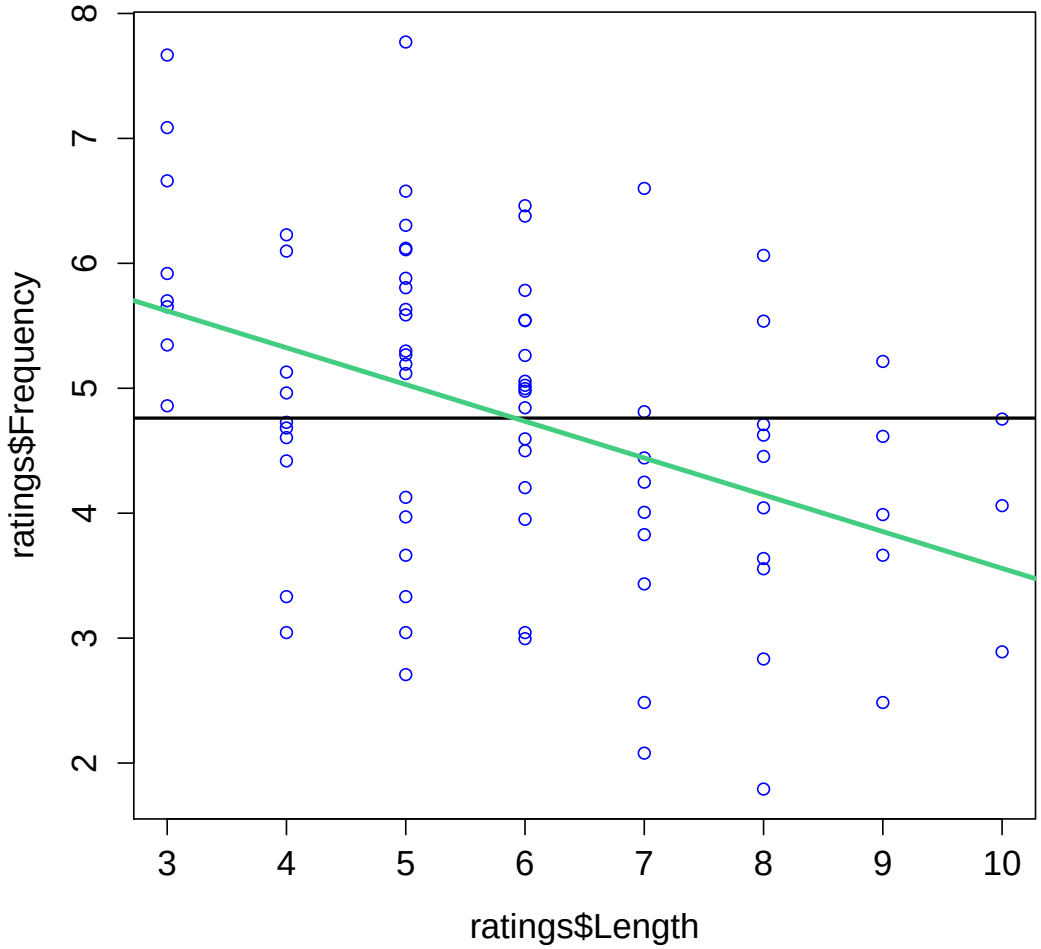


Közönséges legkisebb négyzetek

Kiindulás: (1) kiszámítjuk az y értékek átlagát, (2) minden egyes y érték átlagtól való eltérését (reziduum), és ezek négyzetének összegét.



Keressük azt az egyenest, amelytől az y értékek függőleges négyzetes eltérése (reziduuma, maradéka) a legkisebb.



Egyenes képlete

$$y = a + bx$$

Egyenes képlete

$$y = a + bx$$

Regressziós együtthatók:

a: egyenes metszéspontja az y-tengelyen.
b: egyenes meredeksége.

Keresett érték:

$$OLS = \sum_{k=1}^{n} (y_i - (a + bx_i))^2 = min$$

ahol OLS = *Ordinary Least Square*

Regressziószámítás az R-ben

`lm(függőváltozó~függetlenváltozó)`
kimenet: a és b regressziós együtthatók

Regressziószámítás az R-ben

`lm(függővváltozó~függetlenváltozó)`
kimenet: *a* és *b* regressziós együtthatók

Érdemes az eredményt eltárolni egy változóban, mert így hozzáférünk a számított értékekhez:
`lmcoef = lm(ratings$Frequency~ratings$Length)`

`coef(lmcoef)` vagy `lmcoef$coefficients`: vektor a két együtthatóval.

`fitted(lmcoef)`: az egyeneshez igazított (hipotetikus) *y* értékek.

`resid(lmcoef)`: reziduumok, a hipotetikus *y* értékektől való eltérések.

Egyéb elérhető adatok listázása:
`str(lmcoef)`

Regressziós egyenes ábrázolása

R-függvény:
`abline(intercept,slope)`

1. argumentum: y-tengely metszéspontja, 2. argumentum: meredekség.

Regressziós egyenes ábrázolása

R-függvény:
`abline(intercept,slope)`

1. argumentum: y-tengely metszéspontja, 2. argumentum: meredekség.

```
plot(ratings$Length,ratings$Frequency,cex.axis=1.3,cex.lab=1.3
abline(coef(lmcoef))
```

hiszen a `coef(lmcoef)` parancccsal épp a két szükséges együtthatót kapjuk meg.

Az `abline()` függvény mindig egy már meglévő grafikonokba rajzol egyenest.

Hasznos függvények az ábrázoláshoz

Mindkettő már létrehozott grafikonhoz ad hozzá további információt.
Grafikon koordinátái „ismertek”, és felhasználhatók az elhelyezésben.

`text(x,y,"my text")`: szöveg elhelyezése a grafikonban megadott pozícióban, pl.:
`text(9,6,"y(i)-mean(y)")`
Alapbeállítás: szöveg **középpontja** esik a megadott koordinátákra.

`legend()`: jelmagyarázat
Számos opció, kötelező argumentumok: pozíció
(”center”,”topleft”,”bottom” stb.), magyarázatok:
legend=c(”növény”,”állat”), szín vagy satírozás: col=c(”red”,”blue”), ha lwd (vonalvastagság) definiálva van, akkor vonal kerül elé, és az színes, stb.

Gyakorlás

`ratings` külön a növényekre és az állatokra eltérő színnel: x tengely: név hosszúsága, y-tengely: gyakoriság. Eljárás: először a növények adatpontjait ábrázoljuk, utána

```
par(new=T)
```

majd az állatok adatpontjait hozzáadjuk ehhez az ábrához. Figyelem: a tengelyek terjedelmét manuálisan kell megadni, mert az R automatikusan számolja ki az optimális szélső értékeket, és ezek eltérnek.

Készítsünk jelmagyarázatot (legend) a színek jelentéséről.

Varianciaanalízis (analysis of variance, ANOVA)

Kérdések: (1) van-e különbség a csoportok között (t-próba általánosítása), (2) van-e hatása a vizsgált tényezőnek (regressziószámítás: magyarázó változók hatása a függő változóra).

- **Egy- vs. többtényezős**: ha egy független változó van, egytényezős, ha n, n-tényezős.
- **Független mintás vs. ismételt méréses**: ha az adatok különböző elemeken végzett mérésekből származnak (pl. magyar, cseh és angol beszélők), független mintánk van, ha ha egyazon adatközlőtől többféle adat származik, ismételt méréses dizájnunk van.
- **Egy- vs. többváltozós**: a függő változók száma. ANOVÁ-ban alapértelmezetten egy függő változó van, a MANOVÁ-ban (multivariate ANOVA) legalább kettő.

Alkalmazási területek

- Egy adott kezelés különböző változatainak hatása a kontrollcsoporthoz képest (pl. magasabb dózis, alacsonyabb dózis, placebó).
- Többféle módszer hatékonysága egymáshoz és a kontrollcsoporthoz képest.
- Nominális független változók által kiváltott hatás (pl. mérések különböző napszakokban).

Feltételek

- Egyes csoportokon belül normális eloszlás és
- azonos szórás (varianciák homogenitása),
- megfigyelések egymástól való függetlensége (szfericitás).

Normális eloszlás feltételének megsértését nem szokás sarkalatos problémának tekinteni, mert (1) 30 fölötti elemszám már természetszerűleg normális eloszlású, (2) 10–20 elemnél nem nagy az eltérés, (3) 10-nél kisebb elem esetén nincs igazán értelme eloszlásról beszélni.

Varianciák homogenitása és a megfigyelések egymástól való függetlensége (szfericitás) viszont alapvető, különben az eredmények nem megbízhatóak.

Egytényezős varianciaanalízis

Eljárás: az összes variancia felosztása a faktorok kombinációjából adódó csoportok **közötti** és a csoportokon **belüli** varianciára (innen az elnevezés).

1. csoporton belül: minden egyes csoport varianciája $\rightarrow$ ezek átlaga,
2. csoportok között: minden egyes csoport átlagának varianciája $\rightarrow$ véletlen hiba varianciabecslése = regressziószámítás reziduális varianciája,
3. döntés: ha a **csoportok közötti** variancia nagyobb, mint a **csoportokon belüli** variancia, akkor a tényezőnek (független változónak) van hatása.

Példa

Reiczigel, Harnos & Solymosi, 316. o.: Tápoldat hatékonyságának tesztelése növények növekedésére. Eljárás: növények öntözése tömény, ill. híg tápoldattal, kontroll: víz. Kérdés: serkenthető-e a növények növekedése a tápoldat segítségével?

R-kód:

```
magassag = c(56,48,66,54,57,50,47,58,54,46,60,48)
tapoldat = rep(c("tomeny","hig","viz"),each=4)
novtap = data.frame(magassag,tapoldat)
```

`rep()`: tápoldat típusának ismétlése: opciók: times=4 (teljes sor ismétlése négyszer), each=4 (minden egyes elem ismétlése négyszer).
Fontos: az adatmátrixot a `data.frame()` paranccsal hozzuk létre, ami a *tapoldat* karaktervváltozókat faktorrá alakítja. A faktor egy változótípus, a független változóknak felel meg.

Varianciaelemzés az R-ben

Normális eloszlás tesztelése:
`tapply(novtap$magassag,novtap$tapoldat,shapiro.test)`

`tapply()`: függő változó kiszámítása független változó összes faktorszintjére a megadott függvény szerint, azaz `tapply(függővváltozó,függetlenváltozó(k),függvény)`.
Mindhárom csoport normális eloszlású.

Varianciák homogenitásának ellenőrzése:
`bartlett.test(novtap$magassag,novtap$tapoldat)`: varianciák azonosak.

NB: Bartlett-próba kettőnél több próba összehasonlítására is alkalmazható, de csak normális eloszlás esetén $\leftrightarrow$ `var.test()` (F-próba) csak két mintát tud összehasonlítani. Ha több, nem normális eloszlású próba: `leveneTest()` a `car` könyvtárból.

Varianciaanalízis két függvény alapján:
`aov()`
`lm()`
Különbség: `aov()` csak azonos elemszámú cellák (kiegyensúlyozott elrendezés) esetén alkalmazható. Eltérő csoportelemszámok esetén `lm()` (indoklás Reiczigel et al., 375ff.).

```
h = aov(novtap$magassag~novtap$tapoldat), vagy
h = aov(magassag~tapoldat,data=novtap)
```

Varianciaanalízis két függvény alapján:
`aov()`
`lm()`
Különbség: `aov()` csak azonos elemszámú cellák (kiegyensúlyozott elrendezés) esetén alkalmazható. Eltérő csoportelemszámok esetén `lm()` (indoklás Reiczigel et al., 375ff.).

`h = aov(novtap$magassag~novtap$tapoldat)`, vagy
`h = aov(magassag~tapoldat,data=novtap)`
`summary(h)`.
Kapott F-érték az adott szabadságfokokra nem mutat szignifikáns eltérést a kezelések közötti és kezeléseken belüli átlagos eltérés-négyzetösszegek között $\Rightarrow$ tápoldat alkalmazása nincs hatással a növekedésre.

Igaz ez a víz és a tömény oldat összehasonlítására is?

Post hoc-tesztek

Probléma: az összehasonlítások nagy számával nő az α-hiba lehetősége, azaz annak a valószínűsége, hogy hibás szignifikáns p-értéket kapunk.

Módszerek:

- Páronkénti összehasonlítás t-próbákkal, majd a **Bonferroni-korrektúra** alkalmazása: szignifikancia-határ $\alpha/\frac{k(k-1)}{2}$, azaz konfidenciaintervallum/összes lehetséges párosítás. Hátrány: nagy számú kombináció esetén szinte lehetetlen szignifikáns különbséget kimutatni.
- **Tukey-féle** post-hoc teszt: csak a független mintás varianciaanalízisre alkalmazható, az ismételt mérésesre nem.
- **Dunnett-próba**: általánosabb alkalmazhatóság.

Post hoc-tesztek

1. Tukey-féle post hoc-teszt bemenete az `aov()` kimeneteként kapott objektum:

```
h = aov(novtap$magassag~novtap$tapoldat)
TukeyHSD(h)
```

Post hoc-tesztek

1. Tukey-féle post hoc-teszt bemenete az `aov()` kimeneteként kapott objektum:

```
h = aov(novtap$magassag~novtap$tapoldat)
TukeyHSD(h)
```

Egyik párosítás sem különbözik szignifikánsan.

2. *t*-próba Bonferroni-korrektúrával

Pl. víz és tömény oldat összehasonlítása. Lehetséges kombinációk száma 3, tehát a konfidencia-intervallum határa Bonferroni-korrektúra után $0,005/3 = 0,0167$.

```
hig = novtap$tapoldat == "hig"
t.test(novtap$magassag[!hig]~novtap$tapoldat[!hig]
```

Post hoc-tesztek

1. Tukey-féle post hoc-teszt bemenete az `aov()` kimeneteként kapott objektum:

```
h = aov(novtap$magassag~novtap$tapoldat)
TukeyHSD(h)
```

Egyik párosítás sem különbözik szignifikánsan.

2. *t*-próba Bonferroni-korrektúrával

Pl. víz és tömény oldat összehasonlítása. Lehetséges kombinációk száma 3, tehát a konfidencia-intervallum határa Bonferroni-korrektúra után $0,005/3 = 0,0167$.

```
hig = novtap$tapoldat == "hig"
t.test(novtap$magassag[!hig]~novtap$tapoldat[!hig]
```

p = 0.4462, azaz a különbség messze nem szigifikáns.

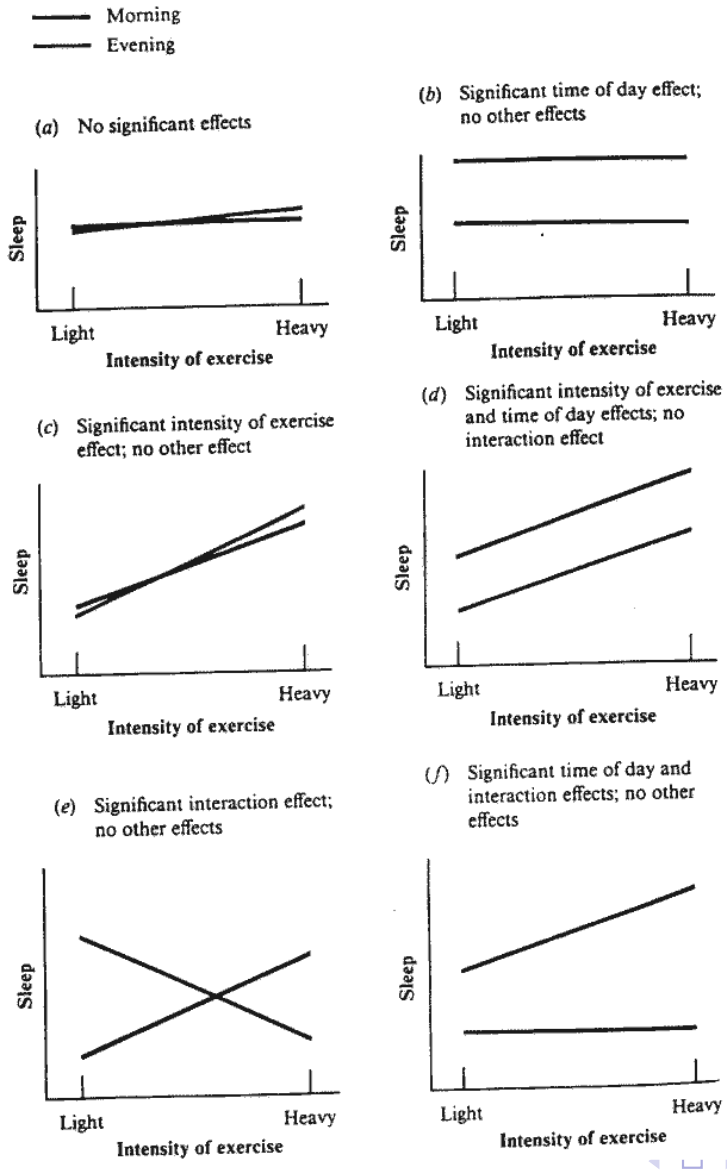
Többtényezős varianciaanalízis

Két vagy több független változó hatása a függő változóra.

Nullhipotézisek: (1) Első tényező (független változó) nincs hatással a függő változóra. (2) Második tényező nincs hatással a függő változóra. (3) Két tényező nincs egymásra hatással, nincs közöttük interakció.

Eljárás: először a két független változó közötti interakciót teszteljük, majd ezek hatását külön-külön.

Interakció


Morning
Evening
(a) No significant effects
Sleep
Light
Heavy
Intensity of exercise
(b) Significant time of day effect; no other effects
Sleep
Light
Heavy
Intensity of exercise
(c) Significant intensity of exercise effect; no other effect
Sleep
Light
Heavy
Intensity of exercise
(d) Significant intensity of exercise and time of day effects; no interaction effect
Sleep
Light
Heavy
Intensity of exercise
(e) Significant interaction effect; no other effects
Sleep
Light
Heavy
Intensity of exercise
(f) Significant time of day and interaction effects; no other effects
Sleep
Light
Heavy
Intensity of exercise

R-kód

Újabb növényeket öntözünk meg tápoldattal és vízzel, de most növényenként két eltérő fajtát tesztelünk.

Kód letölthető innen:
http://biostatkonyv.hu/
R-kódok a 2010-es kiadáshoz, biostat.R, fejezet10.R, 10.3-as példa.
Adatmátrix neve novtap2 legyen.

```
h = aov(magassag~tapoldat*fajta,data=novtap2)
summary(h)
```

R-kód

Újabb növényeket öntözünk meg tápoldattal és vízzel, de most növényenként két eltérő fajtát tesztelünk.

Kód letölthető innen:
http://biostatkonyv.hu/
R-kódok a 2010-es kiadáshoz, biostat.R, fejezet10.R, 10.3-as példa.
Adatmátrix neve novtap2 legyen.

```
h = aov(magassag~tapoldat*fajta,data=novtap2)
summary(h)
```

Tápoldat típusa és fajta nincs hatással egymásra, tehát nincs interakció a két független változó között.

```
h = aov(magassag~tapoldat+fajta,data=novtap2)
summary(h)
```

Egyes p-értékek így még kisebbek.

Értékelés

Döntés H_1 javára: az alkalmazott tápoldat mindkét növényfajta esetében szignifikánsan nagyobb növekedést okoz.
Kérdés: elég-e a két fajta esetében híg tápoldatot alkalmazni a szignifikáns növekedés kiváltásához?

Értékelés

Döntés H_1 javára: az alkalmazott tápoldat mindkét növényfajta esetében szignifikánsan nagyobb növekedést okoz.
Kérdés: elég-e a két fajta esetében híg tápoldatot alkalmazni a szignifikáns növekedés kiváltásához?

Eljárás: 1-es és 2-es fajtára a víz és híg oldat p-értékének összehasonlítása Tukey-féle post hoc-teszttel (összes kombinációt interakciót feltételező modellel kapjuk csak meg).

```
h = aov(magassag~tapoldat*fajta,data=novtap2)
TukeyHSD(h)
```

Értékelés

Döntés H_1 javára: az alkalmazott tápoldat mindkét növényfajta esetében szignifikánsan nagyobb növekedést okoz.
Kérdés: elég-e a két fajta esetében híg tápoldatot alkalmazni a szignifikáns növekedés kiváltásához?

Eljárás: 1-es és 2-es fajtára a víz és híg oldat p-értékének összehasonlítása Tukey-féle post hoc-teszttel (összes kombinációt interakciót feltételező modellel kapjuk csak meg).

```
h = aov(magassag~tapoldat*fajta,data=novtap2)
TukeyHSD(h)
```

	p adj
viz:1-hig:1	0.0181639
viz:2-hig:2	0.0005648

A híg oldat szignifikánsan nagyobb növekedést eredményez mindkét fajta esetében, a tömény és a híg oldat között viszont nem szignifikáns a különbség.

További feladat

ml_vow.RData alapján (letölthető: clara.nytud.hu/~mady/courses/statistics/materials).

Igaz-e az, hogy a felső nyelvállású magánhangzók rövidebbek, mint a középső és alsó nyelvállásúak? (Szükséges oszlopok: dur, hgt.)

Hatással van-e a tartamra a környező mássalhangzó zöngéssége (voi), a magánhangzó-hosszúság (quan), és a magánhangzó minősége (qual)? Melyik tulajdonságok vannak interakcióban egymással?

Az adatok elemzése előtt érdemes a viszonyokat boxplotokon is megszemlélni.

Ismételt méréses módszerek

Humán tudományok örök problémája: egy személytől általában nem egy, hanem többféle adatot gyűjtünk. Ennek elemzésére az egyszerű varianciaanalízis NEM alkalmas, mert ott alapfeltétel a minták függetlensége (ld. független mintás *t*-próba).

A varianciaanalízis függő mintás megfelelője az **ismételt méréses** varianciaanalízis, angolul *repeated measures ANOVA*.

Fontos: az ismételt mérés nem arra vonatkozik, hogy egyazon beszélőtől többször vesszük fel ugyanazt az adatot (pl. mondatokat öt ismétléssel olvasnak fel), hanem hogy **egyazon személlyel** ismételt méréseket végzünk.

Például orvostudományban: egy bizonyos gyógyszer hatása kezelés előtt, a kezelés megkezdése után két héttel, egy hónappal stb.

Eljárás

Egy függő és egy vagy több független változó tesztelése, ahol az ismétlés **belső** tényezői (személyek, növények, akiken/amiken az ismételt méréseket végeztük) közötti különbséget **véletlen** hatásnak tekintjük (*within-subject factor*).
Az alanyok lehetnek két különböző csoport tagjai, amiket összehasonlítunk (pl. különböző nyelvek beszélői, egy növényfaj különböző fajtái stb.), ez a **köztes** tényező (*between-subject factor*).
Alapfeltételek:

- legalább öt alany (személy, növény, tárgy, bármi, amin több mérést végzünk),
- faktorkombinációnként egyetlen adat - azaz ha egyazon faktort többször mértünk (pl. felolvasáskor több ismétlés), ezeket átlagolni kell minden egyes alanyra és cellára,
- kiegyensúlyozott dizájn, azaz ha az egyik faktor két szintjéhez két további faktor tartozik, akkor a másik faktornál is vizsgálni kell ugyanezt a két szintet.

Hátulütők

- R-ben nincs több faktor kombinációjára átlagoló beépített függvény,
- mivel átlagokkal számolunk, az egyes cellákon belüli varianciát nem tudjuk figyelembe venni (erre a *mixed models* kínál kiutat),
- nem tudunk több *within-subject* tényezőt kombinálni ($\rightarrow$ *mixed models*),
- csak a szfericitási feltétel teljesülése esetén alkalmazható ($\rightarrow$ ismételt méréses többváltozós varianciaanalízis, lásd jövő órán)
- nincs post-hoc tesztje, csak *t*-próbák Bonferroni-korrektúrával (konfidenciaszint/összes lehetséges kombináció száma).

A *mixed models* ld. Baayen (2008): *Analizing linguistic data* c. könyvéből, pdf elérhető itt:
http://www.ualberta.ca/ baayen/publications.html, 2008-as publikációk.

Cellánkénti átlagok számítása

`anova.mean.r` nevű R-függvény letöltése innen:
clara.nytud.hu/~mady/courses/statistics/materials

Szkript és függvény közötti különbség: függvényben létrehozott változók (R-objektumok) nem jelennek meg a munkamemóriában. Szkript és függvény egyaránt betölthető a `source("eleresiutvonal")` paranccsal, a szkriptet közvetlenül be is lehet másolni egy szövegszerkesztőből az R-be (copy-paste).

Ha a függvényben szintaktikai hiba van, betöltés helyett hibajelzést kapunk.

Függvény első sora:
`fuggvenynev = function(kotelezoargumentum1, kotelezoargumentum2, ...)`, ahol három pont további opcionális számú opcionális argumentumot jelöl.

Példa

Mondatvégi kétszótagú, /s/-re és /z/-re végződő szavakban megmértük a réshangon belüli zöngés tartomány hosszát. Zöngésebbek-e a mondatvégi /z/-k, mint az /s/-ek?

zfin.RData, letöltés innen:
clara.nytud.hu/∼mady/courses/statistics/materials

```
zmean = anova.mean(zfin$cvoice,zfin$subj,zfin$voiced)
```

Kapott adatmátrix oszlopainak elnevezése:

```
names(zmean) = c("cvoice","subj","voiced")
```

Ismételt méréses varianciaanalízis függvénye

- Függő változó: mássalhangzó zöngésségének tartama (cvoice).
- Független változó: zöngésség (voiced).
- Within-subject factor: beszélő (subj).
- Between-subject factor: nincs.

```
summary(aov(cvoice~ voiced + Error(subj/voiced),
data=zmean))
```

Releváns *p*-érték: `Error: subj:voiced` sor alatt (ez jelzi az alanyokon belüli interakciót).

Ábrázolás:

```
interaction.plot(x-tengely, ismételt_mérés_alanya,
paraméter)
interaction.plot(zmean$voiced,zmean$subj,zmean$cvoice)
```

Több tényező

Többtényezős varianciaanalízis képlete, ha nincs *between-subject factor*, pl. ha megelőző mássalhangzóra is kíváncsiak vagyunk:

```
summary(aov(cvoice ~ voiced*c1 + Error(subj/(voiced*c1)),
data=zmean))
```

Ehhez a cellánkénti átlagokat újra kell számolni:

```
zmean = anova.mean(zfin$cvoice, zfin$subj, zfin$voiced,
zfin$c1)
```

Eredmények

Értelmezés:

`Error: subj:voiced` zöngésségi tartamok beszélőnként, zöngésség függvényében (a *p*-érték változott, mert az átlagokat újraszámoltuk).

`Error: subj:c1` zöngésségi tartamok beszélőnként, a megelőző mássalhangzó függvényében.

`Error: subj:voiced:c1` zöngésségi tartamok beszélőnként, zöngésség és megelőző mássalhangzó interakciója, azaz befolyásolja-e a megelőző mássalhangzó a zöngésség hatását?

Több csoport (*between-subject factors*)

Férfi és női beszélők magánhangzónak 1. és 2. formánsa alapján kiszámoltuk az egyes magánhangzók artikulációs középponttól való távolságát (euklideszi távolság). Erősebben redukálnak-e a férfiak, mint a nők, azaz közelebb vannak-e a magánhangzóik a középponthoz?

Adatok: euk.RData, letöltés: clara.nytud.hu/∼mady.

```
summary(aov(ET ~V.num * nem + Error(beszelo/V.num),
data=euk))
```

beszélők csoportjára nem kapunk *p*-értéket. Miért?

Több csoport (*between-subject factors*)

Férfi és női beszélők magánhangzónak 1. és 2. formánsa alapján kiszámoltuk az egyes magánhangzók artikulációs középponttól való távolságát (euklideszi távolság). Erősebben redukálnak-e a férfiak, mint a nők, azaz közelebb vannak-e a magánhangzóik a középponthoz?

Adatok: euk.RData, letöltés: clara.nytud.hu/∼mady.

```
summary(aov(ET ∼V.num * nem + Error(beszelo/V.num),
data=euk))
```

beszélők csoportjára nem kapunk *p*-értéket. Miért?

Mivel a kódolás számokkal történik, R az adatokat egész számokként (azaz numerikus változóként) értelmezi. Független változó csak faktor lehet! Változót át kell kódolni faktorrá, hogy fusson rajta a varianciaanalízis függvénye.

```
euk$nem = as.factor(euk$nem)
euk$V.num = as.factor(euk$V.num)
```

Gyakorlás

Egy szelektív hulladékgyűjtésre nevelő kampányban hét család vesz részt. A kampány kezdetekor, valamint három és hat hónappal később megmérik az egyhavi hulladéktermelést, ami nem a szelektív gyűjtőkbe kerül. Járt-e hulladékcsökkentő hatással a kampány?

Letöltés: szemet.RData

Feladat: ismételt méréses varianciaanalízis számolása, egyéni trendek megjelenítése az `interaction.plot()` függvénnyel.

A családok beszélők egy része családi házban lakik, mások társasházban. Van-e különbség a csoportok között?