

Adatkezelés és egy változós elemzések R-ben

Abari Kálmán

2021-12-28

Tartalomjegyzék

Üdvözöljük	5
Előszó	7
1. Itt kezdődik	9
1.1. Elindulás	9
1.2. A könyv felépítése	14
1.3. Próbák listája	15
2. Mi az R?	19
2.1. Az R bemutatása	19
2.2. A modern R	23
2.3. Múlt és jelen	25
3. Az R telepítése	29
3.1. A fő komponensek telepítése	29
3.2. A Tidyverse R telepítése	33
3.3. Az R frissítése	34
4. Munka az R-ben	37
4.1. Az RStudio használata	37
5. Az R nyelv	39
5.1. Adatobjektumok	40
6. Beolvasás	41
6.1. Alapvető formátumok	41
7. Adatmanipuláció	43
7.1. Adatkezelés az Alap R-ben	43
8. Leíró statisztika	45
9. Grafika az R-ben	47

10. Hipotézisvizsgálatok	49
11. Publikáció	51
11.1. Reprodukálható kutatás	51
A Megoldások	53
A.1. Megoldások az 1.1.2 feladatokhoz	53
B Színek	55
B.1. Az előre definiált paletta színei	55
B.2. Színek választása az RColorBrewer csomag segítségével	57
B.3. Színek választása a dichromat csomag segítségével	79
B.4. Színek választása egyéb paletta segítségével	88
B.5. A 657 színnév	93
C Óravázlat az R tanításához	99
1. félév	99
2. félév	101

Üdvözöljük

Ez a honlap az *Adatkezelés és egy változós elemzések* c. könyv elektronikus változatát mutatja be. A nyomtatásban megjelent könyvhöz képest számos bővítést tartalmaz:

- a függelék fejezetei, például a kitűzött feladatok megoldásai is itt jelennek meg,
- az R grafikus lehetőségeit tartalmazó fejezet bővebb a *hagyományos grafika* résszel.

A teljes könyv, az adatbázisok és az R kódok megtalálhatók a következő címen¹.

A könyvet Máth János lektorálta, és Friss Kinga illusztrálta.

¹ <https://abarik.github.io/aeer>

Előszó

Kedves Olvasó!

Köszönjük, hogy bizalmat szavaz könyvünknek, és az R megisméréséhez ezt az utat választja. Az első lépések től a komplett adatelemzési feladatok megoldásáig vezetjük az Olvasót, és főként kezdő vagy újrakezdő felhasználókhoz szólunk. Utunk során áttekintjük az adatfeldolgozás minden lépését: az adatok beolvasását, előkészítését, elemzését és az eredmények publikálását is.

Könyvünk összesen 11 fejezetet tartalmaz. Az egyes fejezeteket alkotó alfejezeteket három különböző ikon egyikével jelöltük meg, amelyek jelzőtáblaként szolgálnak az R megismerésének útján. Az egyes ikonok jelentése a következő:



Egy hegység. Az így jelölt fejezet az R alaptudás része, megismerése feltétlenül javasolt. A könyvben megfogalmazott célok ezen fejezetek megismerésével is elérhetők, azaz komplett adatelemzések hajthatók végre csupán ezek végig olvasásával is.



Két hegység. Kiegészítő tudást tartalmazó fejezetek. Újabb eszközök megismerését teszik lehetővé, és/vagy hozzájárulnak az *egy hegység* fejezetek mélyebb megértéséhez.



Három hegység. Az R ismeretek további részletezése, a meglévő eszközök finomabb kezelése, vagy további beállítási lehetőségek olvashatók ezekben a fejezetekben. Elképzelhető, hogy ritkábban felmerülő problémák megoldásához kapunk itt segítséget.

A fejezetek hármas tagolása azt a célt szolgálja, hogy minél hamarabb örömet és sikert okozhasson az R használata, ugyanakkor további olvasással a részletesebb ismeretek utáni vágyunkat is kielégíthessük. **Könyvünk olvasását tehát az 1. fejezet *egy hegység* alfejezetével (1.1 Elindulás) érdemes kezdeni,** ott kapunk ajánlást a folytatásra. A további fejezetek olvasási sorrendje teljes

mértékben az elvégzendő feladattól, tudásunktól és kíváncsiságunktól függ.

A fejezetek végén összefoglaljuk a tanultakat. Megismétljük a legfontosabb fogalmakat és felsoroljuk a megismert függvényeket.



Összefoglalás. Miről szólt az előszó? A könyv 11 fejezetet tartalmaz, az alfejezeteket három különböző ikon egyikével jelöltük: *egy hegypályát*, *két hegytérképet* vagy *három hegytérképet*.



Feladatok. A fejezet végi feladatok megoldásával jelentősen hozzájárulunk a magabiztos R tudás megszerzéséhez. Találunk szórakoztatónak és érdekes feladatokat is.

Örömmel fogadjuk Olvasóink észrevételeit az abari.kalman@gmail.com címen.

1. fejezet

Itt kezdődik



1.1. Elindulás



Ebben a fejezetben:

- bemutatunk egy konkrét adatelemzési példát,
- áttekintjük a könyv tartalmát,
- lehetőséget adunk az előzetes R ismeretek felmérésére,
- és segítünk a megfelelő fejezet kiválasztására a folytatáshoz.

Könyvünk elsődleges célja az R bemutatása kezdő felhasználók számára, de minden bizonnal azok is találni fognak hasznos részeket, akik már rendelkeznek R ismeretekkel. Bevezetést nyújtunk az R által lefedett három nagy terület mindegyikébe: az adatkezelésbe, a grafikus megjelenítésbe és az adatelemzésbe is. A leírtak megértéséhez a statisztikai alapismereteken túl semmilyen előzetes tudás nem szükséges.

Most egy konkrét adatelemzési példa segítségével bemutatjuk, hogy mit nyújt e könyv az Olvasó számára. A bevezető példa megoldása során az előismeretekkel rendelkező Olvasó a saját R tudását is felmérheti, és ezzel egyben segítséget kap-

hat a tudásához és céljaihoz legjobban illeszkedő fejezet kiválasztására, amellyel tovább folytathatja az olvasást.

Bevezető példa: Két tanítási módszer összehasonlítása

Egy 2020-as kutatásunkban (Csapó et al., 2020) 7. osztályos tanulóknak Excel ismereteket oktattunk két különböző megközelítésben. Az egyik csoportban hagyományos, míg a másikban modern (Sprego) tanítási módszert használtunk. A tanulási időszak az Excel ismeretek felmérésével zárult. Az összegyűjtött adatok az `excel_2020.xlsx` állományban állnak rendelkezésre.

Nézzük az adatelemzés lépéseit és egyben könyvünk felépítését!

2. fejezet: Mi az R?

A bevezető példa megoldását R-ben fogjuk elvégezni (és nem más eszközben, mint például az SPSS, jamovi, JASP, SAS stb.). Érdemes tehát ismerni az R céljait és lehetőségeit, jó ha van egy összképünk a használt statisztikai programról. Ezt az áttekintést nyújtja 2 fejezet.

3. fejezet: Az R telepítése.

Az adatelemzés konkrét lépéseinak elvégzéséhez telepített *Alap R* és *RStudio* szükséges. Ha ezek nem állnak rendelkezésre, vagy még nem is találkoztunk ezekkel az eszközökkel, akkor a 3. fejezet nekünk szól.

4. fejezet: Munka az R-ben.

Az adatelemzés végrehajtásához az *RStudio*-t ajánljuk, és azon belül pedig a projektek használatát szorgalmazzuk. A 4. fejezetben megismерjük az *RStudio* legalapvetőbb funkcióit, a parancsállományok létrehozását és futtatását.

A fenti előzmények után elkezdhetjük a bevezető példa megoldását:

1. indítsuk el az *RStudio*-t,
2. hozzunk létre egy új projektet,
3. hozzunk létre egy új RMarkdown állományt,
4. helyezzük el a lentebb szereplő R parancsokat az RMarkdown állomány egyes csonkjaiban.

5. fejezet: Az R nyelv.

Az R parancsok létrehozásának vannak szabályai, amelyeket a munka során be kell tartanunk. Ismernünk kell jó néhány függvényt, és általában el kell tudnunk igazodni az R nyelvben. Az 5. fejezet ezért kulcsfontosságú, tanulmányozzuk alaposan, és lehetőleg minden kitűzött feladatát oldjuk meg.

6. fejezet: Beolvasás

Az adatelemzés első lépése az adatállomány beolvasása. Adataink változatos formában állhatnak rendelkezésre, a 6. fejezetben ezek beolvasására kapunk receptet.

A bevezető példa megoldásához az RMarkdown állomány egyik csonkját bővítsük a lenti sorokkal.

```
# install.packages("rio")                      # rio csomag telepítése
library(rio)                                    # rio csomag betöltése
felmeres <- import(file = "adat/excel_2020.xlsx") # beolvasás
```

7. fejezet: Adatkezelés

A statisztikai elemzés elkezdése előtt számos adatkezelési tevékenységre lehet szükség. Ezt a sokszor rendkívül időigényes folyamatot a 7. fejezetben részletezzük.

A bevezető példa megoldásához az RMarkdown állomány egyik csonkját bővítük a lenti sorokkal. Az adatkezelés legtöbbször a beolvasott állomány *jellemzőinek* lekérésével kezdődik.

```
str(felmeres)          # dataframe szerkezete
names(felmeres)         # változónevek
unique(felmeres$modszer) # különböző értékek - osztaly
```

A karakteres vagy numerikus vektorok faktorrá konvertálása az egyik leggyakoribb előkészítő parancs.

```
felmeres$modszer <- factor(felmeres$modszer)
```

A táblázatok és ábrák megfelelő megjelenéséhez, végezzük el a *faktorschintek sorrendbe állítását*.

```
felmeres$modszer <- factor(felmeres$modszer, levels=c("modern", "hagyományos"))
```

8. fejezet: Mutatók és táblázatok.

Ha az adatainkat már megfelelő formába hoztuk, akkor továbbléphetünk az elemzés felé. A 8. fejezet a leíró statisztikai elemzésekben a mutatók és a táblázatok létrehozását mutatja be.

Most a felmérés eredményeinek statisztikai mutatóit íratjuk ki a két tanítási módszert használó csoportban.

```
# install.packages("psych") # psych csomag telepítése
psych::describeBy(x = felmeres$eredmeny, group = felmeres$modszer, mat=T)
#>      item      group1 vars   n     mean      sd   median   trimmed      mad
#> X11    1       modern   1 13  0.6541958 0.1968892 0.6621212 0.6568871 0.2044191
#> X12    2 hagyományos  1 13  0.3800117 0.1272075 0.3628788 0.3913912 0.1190573
#>           min      max     range      skew kurtosis      se
#> X11 0.32575758 0.9530303 0.6272727 -0.2885768 -1.189491 0.05460724
#> X12 0.08333333 0.5515152 0.4681818 -0.6434813 -0.114030 0.03528102
```

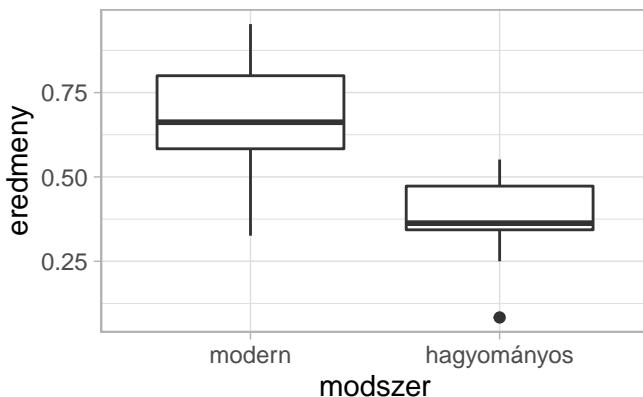
9. fejezet: Grafika.

A grafikus megjelenítés is a leíró statisztikai elemzés része. A 9. fejezetben részletesebben olvashatunk a publikációkész ábrák létrehozásáról.

Most a numerikus változók esetén használt egyik elterjedt ábrázolási formát, a

dobozdiagramot használjuk két tanítási csoport eredményének grafikus összehasonlítására.

```
library(ggplot2)
ggplot(data = felmeres, mapping = aes(x=modszer, y=eredmeny)) + geom_boxplot()
```



10. fejezet: Hipotézisvizsgálatok.

A statisztikai hipotézisvizsgálat minden adatelemzés központi része, a gyűjtött adatokból a populációra nézve következtetést vonhatunk le. A 10. fejezetben a leggyakoribb egy változós elemzéseket mutatjuk be.

Most Mann-Whitney-próbát hajtunk végre a két tanítási módszer eredményességének összehasonlítására.

```
wilcox.test(eredmeny~modszer, data=felmeres)
#>
#> Wilcoxon rank sum exact test
#>
#> data: eredmeny by modszer
#> W = 145, p-value = 0.001248
#> alternative hypothesis: true location shift is not equal to 0
```

11. fejezet: Publikálás.

Az elemzés eredményének publikációkész formába alakítása az adatelemzési folyamat utolsó lépése. A 11. fejezetben megismerjük azokat a legegyszerűbb folyamatokat, amelyekkel többnyire formanyelvtől függetlenül, publikációkész eredményközlést végezhetünk.

A bevezető példa elemzése során kapott eredmények publikálását segíti, ha a következő sorokkal kiegészítjük a fenti, leíró statisztikai elemzéseket. Egy soronként vett kétdimenziós gyakorisági táblázat és egy háttérírra mentett ábra a lenti sorok eredménye. Végül fordítsuk le az RMarkdown állományt a **Knit** nyomógomb segítségével. A kapott HTML (vagy Word/PDF) állomány és a háttéríráron keletkezett képalomány jelentősen segíti az eredmények publikálását.

```
knitr::kable(t(psych::describeBy(x = felmeres$eredmeny, group = felmeres$modszer,
  mat = T)), digits = 3, align = c("c", "c"), format.args = list(decimal.mark = ","))
```

	X11	X12
item	1	2
group1	modern	hagyományos
vars	1	1
n	13	13
mean	0.6541958	0.3800117
sd	0.1968892	0.1272075
median	0.6621212	0.3628788
trimmed	0.6568871	0.3913912
mad	0.2044191	0.1190573
min	0.32575758	0.08333333
max	0.9530303	0.5515152
range	0.6272727	0.4681818
skew	-0.2885768	-0.6434813
kurtosis	-1.189491	-0.114030
se	0.05460724	0.03528102

A bevezető példa megoldásához természetesen a hipotézisvizsgálat szöveges értékelés is hozzátarozik, de ezt most az alfejezet végén szereplő egyik kitűzött feladatra halasztjuk. A hangsúly a könyv vázlatos tartalomjegyzékének bemutatásán volt, részletesebb, de felsorolásszerű tartalomjegyzéket a következő két alfejezetben találunk.

1.1.1. Összefoglalás



Ebben az alfejezetben egy adatelemzési példát oldottunk meg, melynek segítségével illusztrálni tudtuk a további alfejezetek tartalmát. A 2. fejezetben áttekintést fogunk adni az R-ről, a 3.-ban az *Alap R* és *RStudio* telepítését, a 4.-ben az *RStudio* használatát mutatjuk be. Az 5. fejezetben kellő részletességgel ismertetjük az R nyelvet. A további fejezetekben az adatelemzés szokásos lépései tesszük sorra, a 6. fejezetben a beolvasást, a 7. fejezetben az adatok előkészítését, a 8. és 9. fejezetben a leíró statisztikai műveleteket mutatjuk be. A 10. fejezet az egy- és kétváltozós hipotézisvizsgálatoké, az utolsó 11. fejezet az eredmények publikálását foglalja össze.

1.1.2. Feladatok



- Milyen online vagy nyomtat könyvek segítik az R elsajátítását?
Próbáljuk összegyűjteni a magyar nyelvű könyveket is!

2. Térképezzük fel az online videókurzusokat is az R tanulásához!
3. A bevezető példa (*Két tanítási módszer összehasonlítása*) megoldásában a hipotézisvizsgálat alapján adjunk szöveges értékelést!

1.2. A könyv felépítése



Ebben a fejezetben:

- bemutatjuk a könyv részletes felépítését,
- ezzel tovább segítjük a választást a folytatáshoz.

A könyv 11 fejezetből áll, és fejezetenként 3 vagy több alfejezetből. Most röviden bemutatjuk az egyes alfejezetek tartalmát.

Fejezet/alfejezet	Leírás
1. Itt kezdődik	
1.1. Elindulás	A könyv fejezeteinek bemutatása egy konkrét adatelemzésen keresztül
1.2. A könyv felépítése	A könyv egyes alfejezeteinek rövid bemutatása (jelen alfejezet)
1.3. Próbák listája	A könyvben szereplő egy- és kétváltozós statisztikai eljárások listája
2. Mi az R?	
2.1. Az R bemutatása	A parancssoros R jellemzői, az R nyelv, az <i>Alap R</i> és a csomag fogalma
2.2. A modern R	Megtanuljuk a <i>Tidyverse R</i> fogalmát, megtudjuk mi a modern R
2.3. Múlt és jelen	Az R rövid története, alapelvek az R tanulásához, és az R alaptudás elemei
3. Az R telepítése	
3.1. Az Alap R és az RStudio telepítése	Megismерjük az <i>Alap R</i> és az <i>RStudio</i> telepítését
3.2. A Tidyverse R telepítése	A <i>tidyverse</i> csomag(gyűjtemény) telepítése
3.3. Az R frissítése	Az <i>Alap R</i> , az <i>RStudio</i> és a csomagok frissítésének módszerei
4. Munka az R-ben	
4.1. Az RStudio használata	Az <i>RStudio</i> jellemzői és felépítése, a projektek használata és a segítségkérés
4.2. Segítség az R használatához	Segítségkérési lehetőségek az R-ben, a beépített súgó használata

Fejezet/alfejezet	Leírás
4.3. Az Alap R használata	Az <i>Alap R</i> konzola, az <i>R Gui</i> , az <i>R Commander</i> és a kötegelt üzemmód lehetőségei
5. Az R nyelv	
6. Beolvasás	
7. Adatmanipuláció	
8. Mutatók és táblázatok	
9. Grafika	
10. Hipotézisvizsgálatok	
11. Publikáció	



Összefoglalás

Ebben a részben röviden bemutattuk a könyv összes alfejezetét. A kézőbbiekben térképként használhatja az Olvasó az itt ismertetett táblázatot.



Feladatok

1. Az adatfeldolgozás 4 lépése a következő: (1) adatok beolvasása, (2) adatok előkészítése elemzésre, (3) adatok elemzése és (4) az eredmények publikálása. A könyv mely fejezetei tartoznak az adatfeldolgozás fenti lépéseihez?
2. Az R-rel való munka általunk javasolt módja: *RStudio*-ban, projektmódban, R vagy RMarkdown állományokat szerkesztünk és hajtunk végre. Mely fejezetekben találunk hasznos információkat az R ezen használatával kapcsolatban?

1.3. Próbák lista



Ebben a fejezetben:

- áttekintést adunk az egy- és kétváltozós hipotézisvizsgálatokról.

A 10. fejezetben bemutatjuk az egy- és kétváltozós hipotézisvizsgálatok végrehajtását. Ebben a fejezetben felsoroljuk a legfontosabb próbákat, összesen öt táblázatban soroljuk fel őket:

- egy mintát vizsgáló próbák ([1.2 táblázat](#)),
- páros mintát vizsgáló próbák ([1.3 táblázat](#)),
- két független mintát vizsgáló próbák ([1.4 táblázat](#)),
- több összetartozó mintát vizsgáló próbák ([1.5 táblázat](#)),
- több független mintát vizsgáló próbák ([1.6 táblázat](#)).

A táblázatokban megadjuk, hogy a vizsgálatnak mi a célja, vagyis a populációbeli változó(k) melyik paraméterére vonatkoznak a próbák, a várható értékre, a mediánra, a varianciára vagy a valószínűségre. A 10. fejezetben foglalkozunk az eloszlásvizsgálatok közül a normalitást ellenőrző próbákkal is, így a [1.2 táblázat](#) ezeket is számba veszi.

1.2. táblázat: Egy minta vizsgálata

Cél	Próba neve	R függvény
várható érték	egymintás u-próba	<code>BSDA::z.test()</code>
	egymintás t-próba	<code>t.test()</code>
medián	előjel-próba	<code>BSDA::SIGN.test()</code>
	Mood-féle medián-próba	
variancia	egymintás Wilcoxon-próba	<code>wilcox.test()</code>
	khí-négyzet próba	<code>chisq.test()</code>
valószínűség	khí-négyzet próba	<code>chisq.test()</code>
	Shapiro-Wilk próba	<code>shapiro.test()</code>
normalitás	Kolmogorov-Szmirnov próba	<code>DescTools::LillieTest()</code>

1.3. táblázat: Páros minta vizsgálata

Cél	Próba neve	R függvény
várható érték	páros t-próba	<code>t.test(paired=T)</code>
medián	páros előjel-próba	<code>BSDA::SIGN.test()</code>
	páros Wilcoxon-próba	<code>wilcox.test(paired=T)</code>
variancia		<code>var.test()</code>
valószínűség	McNemar-próba	<code>mcnemar.test()</code>
	Cohran-Q próba	<code>mcnemar.test()</code>

1.4. táblázat: Két független minta vizsgálata

Cél	Próba neve	R függvény
várható érték	kétmintás u-próba	<code>BSDA::z.test()</code>
	kétmintás t-próba	<code>t.test()</code>
	Welch-féle d-próba	<code>t.test(var.equal=F)</code>
medián	Mann–Whitney-próba	<code>wilcox.test()</code>
variancia	F-próba	<code>var.test()</code>
valószínűség	khí-négyzet próba	<code>chisq.test()</code>
	Fisher-féle egzakt próba	<code>fisher.test()</code>

1.5. táblázat: Több összetartozó minta vizsgálata

Cél	Próba neve	R függvény
várható érték	egyszempontos összetartozó	<code>ez::ezANOVA()</code>
	mintás varianciaelemzés	
medián	Friedman-próba	<code>friedman.test()</code>

1.6. táblázat: Több független minta vizsgálata

Cél	Próba neve	R függvény
várható érték	egyszempontos varianciaelemzés	<code>aov()</code>
	Welch-féle egyszempontos varianciaelemzés	<code>oneway.test(var.equal=F)</code>
	Kruskal–Wallis-próba	<code>kruskal.test()</code>
medián	Levene-próba	<code>DescTools::LeveneTest()</code>
	Bartlett-próba	<code>bartlett.test()</code>

**Összefoglalás**

Ebben a részben rövid áttekintést adtunk a könyv 10. fejezetében sorra kerülő statisztikai próbákról. Megneveztük a próbákat, R parancsokkal szemléltettük használatukat, valamint jeleztük a céljukat. A táblázatok áttekintésével képet kaphatunk arról, hogy a későbbiekben milyen jellegű statisztikai következtetéseket tudunk levonni az R használatával.

**Feladatok**

1. minden statisztikai próba esetében négy dolgot érdemes tudni:
(1) a statisztikai próba neve, (2) null- és ellenhipotézise, (3) alkalmazási feltételei, és (4) a próba végrehajtásának módja valamely statisztikai programcsomagban. A 10. fejezetben a statisztikai próbák végrehajtását természetesen R-beli eszközökkel mutatjuk be. Ismerjük a fenti táblázatokban megnevezett próbák null- és ellenhipotézisét, valamint az alkalmazási feltételeit? Próbáljuk ezeket felidézni! Hol találunk ezekről információt?
2. Mely próbák maradtak ki ebből a könyvből? Hol találunk ezek R-beli végrehajtására példát?

2. fejezet

Mi az R?



2.1. Az R bemutatása



Ebben a fejezetben:

- megismerjük az R jellemzőit,
- megtudjuk, hogy melyek a parancssoros interfész előnyei,
- megismerjük az *Alap R* fogalmát,
- körülhatároljuk az R nyelv, az *Alap R* és a csomag fogalmát.

2.1.1. Az R jellemzői

Az R egy magas szintű programozási nyelv és környezet, amelynek legfontosabb felhasználása az adatelemzés és az ahhoz kapcsolódó grafikus megjelenítés. Hárrom alapvető jellemzője kiemeli a többi statisztikai programcsomag közül: (1) az R ingyenesen telepíthető és használható; (2) az R nyílt forrású, így bárki hozzájárulhat az R fejlesztéséhez, azaz létrehozhat új *csmagokat*, és ezzel ki-egészítheti az R tudását; és (3) az R felhasználók rendkívül aktív és befogadó online közösséget alkotnak, szinte minden felmerülő kérdésünkre hamar választ kaphatunk.

Álljon itt egy bővített lista azokról a jellemzőkről, amelyek vonzóvá tehetik számunkra az R statisztikai programcsomagot.

- Az R szabad szoftver, bárki ingyenesen letöltheti és használhatja. Ez egy-felől megkönnyíti az oktatási intézmények, tanszékek és oktatók munkáját, hiszen nincs szükség a kereskedelmi programok licenszeléséből adódó pénzügyi vagy más természettű nehézségek kezelésére. Másrészről a hallgatók a statisztika kurzusok során tanultakat otthon vagy később a munkájukban is felhasználhatják.
- Az R platform-független, azaz Windows, Linux és macOS környezetben is használható. Nem kell lemondanunk a kedvenc operációs rendszerünkéről, ha az R-t szeretnénk használni.
- Az R egy teljes értékű programozási nyelv, nem csak egy statisztikai programcsomag önmagában.
- Az R statisztikai módszerek szinte végtelen választékát kínálja. A R-ben felhasználható statisztikai eljárásokat statisztikusok fejlesztik folyamatosan és csomagok formájában teszik elérhetővé. Valószínű, hogy egy új statisztikai módszer leghamarabb az R-ben válik elérhetővé.
- Az R rendkívül gazdag grafikus lehetőségekkel rendelkezik.
- A statisztikai szakirodalomban és az egyetemi oktatók körében egyre elterjedtebb az R, mint közös (statisztikai program)nyelv használata. Ha valamilyen statisztikai problémára keressük a megoldást, vagy csak konszultálunk egy statisztikussal, az R ismerete (akár csak olvasási szinten) rendkívüli előnyt jelenthet.
- Az R igen jól dokumentált, a beépített súgón kívül számos könyv és leírás elérhető el.
- Az R parancssoros interfésszel rendelkezik, amely számos előnnyel jár. Egy rész a szkript állományok létrehozása és végrehajtása a statisztikai elemzések megismételhetőségét biztosítja, másrészről ez az oktatók és a hallgatók könnyebb kommunikációját is lehetővé teszi.
- Az R az adatalemzés eredményének sokszínű publikálását is biztosítja. Az R Markdown¹ formanyelv segítségével HTML, PDF és Word dokumentumot, illetve prezentációs diákat vagy akár kész cikkeket hozhatunk létre. A Shiny² csomag interaktív Webes alkalmazások építését teszi lehetővé.
- Mára az R használata szinte egyet jelent az ingyenesen elérhető RStudio³ használatával, amely egy kényelmes integrált fejlesztői környezetet biztosít a parancsállományok létrehozásához.

Érdemes bepillantani az R árnyékosabb oldalába is. Az R egyik gyengesége, hogy nagy adatbázisok kezeléséhez erős hardverre van szüksége, de a legtöbb felhasználás során ez semmilyen problémát nem okoz. A másik gyengeség, hogy az R elsajátításához nem kevés idő és kitartás szükséges. Jelen könyv éppen ezt a folyamatot kívánja megkönnyíteni és lerövidíteni.

¹ <https://rmarkdown.rstudio.com/>

² <http://shiny.rstudio.com/>

³ <https://www.rstudio.com/>

2.1.2. A R parancssoros

Az R alapvető használata során parancsokat gépelünk be és hajtunk végre. Ez lényegesen eltér a ma megszokott felhasználói programok világától, ahol egy grafikus felhasználói felületen egérrel vagy az ujjunkkal mutogatjuk el a kívánt tevékenységet. Az R egészen más megközelítést vall, használata a kezdeti lépésektről nagyfokú figyelmet és pontosságot követel. A parancsokban kell gondolkodnunk, ám ezt végig áthatja a *tudom mit csinálok* elv, így némi idő elteltével érezni fogjuk, hogy az R megszelídül, már nem köt bele minden „szavunkba”, egyre több dologra tudjuk rávenni, és végül egy rendkívül értékes társsá válik. Jelen könyv ezen az úton szeretné végigvezetni az Olvasót.

Már a tanulás elején szeretnénk tisztázni, hogy az R elsajátításához nem szükséges programozói alaptudás. Az R felhasználók többsége egyáltalán nem programozó, és a minden nap adatelemző munka sem igényli az R nyelv programozói fokú ismeretét. Természetesen, ha rendelkezünk ilyen irányú előtanulmányokkal a tanulási folyamat néhány szakasza lerövidíthető, de könyvünk elsősorban azok számra íródott, akik programozási nyelvekkel korábban nem találkoztak, és nem is vágynak az R ilyen mélységű ismeretére. Az R nyelv elsajátítása során bevezetjük azokat az egyszerű fogalmakat, amelyeket nem nélkülözhetők az adatelemzés során, azonban az R programozásához más szakkönyveket javaslunk olvasásra.

2.1.3. Mi valójában az R?

Az R nyelv fejlesztője az R Core Team⁴. Az R nyelv egy rendkívül népszerű szkriptnyelv, több millióan használják világszerte. Elsősorban adatelemzésre, adatmodellezésre és grafikus megjelenítésre, vagyis arra, amit ma adattudományok (data science) alatt értünk. Azonban az R nyelv önmagában nem szoftver, hanem egy rendkívül rugalmas szkriptnyelv, amely például előírja, hogy milyen szintaktikai szabályok mentén fogalmazhatjuk meg az utasításainkat. Ahhoz, hogy az R nyelvet használni tudjuk, vagyis, hogy a számítógép valóban végre is hajtsa a szintaktikailag helyes utasításinkat, szükség van egy szoftveres környezetre, egy olyan futtató rendszerre, amely a kódunkat értelmezi és végrehajtja.

Az R környezet három fő összetevőt tartalmaz: (1) egy *konzolt*, ahová a parancsainkat begépelhetjük; (2) a parancsok végrehajtásáért felelős *R interpretert*; (3) a *csomagokat*. A konzol és az interpreter biztosítja az R nyelven írt parancsok tényleges végrehajtását. Így tudunk adatokat beolvasni, átlagot számolni, varianciaelemzést futtatni, vagy publikációkész ábrákat létrehozni. A csomagok adatokat és függvényeket tartalmaznak, például a **MASS** csomag 88 adatobjektumot és 78 függvényt tartalmaz. A függvények valamilyen tevékenységet hajtanak végre, és valójában ezeket a csomag-függvényeket használjuk fel a konzolban, ha *bármilyen tevékenységet* szeretnénk végrehajtani (például adatokat beolvasni, átlagot számolni stb.). A könyv írásának időpontjában kb. 17000 csomag volt érhető el az R-hez. Csomagok 3 csoportját különböztetjük meg: *standard csomagok* (14 db), *ajánlott csomagok* (15 db) és *egyéb csomagok* (kb. 17000 db).

⁴ <https://www.r-project.org/contributors.html>

A standard csomagok fejlesztője az R Core Team. A standard csomagok: **base**, **compiler**, **datasets**, **grDevices**, **graphics**, **grid**, **methods**, **parallel**, **splines**, **stats**, **stats4**, **tcltk**, **tools**, **utils**. Az ajánlott csomagok: **KernSmooth**, **MASS**, **Matrix**, **boot**, **class**, **cluster**, **codetools**, **foreign**, **lattice**, **mgcv**, **nlme**, **nnet**, **rpart**, **spatial**, **survival**. Az ajánlot csomagok közül a **foreign** és az **nlme** fejlesztője az R Core Team, a többöt más felhasználók fejlesztették, például a már említett **MASS** csomag fejlesztője Brian Ripley. Csomagot bárki szabadon fejleszthet és terjeszthet, az *egyéb csomagok* csoportját akár mi is gyarapíthatjuk.

A R környezet már igazi szoftver, terjesztésének koordinálását az R Foundation⁵ végzi a CRAN⁶ infrastruktúráján keresztül. Ez biztosítja, hogy számítógépünkre telepíthessük az R környezetet. Ezt a CRAN-ról elérhető R futtatási környezetet *Alap R*-nek nevezzük. Fő komponensei a már említett konzol a parancsok begépelésére, az R értelmező a begépelt parancsok végrehajtására és a csomagok közül a standard és ajánlott csomagok. Az *Alap R* telepítése után már tudunk R parancsokat végrehajtani, és nagyon sok adatelemzési probléma megoldására nyílik módunk, sőt azt mondhatjuk, hogy tetszőleges problémát megoldhatunk kisebb-nagyobb erőfeszítéssel, mert az R egy teljes értékű nyelv. Azonban sokszor érdemesebb az *egyéb csomagok* közül választani, hiszen könnyen elképzelhető, hogy a számtalan csomag között találunk olyat, amely segítségünkre lehet speciális feladataink megoldása során. Valószínű, hogy létezik olyan csomago és benne olyan függvény, amely adatkezelési, adatelemzési, grafikai vagy publikálási feladatunkat jelentősen megkönnyíti. Az *egyéb csomagok* csoportjába tartozó csomagok forrása több tárhely is lehet, ezek közül legjelentősebb az R Foundation által karbantartott CRAN (kb. 15000 csomaggal), a Bioconductor (1741 csomaggal) és a GitHub.

Az R tehát egyszerre több dolgot jelent. Az R egyrészt egy magas szintű programozási nyelv, hamarosan megtanuljuk, hogyan írunk ezen a nyelven értelmes utasításokat. Másrészt a nyelv körüli környezetet is jelenti, amely magába foglalja a konzolt, a parancsaink értelmezését felelős R interpretort, valamint azokat a csomagokat, amelyekkel az R tudása kiegészíthető.



Összefoglalás

Minden statisztikai programcsomag, így az R is, alapvetően a számításigényes statisztikai eljárások kézi végrehajtásától kímél meg minket. Az R nagyon gazdag adatmanipulációs és grafikus funkciókban is, támogatja a reprodukálható adatelemzés végrehajtását. Az R ingyenes, többplatformos és egyik legfontosabb jellemzője, hogy parancsok útján bírhatjuk működésre. Az *Alap R* biztosítja a konzolt a parancsok begépelésére, az R interpretort a parancsok tényleges végrehajtására,

⁵ <https://www.r-project.org/foundation/>

⁶ <https://cran.r-project.org/mirrors.html>

és jónéhány csomagba szervezett eljárást az adatelemzési feladatok elvégzéséhez. Az *Alap R* mindenki számára elérhető csomagot tartalmaz, standard csomagokat és az ajánlott csomagokat, de több tízezer további csomaggal bővíthetjük az R tudását. Az adatelemzési munka során egy R környezet vesz minket körül, amely az R nyelven megírt parancsok értelmezésére és végrehajtására képes *Alap R*-ból, és az ún. *egyéb csomagokból* áll.



Feladatok

1. Keressünk weboldalakat, amelyek az R előnyeit és hátrányait listázzák!
2. Keressük meg, hogy az R optimális futtatásához, milyen hardver követelmények szükségesek!
3. Nézzünk utána, hogy ma kb. hány csomag érhető el az R-hez? Keressünk ábrát, amely bemutatja, hogy az évek során hány csomag volt elérhető az R-hez?
4. Hol áll az R népszerűsége a többi programozási nyelvhez, illetve statisztikai programcsomaghoz képest?
5. Milyen ingyenesen elérhető, grafikus felhasználói felülettel rendelkező statisztikai programcsomagok építenek az R-re?
6. Említettük, hogy az adatelmezési munka nem igényli az R programozói fokú ismeretét, de soroljunk fel néhány könyvet, amelyből az R programozása is megtanulható!

2.2. A modern R



Ebben a fejezetben:

- megismерjük a *Tidyverse R* fogalmát,
- megtudjuk mit értünk modern R alatt.

A 2014-es év az R nyelv életében meghatározó változást hozott. Egyszerűbb megjelenést a **magrittr** csomagban a pipe operátor (`%>%`), amellyel olvashatóbb kódok írására nyílt lehetőség⁷, másrészt a pipe operátorra alapozva Hadley Wickham bemutatta a **dplyr** és **tidyr** csomagokat. Ezzel az R funkcionális⁸ oldalát úgy

⁷ <https://www.r-bloggers.com/magrittr-simplifying-r-code-with-pipes/>

⁸ Az R egy nem túl fiatal, a funkcionális programnyelvekhez hasonlóan építkező programozási nyelv, vagyis egy probléma megoldása tipikusan sokszorosan egymásba ágyazott függvényhívások segítségével történik. Ez sok-sok nyitó és záró kerekzárójellel jár együtt, így a parancsaink áttekintése és karbantartása sokszor nehézségekbe ütközik. Ezt kiküszöbölni az R-ben előszeretettel használnak procedurális eszközöket (például `for`

erősítették meg⁹, hogy a sokszoros egymásba ágyazás során kiküszöbölték a kerekzárójelek írásának problémáját. Az ebben a szellemben készült csomagok listája bővült az idők folyamán, és a *Tidyverse* nevet kapta ez a csomaggyűjtemény. Jelenleg a következő csomagok alkotják: **ggplot2**, **purrr**, **tibble**, **dplyr**, **tidyr**, **stringr**, **readr** és **forcats**. Ezek a csomagok nem egyszerűen új funkciókkal ruházzák fel az *Alap R* tudását, mint általában az *egyéb csomagok*. A *Tidyverse* csomagai konzisztens módon együttműködnek, és egy új megközelítést hoznak az adatelemzési folyamatok végrehajtásában és a kódok írásában. Rövidebb idő alatt hozhatunk létre könnyebben karbantartható kódokat, és a műveleteink végrehajtása is rendszerint gyorsabb. Amikor ebben a megközelítésben hozzuk létre és hajtjuk végre utasításainkat, akkor azt mondjuk hogy a *Tidyverse R*-t használjuk. A *Tidyverse R* nem helyettesíti az *Alap R*-t, és csak bizonyos feladatokra használható. Lássunk tisztán, amit elvégezhetünk *Tidyverse R*-ben, azt az *Alap R*-ben is meg tudnánk tenni, de valószínűleg több gépeléssel, lassabb és rosszabbul karbantartható kódal.

Eddig láttuk, hogy az R használatához szükséges az *Alap R* telepítése, majd a speciális problémáknak megfelelően kiegészíthetjük az R tudását úgy, hogy telepítünk egyet vagy többet az *egyéb csomagok* kategoriájából. Választhatjuk akár a *Tidyverse* csomagjait is telepítésre, ugyanis így lehetőségünk nyílik a *Tidyverse R* használatára. Utasításaink megfogalmazásának ma ez a legmodernebb módja.

A modern R alatt lényegében azokat a funkciókat értjük, amelyek a *Tidyverse*¹⁰ gyűjteményben található csomagokhoz kötődnek. Ezekkel a csomagokkal, gyorsabb, olvashatóbb és könnyebben karbantartható kódokat hozhatunk létre. A *Tidyverse* használata tehát erősen javasolt, de ebben a könyvben a „hagyományos”, *Tidyverse R* előtti lehetőségeket is bemutatjuk.



Összefoglalás

A *Tidyverse R* egy csomaggyűjtemény az *egyéb csomagok* csoportjából, amely újabb szemléletű R parancsok írására ad lehetőséget. Az így készült kódjaink rendszerint gyorsabban futnak és könnyebben karbantarthatók. A modern R a *Tidyverse R* csomagjaival kiegészített *Alap R*, de legfőképp egy új lehetőség parancsaink megfogalmazására.



Feladatok

1. Ki Hadley Wickham?
2. Honnan származik a pipe operátor neve?

ciklusokat), de a kód olvashatóságát és karbantartását igazán ez sem könnyíti meg.

⁹ <https://www.r-bloggers.com/magrittr-the-best-thing-to-have-ever-happened-to-r/>

¹⁰ <https://www.tidyverse.org/>

2.3. Múlt és jelen



Ebben a fejezetben:

- megismerjük az R rövid történetét és annak szereplőit,
- majd egy szubjektív listával segítjük az R tanulását,
- illetve megismerjük az R alaptudás elemeit.

2.3.1. Szereplők és fogalmak

Érdemes néhány szereplőt és fogalmat tisztázni az R világán belül. Az R nyelvet 1992-ben kezdte fejleszteni Ross Ihaka¹¹ és Robert Gentleman¹², 1997-től pedig egy nagyobb csapat, az R Development Core Team¹³ vezeti a fejlesztést (rövidebben *R Core Team*). Ettől az évtől az R hivatalosan a GNU projekt része. Az *R Core Team* tagjai 2002-ben létrehozták a The R Foundation for Statistical Computing¹⁴ (rövidebben *The R Foundation*) közhasznú, nonprofit szervezetet, amelynek célja (1) az R folyamatos fejlesztésének biztosítása, és ehhez kapcsolódóan a nyílt forráskódú számítógépes statisztikai innovációk támogatása, (2) az R fejlesztői közösség (*R Core Team*) hivatalos hangjaként a felhasználók, intézmények és üzleti vállalkozások számára a kommunikáció biztosítása, és (3) az R program és dokumentációk szerzői jogainak kezelése. A szervezet rendszeresen konferenciákat, találkozókat szervez, referált folyóiratot, kézikönyveket és technikai leírásokat ad ki, valamint fenntart egy számítógépes infrastruktúrát (ez a CRAN, amely levelező listákat, FTP- és Webszervereket üzemeltet). Az R Foundation hivatalos oldala – egyben az R hivatalos oldala – a <https://www.r-project.org/>¹⁵. Az R Foundation (és más önkéntesek) által üzemeltetett számítógépes hálózat neve a CRAN (Comprehensive R Archive Network), amely szabad hozzáférést nyújt az R legfrissebb verziójához, az R kiterjesztéseihez (a csomagokhoz) és a részletes dokumentációhoz. A CRAN fő számítógépe Ausztriában található <https://CRAN.R-project.org/>¹⁶, azonban nagyon sok naponta frissülő tükörszerver¹⁷ érhető el világszerte.

2.3.2. Alapelvek

Az R elsődleges célja, hasonlóan más statisztikai programcsomagokhoz, a statisztikai adatelemzés, amelyet négy lépésre bonthatunk:

1. adatok beolvasása,
2. adatok előkészítése elemzésre,
3. adatelemzés,

¹¹ <https://www.stat.auckland.ac.nz/~ihaka/>

¹² <https://www.linkedin.com/in/robert-gentleman-06845098/>

¹³ <https://www.r-project.org/contributors.html>

¹⁴ <https://www.r-project.org/foundation/>

¹⁵ <https://www.r-project.org/>

¹⁶ <https://CRAN.R-project.org/>

¹⁷ <https://cran.r-project.org/mirrors.html>

4. eredmények publikálása.

Az R mára a fenti 4 tevékenység elvégzését teljes körűen támogatja. A könyv célja ezek bemutatása. Mielőtt elkezdjük ezt az izgalmas utat – az R tanulmányozását – néhány alapelvet szeretnék megemlíteni, ami segíthet minket az utazásunk során:

- *Magabiztosság* - Az R nagyon nagy, így a teljes megismerése nem lehet célunk. Mindig lesz valaki, aki az R egyik vagy másik részét jobban, vagy kevésbé ismeri nálunk. Ez természetes, ezen soha ne csodálkozzunk. Az előző ismeretek azonban az R speciális területeire vonatkoznak, az *R alaptudás* (2.3.3 fejezet) minden R-ben jártas felhasználó számára közös. E könyv célja ennek az alaptudásnak az átadása, melynek birtokában már kellő magabiztossággal vághatunk neki az R azon részeinek elsajátításába, amelyek az éppen elénk kerülő speciális feladat megoldásához szükségesek. Hisszük, hogy e könyv elolvásásával, mind az R alaptudás, mind a kellő magabiztosság elérhetővé válik számunkra.
- *Gyakorlás* - Az R alaptudásának megszerzése némi időbe telik, ez tagadhatatlan. A motiváció megtartásához viszonylag jól kell éreznünk magunkat a tanulás és a gyakorlás során. A könyvben ezért minden fejezet végén találunk megoldandó feladatokat, amelyek között szórakoztatónak, érdekes és kihívást jelentő gyakorlatok is szerepelnek.
- *Svájci bicska* - A R nagyon sokféle statisztikai és nem-statisztikai probléma megoldására képes, sőt ugyanarra a problémára nagyon sok különböző eszközt kínál. Ha elsőre nem a legszebb, legoptimálisabb megoldás jut az eszünkbe, ne csüggédjünk, ez a legtöbb esetben nem jelent gondot. Azon se csodálkozzunk, ha korábban megoldott problémánkra idővel újab és újabb megoldási lehetőségeket találunk.

2.3.3. Az R alaptudás

Melyek az R-ben való munkavégzéshez nélkülözhetetlen alapismeretek? Meggyőződésünk, ha a lentebb felsorolt témaörökkel tisztában vagyunk, akkor már magabiztos R tudással rendelkezünk, és bármilyen további R témaörök könnyen elsajátítható lesz. Ezekre az ismeretekre úgy gondolhatunk, mint egy ablakra, amelyen keresztül az R szinte végtelen lehetőségeinek tárháza nyílik meg előtünk. Később visszatérhetünk ehhez a listához, és ellenőrizhetjük, hány elemet tudunk már kipipálni.

Az R alaptudás elemei:

- Az R környezet alapszintű ismerete
 - az *Alap R*, az *RStudio* és a csomagok telepítése
 - projektek használata és R parancsok futtatása az *RStudio*-ban
- Az R nyelv alapszintű ismerete
 - konstansok írása
 - objektumok kezelése
 - egyszerű adattípusok

- alapvető operátorok
- kifejezés fogalma
- a függvényhívás lehetőségei
- összetett adattípusok,
- a vektoraritmetika szabályai
- Az alapvető függvények ismerete
 - csomagkezelő függvények
 - a munkaterület függvényei
 - matematikai függvények
 - input/output függvények
 - indexelés, szűrés, rendezés
 - információ kérés az objektumokról
 - egyszerű típuskonverzió
 - transzformáció
 - ismétlő és összesítő függvények
 - a hagyományos grafika néhány eleme
 - a **ggplot2** alapszintű ismerete
- Egyéb ismeretek
 - szövegszerkesztési és állománykezelési ismeretek
 - a tagolt szöveges állomány fogalma
 - reprodukálható kutatás az RMarkdown segítségével



Összefoglalás

Az R fejlesztését Ross Ihaka és Robert Gentleman kezdte, majd 1997-től egy nagyobb csapat, az *R Development Core Team* vezeti a fejlesztést. Az *R Core Team* tagjai 2002-ben létrehozták a *The R Foundation for Statistical Computing* közhasznú, nonprofit szervezetet, amelynek fő célja az R folyamatos fejlesztésének biztosítása. A szervezet fenntart egy CRAN nevű számítógépes hálózatot, amely szabad hozzáférést biztosít az R legfrissebb verziójához, a csomagokhoz és a részletes dokumentációkhöz.

Az R alaptudás megszerzése elegendő magabiztosságot fog nyújtani az adatalemzési munka során, azonban vegyük figyelembe, hogy ezt csak kellő gyakorlással érhetjük el. Az R sokféle megoldást biztosít ugyanarra a problémára, legyen az statisztikai vagy bármilyen más jellegű probléma.



Feladatok

1. Keressünk olyan statisztikai jellegű témaköröket, amelyekben az R segítségünkre lehet?
2. Keressünk olyan nem-statisztikai jellegű témaköröket, amelyekben az R segítségünkre lehet?

3. Nézzük át néhány online elérhető R könyvet, és hasonlítsuk össze az R alaptudás egyes elemeivel! Melyek az átfedő részek, és hol vannak különbségek?

<https://www.r-bloggers.com/the-history-of-r-updated-for-2020/>

3. fejezet

Az R telepítése



3.1. A fő komponensek telepítése



Ebben a fejezetben:

- megismerjük az *Alap R*, az *RStudio* és a csomagok telepítését.

A korábbi fejezetekben megismertük az R világának néhány fogalmát és szereplőjét. Tudjuk, hogy az R nyelv használatához megfelelő szoftveres környezetre van szükség, amely magába foglalja az *Alap R-t* és az *egyéb csomagok* kategóriájából esetlegesen telepített csomagokat is. Az R már ezen eszközök birtokában is teljes körűen használható, azonban egy újabb ingyenes eszköz, az *RStudio*, kényelmessé és hatékonnyá teszi az adatelemzési munkát.

Könyvünk legfontosabb gondolata: **ma akkor tudjuk a legjobban kihasználni az R lehetőségeit, és ezzel egyidőben a legkényelmesebb módon elvégezni az adatelemzési feladatunkat, ha (1) az *Rstudio-t* használjuk, (2) projekt üzemmódban dolgozunk, és (3) RMarkdown állományokban rögzítjük az R parancsainkat.** Ezt a szemléletet következetesen képviseljük az egyes fejezetekben, és a későbbiekben részletesebben bemutatjuk, hogyan tudjuk mindezt megvalósítani.

A R kényelmes használatához a legelső lépés a szoftveres környezet egyes elemeinek telepítése. Három fő komponens telepítésére lesz szükségünk:

1. *Alap R*, amely tartalmazza a konzolt, az R interpretert, illetve a *standard csomagokat* és az *ajánlott csomagokat*,
2. *RStudio*, amely egy új konzollal „eltakarja” az *Alap R*-t, és kényelmesebb hozzáférést biztosít az *Alap R* interpreteréhez és a csomagjaihoz.
3. Csomagok, amelyek az *egyéb csomagok* nagy halmazából származnak, és telepítésükkel újabb és újabb képességekkel ruházzuk fel az *Alap R*-t.

3.1.1. Az Alap R telepítése

Az *Alap R* telepítéséhez látogassunk el az R hivatalos letöltő oldalára: <https://cran.r-project.org/>. Az operációs rendszerünknek megfelelő link kiválasztásával folytassuk a navigálást.

- A Windows felhasználók a `Download R for Windows` linken, majd a `base` linken kattintva jutnak el a telepítőprogram linkjéhez: `Download R X.X.X for Windows`. A sikeres letöltés után indítsuk el a telepítőt, és az alapértelmezetten felajánlott opciók nyugtázsával végezzük el a telepítést. A telepítést lehetőleg olyan Windows felhasználó alatt végezzük el, amelynek a neve sem ékezetes karaktert, sem szóközt, sem egyéb írásjelet nem tartalmaz.
- A macOS felhasználók a `Download R for (Mac) OS X` linken kattintva jutnak a telepítőhöz: `R-X.X.X.pkg`. A letöltés után indítsuk el a telepítőt, és a `Next` gombok segítségével végezzük el a telepítést.
- A Linux felhasználók az aktuális R verzió telepítéséhez a `Download R for Linux` linken keresztül jutnak el, ahol a megfelelő disztribúció (Debian, Redhat, Suse, Ubuntu) kiválasztása után konkrét információkat kapnak a telepítésről.

3.1.2. Az RStudio telepítése

Az *RStudio* telepítéséhez az operációs rendszerünknek megfelelő telepítőt kell letöltenünk a <https://www.rstudio.com/products/rstudio/download/> oldalról. Az RStudio Desktop (Open Source License) változatára lesz szükségünk, töltük le és telepítük ezt a számítógépünkre. A telepítés során fogadjuk el az alapértelmezett opciókat. Az *RStudio* automatikusan megtalálja és használja a korábban telepített *Alap R* példányunkat, így a későbbiekben elegendő lesz az *RStudio*-t használni, azon keresztül elérhetjük az *Alap R* minden funkcióját.

3.1.3. Csomagok telepítése

A csomagok telepítésére az *Alap R* vagy az *RStudio* elindítása után van módunk. Érdemes a telepítéseket az *RStudio*-ból végezni. A csomag fellelési helye alapján, három különböző tárhelyről mutatjuk be a csomagok telepítését. Láttni fogjuk, hogy a csomagok telepítéséhez R parancsokat fogunk használni. Ha még nem

vagyunk jártasak R parancsok futtatásban, akkor a 4.1 fejezet fellapozásával segítséget kaphatunk a lenti parancsok kipróbálásához, de úgy is eljárhatunk, hogy most kihagyjuk ennek a résznek az áttekintését, és később térünk vissza, amikor valóban felmerül az igény csomagok telepítésére.

Az R csomagok hivatalos helye a CRAN (The Comprehensive R Archive Network)¹. A CRAN számítógépei tárolják a nyílt forráskódú R nyelv és környezet különböző verzióinak kódjait és dokumentációját, így az összes R csomag forrás-kódját is. Egy bírálási folyamat után bármely felhasználó csomagja a CRAN-ból is elérhető lehet.

Az *Alap R* vagy az *RStudio* elindítása után az `install.packages()` függvénytel tölthetünk le és telepíthetünk csomagot a CRAN-ról. Tetszőleges csomag telepítéséhez írjuk a csomag nevét idézőjelekben a függvény argumentumába:

```
install.packages("csomag_neve")
```

A **psych** csomagot, amely a pszichológia kutatások adatainak elemzéséhez nyújt segítséget, például így telepíthetjük:

```
install.packages("psych")      # psych csomag telepítése
```

A csomagok másik fontos forrása a Bioconductor², ahol alaposan tesztelt és igen jól dokumentált bioinformatikai témaúj csomagokat találunk. Az innen elérhető csomagokat – például most a **DESeq2** csomagot az RNS-szekvenálási elemzésekhez – a következő parancsokkal telepíthetjük:

```
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("DESeq2")
```

A csomagok harmadik fő forrása a GitHub. A felhasználók a saját fejlesztésű csomagjaikat rendszerint először a GitHub-on keresztül teszik elérhetővé. Ha ezeket a csomagokat szeretnénk kipróbálni, akkor a felhasználó és a csomag nevének birtokában a következő parancsot kell kiadnunk:

```
devtools::install_github("felhasznalo_neve/csomag_neve")
```

Például a GitHub-ról telepíthető **emo** csomag segítségvel hangulatjeleket szűrhetünk be az RMarkdown állományainkba. Ezzel a sorral telepíthetjük a csomagot:

```
devtools::install_github("hadley/emo")
```

Fontos tudnunk, hogy a csomagok telepítésére egy számítógépen egy adott R verziótól belül csak egyszer van szükség. A telepítő parancsainkat azonban érdekes megőrizni, hogy ha szükséges, akkor egy új R verzióban könnyen tudjuk

¹ <https://cran.r-project.org/web/packages/>

² <https://www.bioconductor.org/>

telepíteni a korábban használt csomagjainkat. Nagyon fontos, hogy a telepítő parancsok futtatása után, tegyük azokat megjegyzésbe, vagyis írunk előjük kettőskereszt (#) karaktert (részletesebb információkat a megjegyzések ról a ?? fejezetben olvashatunk). Ezzel tudjuk megvédeni ezeket a telepítő parancsokat az újbóli, véletlen, felesleges végrehajtástól. Ennek megfelelően a telepítő parancsainkat ilyen formában kell őriznünk:

```
# install.packages("psych")          # psych csomag telepítése
# if (!requireNamespace("BiocManager", quietly = TRUE))
#     install.packages("BiocManager")
# BiocManager::install("DESeq2")
# devtools::install_github("hadley/emo")
```

Vegyük figyelmbe, hogy egy csomag telepítése során más, egyéb csomagok telepítése automatikusan is megtörténhet, tehát egy helyett valójában több csomag is felkerülhet a gépünkre. Az is előfordulhat, hogy egy csomag telepítése csak akkor lesz sikeres, ha más csomagok frissítését engedélyezzük az adott csomag telepítése során. Végül előfordulhat olyan eset is, amikor egy csomag telepítése valamilyen oknál fogva meghiúsul. Erről minden esetben hibaüzenet tájékoztat minket, és ez szinte minden esetben jó kiindulásul szolgál a hibát okozó körmény elhárításában. A legtöbbször egy másik csomag hiánya okozza a sikertelen telepítést, ezért olvassuk ki a hibaüzenetből a hiányolt csomag nevét, és először ennek a telepítését végezzük el.



Összefoglalás

Az R kényelmes használatához először telepítsük az operációs rendszerrünkhöz megfelelő *Alap R*, majd az *RStudio* legújabb verzióját. Az R képességeit csomagok segítségével bővíthetjük, ehhez legtöbbször az `install.packages()` parancsot használjuk.



Feladatok

1. Melyik az R legfrissebb változata, és milyen újdonságokat tartalmaz az előző változathoz képest?
2. Melyik az *RStudio* legfrissebb változata, és milyen újdonságokat tartalmaz az előző változathoz képest?
3. Hogyan deríthető ki, hogy egy csomagban (például a **MASS**) csomagban, hány adatobjektum, és hány függvény található?

3.2. A Tidyverse R telepítése



Ebben a fejezetben:

- megismérjük a *Tidyverse R* telepítését.

A *Tidyverse R* az R meglévő funkcióinak új szemléletű használatát jelenti. A modern R jelenleg egyet jelent a *Tidyverse R*-rel, az ebben a szemléleteben készült parancsaink gyorsak, jól olvashatók és könnyen módosíthatók. A *Tidyverse R* funkció összesen 8 csomagba (**ggplot2**, **purrr**, **tibble**, **dplyr**, **tidyr**, **stringr**, **readr** és **forcats**) vannak szétosztva, minden egyik csomag egy-egy témakört fed le. A fenti csomagok telepítése egyetlen gyűjtőcsomag a **tidyverse** nevű csomag telepítésével is elvégezhető:

```
install.packages("tidyverse") # a Tidyverse R telepítése
```

A *Tidyverse R* telepítését követően a csomagokban lévő függvények használatához a *Tidyverse R* betöltésére is szükség van. Hívjuk meg a **library()** függvényt, amely ebben az esetben igen részletes tájékoztatást ad az újonnan elérhető csomagokról.

```
library(tidyverse)
#> -- Attaching packages ----- tidyverse 1.3.0 --
#> <U+221A> ggplot2 3.3.0      <U+221A> purrr   0.3.3
#> <U+221A> tibble  3.0.0      <U+221A> dplyr   0.8.5
#> <U+221A> tidyr   1.0.2      <U+221A> stringr 1.4.0
#> <U+221A> readr   1.3.1      <U+221A>forcats 0.5.0
#> -- Conflicts ----- tidyverse_conflicts() --
#> #> x dplyr::filter() masks stats::filter()
#> #> x dplyr::lag()    masks stats::lag()
```

A *Tidyverse R* csomagjait jelenleg is intenzíven fejlesztik, így gyakran jelentik meg újabb és újabb verziót. Érdemes ellenőrizni, hogy a *Tidyverse R* csomagjai közül a legfrissebbeket használjuk-e. Ehhez a **tidyverse** csomag **tidyverse_update()** függvényét használjuk.

```
tidyverse::tidyverse_update()
#> The following packages are out of date:
#>
#> * lubridate (1.7.4 -> 1.7.8)
#> * purrr     (0.3.3 -> 0.3.4)
#> * xml2       (1.2.5 -> 1.3.1)
#>
#> Start a clean R session then run:
#> install.packages(c("lubridate", "purrr", "xml2"))
```

Például a fenti esetben 3 csomag frissítését javasolja a **tidyverse_update()**

függvény, és segítséget is ad a telepítőparancs listázásával.



Összefoglalás

A *Tidyverse R* használatához elegendő telepítenünk a **tidyverse** csomagot, amely a többi 8 csomag telepítését automatikusan elvégzi. A telepítést a `install.packages("tidyverse")` parancssal végezzük. Időnként ellenőrizzük a `tidyverse::tidyverse_update()` segítségével, hogy a legfrissebb változatát használjuk-e a *Tidyverse R*-t alkotó csomagoknak.



Feladatok

1. Keressünk rá a *Tidyverse R* csomagjaira, és próbáljuk kideríteni az egyes csomagok fő célját, alkalmazási területeit!
2. Derítsük ki, hogy az R Core Team vagy Hadley Wickham több R csomag szerzője!

3.3. Az R frissítése



Ebben a fejezetben:

- bemutatjuk az *Alap R*, az *RStudio* és a csomagok frissítését.

A R ideális használata során az *RStudio*-ban dolgozunk, és így érjük el az *Alap R* és az egyes csomagok szolgáltatásait. A mai napig mindenkomponenst intenzíven fejlesztik, újabb és újabb funkciókat építenek be, és az esetleges hibákat rendre javítják a frissebb változatokban. Az *Alap R* évente kb. négyeszer frissül, az *RStudio* háromszor, és érdemes időnként azt is ellenőrizni, hogy a gyakran használt csomagjainkból nincs-e frissebb példány.

3.3.1. Az Alap R frissítése

A telepített *Alap R* verzióját az `R.version.string` végrehajtásával ellenőrizhetjük. Amennyiben az R hivatalos oldalán³ találunk frissebb példányt, akkor legalább két módszer segítségével frissíthetjük az *Alap R*-t. Megjegyezzük, hogy az *Alap R* sikeres frissítése után az *RStudio* automatikusan az új példányt fogja használni.

1. módszer: csak Windows alatt. Windows operációs rendszer alatt rendelkezésre áll az `installr` csomag, amelynek pontosan az a feladata, hogy kényelmesen telepíthessük számítógéünkre az R legfrissebb verzióját. Az `installr` a

³ <https://cran.r-project.org/>

régebbi verzióban lévő csomagokat az új változatba is átmozgatja, és ott azok frissítését is elvégzi. A következő parancsok futtatására van szükség.

```
# install.packages("installr") # az installr csomag telepítése
library(installr)           # az installr csomag betöltése
updateR()                   # az Alap R és a csomagok frissítése
```

2. módszer: általános út. Az *Alap R* frissítésének másik módja, hogy telepítünk egy új példányt a régi R mellé. Azaz a korábban látott módon letöltsük és telepítjük az *Alap R* legújabb változatát, pontosan úgy, mintha még nem lenne a géünkön működő R. Ez az új verzió azonban félkarú óriás mindaddig, amíg a régi R verzióban használt összes csomagot nem telepítjük újra az új verzióban is. Ezt magunk egyesével is megtehetjük, ha korábban összegyűjtöttük a csomagtelepítő parancsainkat, legyen szó akár akár a CRAN, a Bioconductor vagy a GitHub oldaláról származó csomagokról. Ha ezek nem állnak rendelkezésre, akkor az *Alap R* frissítésének általános útját három lépésben foglalhatjuk össze.

1. Indítsuk el az *RStudio*-t még az új R verzió telepítése előtt, és futtassuk le a következő sorokat. A futtatás eredménye egy bináris állomány (`csomagok.rds`), amely a régi R összes telepített csomagjának nevét és más információkat tartalmaz. Lépjünk ki az *RStudio*-ból.

```
telepitett.csomagok <- installed.packages(priority="NA")
saverDLS(object = telepitett.csomagok, file = "csmagok.rds")
```

2. Telepítsük az *Alap R* új verzióját.
3. Indítsuk el az *RStudio*-t és futtassuk le a lenti sorokat. A folyamat több percig is eltarthat. Az *RStudio* már az új R verziót használja, így a csomagok az új R tudását egészítik ki.

```
telepitett.csomagok <- readRDS(file = "csmagok.rds")
install.packages(pkgs=telepitett.csomagok[,1])
```

Megjegyezzük, hogy a fenti 2. módszer segítségével csak a CRAN csomagjait tudjuk telepíteni, a Bioconductor és a GitHub oldaláról származó csomagok telepítését magunknak kell megismételni.

3.3.2. Az RStudio frissítése

A telepített *RStudio* példányunk verziószámát a `Help / About RStudio` menüpont segítségével, vagy az `rstudioapi::versionInfo()` parancs futtatásával ellenőrizhetjük. Frissebb verzió létezéséről a `Help / Check for Updates` menüpont ad tájékoztatást. Amennyiben találunk újabb verziót az *RStudio* hivatalos honlapján⁴, töltük le az operációs rendszerünknek megfelelő változatot és indítunk el a telepítőt. Szerencsére a régi *RStudio* beállításait örökli az új példány.

⁴ <https://rstudio.com/>

3.3.3. Csomagok frissítése

A korábban telepített csomagokat az *RStudio Tools/Check for Package Updates* menüpontjával frissíthetjük. A frissíthető csomagok megjelennek egy dialógus dobozban, jelöljük ki az összes csomagot és indítsuk el a telepítési folyamatot. A következő R parancs végrehajtásával is frissíthetjük a csomagjainkat.

```
update.packages(ask=FALSE)
```



Összefoglalás

Az *Alap R*, az *RStudio* és az egyes csomagok időről-időre megújulnak, érdemes évente legalább egy-két alkalommal elvégezni ezek frissítését. Az *Alap R* frissítése lényegében egy új verzió telepítését jelenti, a régi R továbbra is elérhető marad. Az *RStudio* frissítése után csak az új verziót használhatjuk. Az *Alap R* és az *RStudio* friss verziója a hivatalos honlapokról szerezhető be. A csomagok frissítéséhez használjuk a `update.packages(ask=FALSE)` parancsot.

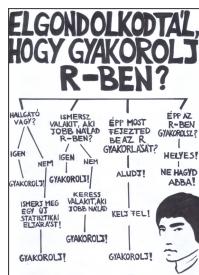


Feladatok

1. Az *RStudio Tools/Check for Package Updates* menüpontjával tájékozódunk a telepített csomagjaink állapotáról. Végezzük el a szükséges frissítéseket!

4. fejezet

Munka az R-ben



4.1. Az RStudio használata



Ebben a fejezetben:

- megismerjük az *RStudio* jellemzőit és felépítését,
- a konzolos és parancsállományos használat különbségeit,
- a parancsállományok és az RMarkdown állományok lehetőségeit,
- a projekt fogalmát és használatát,
- és az *RStudio* billentyűparancsait.

5. fejezet

Az R nyelv



Az előző fejezetekben megismertük az R környezetet, az *Alap R*, az *RStudio* és a csomagok telepítését, megtanultuk a projektek, parancsállományok és RMarkdown állományok létrehozását. Tudjuk, a különböző környezetekben eltérő módszerekkel hajthatjuk végre az R parancsokat: a konzolban az ENTER, a Windowsos *RGui*-ban a Ctrl+R, míg az *RStudio*-ban a Ctrl+Enter billentyűkombinációt kell használnunk. A parancsok végrehajtása közben érdemes észben tartani, ha a folytatás prompt (+) feltűnik, akkor kattintsunk bele a konzolba, és nyomjuk meg az Esc billentyűt, így tudunk kilépni a befejezetlen sor szerkesztéséből.

A fejezet példáinak kipróbáláshoz hozunk létre egy **gyakorlas** nevű új projektet az *RStudio*-ban (*File / New Project*), majd készítsünk egy **gyakorlas.Rmd** RMarkdown állományt (*File / New File / R Markdown*) és egy **gyakorlas.R** parancsállományt (*File / New File / R Script*). A fejezet példáit felváltva gépeljük az RMarkdown állomány R csonkjaiba, illetve a parancsállomány tetszőleges soraiba. A fejezet további részében az R nyelvre koncentrálunk, arra, hogy mit írunk, és nem arra, hogy hová írjuk a parancsokat.

5.1. Adatobjektumok



Ebben a fejezetben:

- áttekintjük az egyszerű számolási lehetőségek R-ben,
- bevezetjük az aritmetikai operátor és a kifejezés fogalmát,
- megismerjük az objektum létrehozását és elnevezését,
- több parancs elhelyezését egy sorban,
- és a megjegyzések használatát.

6. fejezet

Beolvasás



6.1. Alapvető formátumok



Ebben a fejezetben áttekintjük:

- mit nevezünk tagolt szöveges állománynak, hogyan hozzuk létre
- inline és állományos beolvasás közötti különbség
- az inline beolvasás esetei
- tagolt szöveges állomány beolvasása és kiírása (AR és TR)
- fix széles mezővel rendelkező állományok olvasása
- más statisztikai programcsomagok adatállományainak olvasása és írása
- objektumok írása és olvasása bináris állományba

7. fejezet

Adatmanipuláció



7.1. Adatkezelés az Alap R-ben



Ebben a fejezetben:

8. fejezet

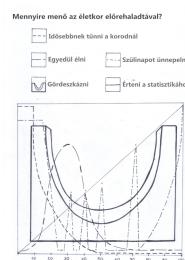
Leíró statisztika



Ebben a fejezetben áttekintjük:

9. fejezet

Grafika az R-ben



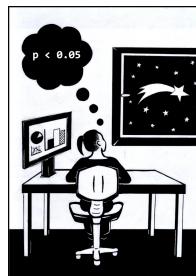
Ebben a fejezetben áttekintjük:



- az R grafikus rendszerei
- a hagyományos grafika alapfogalmai
- magasszintű és alacsonyszintű rajzfüggvények a hagyományos grafikában
- a ggplot2 rendszer alapelve
- ábrák létrehozása ggplot2-ben
- ábrák mentése háttértárra

10. fejezet

Hipotézisvizsgálatok

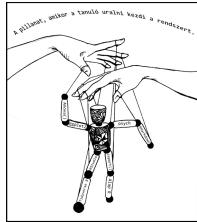


Ebben a fejezetben a statisztika azon klasszikus próbáit foglaltuk össze, amelyek jellemzően egy- vagy kétmintás hipotézisvizsgálatokat jelentenek. Az öt alfejezet a nullhipotézisben szereplő állításoknak és paramétereknek megfelelően a statisztikai próbák különböző csoportjait fedi le:

- várható értékre vonatkozó próbák
- mediánra vonatkozó nemparaméteres próbák
- valószínűségre vonatkozó próbák
- varianciára vonatkozó próbák
- az eloszlás egészére vonatkozó próbák.

11. fejezet

Publikáció



11.1. Reprodukálható kutatás

A kutatás és oktatás világában jelentős elmozdulás figyelhető meg a reprodukálható kutatás felé.

igénye felé. „Az akadémiai kutatás végső terméke a papír, a laboratóriumi jegyzetfüzetekkel és a teljes számítási környezettel együtt az eredmények előállítása érdekében, mint például a kód, az adatok stb., Amelyek felhasználhatók az eredmények reprodukálására és új munka létrehozására a kutatás ”(Wikipedia).

Ennek következménye az, hogy meg kell változtatnunk szokásainkat, és minden kéziratunkat, előadásainkat, házi feladatainkat stb. Tiszta és reprodukálható formában kell elkezdenünk, azaz ha valaki megadja a kódot, akkor ez a személy pontosan reprodukálhatja a dokumentumot. Ez a dokumentum megkönnyíti ezt az átmenetet az R Markdown segítségével.

A függelék

Megoldások

A.1. Megoldások az 1.1.2 feladatokhoz

1. Milyen online vagy nyomtat könyvek segítik az R elsajátítását?
Próbáljuk összegyűjteni a magyar nyelvű könyveket is!

Az R könyvekkel kapcsolatban könnyen az lehet az érzésünk, hogy *túl sok a könyv, és túl kevés az idő*. Valóban tengernyi R könyv vásárolható meg, melyek többsége angol nyelvű. Elegendő a Springer Use R!¹ könyvsorozatára gondolni, amely önmagában több mint 80 címet tartalmaz. Az Amazon-ról elérhető, 2020. januárja után megjelent² könyvek száma is túl van a kétszázon.

Különösen értékesek lehetnek az online elérhető könyvek. Az idegen nyelvű könyvek tematikus gyűjteménye a Big Book of R³, míg az R erőforrásokról saját listát⁴ is karbantartok.

A magyar nyelvű könyvek közül külön is felsorolunk néhányat:

- Reiczigel Jenő, Harnos Andrea, Solymosi Norbert (2021). *Biostatisztika nem statisztikusoknak*. Pars Kft., Nagykovácsi.
- Dinya Elek, Solymosi Norbert (2016). *Biometria a klinikumban 2. Feladatok megoldása R-környezetben*. Medicina Kiadó.
- Münnich Ákos, Nagy Ágnes, Abasi Kálmán. *Többváltozós statisztika pszichológus hallgatók számára*. Bölcsész Konzorcium, Debrecen, 2006. (<http://psycho.unideb.hu/statisztika>) ISBN 963 9704 04 0

2. Térképezzük fel az online videókurzusokat is az R tanulásához!

¹ <https://www.springer.com/series/6991>

² https://www.amazon.com/s?k=r+statistics&i=stripbooks&rh=n%3A5&rs=relevanceexrank&Adv-Srch-Books-Submit.x=39&Adv-Srch-Books-Submit.y=5&field-datemod=1&field-dateop=After&field-dateyear=2020&unfiltered=1&ref=sr_adv_b

³ <https://www.bigbookofr.com/>

⁴ <https://abarik.github.io/roforrasok/>

A videókurzusok többsége is angol nyelvű, a Youtube⁵-ról ingyenesen, de a Udemy⁶ vagy a datacamp⁷ oldaláról előfizetés ellenében több száz kurzust is elérhetünk.

Az R-rel való ismerkedést kezdhetjük az R a gyakorlatban⁸ videósorozatommal is.

3. A bevezető példa (*Két tanítási módszer összehasonlítása*) megoldásában a hipotézisvizsgálat alapján adjunk szöveges értékelést!

A Mann-Whitney-próba alapján azt mondhatjuk, elegendő bizonyítékot találtunk arra, hogy a modern (Sprego) módszer eredményesebb ($Mdn=0,36$), mint a hagyományos ($Mdn=0,66$) tanítási módszer ($W=24$; $p=0,001$).

⁵ https://www.youtube.com/results?search_query=r+statistics

⁶ <https://www.udemy.com/>

⁷ <https://www.datacamp.com/>

⁸ <https://www.youtube.com/playlist?list=PLnmeQHnHYqv6ENGrdTXiE9YJrvHuxH2C9>

B függelék

Színek

Az R grafikus elemeinek a színét magunk is megválaszthatjuk. Ehhez nyújt segítséget ez a függelék.

Megismerjük

- az előre definiált paletta színeit ([B.1 fejezet](#)),
- színek választását az **RColorBrewer** csomag segítségével ([B.2 fejezet](#)),
- színek választása a **dichromat** csomag segítségével ([B.3 fejezet](#)),
- színek választását egyéb paletták segítségével ([B.4 fejezet](#)),
- és végül az előre definiált 657 színnevet ([B.5 fejezet](#)).

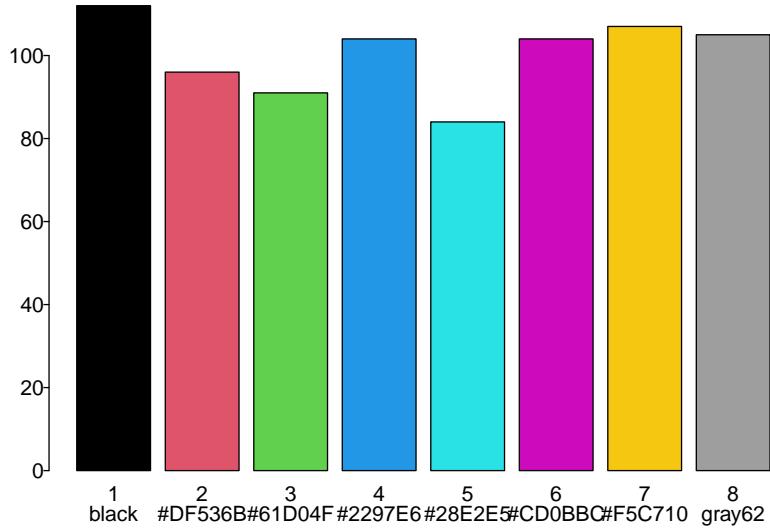
B.1. Az előre definiált paletta színei

Az előre definiált paletta 8 színt tartalmaz. A [B.1](#) ábra az alapértelmezett színeket tartalmazza. Az előre definiált paletta egyes színeit sorszámmokkal (1, 2 stb.) tudjuk elérni, amelyeket rendszerint a rajzfüggvények `col=` argumentumában kell elhelyezni. Az 1. szín a palettán a fekete, a második a piros, és így tovább.

```
set.seed(0)
x <- rpois(n = 50, lambda = 100)
par(las = 1, mgp = c(0, 0.2, 0), tcl = -0.2, mar = c(3, 2, 1, 1))
bar <- barplot(x[1:8], col = 1:8, names.arg = 1:8)
mtext(side = 1, at = bar, text = palette(), line = 1)
```

Más színeket is alapértelmezetté tehetünk, sőt a színek számát is megnövelhetjük a palettán. Ennek a legegyszerűbb módja, ha a `palette()` függvény argumentumában színkódokat tartalmazó karakteres vektort adunk meg.

```
szinek <- c("#E84F2C", "#E31307", "#E84F2C", "#E4BA51", "#E3B786", "#825846",
"#59392A", "#564C30", "#897D6E", "#627C82", "#93AF8A", "#A0BA5E", "#63BA5E",
```



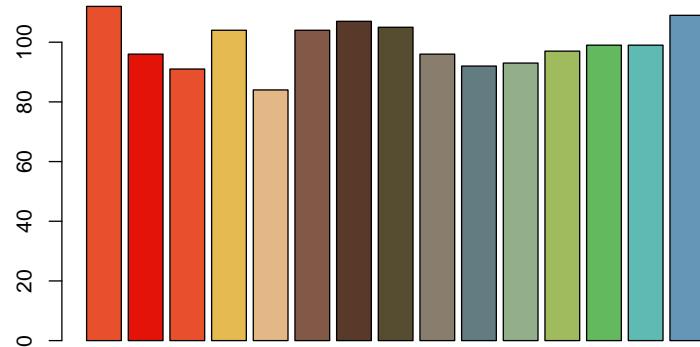
B.1. ábra. Az alapértelmezett paletta 8 színének sorszáma és neve

```
"#5EBAB2", "#6596B7")
palette(value = szinek)
```

A rajzfüggvények ezután a paletta új színeit használják.

```
barplot(x[1:15], col = 1:15)
```

B.2. SZÍNEK VÁLASZTÁSA AZ RCOLORBREWER CSOMAG SEGÍTSÉGÉVEL57



Ha megváltoztattuk a paletta színeit, akkor az alapértelmezett színekhez a `palette("default")` parancsal térhetünk vissza.

```
palette(value = "default") # alapértelmezett paletta visszaállítása
```

B.2. Színek választása az RColorBrewer csomag segítségével

Az **RColorBrewer** csomag `brewer.pal()` függvénye szolgál az előre definiált színpaletták alapján színkódokat tartalmazó vektor létrehozására. A függvény általános alakja:

```
library(RColorBrewer)
brewer.pal(n, name)
```

Az `n=` a kívánt színek számát határozza meg, amely háromnál nem lehet kevesebb. A `name=` a színpaletta nevét tartalmazza. A választható neveket a `brewer.pal.info` adattábla tartalmazza, amely a palettából elérhető összes szín számát és a paletta típusát is tartalmazza. Ez utóbbi a `category` oszlopból olvasható, amelynek lehetséges értékei: `seq`, `div` és `qual`. A szekvenciális palettákat (`seq`) rendezett adatok ábrázolására használhatjuk: a világosabb színek a kisebb értékeket, a sötétebbek a nagyobbakat szemléltethetik. A divergens (`div`) paletták a középső részt világosabb színekkel, a szélső értékeket sötétebb színekkel jelenítik meg. A kvalitatív (`qual`) paletta kategorikus változók megjelenítésére használható.

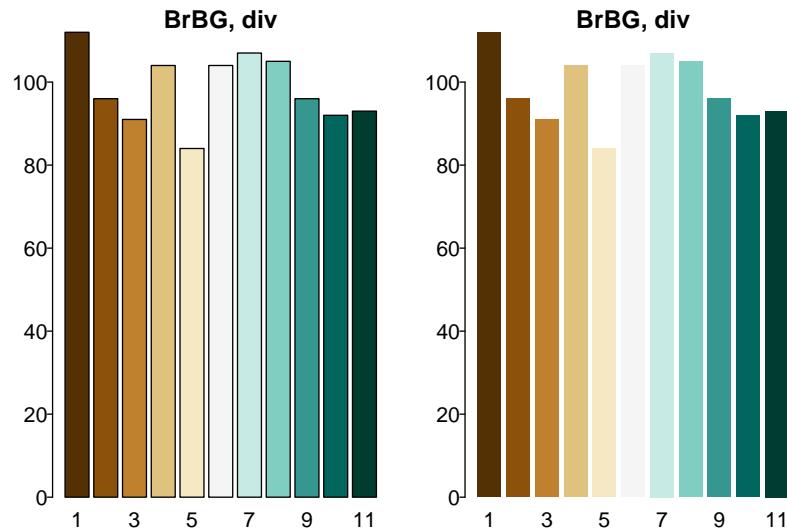
```
library(RColorBrewer)
brewer.pal.info
#>      maxcolors category colorblind
#> BrBG          11    div     TRUE
#> PiYG          11    div     TRUE
#> PRGn          11    div     TRUE
#> PuOr          11    div     TRUE
#> RdBu          11    div     TRUE
#> RdGy          11    div     FALSE
#> RdYlBu        11    div     TRUE
#> RdYlGn        11    div     FALSE
#> Spectral       11    div     FALSE
#> Accent         8     qual   FALSE
#> Dark2          8     qual   TRUE
#> Paired         12    qual   TRUE
#> Pastel1        9     qual   FALSE
#> Pastel2        8     qual   FALSE
#> Set1           9     qual   FALSE
#> Set2           8     qual   TRUE
#> Set3           12    qual   FALSE
#> Blues          9     seq    TRUE
#> BuGn          9     seq    TRUE
#> BuPu          9     seq    TRUE
#> GnBu          9     seq    TRUE
#> Greens         9     seq    TRUE
#> Greys          9     seq    TRUE
#> Oranges         9    seq    TRUE
#> OrRd          9     seq    TRUE
#> PuBu          9     seq    TRUE
#> PuBuGn        9     seq    TRUE
#> PuRd          9     seq    TRUE
#> Purples        9    seq    TRUE
#> RdPu          9     seq    TRUE
#> Reds           9     seq    TRUE
#> YlGn          9     seq    TRUE
#> YlGnBu        9     seq    TRUE
#> YlOrBr        9     seq    TRUE
#> YlOrRd        9     seq    TRUE
```

A továbbiakban a `brewer.pal()` függvény használatára mutatunk példát:

```
# az x adatvektor beállítása
set.seed(0)
x <- rpois(n = 50, lambda = 100)
# grafikus paraméterek beállítása
par(las = 1, mgp = c(0, 0.2, 0), tcl = -0.2, mar = c(3, 2, 1, 1))
```

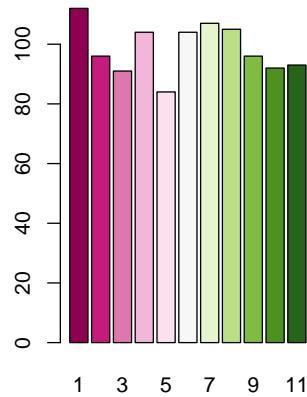
B.2. SZÍNEK VÁLASZTÁSA AZ RCOLORBREWER CSOMAG SEGÍTSÉGÉVEL59

```
library(RColorBrewer)
par(mfrow = c(1, 2))
barplot(x[1:11], col = brewer.pal(n = 11, name = "BrBG"), names.arg = 1:11,
        main = "BrBG, div")
barplot(x[1:11], col = brewer.pal(n = 11, name = "BrBG"), names.arg = 1:11,
        main = "BrBG, div", border = NA)
```

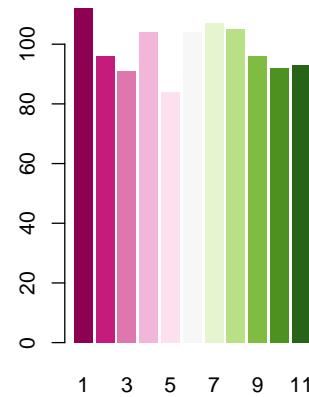


```
par(mfrow = c(1, 2))
barplot(x[1:11], col = brewer.pal(n = 11, name = "PiYG"), names.arg = 1:11,
        main = "PiYG, div")
barplot(x[1:11], col = brewer.pal(n = 11, name = "PiYG"), names.arg = 1:11,
        main = "PiYG, div", border = NA)
```

PiYG, div

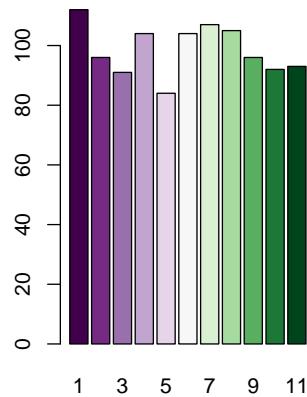


PiYG, div

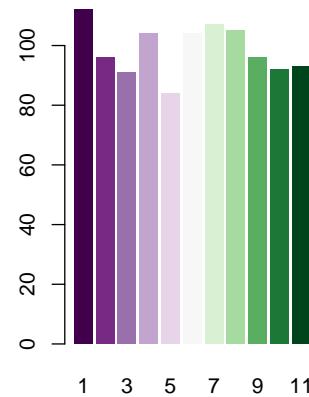


```
par(mfrow = c(1, 2))
barplot(x[1:11], col = brewer.pal(n = 11, name = "PRGn"), names.arg = 1:11,
        main = "PRGn, div")
barplot(x[1:11], col = brewer.pal(n = 11, name = "PRGn"), names.arg = 1:11,
        main = "PRGn, div", border = NA)
```

PRGn, div



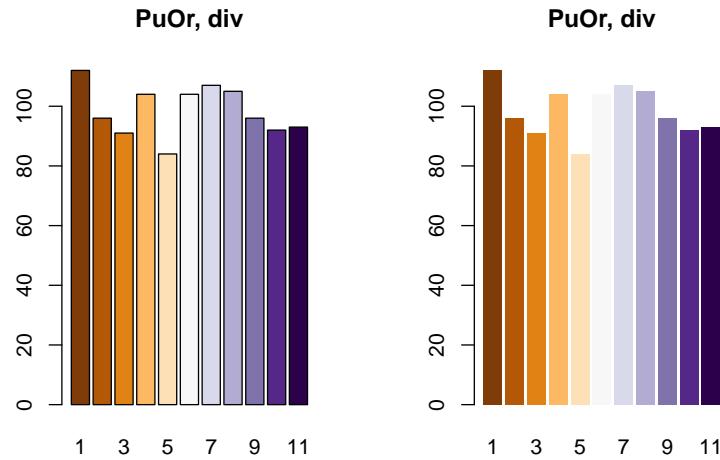
PRGn, div



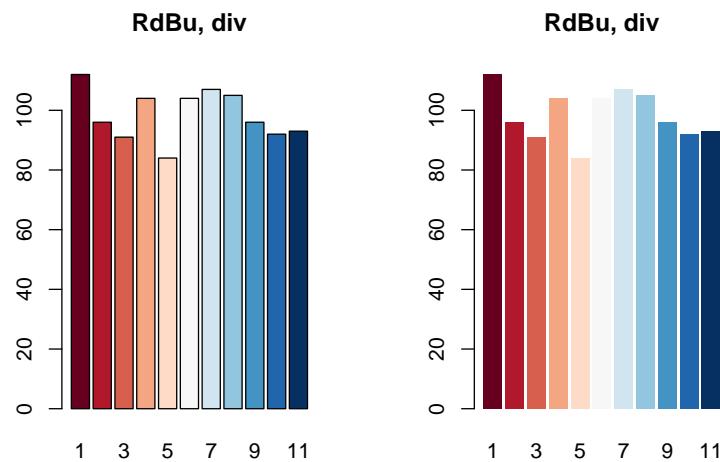
```
par(mfrow = c(1, 2))
barplot(x[1:11], col = brewer.pal(n = 11, name = "PuOr"), names.arg = 1:11,
        main = "PuOr, div")
```

B.2. SZÍNEK VÁLASZTÁSA AZ RCOLORBREWER CSOMAG SEGÍTSÉGÉVEL61

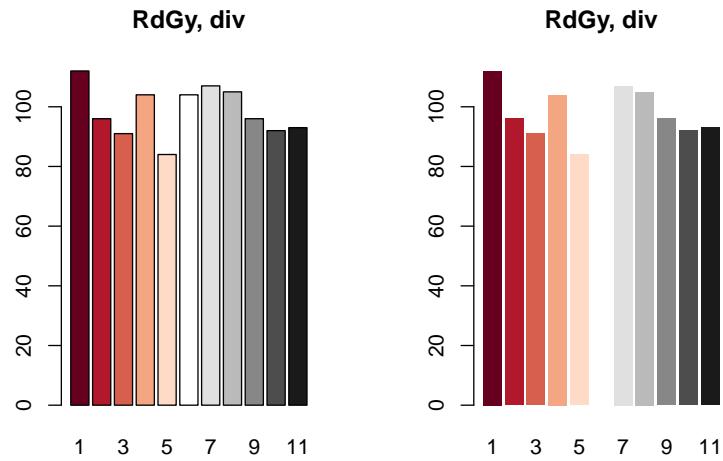
```
barplot(x[1:11], col = brewer.pal(n = 11, name = "PuOr"), names.arg = 1:11,
        main = "PuOr, div", border = NA)
```



```
par(mfrow = c(1, 2))
barplot(x[1:11], col = brewer.pal(n = 11, name = "RdBu"), names.arg = 1:11,
        main = "RdBu, div")
barplot(x[1:11], col = brewer.pal(n = 11, name = "RdBu"), names.arg = 1:11,
        main = "RdBu, div", border = NA)
```

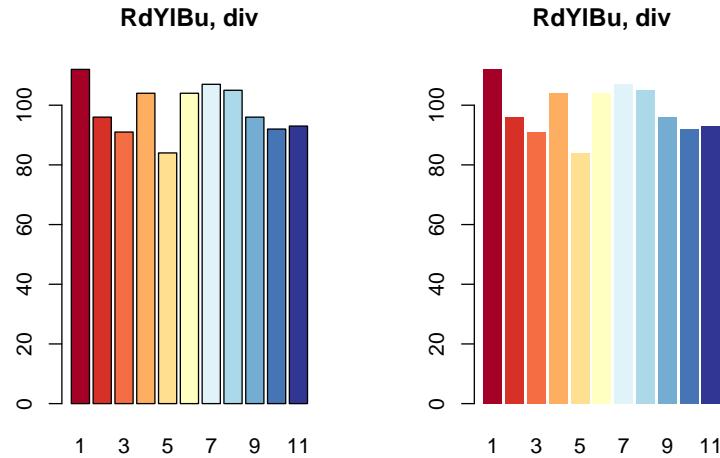


```
par(mfrow = c(1, 2))
barplot(x[1:11], col = brewer.pal(n = 11, name = "RdGy"), names.arg = 1:11,
        main = "RdGy, div")
barplot(x[1:11], col = brewer.pal(n = 11, name = "RdGy"), names.arg = 1:11,
        main = "RdGy, div", border = NA)
```

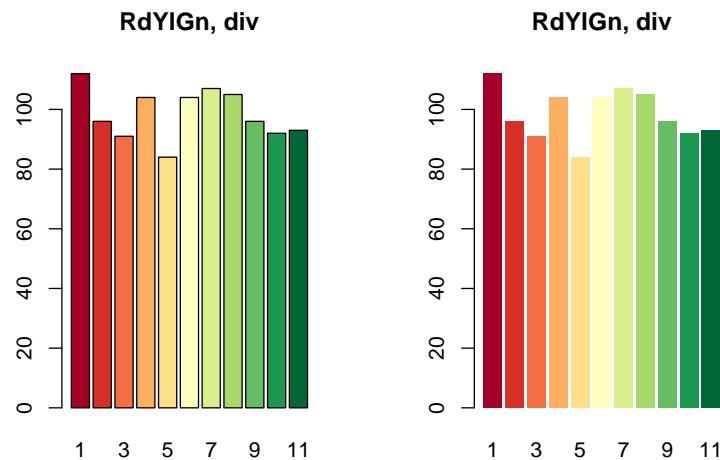


```
par(mfrow = c(1, 2))
barplot(x[1:11], col = brewer.pal(n = 11, name = "RdYlBu"), names.arg = 1:11,
        main = "RdYlBu, div")
barplot(x[1:11], col = brewer.pal(n = 11, name = "RdYlBu"), names.arg = 1:11,
        main = "RdYlBu, div", border = NA)
```

B.2. SZÍNEK VÁLASZTÁSA AZ RCOLORBREWER CSOMAG SEGÍTSÉGÉVEL63

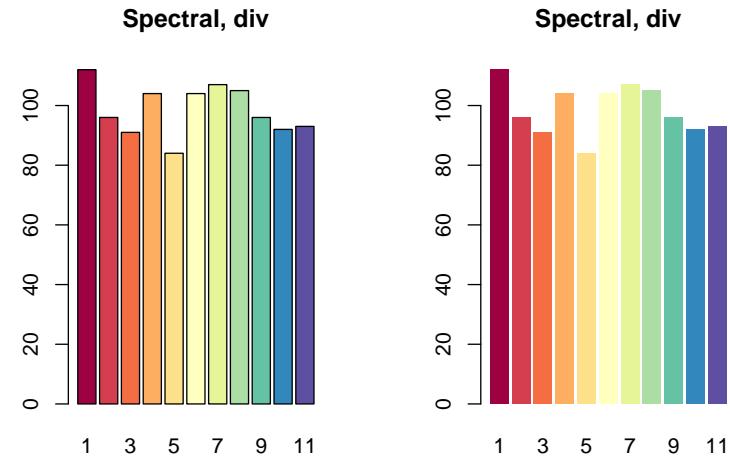


```
par(mfrow = c(1, 2))
barplot(x[1:11], col = brewer.pal(n = 11, name = "RdYlGn"), names.arg = 1:11,
        main = "RdYlGn, div")
barplot(x[1:11], col = brewer.pal(n = 11, name = "RdYlGn"), names.arg = 1:11,
        main = "RdYlGn, div", border = NA)
```

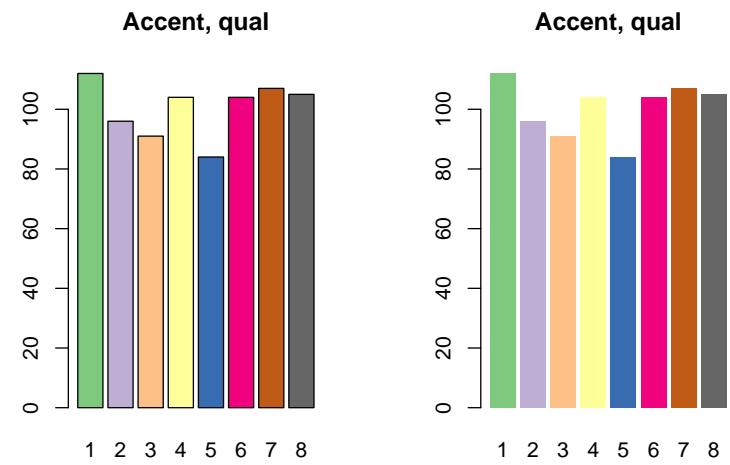


```
par(mfrow = c(1, 2))
barplot(x[1:11], col = brewer.pal(n = 11, name = "Spectral"), names.arg = 1:11,
        main = "Spectral, div")
```

```
barplot(x[1:11], col = brewer.pal(n = 11, name = "Spectral"), names.arg = 1:11,
        main = "Spectral, div", border = NA)
```

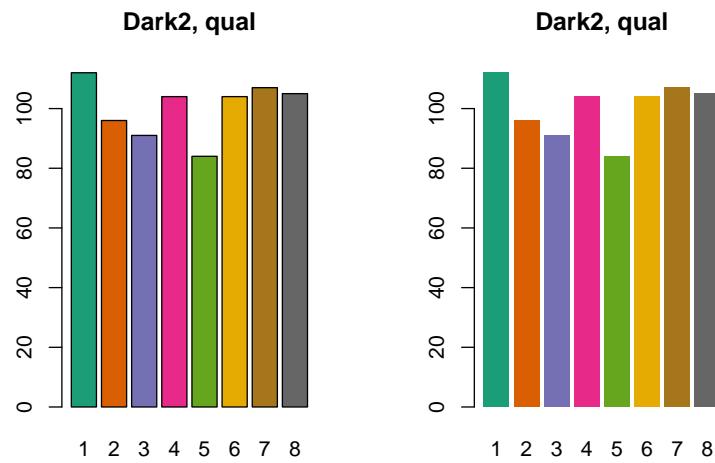


```
par(mfrow = c(1, 2))
barplot(x[1:8], col = brewer.pal(n = 8, name = "Accent"), names.arg = 1:8, main = "Accent",
        border = NA)
barplot(x[1:8], col = brewer.pal(n = 8, name = "Accent"), names.arg = 1:8, main = "Accent",
        border = NA)
```



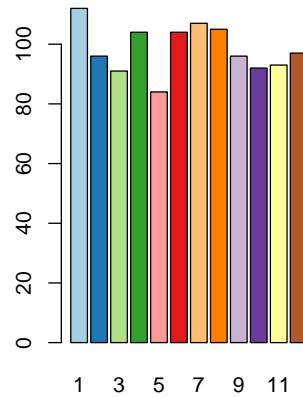
B.2. SZÍNEK VÁLASZTÁSA AZ RCOLORBREWER CSOMAG SEGÍTSÉGÉVEL65

```
par(mfrow = c(1, 2))
barplot(x[1:8], col = brewer.pal(n = 8, name = "Dark2"), names.arg = 1:8, main = "Dark2, qual")
barplot(x[1:8], col = brewer.pal(n = 8, name = "Dark2"), names.arg = 1:8, main = "Dark2, qual",
        border = NA)
```

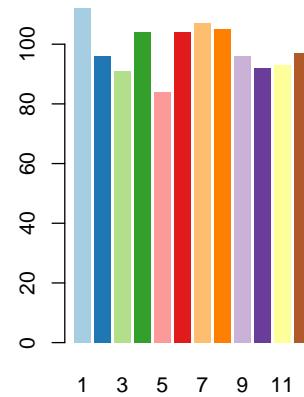


```
par(mfrow = c(1, 2))
barplot(x[1:12], col = brewer.pal(n = 12, name = "Paired"), names.arg = 1:12,
        main = "Paired, qual")
barplot(x[1:12], col = brewer.pal(n = 12, name = "Paired"), names.arg = 1:12,
        main = "Paired, qual", border = NA)
```

Paired, qual

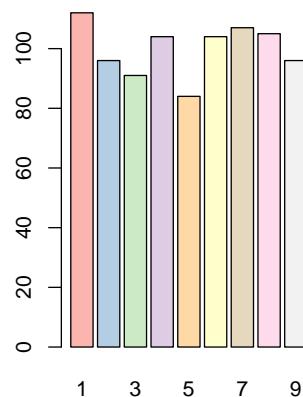


Paired, qual

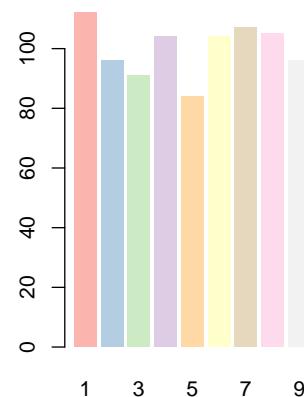


```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "Pastel1"), names.arg = 1:9,
        main = "Pastel1, qual")
barplot(x[1:9], col = brewer.pal(n = 9, name = "Pastel1"), names.arg = 1:9,
        main = "Pastel1, qual", border = NA)
```

Pastel1, qual



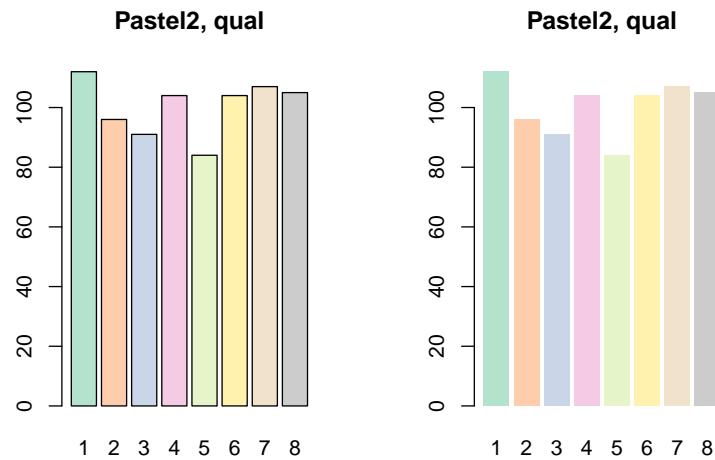
Pastel1, qual



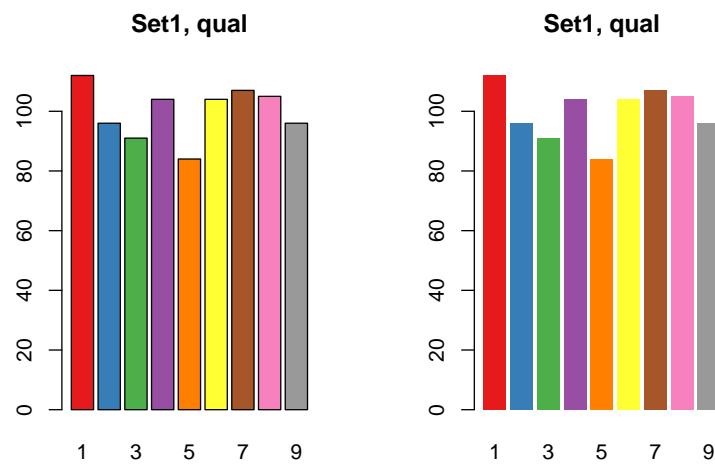
```
par(mfrow = c(1, 2))
barplot(x[1:8], col = brewer.pal(n = 8, name = "Pastel2"), names.arg = 1:8,
        main = "Pastel2, qual")
```

B.2. SZÍNEK VÁLASZTÁSA AZ RCOLORBREWER CSOMAG SEGÍTSÉGÉVEL67

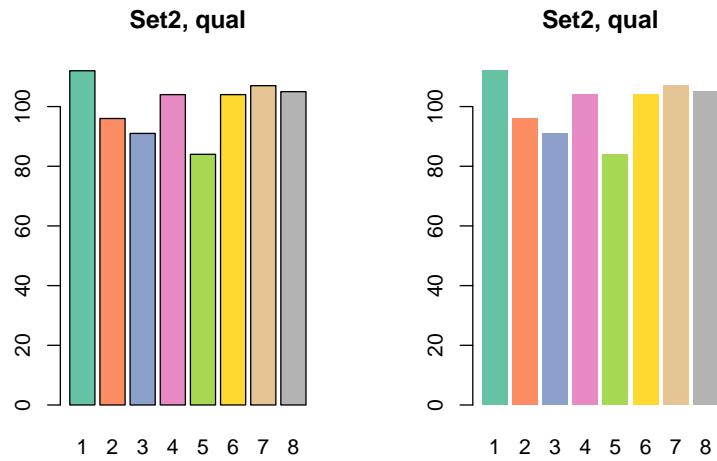
```
barplot(x[1:8], col = brewer.pal(n = 8, name = "Pastel2"), names.arg = 1:8,
        main = "Pastel2, qual", border = NA)
```



```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "Set1"), names.arg = 1:9, main = "Set1, qual")
barplot(x[1:9], col = brewer.pal(n = 9, name = "Set1"), names.arg = 1:9, main = "Set1, qual",
        border = NA)
```

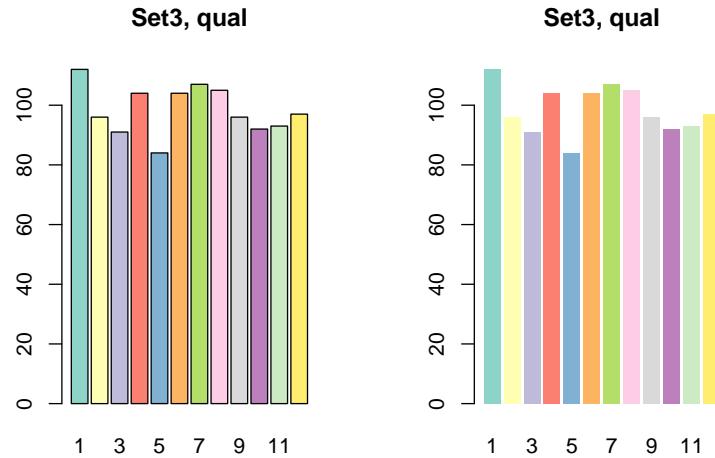


```
par(mfrow = c(1, 2))
barplot(x[1:8], col = brewer.pal(n = 8, name = "Set2"), names.arg = 1:8, main = "Set2",
barplot(x[1:8], col = brewer.pal(n = 8, name = "Set2"), names.arg = 1:8, main = "Set2",
        border = NA)
```

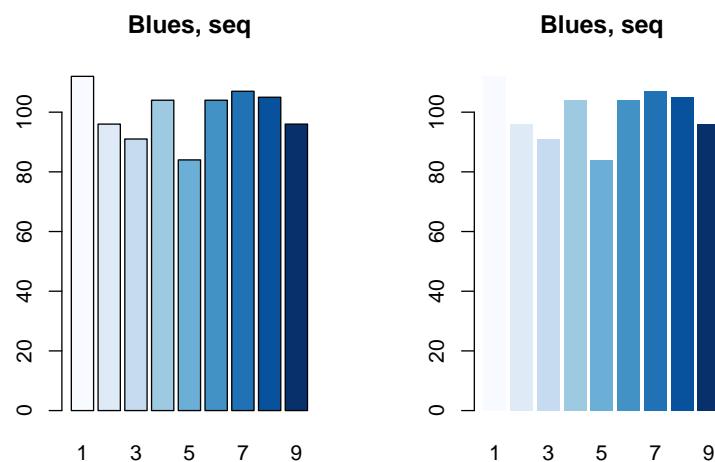


```
par(mfrow = c(1, 2))
barplot(x[1:12], col = brewer.pal(n = 12, name = "Set3"), names.arg = 1:12,
        main = "Set3, qual")
barplot(x[1:12], col = brewer.pal(n = 12, name = "Set3"), names.arg = 1:12,
        main = "Set3, qual", border = NA)
```

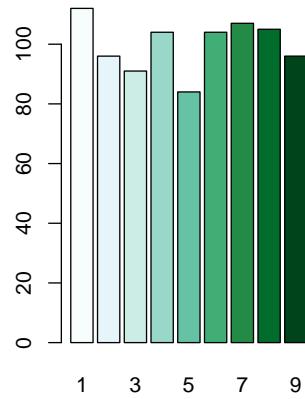
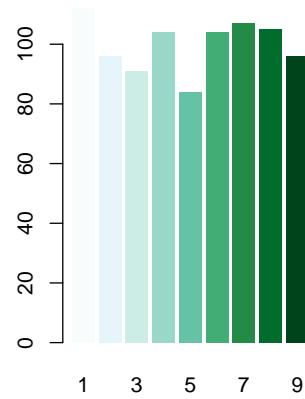
B.2. SZÍNEK VÁLASZTÁSA AZ RCOLORBREWER CSOMAG SEGÍTSÉGÉVEL69



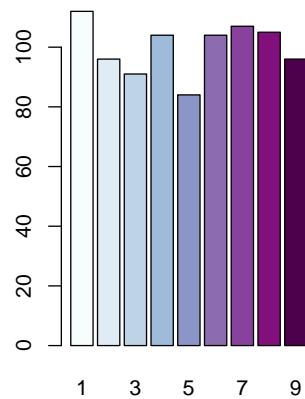
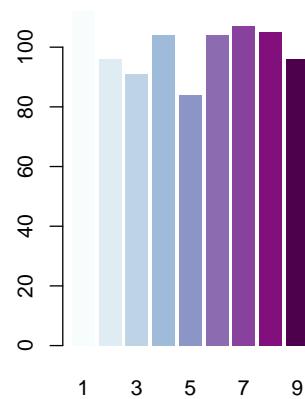
```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "Blues"), names.arg = 1:9, main = "Blues, seq")
barplot(x[1:9], col = brewer.pal(n = 9, name = "Blues"), names.arg = 1:9, main = "Blues, seq",
        border = NA)
```



```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "BuGn"), names.arg = 1:9, main = "BuGn, seq")
barplot(x[1:9], col = brewer.pal(n = 9, name = "BuGn"), names.arg = 1:9, main = "BuGn, seq",
        border = NA)
```

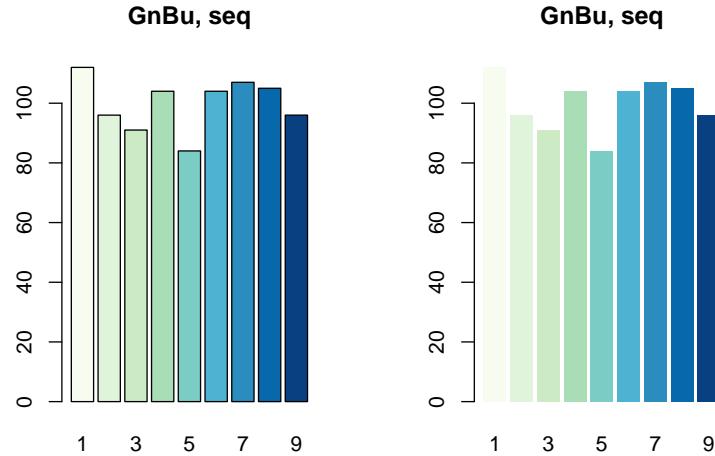
BuGn, seq**BuGn, seq**

```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "BuPu"), names.arg = 1:9, main = "BuPu",
barplot(x[1:9], col = brewer.pal(n = 9, name = "BuPu"), names.arg = 1:9, main = "BuPu",
        border = NA)
```

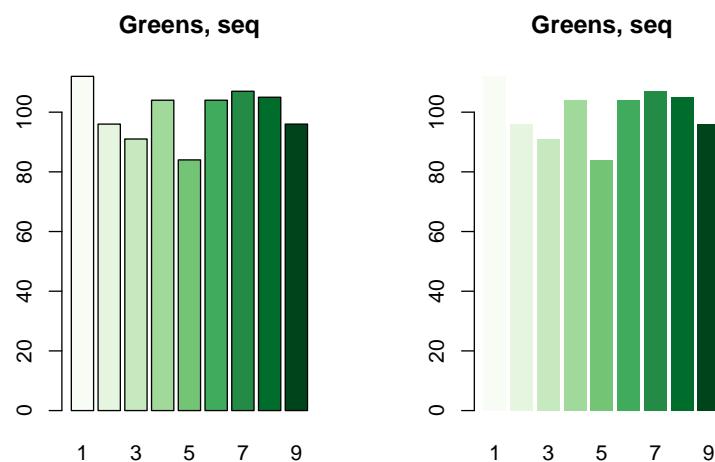
BuPu, seq**BuPu, seq**

```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "GnBu"), names.arg = 1:9, main = "GnBu",
barplot(x[1:9], col = brewer.pal(n = 9, name = "GnBu"), names.arg = 1:9, main = "GnBu",
        border = NA)
```

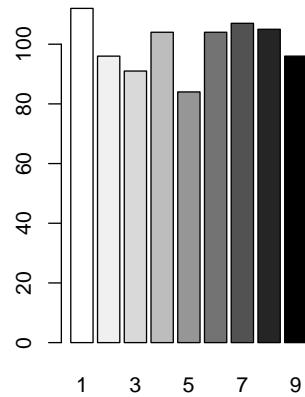
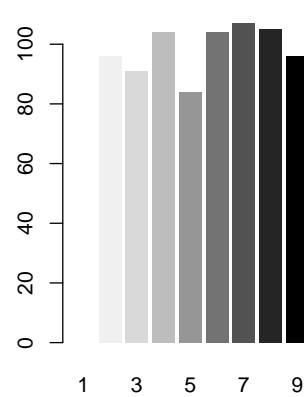
B.2. SZÍNEK VÁLASZTÁSA AZ RCOLORBREWER CSOMAG SEGÍTSÉGÉVEL 71



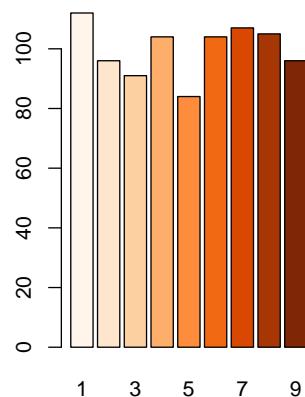
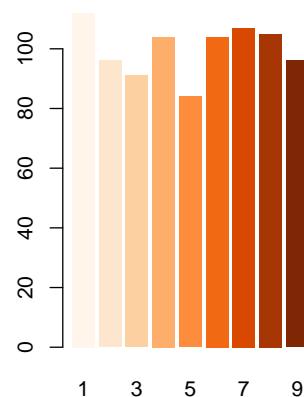
```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "Greens"), names.arg = 1:9, main = "Greens, seq")
barplot(x[1:9], col = brewer.pal(n = 9, name = "Greens"), names.arg = 1:9, main = "Greens, seq",
       border = NA)
```



```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "Greys"), names.arg = 1:9, main = "Greys, seq")
barplot(x[1:9], col = brewer.pal(n = 9, name = "Greys"), names.arg = 1:9, main = "Greys, seq",
       border = NA)
```

Greys, seq**Greys, seq**

```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "Oranges"), names.arg = 1:9,
        main = "Oranges, seq")
barplot(x[1:9], col = brewer.pal(n = 9, name = "Oranges"), names.arg = 1:9,
        main = "Oranges, seq", border = NA)
```

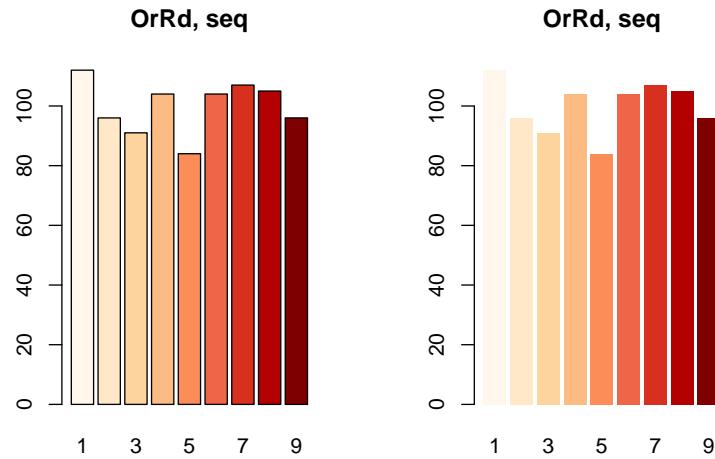
Oranges, seq**Oranges, seq**

```
par(mfrow = c(1, 2))
```

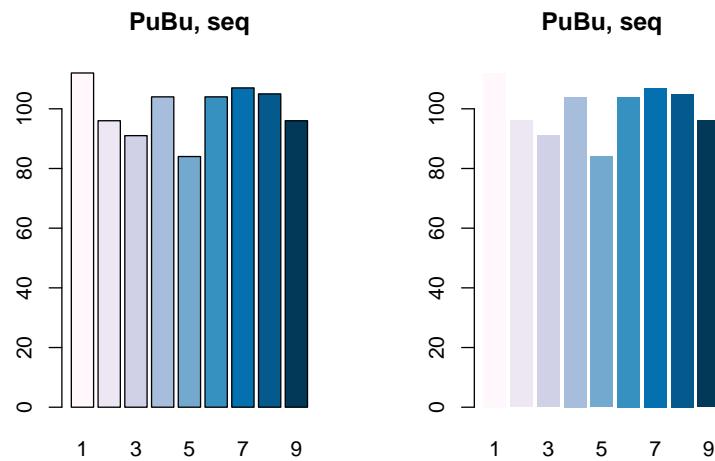
```
barplot(x[1:9], col = brewer.pal(n = 9, name = "OrRd"), names.arg = 1:9, main = "OrRd",
barplot(x[1:9], col = brewer.pal(n = 9, name = "OrRd"), names.arg = 1:9, main = "OrRd")
```

B.2. SZÍNEK VÁLASZTÁSA AZ RCOLORBREWER CSOMAG SEGÍTSÉGÉVEL 73

```
border = NA)
```

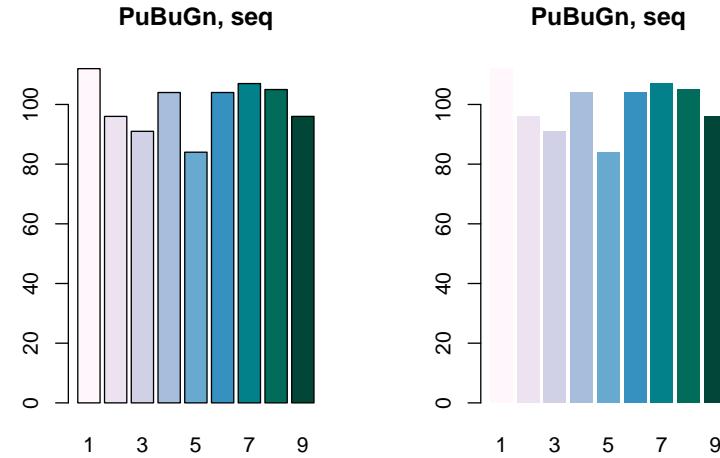


```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "PuBu"), names.arg = 1:9, main = "PuBu, seq")
barplot(x[1:9], col = brewer.pal(n = 9, name = "PuBu"), names.arg = 1:9, main = "PuBu, seq",
       border = NA)
```

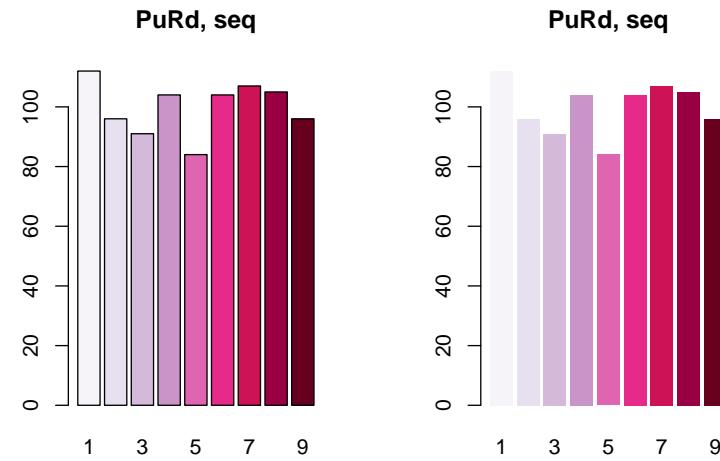


```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "PuBuGn"), names.arg = 1:9, main = "PuBuGn, seq")
```

```
barplot(x[1:9], col = brewer.pal(n = 9, name = "PuBuGn"), names.arg = 1:9, main = "PuBuGn, seq", border = NA)
```

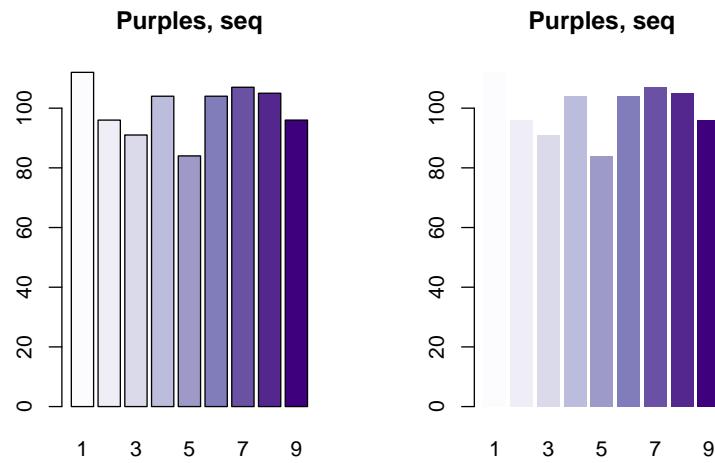


```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "PuRd"), names.arg = 1:9, main = "PuRd, seq", border = NA)
barplot(x[1:9], col = brewer.pal(n = 9, name = "PuRd"), names.arg = 1:9, main = "PuRd, seq", border = NA)
```

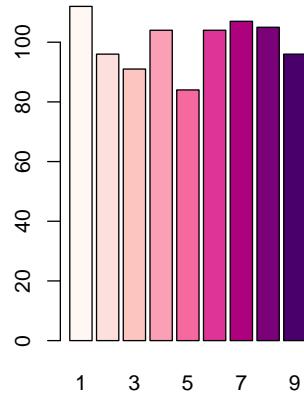
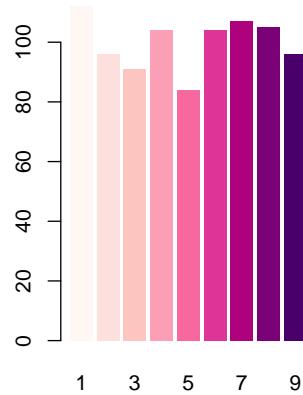


B.2. SZÍNEK VÁLASZTÁSA AZ RCOLORBREWER CSOMAG SEGÍTSÉGÉVEL 75

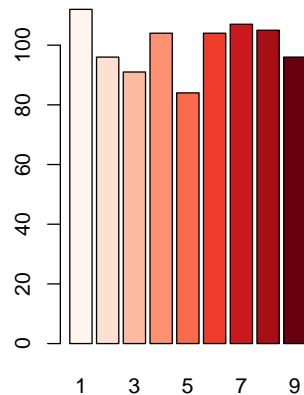
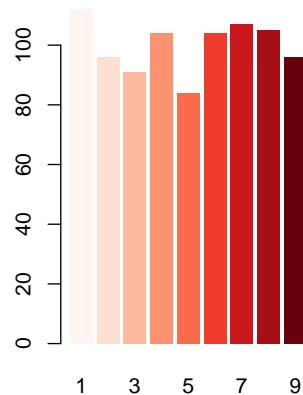
```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "Purples"), names.arg = 1:9,
        main = "Purples, seq")
barplot(x[1:9], col = brewer.pal(n = 9, name = "Purples"), names.arg = 1:9,
        main = "Purples, seq", border = NA)
```



```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "RdPu"), names.arg = 1:9, main = "RdPu, seq")
barplot(x[1:9], col = brewer.pal(n = 9, name = "RdPu"), names.arg = 1:9, main = "RdPu, seq",
        border = NA)
```

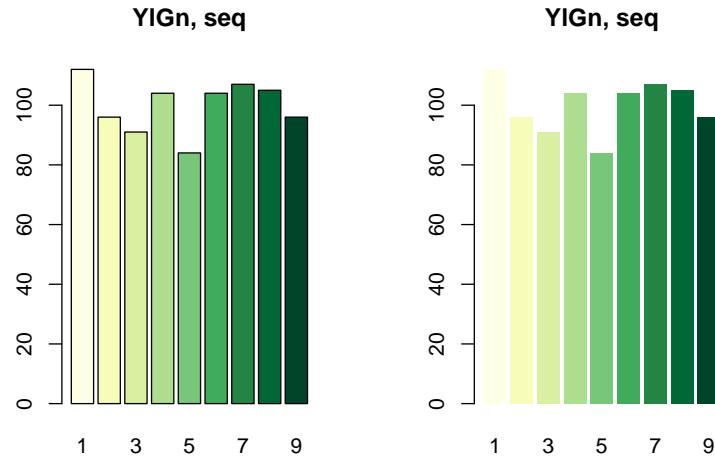
RdPu, seq**RdPu, seq**

```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "Reds"), names.arg = 1:9, main = "Reds,
barplot(x[1:9], col = brewer.pal(n = 9, name = "Reds"), names.arg = 1:9, main = "Reds,
border = NA)
```

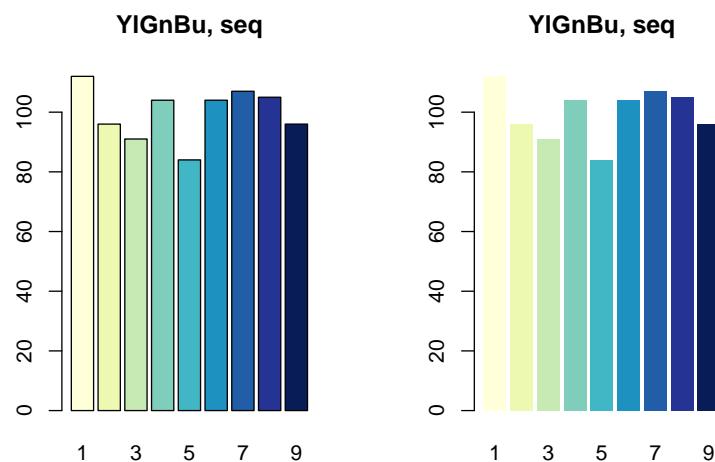
Reds, seq**Reds, seq**

```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "YlGn"), names.arg = 1:9, main = "YlGn,
barplot(x[1:9], col = brewer.pal(n = 9, name = "YlGn"), names.arg = 1:9, main = "YlGn,
border = NA)
```

B.2. SZÍNEK VÁLASZTÁSA AZ RCOLORBREWER CSOMAG SEGÍTSÉGÉVEL 77

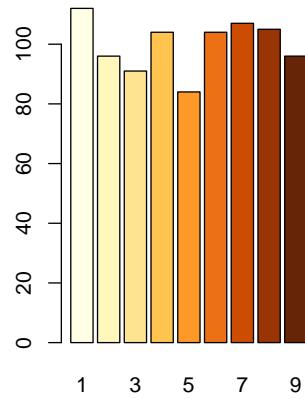


```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "YlGnBu"), names.arg = 1:9, main = "YlGnBu, seq")
barplot(x[1:9], col = brewer.pal(n = 9, name = "YlGnBu"), names.arg = 1:9, main = "YlGnBu, seq",
        border = NA)
```

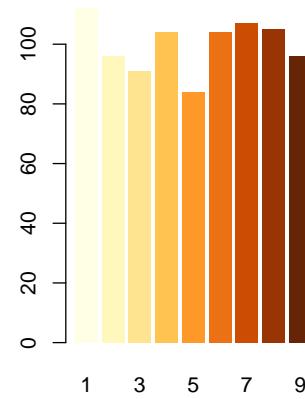


```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "YlOrBr"), names.arg = 1:9, main = "YlOrBr, seq")
barplot(x[1:9], col = brewer.pal(n = 9, name = "YlOrBr"), names.arg = 1:9, main = "YlOrBr, seq",
        border = NA)
```

YlOrBr, seq

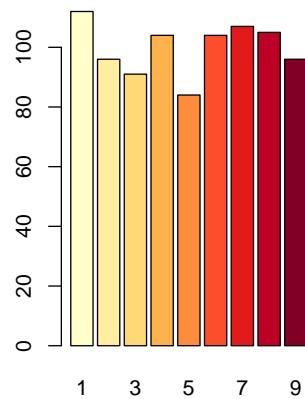


YlOrBr, seq

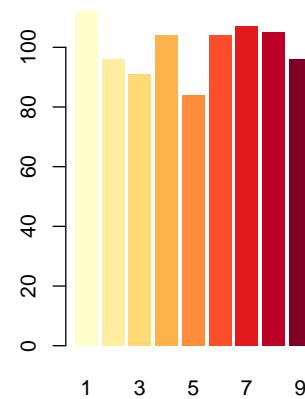


```
par(mfrow = c(1, 2))
barplot(x[1:9], col = brewer.pal(n = 9, name = "YlOrRd"), names.arg = 1:9, main = "YlOrRd, seq")
barplot(x[1:9], col = brewer.pal(n = 9, name = "YlOrRd"), names.arg = 1:9, main = "YlOrBr, seq",
        border = NA)
```

YlOrRd, seq



YlOrRd, seq



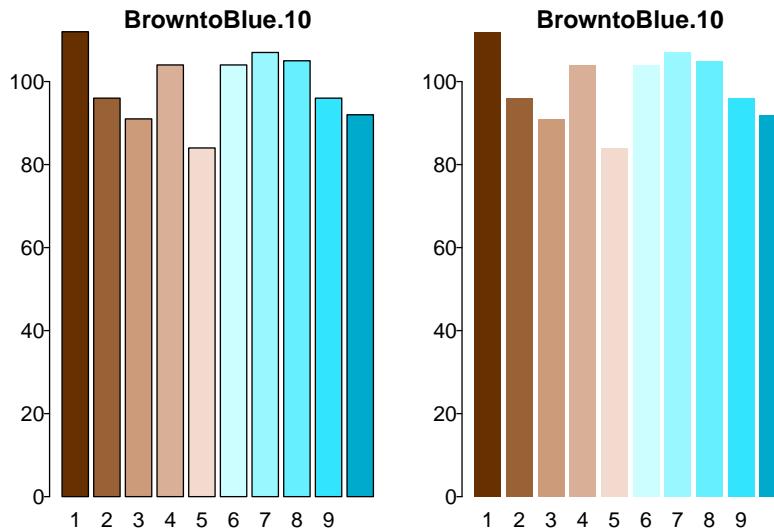
B.3. Színek választása a dichromat csomag segítéssel

A dichromat csomag színsémai közül is választhatunk. A színsémákat a colorschemes lista tartalmazza. A lisatelemek neve:

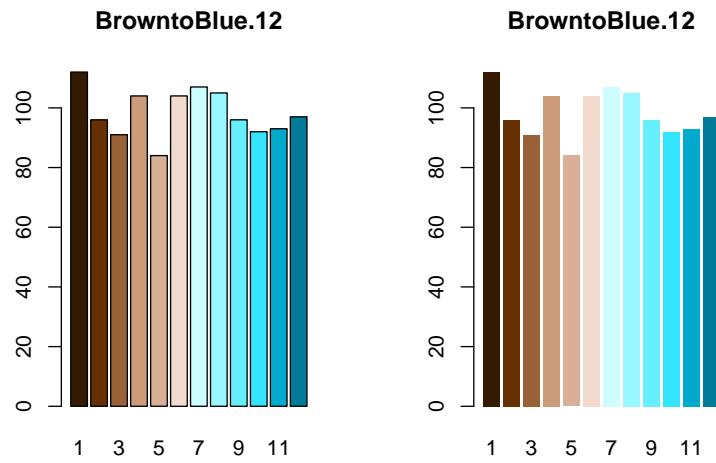
```
library(dichromat)
names(colorschemes)
#> [1] "BrownToBlue.10"           "BrownToBlue.12"           "BlueToDarkOrange.12"
#> [4] "BlueToDarkOrange.18"      "DarkRedToBlue.12"         "DarkRedToBlue.18"
#> [7] "BlueToGreen.14"           "BlueToGray.8"             "BlueToOrangeRed.14"
#> [10] "BlueToOrange.10"          "BlueToOrange.12"           "BlueToOrange.8"
#> [13] "LightBlueToDarkBlue.10" "LightBlueToDarkBlue.7"     "Categorical.12"
#> [16] "GreenToMagenta.16"       "SteppedSequential.5"
```

Példák színek választására:

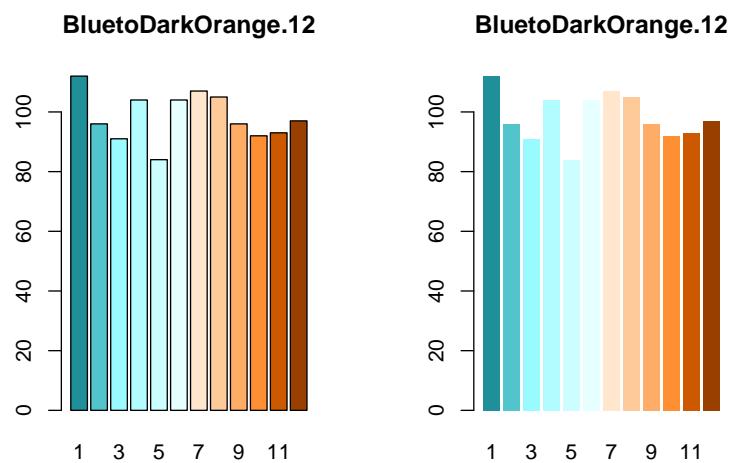
```
# az x adatvektor beállítása
set.seed(0)
x <- rpois(n = 50, lambda = 100)
# grafikus paraméterek beállítása
par(las = 1, mgp = c(0, 0.2, 0), tcl = -0.2, mar = c(3, 2, 1, 1))
library(dichromat)
par(mfrow=c(1,2))
barplot(x[1:10], col = colorschemes$BrownToBlue.10, names.arg = 1:10, main = "BrownToBlue.10")
barplot(x[1:10], col = colorschemes$BrownToBlue.10, names.arg = 1:10, main = "BrownToBlue.10", b
```



```
par(mfrow=c(1,2))
barplot(x[1:12], col = colorschemes$BrownToBlue.12, names.arg = 1:12, main = "BrownToBlue.12")
barplot(x[1:12], col = colorschemes$BrownToBlue.12, names.arg = 1:12, main = "BrownToBlue.12")
```

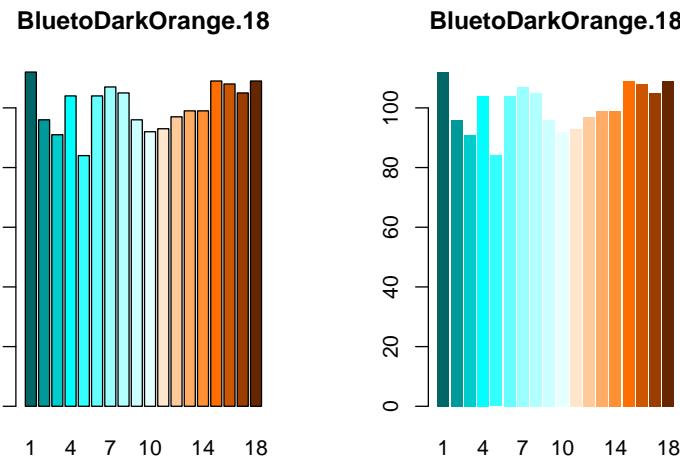


```
par(mfrow=c(1,2))
barplot(x[1:12], col = colorschemes$BluetooDarkOrange.12, names.arg = 1:12, main = "BluetooDarkOrange.12")
barplot(x[1:12], col = colorschemes$BluetooDarkOrange.12, names.arg = 1:12, main = "BluetooDarkOrange.12")
```

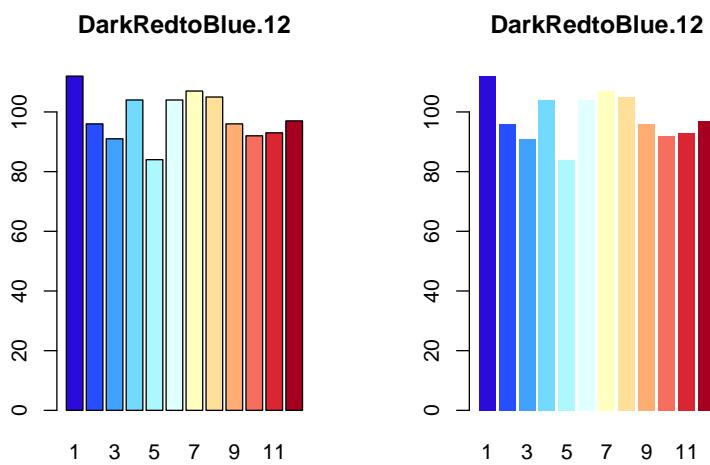


B.3. SZÍNEK VÁLASZTÁSA A DICHROMAT CSOMAG SEGÍTSÉGÉVEL 81

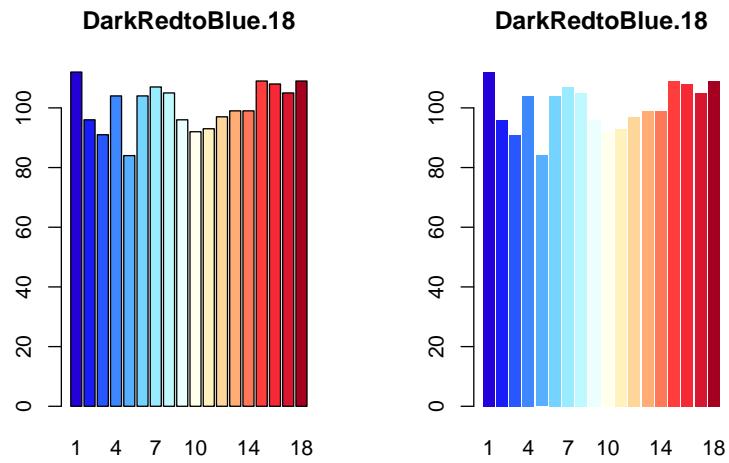
```
par(mfrow=c(1,2))
barplot(x[1:18], col = colorschemes$Bluetodarkorange.18, names.arg = 1:18, main = "Bluetodarkorange.18")
barplot(x[1:18], col = colorschemes$Bluetodarkorange.18, names.arg = 1:18, main = "Bluetodarkorange.18")
```



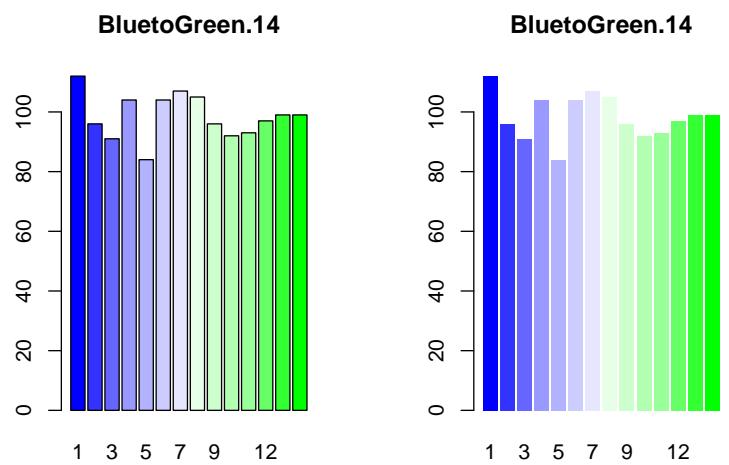
```
par(mfrow=c(1,2))
barplot(x[1:12], col = colorschemes$DarkRedtoblue.12, names.arg = 1:12, main = "DarkRedtoblue.12")
barplot(x[1:12], col = colorschemes$DarkRedtoblue.12, names.arg = 1:12, main = "DarkRedtoblue.12")
```



```
par(mfrow=c(1,2))
barplot(x[1:18], col = colorschemes$DarkRedtoBlue.18, names.arg = 1:18, main = "DarkRedtoBlue.18")
barplot(x[1:18], col = colorschemes$DarkRedtoBlue.18, names.arg = 1:18, main = "DarkRedtoBlue.18")
```

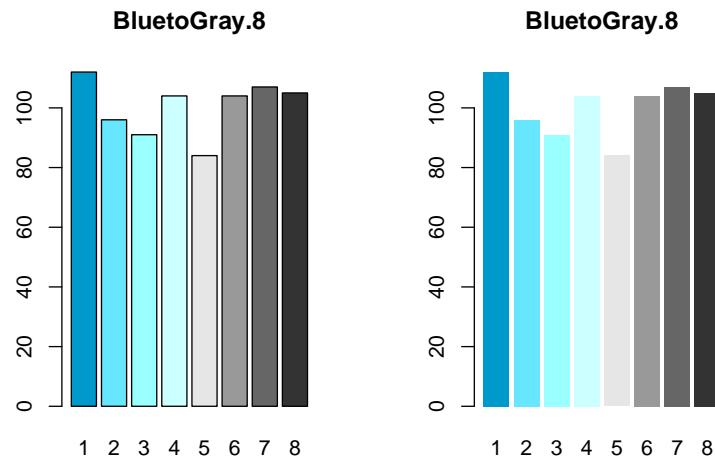


```
par(mfrow=c(1,2))
barplot(x[1:14], col = colorschemes$Bluetogreen.14, names.arg = 1:14, main = "Bluetogreen.14")
barplot(x[1:14], col = colorschemes$Bluetogreen.14, names.arg = 1:14, main = "Bluetogreen.14")
```

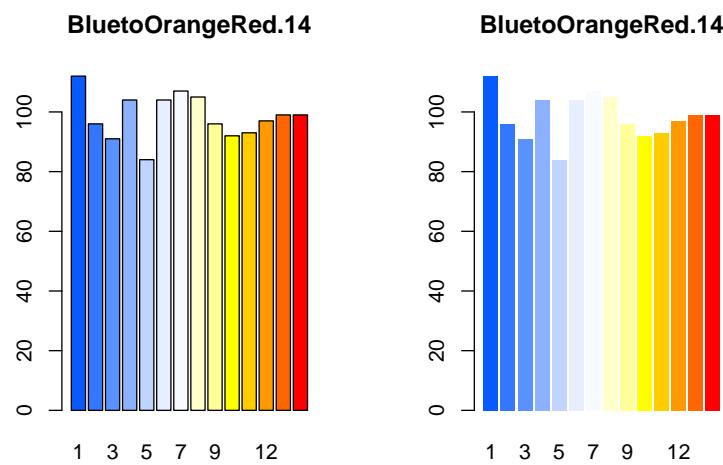


B.3. SZÍNEK VÁLASZTÁSA A DICHROMAT CSOMAG SEGÍTSÉGÉVEL 83

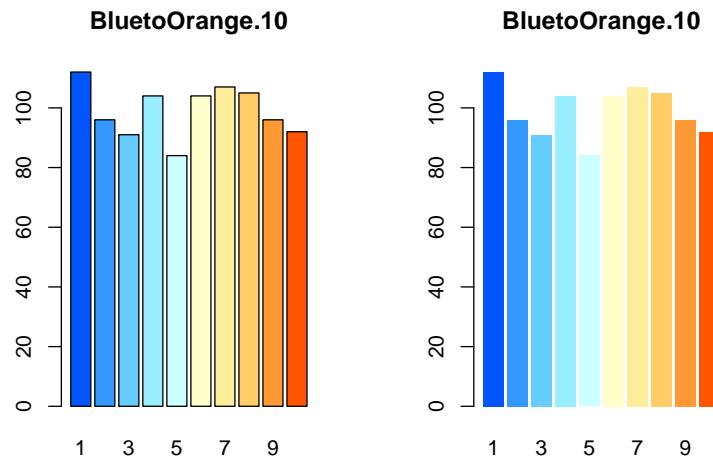
```
par(mfrow=c(1,2))
barplot(x[1:8], col = colorschemes$BluetoGray.8, names.arg = 1:8, main = "BluetoGray.8")
barplot(x[1:8], col = colorschemes$BluetoGray.8, names.arg = 1:8, main = "BluetoGray.8", border =
```



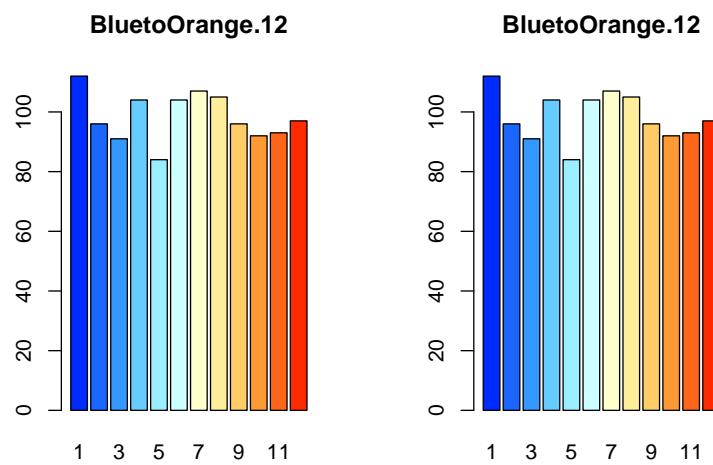
```
par(mfrow=c(1,2))
barplot(x[1:14], col = colorschemes$BluetoOrangeRed.14, names.arg = 1:14, main = "BluetoOrangeRed.14")
barplot(x[1:14], col = colorschemes$BluetoOrangeRed.14, names.arg = 1:14, main = "BluetoOrangeRed.14", border =
```



```
par(mfrow=c(1,2))
barplot(x[1:10], col = colorschemes$BluetooOrange.10, names.arg = 1:10, main = "BluetooOrange.10")
barplot(x[1:10], col = colorschemes$BluetooOrange.10, names.arg = 1:10, main = "BluetooOrange.10")
```

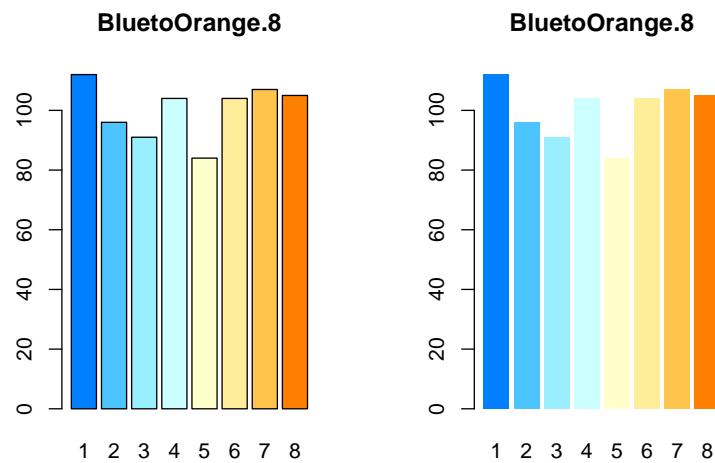


```
par(mfrow=c(1,2))
barplot(x[1:12], col = colorschemes$BluetooOrange.12, names.arg = 1:12, main = "BluetooOrange.12")
barplot(x[1:12], col = colorschemes$BluetooOrange.12, names.arg = 1:12, main = "BluetooOrange.12")
```

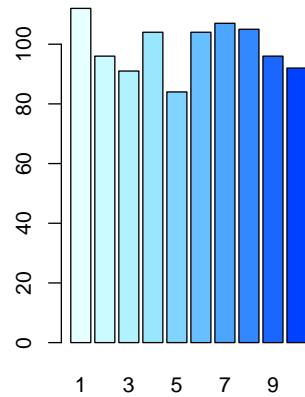
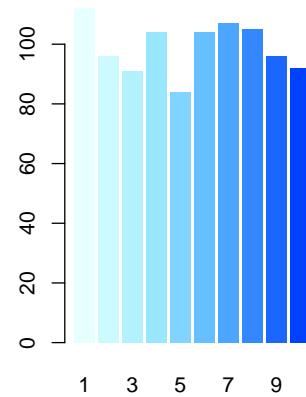


B.3. SZÍNEK VÁLASZTÁSA A DICHROMAT CSOMAG SEGÍTSÉGÉVEL 85

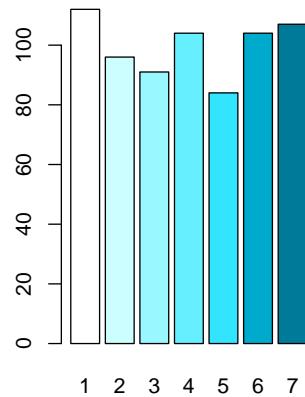
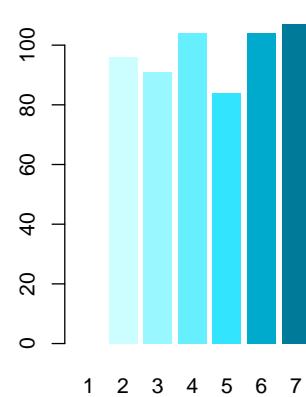
```
par(mfrow=c(1,2))
barplot(x[1:8], col = colorschemes$BluetooOrange.8, names.arg = 1:8, main = "BluetooOrange.8")
barplot(x[1:8], col = colorschemes$BluetooOrange.8, names.arg = 1:8, main = "BluetooOrange.8", border = NA)
```



```
par(mfrow=c(1,2))
barplot(x[1:10], col = colorschemes$LightBluetooDarkBlue.10, names.arg = 1:10,
        main = "LightBluetooDarkBlue.10")
barplot(x[1:10], col = colorschemes$LightBluetooDarkBlue.10, names.arg = 1:10,
        main = "LightBluetooDarkBlue.10", border = NA)
```

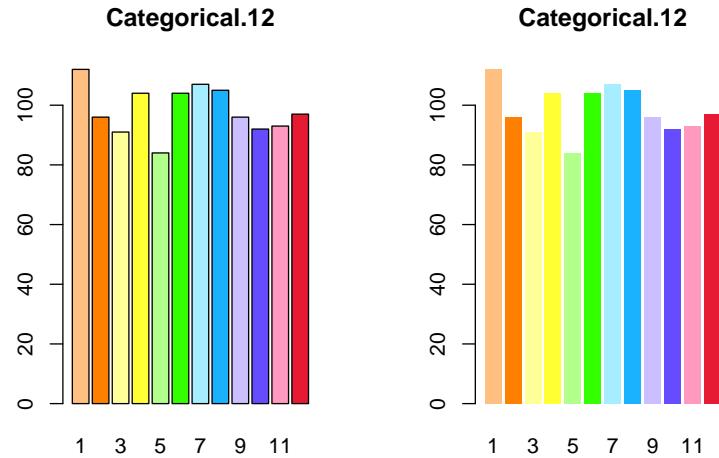
LightBluetooDarkBlue.10**LightBluetooDarkBlue.10**

```
par(mfrow=c(1,2))
barplot(x[1:7], col = colorschemes$LightBluetooDarkBlue.7, names.arg = 1:7, main = "LightBluetooDarkBlue.7")
barplot(x[1:7], col = colorschemes$LightBluetooDarkBlue.7, names.arg = 1:7, main = "LightBluetooDarkBlue.7")
```

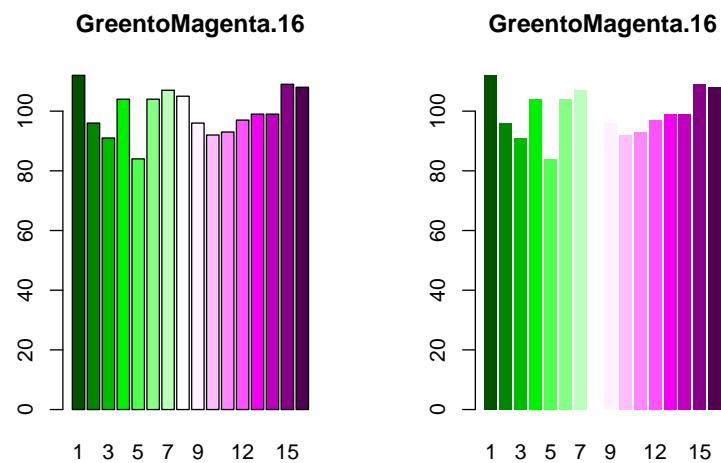
LightBluetooDarkBlue.7**LightBluetooDarkBlue.7**

```
par(mfrow=c(1,2))
barplot(x[1:12], col = colorschemes$Categorical.12, names.arg = 1:12, main = "Categorical.12")
barplot(x[1:12], col = colorschemes$Categorical.12, names.arg = 1:12, main = "Categorical.12")
```

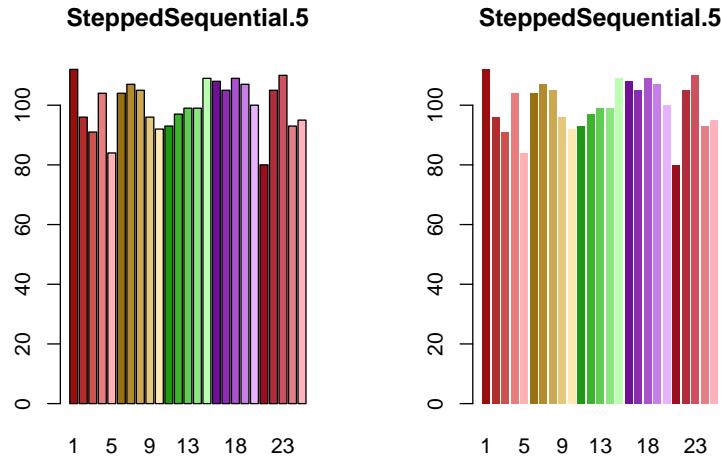
B.3. SZÍNEK VÁLASZTÁSA A DICHROMAT CSOMAG SEGÍTSÉGÉVEL 87



```
par(mfrow=c(1,2))
barplot(x[1:16], col = colorschemes$GreentoMagenta.16, names.arg = 1:16, main = "GreentoMagenta.16")
barplot(x[1:16], col = colorschemes$GreentoMagenta.16, names.arg = 1:16, main = "GreentoMagenta.16")
```



```
par(mfrow=c(1,2))
barplot(x[1:25], col = colorschemes$SteppedSequential.5, names.arg = 1:25, main = "SteppedSequential.5")
barplot(x[1:25], col = colorschemes$SteppedSequential.5, names.arg = 1:25, main = "SteppedSequential.5")
```



B.4. Színek választása egyéb paletta segítségével

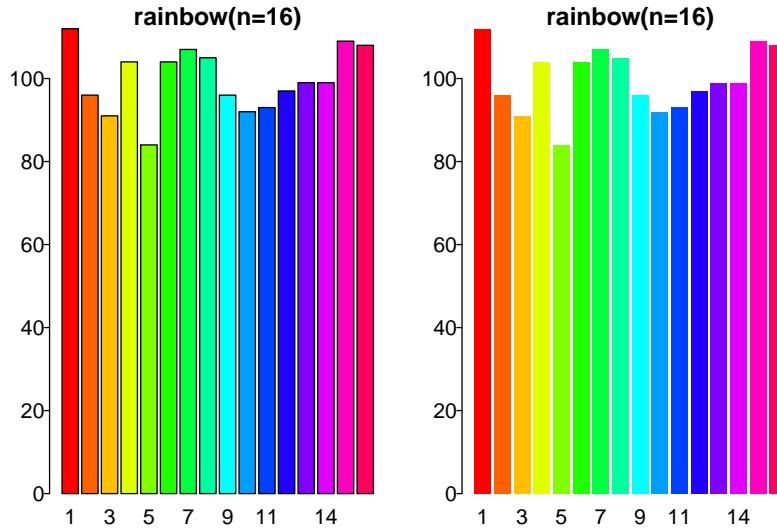
Színpaletta létrehozásához több beépített lehetőségek közül is választhatunk.

```
rainbow(n, start=0, end, alpha = 1)
heat.colors(n, alpha = 1)
terrain.colors(n, alpha = 1)
topo.colors(n, alpha = 1)
cm.colors(n, alpha = 1)
```

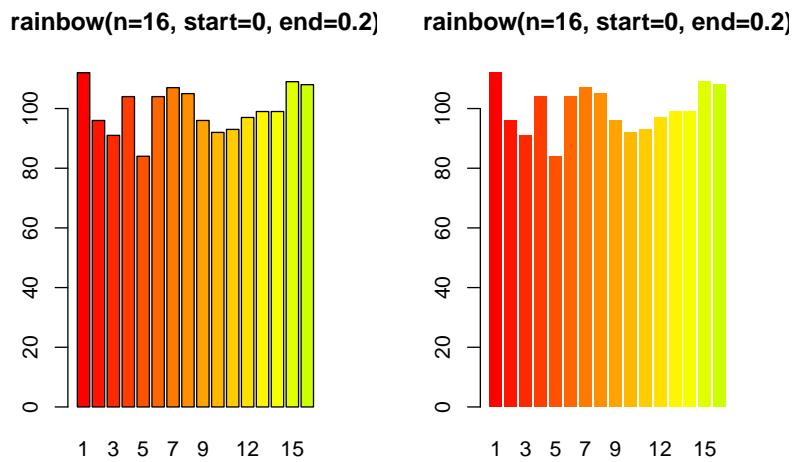
Az n= argumentum a létrehozandó színek számát jelenti.

Példák színpaletta létrehozására:

```
# az x adatvektor beállítása
set.seed(0)
x <- rpois(n = 50, lambda = 100)
# grafikus paraméterek beállítása
par(las = 1, mgp = c(0, 0.2, 0), tcl = -0.2, mar = c(3, 2, 1, 1))
par(mfrow=c(1,2))
barplot(x[1:16], col = rainbow(16), names.arg = 1:16, main = "rainbow(n=16)")
barplot(x[1:16], col = rainbow(16), names.arg = 1:16, main = "rainbow(n=16)", border =
```



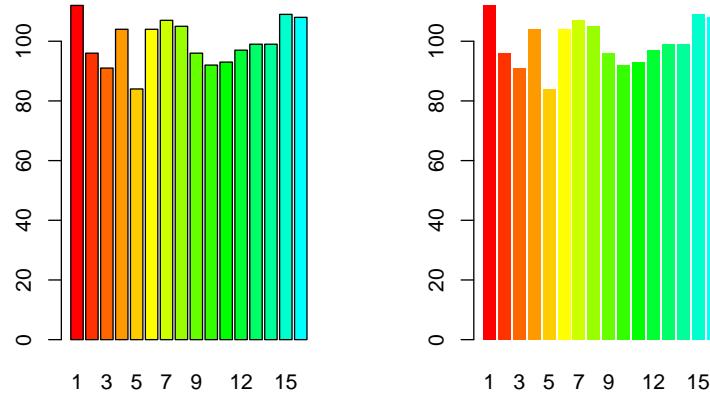
```
par(mfrow=c(1,2))
barplot(x[1:16], col = rainbow(16, start = 0, end = 0.2), names.arg = 1:16,
        main = "rainbow(n=16, start=0, end=0.2)")
barplot(x[1:16], col = rainbow(16, start = 0, end = 0.2), names.arg = 1:16,
        main = "rainbow(n=16, start=0, end=0.2)", border = NA)
```



```
par(mfrow=c(1,2))
barplot(x[1:16], col = rainbow(16, start = 0, end = 0.5), names.arg = 1:16,
        main = "rainbow(n=16, start=0, end=0.5)")
```

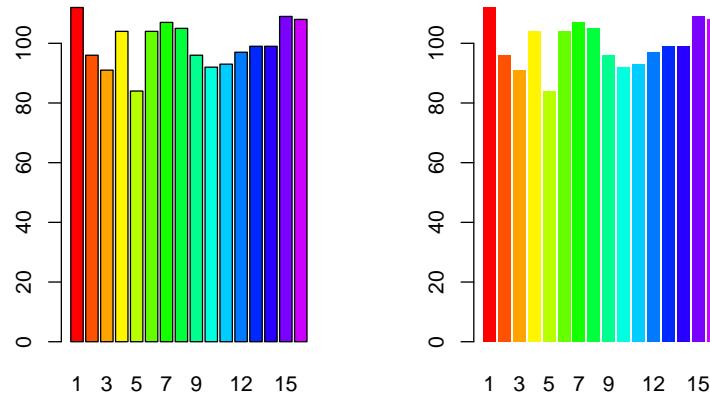
```
barplot(x[1:16], col = rainbow(16, start = 0, end = 0.5), names.arg = 1:16,
        main = "rainbow(n=16, start=0, end=0.5)", border = NA)
```

rainbow(n=16, start=0, end=0.5) rainbow(n=16, start=0, end=0.5)

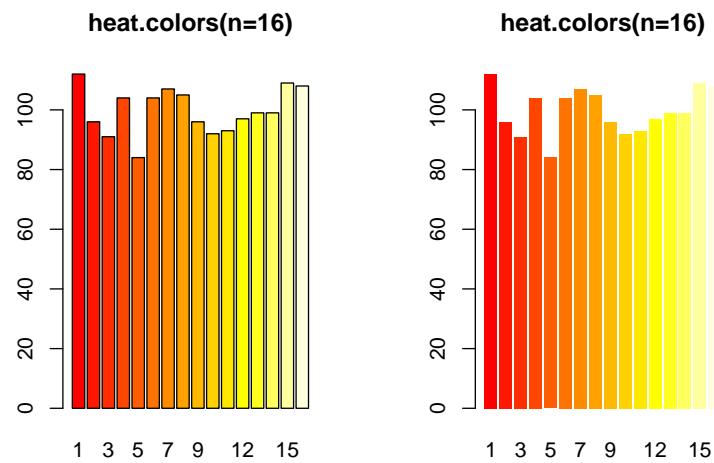


```
par(mfrow=c(1,2))
barplot(x[1:16], col = rainbow(16, start = 0, end = 0.8), names.arg = 1:16,
        main = "rainbow(n=16, start=0, end=0.8)")
barplot(x[1:16], col = rainbow(16, start = 0, end = 0.8), names.arg = 1:16,
        main = "rainbow(n=16, start=0, end=0.8)", border = NA)
```

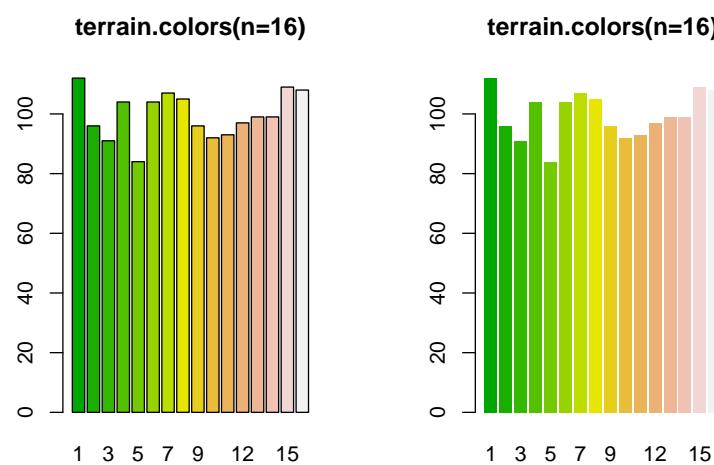
rainbow(n=16, start=0, end=0.8) rainbow(n=16, start=0, end=0.8)



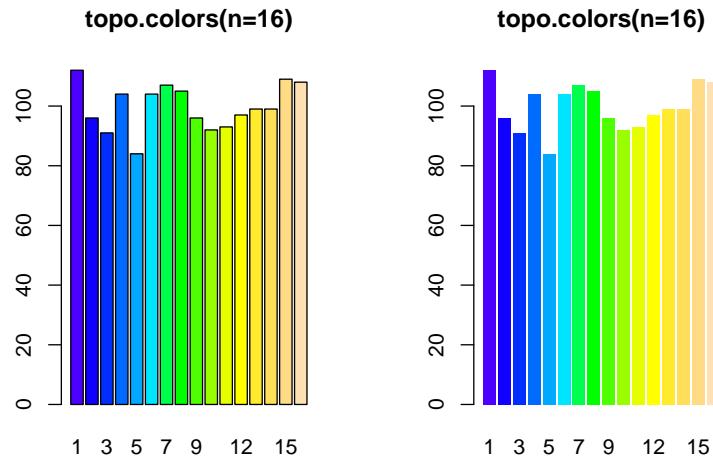
```
par(mfrow=c(1,2))
barplot(x[1:16], col = heat.colors(16), names.arg = 1:16, main = "heat.colors(n=16)")
barplot(x[1:16], col = heat.colors(16), names.arg = 1:16, main = "heat.colors(n=16)", border = NA)
```



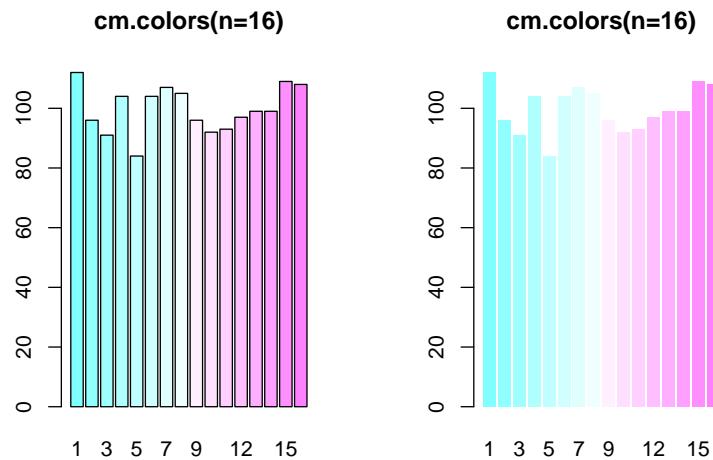
```
par(mfrow=c(1,2))
barplot(x[1:16], col = terrain.colors(16), names.arg = 1:16, main = "terrain.colors(n=16)")
barplot(x[1:16], col = terrain.colors(16), names.arg = 1:16, main = "terrain.colors(n=16)", border = NA)
```



```
par(mfrow=c(1,2))
barplot(x[1:16], col = topo.colors(16), names.arg = 1:16, main = "topo.colors(n=16)")
barplot(x[1:16], col = topo.colors(16), names.arg = 1:16, main = "topo.colors(n=16)", border=1)
```



```
par(mfrow=c(1,2))
barplot(x[1:16], col = cm.colors(16), names.arg = 1:16, main = "cm.colors(n=16)")
barplot(x[1:16], col = cm.colors(16), names.arg = 1:16, main = "cm.colors(n=16)", border=1)
```



B.5. A 657 színnév

```

colors()
#> [1] "white"
#> [4] "antiquewhite1"
#> [7] "antiquewhite4"
#> [10] "aquamarine2"
#> [13] "azure"
#> [16] "azure3"
#> [19] "bisque"
#> [22] "bisque3"
#> [25] "blanchedalmond"
#> [28] "blue2"
#> [31] "blueviolet"
#> [34] "brown2"
#> [37] "burlywood"
#> [40] "burlywood3"
#> [43] "cadetblue1"
#> [46] "cadetblue4"
#> [49] "chartreuse2"
#> [52] "chocolate"
#> [55] "chocolate3"
#> [58] "coral1"
#> [61] "coral4"
#> [64] "cornsilk1"
#> [67] "cornsilk4"
#> [70] "cyan2"
#> [73] "darkblue"
#> [76] "darkgoldenrod1"
#> [79] "darkgoldenrod4"
#> [82] "darkgrey"
#> [85] "darkolivegreen"
#> [88] "darkolivegreen3"
#> [91] "darkorange1"
#> [94] "darkorange4"
#> [97] "darkorchid2"
#> [100] "darkred"
#> [103] "darkseagreen1"
#> [106] "darkseagreen4"
#> [109] "darkslategray1"
#> [112] "darkslategray4"
#> [115] "darkviolet"
#> [118] "deeppink2"
#> [121] "deepskyblue"
#> [124] "deepskyblue3"
#> [1] "aliceblue"
#> [4] "antiquewhite2"
#> [7] "aquamarine"
#> [10] "aquamarine3"
#> [13] "azure1"
#> [16] "azure4"
#> [19] "bisque1"
#> [22] "bisque4"
#> [25] "blue"
#> [28] "blue3"
#> [31] "brown"
#> [34] "brown3"
#> [37] "burlywood1"
#> [40] "burlywood4"
#> [43] "cadetblue2"
#> [46] "chartreuse"
#> [49] "chartreuse3"
#> [52] "chocolate1"
#> [55] "chocolate4"
#> [58] "coral2"
#> [61] "cornflowerblue"
#> [64] "cornsilk2"
#> [67] "cyan"
#> [70] "cyan3"
#> [73] "darkcyan"
#> [76] "darkgoldenrod2"
#> [79] "darkgray"
#> [82] "darkkhaki"
#> [85] "darkolivegreen1"
#> [88] "darkolivegreen4"
#> [91] "darkorange2"
#> [94] "darkorchid"
#> [97] "darkorchid3"
#> [100] "darksalmon"
#> [103] "darkseagreen2"
#> [106] "darkslateblue"
#> [109] "darkslategray2"
#> [112] "darkslategray4"
#> [115] "deeppink"
#> [118] "deeppink3"
#> [121] "deepskyblue1"
#> [124] "deepskyblue4"
#> [1] "antiquewhite"
#> [4] "antiquewhite3"
#> [7] "aquamarine1"
#> [10] "aquamarine4"
#> [13] "azure2"
#> [16] "beige"
#> [19] "bisque2"
#> [22] "black"
#> [25] "blue1"
#> [28] "blue4"
#> [31] "brown1"
#> [34] "brown4"
#> [37] "burlywood2"
#> [40] "cadetblue"
#> [43] "cadetblue3"
#> [46] "chartreuse1"
#> [49] "chartreuse4"
#> [52] "chocolate2"
#> [55] "coral"
#> [58] "coral3"
#> [61] "cornsilk"
#> [64] "cornsilk3"
#> [67] "cyan1"
#> [70] "cyan4"
#> [73] "darkgoldenrod"
#> [76] "darkgoldenrod3"
#> [79] "darkgreen"
#> [82] "darkmagenta"
#> [85] "darkolivegreen2"
#> [88] "darkorange"
#> [91] "darkorange3"
#> [94] "darkorchid1"
#> [97] "darkorchid4"
#> [100] "darkseagreen"
#> [103] "darkseagreen3"
#> [106] "darkslategray"
#> [109] "darkslategray3"
#> [112] "darkturquoise"
#> [115] "deeppink1"
#> [118] "deeppink4"
#> [121] "deepskyblue2"
#> [124] "dimgray"

```

#> [127] "dimgray"	"dodgerblue"	"dodgerblue1"
#> [130] "dodgerblue2"	"dodgerblue3"	"dodgerblue4"
#> [133] "firebrick"	"firebrick1"	"firebrick2"
#> [136] "firebrick3"	"firebrick4"	"floralwhite"
#> [139] "forestgreen"	"gainsboro"	"ghostwhite"
#> [142] "gold"	"gold1"	"gold2"
#> [145] "gold3"	"gold4"	"goldenrod"
#> [148] "goldenrod1"	"goldenrod2"	"goldenrod3"
#> [151] "goldenrod4"	"gray"	"gray0"
#> [154] "gray1"	"gray2"	"gray3"
#> [157] "gray4"	"gray5"	"gray6"
#> [160] "gray7"	"gray8"	"gray9"
#> [163] "gray10"	"gray11"	"gray12"
#> [166] "gray13"	"gray14"	"gray15"
#> [169] "gray16"	"gray17"	"gray18"
#> [172] "gray19"	"gray20"	"gray21"
#> [175] "gray22"	"gray23"	"gray24"
#> [178] "gray25"	"gray26"	"gray27"
#> [181] "gray28"	"gray29"	"gray30"
#> [184] "gray31"	"gray32"	"gray33"
#> [187] "gray34"	"gray35"	"gray36"
#> [190] "gray37"	"gray38"	"gray39"
#> [193] "gray40"	"gray41"	"gray42"
#> [196] "gray43"	"gray44"	"gray45"
#> [199] "gray46"	"gray47"	"gray48"
#> [202] "gray49"	"gray50"	"gray51"
#> [205] "gray52"	"gray53"	"gray54"
#> [208] "gray55"	"gray56"	"gray57"
#> [211] "gray58"	"gray59"	"gray60"
#> [214] "gray61"	"gray62"	"gray63"
#> [217] "gray64"	"gray65"	"gray66"
#> [220] "gray67"	"gray68"	"gray69"
#> [223] "gray70"	"gray71"	"gray72"
#> [226] "gray73"	"gray74"	"gray75"
#> [229] "gray76"	"gray77"	"gray78"
#> [232] "gray79"	"gray80"	"gray81"
#> [235] "gray82"	"gray83"	"gray84"
#> [238] "gray85"	"gray86"	"gray87"
#> [241] "gray88"	"gray89"	"gray90"
#> [244] "gray91"	"gray92"	"gray93"
#> [247] "gray94"	"gray95"	"gray96"
#> [250] "gray97"	"gray98"	"gray99"
#> [253] "gray100"	"green"	"green1"
#> [256] "green2"	"green3"	"green4"
#> [259] "greenyellow"	"grey"	"grey0"

#> [262] "grey1"	"grey2"	"grey3"
#> [265] "grey4"	"grey5"	"grey6"
#> [268] "grey7"	"grey8"	"grey9"
#> [271] "grey10"	"grey11"	"grey12"
#> [274] "grey13"	"grey14"	"grey15"
#> [277] "grey16"	"grey17"	"grey18"
#> [280] "grey19"	"grey20"	"grey21"
#> [283] "grey22"	"grey23"	"grey24"
#> [286] "grey25"	"grey26"	"grey27"
#> [289] "grey28"	"grey29"	"grey30"
#> [292] "grey31"	"grey32"	"grey33"
#> [295] "grey34"	"grey35"	"grey36"
#> [298] "grey37"	"grey38"	"grey39"
#> [301] "grey40"	"grey41"	"grey42"
#> [304] "grey43"	"grey44"	"grey45"
#> [307] "grey46"	"grey47"	"grey48"
#> [310] "grey49"	"grey50"	"grey51"
#> [313] "grey52"	"grey53"	"grey54"
#> [316] "grey55"	"grey56"	"grey57"
#> [319] "grey58"	"grey59"	"grey60"
#> [322] "grey61"	"grey62"	"grey63"
#> [325] "grey64"	"grey65"	"grey66"
#> [328] "grey67"	"grey68"	"grey69"
#> [331] "grey70"	"grey71"	"grey72"
#> [334] "grey73"	"grey74"	"grey75"
#> [337] "grey76"	"grey77"	"grey78"
#> [340] "grey79"	"grey80"	"grey81"
#> [343] "grey82"	"grey83"	"grey84"
#> [346] "grey85"	"grey86"	"grey87"
#> [349] "grey88"	"grey89"	"grey90"
#> [352] "grey91"	"grey92"	"grey93"
#> [355] "grey94"	"grey95"	"grey96"
#> [358] "grey97"	"grey98"	"grey99"
#> [361] "grey100"	"honeydew"	"honeydew1"
#> [364] "honeydew2"	"honeydew3"	"honeydew4"
#> [367] "hotpink"	"hotpink1"	"hotpink2"
#> [370] "hotpink3"	"hotpink4"	"indianred"
#> [373] "indianred1"	"indianred2"	"indianred3"
#> [376] "indianred4"	"ivory"	"ivory1"
#> [379] "ivory2"	"ivory3"	"ivory4"
#> [382] "khaki"	"khaki1"	"khaki2"
#> [385] "khaki3"	"khaki4"	"lavender"
#> [388] "lavenderblush"	"lavenderblush1"	"lavenderblush2"
#> [391] "lavenderblush3"	"lavenderblush4"	"lawngreen"
#> [394] "lemonchiffon"	"lemonchiffon1"	"lemonchiffon2"

```

#> [397] "lemonchiffon3"           "lemonchiffon4"           "lightblue"
#> [400] "lightblue1"              "lightblue2"              "lightblue3"
#> [403] "lightblue4"              "lightcoral"              "lightcyan"
#> [406] "lightcyan1"              "lightcyan2"              "lightcyan3"
#> [409] "lightcyan4"              "lightgoldenrod"          "lightgoldenrod1"
#> [412] "lightgoldenrod2"          "lightgoldenrod3"          "lightgoldenrod4"
#> [415] "lightgoldenrodyellow"    "lightgray"               "lightgreen"
#> [418] "lightgrey"               "lightpink"               "lightpink1"
#> [421] "lightpink2"              "lightpink3"               "lightpink4"
#> [424] "lightsalmon"             "lightsalmon1"            "lightsalmon2"
#> [427] "lightsalmon3"             "lightsalmon4"            "lightseagreen"
#> [430] "lightskyblue"            "lightskyblue1"            "lightskyblue2"
#> [433] "lightskyblue3"            "lightskyblue4"            "lightslateblue"
#> [436] "lightslategray"          "lightslategrey"          "lightsteelblue"
#> [439] "lightsteelblue1"          "lightsteelblue2"          "lightsteelblue3"
#> [442] "lightsteelblue4"          "lightyellow"              "lightyellow1"
#> [445] "lightyellow2"             "lightyellow3"              "lightyellow4"
#> [448] "limegreen"               "linen"                  "magenta"
#> [451] "magenta1"                "magenta2"                "magenta3"
#> [454] "magenta4"                "maroon"                 "maroon1"
#> [457] "maroon2"                 "maroon3"                 "maroon4"
#> [460] "mediumaquamarine"        "mediumblue"              "mediumorchid"
#> [463] "mediumorchid1"            "mediumorchid2"            "mediumorchid3"
#> [466] "mediumorchid4"            "mediumpurple"             "mediumpurple1"
#> [469] "mediumpurple2"            "mediumpurple3"            "mediumpurple4"
#> [472] "mediumseagreen"           "mediumslateblue"          "mediumspringgreen"
#> [475] "mediumturquoise"          "mediumvioletred"          "midnightblue"
#> [478] "mintcream"                "mistyrose"                "mistyrose1"
#> [481] "mistyrose2"                "mistyrose3"                "mistyrose4"
#> [484] "moccasin"                 "navajowhite"              "navajowhite1"
#> [487] "navajowhite2"              "navajowhite3"              "navajowhite4"
#> [490] "navy"                     "navyblue"                 "oldlace"
#> [493] "olivedrab"                "olivedrab1"                "olivedrab2"
#> [496] "olivedrab3"                "olivedrab4"                "orange"
#> [499] "orange1"                  "orange2"                  "orange3"
#> [502] "orange4"                  "orangered"                "orangered1"
#> [505] "orangered2"                "orangered3"                "orangered4"
#> [508] "orchid"                   "orchid1"                  "orchid2"
#> [511] "orchid3"                  "orchid4"                  "palegoldenrod"
#> [514] "palegreen"                 "palegreen1"                "palegreen2"
#> [517] "palegreen3"                 "palegreen4"                "paleturquoise"
#> [520] "paleturquoise1"            "paleturquoise2"            "paleturquoise3"
#> [523] "paleturquoise4"            "palevioletred"             "palevioletred1"
#> [526] "palevioletred2"            "palevioletred3"            "palevioletred4"
#> [529] "papayawhip"                "peachpuff"                 "peachpuff1"

```

#> [532] "peachpuff2"	"peachpuff3"	"peachpuff4"
#> [535] "peru"	"pink"	"pink1"
#> [538] "pink2"	"pink3"	"pink4"
#> [541] "plum"	"plum1"	"plum2"
#> [544] "plum3"	"plum4"	"powderblue"
#> [547] "purple"	"purple1"	"purple2"
#> [550] "purple3"	"purple4"	"red"
#> [553] "red1"	"red2"	"red3"
#> [556] "red4"	"rosybrown"	"rosybrown1"
#> [559] "rosybrown2"	"rosybrown3"	"rosybrown4"
#> [562] "royalblue"	"royalblue1"	"royalblue2"
#> [565] "royalblue3"	"royalblue4"	"saddlebrown"
#> [568] "salmon"	"salmon1"	"salmon2"
#> [571] "salmon3"	"salmon4"	"sandybrown"
#> [574] "seagreen"	"seagreen1"	"seagreen2"
#> [577] "seagreen3"	"seagreen4"	"seashell"
#> [580] "seashell1"	"seashell2"	"seashell3"
#> [583] "seashell4"	"sienna"	"sienna1"
#> [586] "sienna2"	"sienna3"	"sienna4"
#> [589] "skyblue"	"skyblue1"	"skyblue2"
#> [592] "skyblue3"	"skyblue4"	"slateblue"
#> [595] "slateblue1"	"slateblue2"	"slateblue3"
#> [598] "slateblue4"	"slategray"	"slategray1"
#> [601] "slategray2"	"slategray3"	"slategray4"
#> [604] "slategrey"	"snow"	"snow1"
#> [607] "snow2"	"snow3"	"snow4"
#> [610] "springgreen"	"springgreen1"	"springgreen2"
#> [613] "springgreen3"	"springgreen4"	"steelblue"
#> [616] "steelblue1"	"steelblue2"	"steelblue3"
#> [619] "steelblue4"	"tan"	"tan1"
#> [622] "tan2"	"tan3"	"tan4"
#> [625] "thistle"	"thistle1"	"thistle2"
#> [628] "thistle3"	"thistle4"	"tomato"
#> [631] "tomato1"	"tomato2"	"tomato3"
#> [634] "tomato4"	"turquoise"	"turquoise1"
#> [637] "turquoise2"	"turquoise3"	"turquoise4"
#> [640] "violet"	"violetred"	"violetred1"
#> [643] "violetred2"	"violetred3"	"violetred4"
#> [646] "wheat"	"wheat1"	"wheat2"
#> [649] "wheat3"	"wheat4"	"whitesmoke"
#> [652] "yellow"	"yellow1"	"yellow2"
#> [655] "yellow3"	"yellow4"	"yellowgreen"

C függelék

Óravázlat az R tanításához

Az R oktatásához és tanulásához egy lehetséges tanmenet vázlatát mutatjuk be. Az R tanulmányozását két féléves rendszerben, félévenként 10 duplaórában képzeltük el. A lent felsorolt óravázlat összesen 20 duplaórában sorolja fel azokat az ismeretelemeket, amelyek az R gyakorlati felhasználásához nem nélkülözhetők.

1. félév

1. óra

Az *Alap R* és az *RStudio* letöltése és telepítése, dokumentációk az R használatához. Az *RGui* áttekintése, a konzol használata, a szkriptablak használata, parancsok végrehajtása. Az *RStudio* használata.

2. óra

Számok írása, numerikus operátorok. Karakteres konstansok az R-ben. Logikai konstansok és a logikai műveletek. Objektum létrehozása, értékének megváltoztatása. Objektumokat is tartalmazó kifejezések.

3. óra

Függvények hívása, például a `log()` és `exp()` függvények bemutatása, egyéb matematikai függvények. Függvényparaméterek elnevezése, sorrendjük cseréje. Kifejezés definíciója. Vektor adatszerkezet definíciója. Vektor létrehozása: `c(), : (kettőspont operátor)`, `seq()`, `rep()`

4. óra

A faktor adatszerkezet definíciója, faktor létrehozása a `factor()` függvénnnyel. Objektumok adattípusának megtekintése és az objektum hossza: `class()`, `typeof()`, `length()`. Adattábla definíciója, létrehozása a `data.frame()` függvénnnyel. Mátrix adatszerkezet, létrehozása a `matrix()` függvénnnyel. Tömb adatszerkezet, létrehozása az `array()` függvénnnyel. Lista adatszerkezet, létrehozása a `list()` függvénnnyel. Indexelés vektor, mátrix, tömb és adattábla esetén.

Hivatkozás listaelemekre lista és adattábla esetén.

5. óra

Tagolt szöveges állomány létrehozása táblázatkezelővel és szövegszerkesztővel. Tagolt szöveges állomány beolvasása a `read.table()` függvénytel. A paraméterek megbeszélése (`file=`, `sep=`, `header=`, `dec=`, `quote=`, `stringsAsFactors=`, `comment.char=`, `na.strings=`, `strip.white=`, `fileEncode=`). Táblázatkezelők és más statisztikai programcsomagok saját formátomú állományainak beolvasása a `rio::import()` segítségével. A beolvasás helyességének ellenőrzése az `str()` függvénytel. A `read.table()` argumentumainak hatása az `str()` outputjára. Adatállomány kiírása a `write.table()` és a `rio::export()` segítségével.

6. óra

Egyszerű típuskonverziók (numerikus vektorból faktor és karakteres vektorból faktor előállítása a `factor()` függvénytel, valamint az `as.character()` és `as.numeric()` használata). Kapcsolat a statisztikai skálák és a változók típusai között. Faktor szintjeinek átnevezése, a szintek sorrendjének meghatározása. Vektor rendezése a `sort()` és `order()` függvénytel. Adattábla rendezése az `order()` függvénytel. Adattábla szűrése egyszerű logikai kifejezéssel és összetett logikai kifejezéssel.

7. óra

Az R környezet fogalmai: munkakönyvtár, munkaterület, csomag. Munkakönyvtár kezelése Objektum létrehozása és eltávolítása a munkaterületről, objektumok listája. Csomag telepítése, telepített csomagok listája, csomag betöltése, csomag eltávolítása, betöltött csomagok listája. Az R Commander telepítése és használata, bemutatása. A jamovi telepítése és használata. Az R GUI, RStudio, jamovi és az R Commander összehasonlítása.

8. óra

Leíró statisztikai mérőszámok meghatározása: `summary()`, `mean()`, `median()`, `sd()` stb. Hiányzó értékek kezelése (`na.rm=T`, `is.na()`, `na.omit()`). Műveletek több változóra és csoportra (`apply()`, `sapply()`, `tapply()`, `aggregate()`, `by()`). A `psych::describe()`, `psych::describeBy()` és a `DescTools::Desc()` függvényének bemutatása.

9. óra

Gyakorisági táblázatok létrehozása a `table()` és `xtabs()` függvénytel. Hiányzó értékek kezelése (`useNA="ifany"`). Relatív gyakorisági táblázatok. Kumulált gyakorisági táblázatok. Soronként vagy oszloponként vett kétdimenziós gyakorisági táblázatok. Három vagy többdimenziós táblázatok (`array()` és `ftable()`). Gyakorisági táblázatok a `DescTools::Desc()` függvényel.

10. óra

A hagyományos grafika magasszintű függvényei. Oszlopdiagram rajzolása (`barplot()`). Egydimenziós pontdiagram létrehozása (`stripchart()`). Graffkus paraméterek fogalma és beállítása a `par()` függvénytel. Feliratok (`main=`, `sub=`, `xlab=`, `ylab=`, `las=`), tengelyek (`xlim=`, `ylim=`, `mgp=`, `tcl=`), pontkarakterek (`pch=`), margók (`mar=`), színek (`col=`) beállítása. Pontdiagram

és vonaldiagram rajzolása (`plot()`). Dobozdiagram rajzolása (`boxplot()`). Hisztogram rajzolása (`hist()`). További grafikus paraméterek: rajzterület felosztása (`mfrow=`, `mfcol=`). Grafikus eszközök típusai (`windows()`, `png()`, `jpeg()`, `pdf()`). Képallokányok létrehozása és beállítási lehetőségek (`res=`, `width=`, `height=`). Grafikus eszközök bezárása.

2. félév

11. óra

A reprodukálható kutatás elvének bemutatása. Az RStudio és a jamovi lehetőségei a reprodukálható kutatásban. RMarkdown állomány (`*.Rmd`) létrehozása, szintaxisának bemutatása. A fejléc lehetőségei. A szöveges rész formázása Markdown segítségével. Az R csonkok lehetőségei. PDF, Word és HTML állomány generálása. Egyedi template-ek használata.

12. óra

Információk az adatobjektumokról (`str()`, `head()`, `dim()`, `nrow()`, `ncol()`, `names()`, `colnames()`, `rownames()`, `levels()`, `nlevels()`). Egyszerű típuskonverzió a karakters, numerikus, logikai és faktor típusok között (`factor()`, `as.factor()`, `as.numeric()`, `as.character()`, `as.logical()`). Mátrix és adattábla sor- és oszlopmanipulációja: sor és oszlopnevek megadása, átnevezése, sorok és oszlopok törlése, beszúrása és cseréje (pl. `rbind()`, `cbind()`). Numerikusból numerikus transzformáció (tetszőleges matematikai függvény, `transform()`, `recode()`). Numerikusból faktor transzformáció (`cut()`, `car::recode()`). Faktorból faktor transzformáció (`car::recode()`).

13. óra

Adatelőkészítés egy- és kétmintás próbákhoz. Adatbázis létrehozás a `data.frame()` függvényel a varianciaelemzés számára (kiegyensúlyozott és nem kiegyensúlyozott esetek). A hosszú és széles adatbázisok fogalma. Adatbázis átalakítása a két formátum között (`reshape()`, `melt()`, `dcast()`). Adattáblák összefűzése (`merge()`).

14. óra

Egymintás és kétmintás próbák vérehajtása Argumentumok és outputok értelmezése. Például a `BSDA::z.test()`, `t.test()`, `prop.test()`, `binom.test()`, `fisher.test()`, `mcnemar.test()`, `wilcox.test()`, `kruskal.test()` és `friedman.test()` függvényekkel. A varianciaelemzés különböző változatainak vérehajtása. Argumentumok és outputok értelmezése az `aov()` és `lm()` függvények esetében. Korreláció- és regressziószámítás vérehajtása. Argumentumok és outputok értelmezése a `cor()`, `cor.test()` és `lm()` esetében.

15. óra

Véletlen a `sample()` függvény segítségével. Eloszlások függvényei (`r...()`, `d...()`, `q...()`, `p...()`). Kritikus értékek és p-értékek meghatározása a `q...()`, `p...()` függvényekkel. Eloszlások sűrűségfüggvényeinek rajzolása a `curve()` függvény segítségével.

16. óra

A Tidyverse R bemutatása. A pipe operátor (`%>%`) használata. A tibble adatobjektumok kezelése. Adatállományok beolvasás és kiírása. Adatobjektom vizsgálata, sor- és oszlopnevek manipulációja. Oszlopok leválogatása, beszúrása és törlése. Adatobjektum rendezése.

17. óra

A Tidyverse R bemutatása. Sorok szűrése, törlése és beszúrása. Adatok összesítése, csoportosítása és transzformációja, valamint adattáblák összefűzése a **dplyr** csomag segítségével. Kategorikus változók kezelése a **forcats** csomag segítségével.

18. óra

A Tidyverse R bemutatása. Széles-hosszú átalakítás a **tidyr** csomag segítségével (`pivot_wider()`, `pivot_longer()`). Hiányzó, duplikált és kiugró értékek kezelése a **dplyr** csomag segítségével. Karakterláncok kezelése a **stringr** csomag segítségével.

19. óra

A ggplot2 grafikus rendszer elemei: az adat, az alakzat (geom) és a megjelenés (aes) rétegek. Az `x=`, `y=`, `alpha=`, `colour=`, `fill=`, `group=`, `shape=` és `size=` paraméterek használata az `aes()` függvényben. Jelmagyarázat megjelenítése. Több ábra megjelenítése kategorikus változó alapján (faceting). Ábra elemeinek beállítása: feliratok, tengelyek, színek és téma. Ábrák mentése.

20. óra

Az alavető ábratípusok megjelenítése és paramétereit: oszlopdiagram, egydimenziós pontdiagram, hisztogram, dobozdiagram, pontdiagram, vonaldiagram, Q-Q ábra, hegedűdiagram, lollipop ábra és terület diagram.

20+1. óra

Az R programozási lehetőségei: szekvencia, feltételes utasítások, ciklusok. Függvények létrehozása. Objektum-orientált lehetőségek.

Irodalomjegyzék

Csapó, G., Csénoch, M., and Abari, K. (2020). Sprego: case study on the effectiveness of teaching spreadsheet management with schema construction. *Education and Information Technologies*, 25(3):1585–1605.