Adatok előkészítése
Data Preparation (3)

Lukovszki Csaba

2021. március

Tartalomjegyzék

1. Adatok betöltése

```
rm(list = ls())
data <- read.csv("data/finreport.csv", stringsAsFactors = T)
```

2. Adatok megismerése

Az adathalmazunk USA-beli cégek pénzügyi riportjait tartalmazza. Minde egyes riport a következő adatokat tartalmazza:

- `ID`: Riport index
- `Name`: Cég neve

- `Industry`: Iparág
- `Inception`: Év
- `Employees`: Alkalmazottak száma
- `State`: Állam
- `City`: Város
- 'Revenue": Bevétel
- `Expenses`: Költségek
- `Profit`: Profit, eredmény
- `Growth`: Növekedés százalékban megadva

```
str(data)
```

```
## 'data.frame':    500 obs. of  11 variables:
##  $ ID       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Name     : Factor w/ 500 levels "Abstractedchocolat",..: 297 451 168 40 485 199 435 33
##  $ Industry : Factor w/ 8 levels "","Construction",..: 8 6 7 6 8 6 3 2 6 3 ...
##  $ Inception: int  2006 2009 2012 2011 2013 2013 2009 2013 2009 2010 ...
##  $ Employees: int  25 36 NA 66 45 60 116 73 55 25 ...
##  $ State    : Factor w/ 43 levels "","AL","AZ","CA",..: 37 34 36 4 42 28 23 30 4 9 ...
##  $ City     : Factor w/ 297 levels "Addison","Alexandria",..: 94 181 105 195 151 154 53 2
##  $ Revenue  : Factor w/ 499 levels "","$1,614,585",..: 480 195 486 247 403 142 309 1 97 1
##  $ Expenses : Factor w/ 498 levels "","1,026,548 Dollars",..: 7 486 4 249 228 248 58 1 40
##  $ Profit   : int  8553827 13212508 8701897 10727561 4193069 8179177 3259485 NA 5274553 1
##  $ Growth   : Factor w/ 33 levels "","-2%","-3%",..: 15 17 12 15 15 19 13 1 27 17 ...
```

```
#summary(data)
```

A betöltés során a karakter típusú értékek automatikusan faktorokká lettek konvertálva, amit a beolvasás során a `stringsAsFactors = T` paraméter értékével határoztunk meg. Ennek eredményeképpen a következő attribútumokat vissza kell alakítanunk:

- Bevétel (`Revenue`)
- Költségek (`Expenses`)
- Növekedési faktor (`Growth`)

Ennek következében jó stratégia, ha az adatok beolvasása során mindig a `stringsAsFactors = F` paramétert választjuk, és a faktorokká alakítást az adott feladatra optimalizáltan, attribútumonként végezzük el.

2.1. Faktorok visszaalakítása

Amennyiben a faktorokat vissza szeretnénk konvertálni karakter formátumúvá, figyelmesen kell eljárnunk. Lássuk ugyanis a következő példát:

```
a <- c(5, 6, 7, 8, 5, 6, 7)
# alakítsuk faktorrá
b <- as.factor(a)
b
```

```
## [1] 5 6 7 8 5 6 7
## Levels: 5 6 7 8
```

```
# alakítsuk vissza numerikus értékekké
as.numeric(b)
```

```
## [1] 1 2 3 4 1 2 3
```

A numerikus értékekre való visszaalakítás során a konverzió nem a faktor értékei, hanem a faktor indexei alapján történt, ezért az értékük elveszett! Erre azért van szükség, hogy legyen mód az indexek kinyerésére.

Az értékek szerinti konvertálásra megoldást az nyújt, ha a faktorokat először karakter értékekké konvertáljuk, mivel ez esetben a konverzió a faktor egyes értékei és az indexei alapján történik.

```
# először karakter típussá konvertáljuk
as.numeric(as.character(b))
```

```
## [1] 5 6 7 8 5 6 7
```

Most alakítsuk vissza a bevétel `Revenue`, költség `Expenses` attribútumokat sztringgé.

```
data$Revenue <- as.character(data$Revenue)
data$Expenses <- as.character(data$Expenses)
str(data)
```

```
## 'data.frame':    500 obs. of  11 variables:
##  $ ID       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Name     : Factor w/ 500 levels "Abstractedchocolat",..: 297 451 168 40 485 199 435 33
##  $ Industry : Factor w/ 8 levels "","Construction",..: 8 6 7 6 8 6 3 2 6 3 ...
##  $ Inception: int  2006 2009 2012 2011 2013 2013 2009 2013 2009 2010 ...
##  $ Employees: int  25 36 NA 66 45 60 116 73 55 25 ...
##  $ State    : Factor w/ 43 levels "","AL","AZ","CA",..: 37 34 36 4 42 28 23 30 4 9 ...
##  $ City     : Factor w/ 297 levels "Addison","Alexandria",..: 94 181 105 195 151 154 53 2
##  $ Revenue  : chr  "$9,684,527" "$14,016,543" "$9,746,272" "$15,359,369" ...
##  $ Expenses : chr  "1,130,700 Dollars" "804,035 Dollars" "1,044,375 Dollars" "4,631,808 D
##  $ Profit   : int  8553827 13212508 8701897 10727561 4193069 8179177 3259485 NA 5274553 1
##  $ Growth   : Factor w/ 33 levels "","-2%","-3%",..: 15 17 12 15 15 19 13 1 27 17 ...
```

---

1. Hozzon létre egy karakter vektort, mely kétjegyű számokat tartalmaz. Faktorizálja a vektort, majd a faktor vektort alakítsa át numerikus vektorrá!
2. A vizsgált adathalmaz `Growth` attribútumát alakítsa vissza karakter típusúvá.

2.2. Adatok konvertlása

Az adathalmazunkban a bevétel (`Revenue`), költségek (`Expenses`) és a növekedési faktor (`Growth`) is numerikus adatokat tartalmaz, viszont egyedi formátuma miatt karakterként került tárolásra.

A következőkben a fenti mezőket numerikus értékekké konvertáljuk. Ennek első lépése a konvertálást megnehezítő karakterek elhagyása, melyet a `sub()` és `gsub()` függvények segítségével érhetünk el.

A `sub(pattern, replacement, x )` használata esetében a `pattern` karaktereket helyettesíti a függvény a `replacement` értékkel az `x` vektoron. A `pattern` értéknek reguláris kifejezés is megadható, ezért a reguláris kifejezésben is használt karaktereket `\\` előtaggal kell megadni.

A következő példában alakítsuk numerikus értékké a bevételeket tartalmazó attribútumokat.

```
# 1. $ karakter törlése
# a $ egy speciális karakter, ezért \\$ formában kell hivatkozni
data$Revenue <- gsub("\\$", "", data$Revenue)
```

```r
# 2. a ',' karakter törlése
data$Revenue <- gsub(",", "", data$Revenue)
# 3. a karakter numerikus értékké alakítása
data$Revenue <- as.numeric(data$Revenue)
str(data)
```

```
## 'data.frame':    500 obs. of  11 variables:
##  $ ID       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Name     : Factor w/ 500 levels "Abstractedchocolat",..: 297 451 168 40 485 199 435 33
##  $ Industry : Factor w/ 8 levels "","Construction",..: 8 6 7 6 8 6 3 2 6 3 ...
##  $ Inception: int  2006 2009 2012 2011 2013 2013 2009 2013 2009 2010 ...
##  $ Employees: int  25 36 NA 66 45 60 116 73 55 25 ...
##  $ State    : Factor w/ 43 levels "","AL","AZ","CA",..: 37 34 36 4 42 28 23 30 4 9 ...
##  $ City     : Factor w/ 297 levels "Addison","Alexandria",..: 94 181 105 195 151 154 53 2
##  $ Revenue  : num  9684527 14016543 9746272 15359369 8567910 ...
##  $ Expenses : chr  "1,130,700 Dollars" "804,035 Dollars" "1,044,375 Dollars" "4,631,808 D
##  $ Profit   : int  8553827 13212508 8701897 10727561 4193069 8179177 3259485 NA 5274553 1
##  $ Growth   : Factor w/ 33 levels "","-2%","-3%",..: 15 17 12 15 15 19 13 1 27 17 ...
```

---

1. A vizsgált adathalmazon alakítsuk numerikus értékekké a költségek (`Expenses`) és növekedés (`Growth`) attribútumokat!

---

3. Hiányzó értékek

A vizsgált adathalmazunk tartalmaz hiányzó adatokat. Ezek egyrészt, mint üres karakter változók, másrészt, mint `NA` értékek jelennek meg az adathalmazban. A hiányzó adatok pótlására a következő stratégiákat használhatjuk:

1. Egyértelmű adatok megadása: Ezt a stratégiát használjuk, amennyiben 100/ biztonsággal meghatározhatjuk a hiányzó értékeket. Például a következő esetekben
   - `State`: A városnevek ismeretében az államok nevei egyértelműen meghatározhatók.
   - `Industry`: Más forrásokból az adott cég esetében az iparág egyértelműen megkereshető.
   - `Revenue`, `Expenses`, `Profit`: Egy érték hiányzása esetében a $Revenue - Expenses = Profit$ összefüggésből egyértelműen számítható a hiányzó érték.
2. Adatok helyettesítése: Egyes esetkben az adathalmaz hasonló csoportjára jellemző értékekkel a becsülhetők.
   a. Átlaggal
   b. Mediánnal: A medián kevésbé érzékeny kiugró értékekre
   c. Adatmodell alapján: Korrelációk és hasonlóságok alapján. Például az ismeretlen adat függését vizsgálva más értékektől a hiányzó értéket predikálhatjuk.

   - `Employees`: Iparági középértékkel való helyettesítés
   - `Revenue`, `Expenses`, `Profit`: Amennyiben több mint egy hiányzik ezen értékek közül az iparági átlaggal helyettesíthető, hogy a maradék egy értéket már számítani lehessen.
3. Adatok törlése: Amennyiben az adattal nem tudunk mit kezdeni és szükséges az adat, a

hiányzó érték törlendő. Ebben az esetben viszont figyelembe kell venni, hogy az adathalmazunk kevésbé lesz szignifikáns.

- `Industry`: Amennyiben az adat szükséges az adatok feldolgozásához, de nem találjuk a helyes értéket, az adott objektum törölhető.

4. Dummy értékek bevezetése: Olyan egyértelmű, adatok hiányára utaló adat bevezetése, melyet később, a modellalkotás során kezelünk egyedileg.
5. Változtatás nélkül: Az adatokat változtatás nélkül is hagyhatjuk.

- `Inspection`: Amennyiben az adat nem szükséges az adatok feldolgozásához, az adatokat változtatás nélkül hagyhatjuk.

3.1. Értékek lokalizációja

Az adathalmazunkban a hiányzó adatokat az üres karakter és az `NA` értékek is reprezentálják. Az `NA` értékeket és az üres karakter értékeket másként kell szűrnünk. Lehet NA, vagy üres mező

```
# NA értékek keresése
is.na(data[3,])
```

```
##      ID  Name Industry Inception Employees State  City Revenue Expenses Profit
## 3 FALSE FALSE    FALSE     FALSE      TRUE FALSE FALSE   FALSE    FALSE  FALSE
##   Growth
## 3  FALSE
```

```
# Az is.na() nem találja az üres karaktereket
is.na(data[14,])
```

```
##       ID  Name Industry Inception Employees State  City Revenue Expenses Profit
## 14 FALSE FALSE    FALSE     FALSE     FALSE FALSE FALSE   FALSE    FALSE  FALSE
##    Growth
## 14  FALSE
```

```
# Az üres karakter értékek keresése
data[14,] == ""
```

```
##       ID  Name Industry Inception Employees State  City Revenue Expenses Profit
## 14 FALSE FALSE     TRUE     FALSE     FALSE FALSE FALSE   FALSE    FALSE  FALSE
##    Growth
## 14  FALSE
```

```
# Az összes üres értéket tartalmazó sor
which(!apply(!apply(data, 1, `==`, ""), 2, prod))
```

```
## [1]  11  14  15  84 267 379
```

Az egységes kezelés érdekében érdemes az üres karakter értékeket is `NA` típusúvá alakítani. Ez a következőképpen tehetjük meg.

```
data[data == ""] <- NA
```

Keressük meg az összes `NA` értéket tartalmazó sor számát!

```
# NA értékeket tartalmazó sorok
which(!apply(!is.na(data), 1, prod))
```

```
##  [1]   3   8  11  14  15  17  22  44  84 267 332 379
```

```
which(complete.cases(data) == F)
```

```
##  [1]   3   8  11  14  15  17  22  44  84 267 332 379
```

A következőképpen listázhatjuk a hiányzó értékeket tartalmazó objektumokat.

```
data[!complete.cases(data),]
```

```
##      ID            Name           Industry Inception Employees State
## 3     3        Greenfax             Retail      2012        NA    SC
## 8     8       Rednimdox       Construction      2013        73    NY
## 11   11 Canecorporation             Health      2012         6  <NA>
## 14   14        Techline               <NA>      2006        65    CA
## 15   15         Cityace               <NA>      2010        25    CO
## 17   17         Ganzlax         IT Services      2011        75    NJ
## 22   22      Lathotline             Health        NA       103    VA
## 44   44       Ganzgreen       Construction      2010       224    TN
## 84   84      Drilldrill           Software      2010        30  <NA>
## 267 267      Circlechop           Software      2010        14  <NA>
## 332 332     Westminster Financial Services      2010        NA    MI
## 379 379       Stovepuck             Retail      2013        73  <NA>
##              City  Revenue Expenses   Profit Growth
## 3      Greenville  9746272  1044375  8701897     16
## 8        Woodside       NA       NA       NA     NA
## 11       New York 10597009  7591189  3005820      7
## 14      San Ramon 13898119  5470303  8427816     23
## 15     Louisville  9254614  6249498  3005116      6
## 17         Iselin 14001180       NA 11901180     18
## 22         McLean  9418303  7567233  1851070      2
## 44       Franklin       NA       NA       NA      9
## 84  San Francisco  7800620  2785799  5014821     17
## 267 San Francisco  9067070  5929828  3137242     20
## 332          Troy 11861652  5245126  6616526     15
## 379      New York 13814975  5904502  7910473     10
```

3.2. Adathalmaz szűrése

Az adathalmazunkat tetszőleges érték alapján szűrhetjük. A következőkben készítsünk egy szűrést a `Revenue` értékre.

```
data[data$Revenue == 9746272, ]
```

```
##      ID     Name Industry Inception Employees State       City Revenue Expenses
## 3     3 Greenfax   Retail      2012        NA    SC Greenville 9746272  1044375
## NA   NA     <NA>     <NA>        NA        NA  <NA>       <NA>      NA       NA
## NA.1 NA     <NA>     <NA>        NA        NA  <NA>       <NA>      NA       NA
##       Profit Growth
## 3    8701897     16
## NA        NA     NA
## NA.1      NA     NA
```

Az eredményben láthatunk két olyan sort is, melyet nem várunk. Ennek megértéséhez nézzük át a következő relációs műveleteket.

```
NA == NA
```

```
## [1] NA
```

```
23 == NA
```

```
## [1] NA
```

```
TRUE == NA
```

```
## [1] NA
```

Látható, hogy az `NA` érték az értékek hiányára utal, ezért reláció esetében is ezt kapjuk eredményül. A szűrésben használt reláció tartalmaz `FALSE`, `TRUE` és `NA` értékeket. A szűrés visszaadja azokat az értékeket, ahol az érték nem `FALSE`. A szűrés során az `NA` értékek is megjelenítésre kerülnek.

```
#data$Revenue == 9746272
```

Amennyiben csak a `TRUE` értékek által hivatkozott sorokat szeretnénk visszanyerni, használjuk a `which()` függvényt az indexek meghatározására.

```
data[which(data$Revenue == 9746272), ]
```

```
##   ID     Name Industry Inception Employees State       City Revenue Expenses
## 3  3 Greenfax   Retail      2012        NA    SC Greenville 9746272  1044375
##    Profit Growth
## 3 8701897     16
```

Amennyiben `NA` értékekre szeretnénk szűrni, használjuk az `is.na()` függvényt a szűrés során. A következőkben keressük meg azokat az objektumokat, melyekben az iparág értéke `NA`.

```
data[is.na(data$Industry),]
```

```
##    ID     Name Industry Inception Employees State       City  Revenue Expenses
## 14 14 Techline     <NA>      2006        65    CA  San Ramon 13898119  5470303
## 15 15  Cityace     <NA>      2010        25    CO Louisville  9254614  6249498
##     Profit Growth
## 14 8427816     23
## 15 3005116      6
```

4. Adatok mentése

Az adatok manipulációja esetében a legelső és legfontosabb dolgunk, hogy az eredeti adatunkat bármikor elérhetővé tegyük. Ezt egy egyszerű `backup` adathalmaz létrehozásával érhetjük el.

```
# adatok mentése
data_backup <- data
# adatok visszatöltése
data <- data_backup
```

5. Az összes NA adat törlése

Amennyiben adatokat törlünk az adathalmazunkból, azt a legegyszerűbben az `na.omit()` függvénnyel érhetjük el. A függvény kiszűri azokat az eseteket, objektumokat, melyek tartalmaznak `NA` értéket bármelyik mezőjükben is. A következőkben erre láthatunk példát.

```
# az összes NA-t tartalmazó sor törlése
data <- na.omit(data)
# a nem teljes objektumok listázása
# látható, hogy nincs ilyen adat
data[!complete.cases(data),]
```

```
##  [1] ID        Name      Industry  Inception Employees State     City
##  [8] Revenue   Expenses  Profit    Growth
## <0 rows> (or 0-length row.names)
```

```
# töltsük vissza az eredeti adathalmazt
data <- data_backup
```

6. Egyértelmű adatok megadása

Egyértelműen meadhatók azok az adatok, melyek meglévő adatok segítségével pótolhatók, mint például az állam (`State`) és a költségek (`Expenses`), amennyiben a bevétel és profit meg van adva.

A következőkben pótoljuk a hiányzó államok nevét. `New York` város esetében állítsuk az államot `NY`-ra.

```
# Adatok listázása, ahol a State nincs kitöltve
data[is.na(data$Stat),]
```

```
##      ID            Name Industry Inception Employees State          City
## 11   11 Canecorporation   Health      2012         6  <NA>      New York
## 84   84      Drilldrill Software      2010        30  <NA> San Francisco
## 267 267      Circlechop Software      2010        14  <NA> San Francisco
## 379 379       Stovepuck   Retail      2013        73  <NA>      New York
##      Revenue Expenses  Profit Growth
## 11  10597009  7591189 3005820      7
## 84   7800620  2785799 5014821     17
## 267  9067070  5929828 3137242     20
## 379 13814975  5904502 7910473     10
```

```
# New York <-- NY
data[is.na(data$Stat) & data$City == "New York", "State"] <- "NY"
```

Kutatások során is megállapíthatjuk a hiányzó értékeket. Némi keresés után kiderül, hogy a Techline San Ramon-ban fémiparban érdekelt, ezért a `Construction` iparágba, a Cityace Louisville-ben az egészségiparral foglalkozik, ezért a `Health` iparágba sorolhatjuk.

```
data[data$Name == "Techline",]$Industry <- "Construction"
data[!complete.cases(data),]
```

```
##      ID        Name           Industry Inception Employees State          City
## 3     3    Greenfax             Retail      2012        NA    SC    Greenville
## 8     8   Rednimdox       Construction      2013        73    NY      Woodside
## 15   15     Cityace              <NA>      2010        25    CO    Louisville
## 17   17     Ganzlax        IT Services      2011        75    NJ        Iselin
## 22   22  Lathotline             Health        NA       103    VA        McLean
## 44   44   Ganzgreen       Construction      2010       224    TN      Franklin
## 84   84  Drilldrill           Software      2010        30  <NA> San Francisco
```

```
## 267 267  Circlechop           Software      2010        14  <NA> San Francisco
## 332 332 Westminster Financial Services      2010        NA    MI          Troy
##      Revenue Expenses   Profit Growth
## 3    9746272  1044375  8701897     16
## 8         NA       NA       NA     NA
## 15   9254614  6249498  3005116      6
## 17  14001180       NA 11901180     18
## 22   9418303  7567233  1851070      2
## 44        NA       NA       NA      9
## 84   7800620  2785799  5014821     17
## 267  9067070  5929828  3137242     20
## 332 11861652  5245126  6616526     15
```

---

1. A vizsgált adathalmazban a hiányzó államot `San Francisco` esetében állítsuk be `CA` (California) értékűre.
2. Abban az esetben, ha a `Revenue`, `Expenses` és `Profit` értékek közül csak az `Expenses` hiányzik, az értékét állítsa be a bevétel és a profit különbségére ($Revenue - Profit$).
3. A táblázatban a `Cityace` cég iparágát állítsuk be `Health` értékűre.

---

7. Hiány pótlása mediánnal

A következőkben a hiányzó létszámadatokat pótoljuk az iparágban tapasztalható medián értékkel.

```
data[is.na(data$Employees),]
```

```
##      ID        Name           Industry Inception Employees State       City
## 3     3    Greenfax             Retail      2012        NA    SC Greenville
## 332 332 Westminster Financial Services      2010        NA    MI       Troy
##      Revenue Expenses  Profit Growth
## 3    9746272  1044375 8701897     16
## 332 11861652  5245126 6616526     15
```

```
# 3, Greenfax
name_flt <- data$Name == "Greenfax"
ind_empl <- data$Industry[name_flt]
med_empl <- median(data[data$Industry == ind_empl, "Employees"], na.rm = TRUE)
data[name_flt, "Employees"] <- med_empl
```

---

1. Az adathalmazon a pótolja a hiányzó alkalmazotti létszámot az iparágra jellemző értékkel!

---

8. Adatok törlése

Az adatok közül szelektíven is törölhetünk. A következőkben töröljük azokat az adatokat, melyek esetében a bevétel, kiadás és a profit értékek nincsennek kitöltve.

```
data <- data[!(is.na(data$Revenue) & is.na(data$Expenses)),]
data[!complete.cases(data),]
```

```
##    ID       Name Industry Inception Employees State   City Revenue Expenses
## 22 22 Lathotline   Health        NA       103    VA McLean 9418303  7567233
##     Profit Growth
## 22 1851070      2
```

8.1. Indexek helyreállítása

A törlés után a sorok indexei és a sorok azonosítói nem maradtak folytonosak. A 8 és 44 indexek hiányoznak. Törlés után a megfelelő adatindexelés végett a sorok indexeit helyre kell állítani.

```
# A 13. index után az indexek távolsága nagyobb, mint 1
which(diff(as.numeric(rownames(data))) > 1)
```

```
## [1]  7 42
```

```
id.step <- which(diff(as.numeric(rownames(data))) > 1)
# Az indexek távolsága 2
diff(data$ID)[id.step]
```

```
## [1] 2 2
```

```
# A 8 és 44 hiányzik
rownames(data)[c(6:10, 40:46)]
```

```
##  [1] "6"  "7"  "9"  "10" "11" "41" "42" "43" "45" "46" "47" "48"
```

```
# A sorok indexeinek újra generálása
rownames(data) <- 1:nrow(data)
which(diff(as.numeric(rownames(data))) > 1)
```

```
## integer(0)
```

```
# A másik megoldás az indexek nullázása
# Ekkor automatikusan újragenerálódnak az indexek
rownames(data) <- NULL
which(diff(as.numeric(rownames(data))) > 1)
```

```
## integer(0)
```

9. Adatok figyelmen kívül hagyása

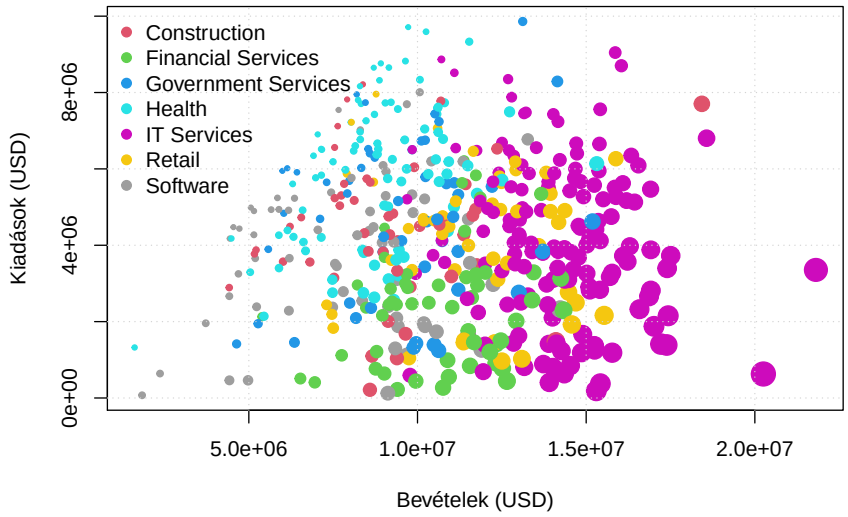
Az `Inception` mező értékét változatlanul hagyjuk, mivel itt nem foglalkozunk ezzel az értékkel.

10. Eredmények megjelenítése

A következőkben ábrázoljuk az egyes cégek adatit egy ponthalmazon, melyben a bevételek függvényében jelenítjük meg a kiadásokat. A megjelenített pontok nagyságát a profit alapján állapítjuk meg, valamint az egyes iparágak alapján színezzük az ábrát. Az ábrához készítünk jelmagyarázatot is.

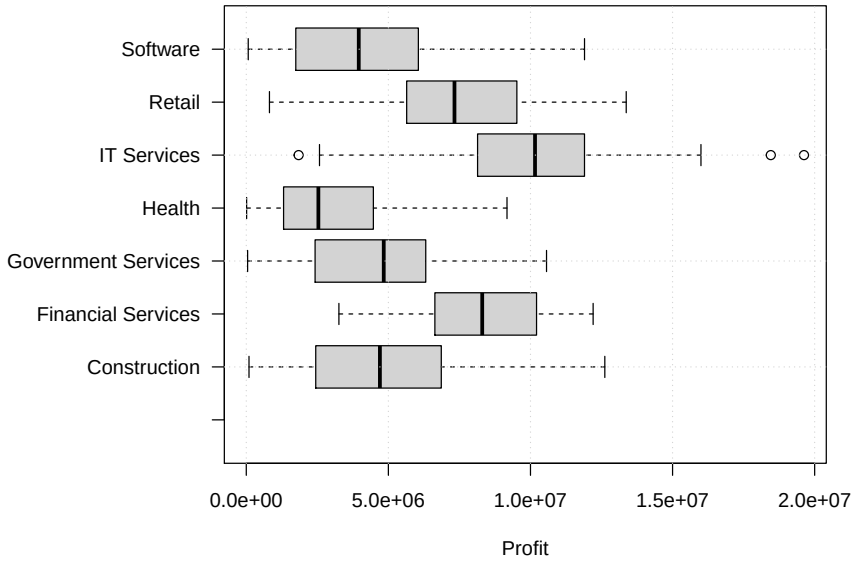
```
plot(data$Revenue, data$Expenses,
     col = data$Industry,
     cex = 0.5 + 2*data$Profit/max(data$Profit, na.rm = TRUE),
```

```
     pch = 19,
     xlab = "Bevételek (USD)", ylab = "Kiadások (USD)")
grid()
legend("topleft", legend = sort(unique(data$Industry)),
       pch = 19, col = sort(unique(data$Industry)),
       bg = rgb(0,0,0,0), box.lty = 0)
```



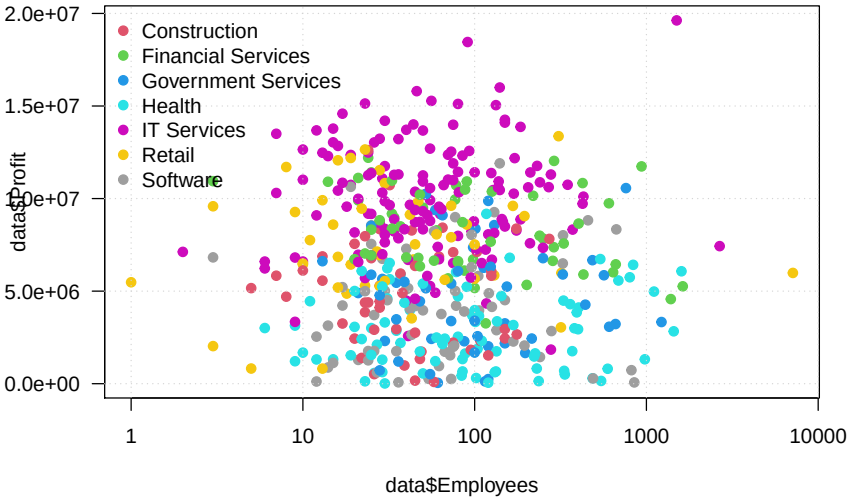
A következőkben boxplot segítségével ábrázoljuk az egyes iparági profit eloszlását.

```
par(mar = c(5, 9, 1, 1))
boxplot(data$Profit ~ data$Industry, las = 1, horizontal = T,
        xlab = "Profit", ylab = "")
grid()
```



---

1. A következőkben ábrázolja ponthalmaz segítségével, hogy a profit hogyan függ az alkalmazottak létszámától. A vízszintes tengelyen az alkalmazottak számát logikus léptékben jelenítsük meg. Az egyes pontokat színezzük az iparágak szerint. (Az ábra szemléltetőeszközként szolgál.)



2. Boxplot segítségével ábrázoljuk, hogy az egyes iparágakban milyen az alkalmazotti létszám eloszlása. (Az ábra szemléltetőeszközként szolgál.)