Andmestik

Allan Sims

31.01.2025

Andmestik

Metsanduses kogutakse pidevalt mitmesuguseid andmeid, mis võivad erineda nii oma tüübilt kui ka struktuurilt. Andmekogumise protsessi üldnimetus on **takseerimine**, mis hõlmab mitmesuguseid meetodeid nagu mõõtmine, loendamine, määramine ja muud sarnased tegevused. Iga takseerimisviis määrab ühtlasi ka kogutavate andmete tüübi.

- **Mõõtmisel** saadakse enamasti **pidev tunnus**. See on tunnus, mille väärtused võivad olla mistahes arvud teatud vahemikus. See tähendab, et pideva tunnuse puhul ei ole väärtused piiratud kindlate, eraldiseisvate arvudega, vaid võivad muutuda sujuvalt ja pidevalt. Näiteks mõõdetud puu kõrgus või diameeter.
- **Loendamisel** saadakse **diskreetne tunnus**. See on tunnus, mille väärtused saavad olla ainult teatud kindlad, eraldiseisvad arvud, tavaliselt täisarvud. See tähendab, et diskreetsete tunnuste puhul ei saa väärtused olla sujuvad või pidevad, vaid peavad olema selgelt eristuvad. Näiteks puude ja aastarõngaste loendamine. Diskreetne on ka rinnaspindala mõõtmisel relaskoobiga loendamine, kus loendatakse puid, kuid teatud juhtudel kasutatakse ka konstanti 0,5.
- **Määramisel** saadakse üldjuhul **nominaalne tunnus**. See on selline tunnus, mis väljendab kvalitatiivseid ehk mittearvulisi andmeid, mida ei saa järjestada. Näiteks puuliik, kasvukohatüüp vms. Selle alla liigitub ka **ordinaalne tunnus** ehk järjestustunnus on tunnus, mille väärtustel on loogiline järjestus, kuid väärtuste vahed ei ole tingimata võrdsed. Näiteks on metsanduses selleks boniteediklass.

Statistikas tähendab **andmestik** vaatlusandmete kogumit, mis on kogutud mingi uuringu või eksperimendi käigus. See on toorandmete hulk, mida analüüsitakse, et avastada mustreid, seoseid ja teha järeldusi. Andmete tüüp mängib olulist rolli statistilises analüüsis, kuna see määrab, milliseid meetodeid ja visualiseerimisviise saab kasutada. Õige andmetüübi tuvastamine on seega esimene samm edukaks andmetöötluseks ja tulemuste tõlgendamiseks.

Andmestik on organiseeritud andmete kogum, mis on tavaliselt salvestatud tabelitesse, andmebaasidesse või muudesse struktuuridesse.

- **Tüübid:**
  - **Vektor:** Sama tüüpi väärtuste jada (nt arvude vektor, tekstiväärtuste vektor).
  - **Maatriks:** Sama tüüpi väärtuste kogum, kuid mis on mitmemõõtmeline.
  - **Andmetabel:** Struktureeritud andmete kogum, kus iga rida esindab ühte objekti ja iga veerg esindab objekti atribuuti.

Andmestiku struktuur R keskkonnas

Vektor

Vektor on andmetüüp, mis võimaldab talletada mitut ühesugust andmetüüpi elementi ühes andmestruktuuris. Vektoris saab hoida näiteks numbreid, teksti või loogilisi väärtusi. Vektorit loetakse ühemõõtmeliseks andmestruktuuriks, kuna kõik selles sisalduvad elemendid asuvad ühes reas.

Vektorit saab luua kasutades funktsiooni `c()`, kuhu tuleb sisestada eraldatud komadega elemendid. Näiteks:

```r
# Näide R keskkonnas

num_vektor <- c(1, 2, 3, 4, 5)
```

Vektori elementidele saab viidata indeksite abil, alustades indeksist 1. Näiteks:

```r
# Näide R keskkonnas

# väljastab vektori num_vektor kolmanda elemendi väärtuse
num_vektor[3]
```

```
## [1] 3
```

Vektori pikkus ehk elementide arv on võimalik saada funktsiooniga `length()`. Vektorit saab ühendada teise vektoriga funktsiooniga `c()`. Näiteks:

```r
# Näide R keskkonnas

num_vektor_2 <- c(6, 7, 8)
kombineeritud_vektor <- c(num_vektor, num_vektor_2)
```

Võimalik on ka vektorit luua jada genereerimise abil, kasutades funktsiooni `seq()`. Näiteks:

```r
# Näide R keskkonnas

# loob vektori, mis sisaldab arve vahemikus 1 kuni 10 sammuga 2
jarg_vector <- seq(1, 10, by = 2)
```

Vektoritega on võimalik teostada erinevaid arvutusi, teisendusi ning andmete filtreerimist. Vektorid on olulised R programmeerimises, kuna nende abil on mugav töödelda andmeid ja teostada statistilisi analüüse.

Maatriks

Andmetüüp maatriks on kahe- või mitmemõõtmeline andmete struktuur, kus andmed on paigutatud mitme rea ja veeruna, moodustades seeläbi mitmemõõtmelise andmestiku. Maatriksis olevad andmed peavad olema kõik samatüübilised, näiteks kõik arvud või kõik tekstid.

Maatriksi loomiseks kasutatakse funktsioone `matrix()` või `array()`, millest esimene võimaldab luua kahemõõtmelist ning teine juba mitmemõõtmelisi maatrikseid. Nendele funktsioonidele saab määrata erinevaid parameetreid, näiteks andmete sisu, ridade ja veergude arvu ning rea- ja veerunimed.

```r
# Näide R keskkonnas
# funktsioon runif() loob juhuslike andmete vektori
# Loome kahemõõtmelise maatriksi juhuslikest arvudest
minu_maatriks <- matrix(runif(12), nrow=3, ncol=4)
print(minu_maatriks)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.8790838 0.2078004 0.5245263 0.1616878
## [2,] 0.2769429 0.5608873 0.8876623 0.4387479
## [3,] 0.4892830 0.3451458 0.5606076 0.4828189
```

```r
# Loome kolmemõõtmelise maatriksi juhuslikest arvudest
minu_array <- array(runif(12), dim = c(2,2,3))
```

Maatriksit saab kasutada erinevate arvutusoperatsioonide tegemiseks, nagu maatriksite korrutamine, liitmine, lahutamine jne. Samuti on võimalik maatriksit indekseerida ja välja printida konkreetseid ridu ja veerge.

Andmetabel

R-i andmetüüp andmetabel (data.frame) on üks olulisemaid andmestruktuure andmete haldamiseks ja analüüsimiseks R-keskkonnas. Tabelis on andmed korraldatud ridade (vaatluste) ja veergude (tunnuste) kaupa. Iga veerg andmetabelis vastab ühele muutujale, mis võib olla numbriline, tekstiline või faktor. Iga rida esindab ühte andmepunkti, mis sisaldab väärtusi vastavatele muutujatele.

Andmetabeli loomiseks saab kasutada funktsiooni `data.frame()`. Andmed saab andmetabelisse importida erinevatest andmeallikatest (nt CSV-failid, andmebaasid) või genereerida otse R-s.

```
# Näide R keskkonnas

# Loome data.frame andmestiku
minu_data_frame <- data.frame(x = runif(5), y = runif(5), z = 1)
```

Andmetabeli puhul on võimalik teha erinevaid andmehalduse ja analüüsi operatsioone nagu veergude ja ridade lisamine, kustutamine, filtreerimine, rühmitamine, sorteerimine ja statistiliste analüüside tegemine. Andmetabelil on oluline oskuslikult hallata ja kasutada, kuna see on väga levinud andmestruktuur R-keskkonnas ning võimaldab mugavalt töötada suurte andmekogumitega.

List

R-i andmetüüp list on objektide kogu, mis võimaldab ühte muutujasse salvestada erineva pikkuse ja tüübiga andmeid. Üks list võib koosneda erinevatest vektoritest, maatriksitest või tabelitest.

```
# Näide R keskkonnas

# Loome list andmestiku
minu_list <- list(
  x = runif(5),
  y = array(runif(16), dim = c(2,2,2)),
  z = c("MA","KU"))
```

Andmestike importimine R keskkonda

Andmete importimine CSV või Exceli failist R-i keskkonda on väga lihtne. Siin on näited mõlemast.

Andmete importimine CSV failist

Enne koodi käivitamist veendu, et CSV fail on salvestatud samas kaustas, kus Sa töötad R-is või et oled määranud õige tee failini.

```
# Näide R keskkonnas

andmed <- read.csv("failinimi.csv")
```

Kui CSV failil on teistsugune eraldusmärk (nt semikoolon), siis saab selle täpsustada lisades argumenti `sep=";"`.

```
# Näide R keskkonnas

andmed <- read.csv("failinimi.csv", sep=";")
```

Andmete importimine MS Exceli failist

Esmalt tuleb installida ja laadida sisse `openxlsx` pakett, mis võimaldab Exceli failidest andmete lugemist.

```
# Laadime peatükis kasutatavad paketid
library(dplyr)
library(readxl)
```

Seejärel kasutada funktsiooni `read_excel()`, et importida andmed Exceli failist.

```
# Näide R keskkonnas

andmed <- read_excel("failinimi.xlsx")
```

Arvutamine R-i keskkonnas

```
# Liitmine
1 + 1
```

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

```
# Lahutamine
5 - 3
```

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

```
# Korrutamine
2 * 4
```

```
## [1] 8
```

```
# Jagamine
10 / 2
```

```
## [1] 5
```

```
# Astendamine
2 ^ 3
```

```
## [1] 8
```

```
# Jäägi leidmine
10 %% 3
```

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

R keskkonnas on võimalik teha arvutusi tsüklina

```
# Loome vektori 1st 10ni

for(x in 1:10)
{
  y <- x + 1
}
```