

# Adatok beolvasása

Csima Judit

BME, VIK,  
Számítástudományi és Információelméleti Tanszék

2016. március 3. és március 10.

# Adatok beolvasása, kiírása

- `read.table`, `read.csv`: tabulált adatok beolvasására
- `readLines`: text file beolvasására soronként
- kiírásra `write.table`, `write.csv` illetve `writeLines`
- vannak még más író/olvasó parancsok is, de ezek a legfontosabbak

## read.table és read.csv paraméterei

- `file`: mit olvasunk be
- `header`: logikai változó, van-e fejléc
- `sep`: mi a szeparátor (default a space illetve a vessző)
- `colClasses`: egy character típusú vektor, mik az oszlopok osztályai
- `nrows`: hány sor van/kell
- `comment.char`: mi a komment jele (default a #)

Általában elég a `file` értékét megadni, a többinek van default értéke vagy megpróbálja kitalálni maga.

Az eredmény data frame lesz

## read.csv használata

- lehet konkrét helyről közvetlenül beolvasni  
`data = read.csv ("http://www.cs.bme.hu/~csima/dm15/001.csv")`
- lehet letölteni előbb magunkhoz és a file-t beolvasni  
`data = read.csv ("../public_html/dm14/001.csv")`  
ekkor a working directory-hoz képest relatív a címzés

## Text file soronkénti beolvasása

- `readLines`: beolvas egy file-t (vagy annak első néhány sorát) egy character típusú vektorba
- a "honnan" paraméter lehet url cím is:

```
> readLines("http://cs.bme.hu/dm", 5)
```

```
[1] "<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN"
```

```
[2] "<html>"
```

```
[3] " <head>"
```

```
[4] " <meta http-equiv=content-type content=
```

```
[5] " charset=ISO-8859-1"
```

## Mielőtt beolvassuk az adatokat, számoljunk....

Ha van 1,500,000 sorunk és 120 oszlopunk, minden adat szám és egy szám 8 byte, akkor ez összesen

$$\begin{aligned} 1,500,000 \times 120 \times 8 \text{ bytes} &= 1440000000 \text{ bytes} \\ &= 1,373.29 \text{ MB} = 1.34 \text{ GB} \end{aligned}$$