

2: Tidy data

Videregående kvantitative metoder i studiet af politisk adfærd

Frederik Hjorth
fh@ifs.ku.dk
fghjorth.github.io
@fghjorth

Institut for Statskundskab
Københavns Universitet

13. september 2017

1 Opsamling fra sidst

2 Tidy data

3 Piping

4 Øvelse

5 Kig fremad

Leeper, *Really Introductory Introduction*:

- Getting started
 - brug af R som regnemaskine: fx. `(2+4)/7`
 - parsing errors ctr. syntax errors
 - nye vektorer: fx. `dice <- c(2,2,3,4)`
 - ekstrahering fra vektorer: fx. `dice[1:3]`
 - ny data frame: fx. `df <- data.frame(dice,number=1:4)`
 - data framens struktur: `str(df)`
 - centrale tendenser: `summary(df)`

- Real data
 - installer pakker: `install.packages()`
 - indlæs pakker: `library()`
 - importér data: `import()` fra rio-pakken

- Randomness
 - sample fra en vektor: `sample()`
- Plots
 - pakke: `ggplot2`
 - fx. `ggplot(iris,aes(x=Sepal.Length)) + geom_histogram`

- Basic programming tools
 - funktioner: fx. `ftoc <- function(f){ c<-((f-35)*5)/9 ; print(c) }`
 - for loops: fx. `for (i in 1:10) print(i*i)`

Swirl øvelse 1.1-1.4

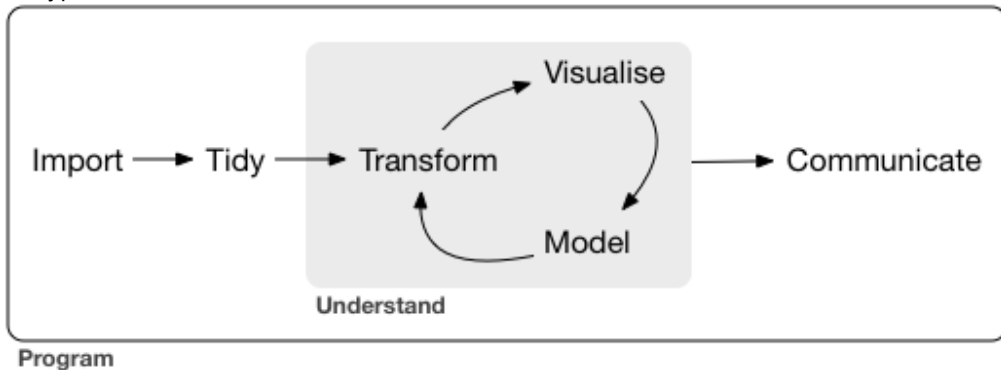
- assignment
- vektorer
- 'recycling' af vektorer
- working directories
- talrækker
- atomiske vektorer ctr. lister
- logiske operatorer og vektorer
- character vektorer
- `paste()`

Tænkeren bag 'tidy data': Hadley Wickham



i alt forfatter til ≈ 63 R-pakker (a.k.a. 'hadleyverse')

Et typisk data science workflow:



Centrale Hadley-pakker til databehandling:

- ggplot2 (visualisering)
- dplyr (databehandling)
- readr (import af csv o. lign.)
- purrr (funktioner og loops)
- tibble (smartere version af data frames)

→ alle disse kan loades samtidig med
`library(tidyverse)`



Andre nyttige hadleyverse-pakker:

- haven (import af data fra Stata, SPSS etc.)
- rvest (web scraping)
- stringr (behandling af character-objekter)
- lubridate (dato-objekter)
- forcats (faktorer)

Messy data:

	treatmenta	treatmentb
John Smith	–	2
Jane Doe	16	11
Mary Johnson	3	1

Tidy data:

person	treatment	result
John Smith	a	–
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

Principper i tidy data:

- ① Hver variabel udgør en kolonne
- ② Hver observation udgør en række
- ③ Hver enhed udgør en tabel

NB: good for data entry \neq tidy \rightarrow det typiske datasæt er messy!

De fire verber i databehandling:

- ① *filter*: udvælg observationer
- ② *transform*: tilføj eller omkode variable
- ③ *aggregate*: opsummere værdier til én eller færre
- ④ *sort*: ændre observationers rækkefølge

→ vi kan foretage (næsten) al databehandling med dplyr

Implementering af de fire verber i dplyr:

	verbum	funktion i dplyr
1	filter	<code>filter()</code> til rækker, <code>select()</code> til kolonner
2	transform	<code>mutate()</code>
3	aggregate	<code>group_by()</code> og <code>summarise()</code>
4	sort	<code>arrange()</code>

Uvurderligt opslagsværk til databehandling: *data wrangling cheat sheet*

Data Wrangling with dplyr and tidyr

Cheat Sheet



Syntax - Helpful conventions for wrangling

dplyr::tbl_df(iris)

Converts data to tbl class. tbl's are easier to examine than data frames. R displays only the data that fits onscreen.

```
Source: local data frame [158 x 5]
  Sepal.Length Sepal.Length Petal.Length
1         5.1         3.5         1.4
2         4.9         3.0         1.4
3         4.7         3.2         1.3
4         4.6         3.1         1.5
5         5.0         3.6         1.4
***
Variables shown: Petal.Length (dbl),
Variable(s) hidden: Species (fctr)
```

dplyr::glimpse(iris)

Information dense summary of tbl data.

utils::View(iris)

View data set in spreadsheet-like display (note capital V).

```
head(iris)
  Sepal.Length Petal.Length Species
1         5.1         1.4 setosa
2         4.9         1.4 setosa
3         4.7         1.3 setosa
4         4.6         1.5 setosa
5         5.0         1.4 setosa
6         5.4         1.7 versicolour
```

dplyr::%>%

Passes object on left hand side as first argument (or argument) of function on right hand side.

```
x %>% f(x) is the same as f(x, x)
y %>% f(x, ..., x) is the same as f(x, y, ..., x)
```

"Piping" with %>% makes code more readable, e.g.

```
iris %>%
  group_by(Species) %>%
  summarise(m = mean(Sepal.Length)) %>%
  arrange(m)
```

©2016 Pearson Education, Inc. All rights reserved. RStudio is a trademark of RStudio, Inc. R is a trademark of R Foundation for Statistical Computing. RStudio is a trademark of RStudio, Inc. R is a trademark of R Foundation for Statistical Computing.

Tidy Data - A foundation for wrangling in R

In a tidy data set:

Each variable is saved in its own column

&

Each observation is saved in its own row

No other format works as intuitively with R

Tidy data complements R's vectorized operations. It will automatically preserve observations as you manipulate variables.

No other format works as intuitively with R

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

M * A = A

Reshaping Data - Change the layout of a data set

tidyr::gather(cases, "year", "m", 2:4)

Gather columns into rows.

tidyr::spread(pollution, size, amount)

Spread rows into columns.

tidyr::separate(storms, date, c("y", "m", "d"))

Separate one column into several.

tidyr::unite(data, col_1, ..., col_n, sep)

Unite several columns into one.

Subset Observations (Rows)

dplyr::filter(iris, Sepal.Length > 7)

Extract rows that meet logical criteria.

dplyr::distinct(iris)

Remove duplicate rows.

dplyr::sample_frac(iris, 0.5, replace = TRUE)

Randomly select fraction of rows.

dplyr::sample_n(iris, 10, replace = TRUE)

Randomly select n rows.

dplyr::slice(iris, 10:15)

Select rows by position.

dplyr::top_n(storms, 2, date)

Select and order top n entries (by group if grouped data).

Logic in R - Comparison, Base R Logic

```
> 1 < 2 # TRUE
> 1 <= 2 # TRUE
> 1 > 2 # FALSE
> 1 >= 2 # FALSE
> 1 == 2 # FALSE
> 1 != 2 # TRUE
> 1 %in% 2 # FALSE
> 2 %in% 1 # TRUE
> 1 %in% 1:2 # TRUE
> 1:2 %in% 1 # TRUE
> 1:2 %in% 2 # TRUE
> 1:2 %in% 1:3 # TRUE
> 1:2 %in% 3:4 # FALSE
> 1:2 %in% 3:5 # FALSE
> 1:2 %in% 3:6 # FALSE
> 1:2 %in% 3:7 # FALSE
> 1:2 %in% 3:8 # FALSE
> 1:2 %in% 3:9 # FALSE
> 1:2 %in% 3:10 # FALSE
> 1:2 %in% 3:11 # FALSE
> 1:2 %in% 3:12 # FALSE
> 1:2 %in% 3:13 # FALSE
> 1:2 %in% 3:14 # FALSE
> 1:2 %in% 3:15 # FALSE
> 1:2 %in% 3:16 # FALSE
> 1:2 %in% 3:17 # FALSE
> 1:2 %in% 3:18 # FALSE
> 1:2 %in% 3:19 # FALSE
> 1:2 %in% 3:20 # FALSE
> 1:2 %in% 3:21 # FALSE
> 1:2 %in% 3:22 # FALSE
> 1:2 %in% 3:23 # FALSE
> 1:2 %in% 3:24 # FALSE
> 1:2 %in% 3:25 # FALSE
> 1:2 %in% 3:26 # FALSE
> 1:2 %in% 3:27 # FALSE
> 1:2 %in% 3:28 # FALSE
> 1:2 %in% 3:29 # FALSE
> 1:2 %in% 3:30 # FALSE
> 1:2 %in% 3:31 # FALSE
> 1:2 %in% 3:32 # FALSE
> 1:2 %in% 3:33 # FALSE
> 1:2 %in% 3:34 # FALSE
> 1:2 %in% 3:35 # FALSE
> 1:2 %in% 3:36 # FALSE
> 1:2 %in% 3:37 # FALSE
> 1:2 %in% 3:38 # FALSE
> 1:2 %in% 3:39 # FALSE
> 1:2 %in% 3:40 # FALSE
> 1:2 %in% 3:41 # FALSE
> 1:2 %in% 3:42 # FALSE
> 1:2 %in% 3:43 # FALSE
> 1:2 %in% 3:44 # FALSE
> 1:2 %in% 3:45 # FALSE
> 1:2 %in% 3:46 # FALSE
> 1:2 %in% 3:47 # FALSE
> 1:2 %in% 3:48 # FALSE
> 1:2 %in% 3:49 # FALSE
> 1:2 %in% 3:50 # FALSE
> 1:2 %in% 3:51 # FALSE
> 1:2 %in% 3:52 # FALSE
> 1:2 %in% 3:53 # FALSE
> 1:2 %in% 3:54 # FALSE
> 1:2 %in% 3:55 # FALSE
> 1:2 %in% 3:56 # FALSE
> 1:2 %in% 3:57 # FALSE
> 1:2 %in% 3:58 # FALSE
> 1:2 %in% 3:59 # FALSE
> 1:2 %in% 3:60 # FALSE
> 1:2 %in% 3:61 # FALSE
> 1:2 %in% 3:62 # FALSE
> 1:2 %in% 3:63 # FALSE
> 1:2 %in% 3:64 # FALSE
> 1:2 %in% 3:65 # FALSE
> 1:2 %in% 3:66 # FALSE
> 1:2 %in% 3:67 # FALSE
> 1:2 %in% 3:68 # FALSE
> 1:2 %in% 3:69 # FALSE
> 1:2 %in% 3:70 # FALSE
> 1:2 %in% 3:71 # FALSE
> 1:2 %in% 3:72 # FALSE
> 1:2 %in% 3:73 # FALSE
> 1:2 %in% 3:74 # FALSE
> 1:2 %in% 3:75 # FALSE
> 1:2 %in% 3:76 # FALSE
> 1:2 %in% 3:77 # FALSE
> 1:2 %in% 3:78 # FALSE
> 1:2 %in% 3:79 # FALSE
> 1:2 %in% 3:80 # FALSE
> 1:2 %in% 3:81 # FALSE
> 1:2 %in% 3:82 # FALSE
> 1:2 %in% 3:83 # FALSE
> 1:2 %in% 3:84 # FALSE
> 1:2 %in% 3:85 # FALSE
> 1:2 %in% 3:86 # FALSE
> 1:2 %in% 3:87 # FALSE
> 1:2 %in% 3:88 # FALSE
> 1:2 %in% 3:89 # FALSE
> 1:2 %in% 3:90 # FALSE
> 1:2 %in% 3:91 # FALSE
> 1:2 %in% 3:92 # FALSE
> 1:2 %in% 3:93 # FALSE
> 1:2 %in% 3:94 # FALSE
> 1:2 %in% 3:95 # FALSE
> 1:2 %in% 3:96 # FALSE
> 1:2 %in% 3:97 # FALSE
> 1:2 %in% 3:98 # FALSE
> 1:2 %in% 3:99 # FALSE
> 1:2 %in% 3:100 # FALSE
> 1:2 %in% 3:101 # FALSE
> 1:2 %in% 3:102 # FALSE
> 1:2 %in% 3:103 # FALSE
> 1:2 %in% 3:104 # FALSE
> 1:2 %in% 3:105 # FALSE
> 1:2 %in% 3:106 # FALSE
> 1:2 %in% 3:107 # FALSE
> 1:2 %in% 3:108 # FALSE
> 1:2 %in% 3:109 # FALSE
> 1:2 %in% 3:110 # FALSE
> 1:2 %in% 3:111 # FALSE
> 1:2 %in% 3:112 # FALSE
> 1:2 %in% 3:113 # FALSE
> 1:2 %in% 3:114 # FALSE
> 1:2 %in% 3:115 # FALSE
> 1:2 %in% 3:116 # FALSE
> 1:2 %in% 3:117 # FALSE
> 1:2 %in% 3:118 # FALSE
> 1:2 %in% 3:119 # FALSE
> 1:2 %in% 3:120 # FALSE
> 1:2 %in% 3:121 # FALSE
> 1:2 %in% 3:122 # FALSE
> 1:2 %in% 3:123 # FALSE
> 1:2 %in% 3:124 # FALSE
> 1:2 %in% 3:125 # FALSE
> 1:2 %in% 3:126 # FALSE
> 1:2 %in% 3:127 # FALSE
> 1:2 %in% 3:128 # FALSE
> 1:2 %in% 3:129 # FALSE
> 1:2 %in% 3:130 # FALSE
> 1:2 %in% 3:131 # FALSE
> 1:2 %in% 3:132 # FALSE
> 1:2 %in% 3:133 # FALSE
> 1:2 %in% 3:134 # FALSE
> 1:2 %in% 3:135 # FALSE
> 1:2 %in% 3:136 # FALSE
> 1:2 %in% 3:137 # FALSE
> 1:2 %in% 3:138 # FALSE
> 1:2 %in% 3:139 # FALSE
> 1:2 %in% 3:140 # FALSE
> 1:2 %in% 3:141 # FALSE
> 1:2 %in% 3:142 # FALSE
> 1:2 %in% 3:143 # FALSE
> 1:2 %in% 3:144 # FALSE
> 1:2 %in% 3:145 # FALSE
> 1:2 %in% 3:146 # FALSE
> 1:2 %in% 3:147 # FALSE
> 1:2 %in% 3:148 # FALSE
> 1:2 %in% 3:149 # FALSE
> 1:2 %in% 3:150 # FALSE
> 1:2 %in% 3:151 # FALSE
> 1:2 %in% 3:152 # FALSE
> 1:2 %in% 3:153 # FALSE
> 1:2 %in% 3:154 # FALSE
> 1:2 %in% 3:155 # FALSE
> 1:2 %in% 3:156 # FALSE
> 1:2 %in% 3:157 # FALSE
> 1:2 %in% 3:158 # FALSE
> 1:2 %in% 3:159 # FALSE
> 1:2 %in% 3:160 # FALSE
> 1:2 %in% 3:161 # FALSE
> 1:2 %in% 3:162 # FALSE
> 1:2 %in% 3:163 # FALSE
> 1:2 %in% 3:164 # FALSE
> 1:2 %in% 3:165 # FALSE
> 1:2 %in% 3:166 # FALSE
> 1:2 %in% 3:167 # FALSE
> 1:2 %in% 3:168 # FALSE
> 1:2 %in% 3:169 # FALSE
> 1:2 %in% 3:170 # FALSE
> 1:2 %in% 3:171 # FALSE
> 1:2 %in% 3:172 # FALSE
> 1:2 %in% 3:173 # FALSE
> 1:2 %in% 3:174 # FALSE
> 1:2 %in% 3:175 # FALSE
> 1:2 %in% 3:176 # FALSE
> 1:2 %in% 3:177 # FALSE
> 1:2 %in% 3:178 # FALSE
> 1:2 %in% 3:179 # FALSE
> 1:2 %in% 3:180 # FALSE
> 1:2 %in% 3:181 # FALSE
> 1:2 %in% 3:182 # FALSE
> 1:2 %in% 3:183 # FALSE
> 1:2 %in% 3:184 # FALSE
> 1:2 %in% 3:185 # FALSE
> 1:2 %in% 3:186 # FALSE
> 1:2 %in% 3:187 # FALSE
> 1:2 %in% 3:188 # FALSE
> 1:2 %in% 3:189 # FALSE
> 1:2 %in% 3:190 # FALSE
> 1:2 %in% 3:191 # FALSE
> 1:2 %in% 3:192 # FALSE
> 1:2 %in% 3:193 # FALSE
> 1:2 %in% 3:194 # FALSE
> 1:2 %in% 3:195 # FALSE
> 1:2 %in% 3:196 # FALSE
> 1:2 %in% 3:197 # FALSE
> 1:2 %in% 3:198 # FALSE
> 1:2 %in% 3:199 # FALSE
> 1:2 %in% 3:200 # FALSE
> 1:2 %in% 3:201 # FALSE
> 1:2 %in% 3:202 # FALSE
> 1:2 %in% 3:203 # FALSE
> 1:2 %in% 3:204 # FALSE
> 1:2 %in% 3:205 # FALSE
> 1:2 %in% 3:206 # FALSE
> 1:2 %in% 3:207 # FALSE
> 1:2 %in% 3:208 # FALSE
> 1:2 %in% 3:209 # FALSE
> 1:2 %in% 3:210 # FALSE
> 1:2 %in% 3:211 # FALSE
> 1:2 %in% 3:212 # FALSE
> 1:2 %in% 3:213 # FALSE
> 1:2 %in% 3:214 # FALSE
> 1:2 %in% 3:215 # FALSE
> 1:2 %in% 3:216 # FALSE
> 1:2 %in% 3:217 # FALSE
> 1:2 %in% 3:218 # FALSE
> 1:2 %in% 3:219 # FALSE
> 1:2 %in% 3:220 # FALSE
> 1:2 %in% 3:221 # FALSE
> 1:2 %in% 3:222 # FALSE
> 1:2 %in% 3:223 # FALSE
> 1:2 %in% 3:224 # FALSE
> 1:2 %in% 3:225 # FALSE
> 1:2 %in% 3:226 # FALSE
> 1:2 %in% 3:227 # FALSE
> 1:2 %in% 3:228 # FALSE
> 1:2 %in% 3:229 # FALSE
> 1:2 %in% 3:230 # FALSE
> 1:2 %in% 3:231 # FALSE
> 1:2 %in% 3:232 # FALSE
> 1:2 %in% 3:233 # FALSE
> 1:2 %in% 3:234 # FALSE
> 1:2 %in% 3:235 # FALSE
> 1:2 %in% 3:236 # FALSE
> 1:2 %in% 3:237 # FALSE
> 1:2 %in% 3:238 # FALSE
> 1:2 %in% 3:239 # FALSE
> 1:2 %in% 3:240 # FALSE
> 1:2 %in% 3:241 # FALSE
> 1:2 %in% 3:242 # FALSE
> 1:2 %in% 3:243 # FALSE
> 1:2 %in% 3:244 # FALSE
> 1:2 %in% 3:245 # FALSE
> 1:2 %in% 3:246 # FALSE
> 1:2 %in% 3:247 # FALSE
> 1:2 %in% 3:248 # FALSE
> 1:2 %in% 3:249 # FALSE
> 1:2 %in% 3:250 # FALSE
> 1:2 %in% 3:251 # FALSE
> 1:2 %in% 3:252 # FALSE
> 1:2 %in% 3:253 # FALSE
> 1:2 %in% 3:254 # FALSE
> 1:2 %in% 3:255 # FALSE
> 1:2 %in% 3:256 # FALSE
> 1:2 %in% 3:257 # FALSE
> 1:2 %in% 3:258 # FALSE
> 1:2 %in% 3:259 # FALSE
> 1:2 %in% 3:260 # FALSE
> 1:2 %in% 3:261 # FALSE
> 1:2 %in% 3:262 # FALSE
> 1:2 %in% 3:263 # FALSE
> 1:2 %in% 3:264 # FALSE
> 1:2 %in% 3:265 # FALSE
> 1:2 %in% 3:266 # FALSE
> 1:2 %in% 3:267 # FALSE
> 1:2 %in% 3:268 # FALSE
> 1:2 %in% 3:269 # FALSE
> 1:2 %in% 3:270 # FALSE
> 1:2 %in% 3:271 # FALSE
> 1:2 %in% 3:272 # FALSE
> 1:2 %in% 3:273 # FALSE
> 1:2 %in% 3:274 # FALSE
> 1:2 %in% 3:275 # FALSE
> 1:2 %in% 3:276 # FALSE
> 1:2 %in% 3:277 # FALSE
> 1:2 %in% 3:278 # FALSE
> 1:2 %in% 3:279 # FALSE
> 1:2 %in% 3:280 # FALSE
> 1:2 %in% 3:281 # FALSE
> 1:2 %in% 3:282 # FALSE
> 1:2 %in% 3:283 # FALSE
> 1:2 %in% 3:284 # FALSE
> 1:2 %in% 3:285 # FALSE
> 1:2 %in% 3:286 # FALSE
> 1:2 %in% 3:287 # FALSE
> 1:2 %in% 3:288 # FALSE
> 1:2 %in% 3:289 # FALSE
> 1:2 %in% 3:290 # FALSE
> 1:2 %in% 3:291 # FALSE
> 1:2 %in% 3:292 # FALSE
> 1:2 %in% 3:293 # FALSE
> 1:2 %in% 3:294 # FALSE
> 1:2 %in% 3:295 # FALSE
> 1:2 %in% 3:296 # FALSE
> 1:2 %in% 3:297 # FALSE
> 1:2 %in% 3:298 # FALSE
> 1:2 %in% 3:299 # FALSE
> 1:2 %in% 3:300 # FALSE
> 1:2 %in% 3:301 # FALSE
> 1:2 %in% 3:302 # FALSE
> 1:2 %in% 3:303 # FALSE
> 1:2 %in% 3:304 # FALSE
> 1:2 %in% 3:305 # FALSE
> 1:2 %in% 3:306 # FALSE
> 1:2 %in% 3:307 # FALSE
> 1:2 %in% 3:308 # FALSE
> 1:2 %in% 3:309 # FALSE
> 1:2 %in% 3:310 # FALSE
> 1:2 %in% 3:311 # FALSE
> 1:2 %in% 3:312 # FALSE
> 1:2 %in% 3:313 # FALSE
> 1:2 %in% 3:314 # FALSE
> 1:2 %in% 3:315 # FALSE
> 1:2 %in% 3:316 # FALSE
> 1:2 %in% 3:317 # FALSE
> 1:2 %in% 3:318 # FALSE
> 1:2 %in% 3:319 # FALSE
> 1:2 %in% 3:320 # FALSE
> 1:2 %in% 3:321 # FALSE
> 1:2 %in% 3:322 # FALSE
> 1:2 %in% 3:323 # FALSE
> 1:2 %in% 3:324 # FALSE
> 1:2 %in% 3:325 # FALSE
> 1:2 %in% 3:326 # FALSE
> 1:2 %in% 3:327 # FALSE
> 1:2 %in% 3:328 # FALSE
> 1:2 %in% 3:329 # FALSE
> 1:2 %in% 3:330 # FALSE
> 1:2 %in% 3:331 # FALSE
> 1:2 %in% 3:332 # FALSE
> 1:2 %in% 3:333 # FALSE
> 1:2 %in% 3:334 # FALSE
> 1:2 %in% 3:335 # FALSE
> 1:2 %in% 3:336 # FALSE
> 1:2 %in% 3:337 # FALSE
> 1:2 %in% 3:338 # FALSE
> 1:2 %in% 3:339 # FALSE
> 1:2 %in% 3:340 # FALSE
> 1:2 %in% 3:341 # FALSE
> 1:2 %in% 3:342 # FALSE
> 1:2 %in% 3:343 # FALSE
> 1:2 %in% 3:344 # FALSE
> 1:2 %in% 3:345 # FALSE
> 1:2 %in% 3:346 # FALSE
> 1:2 %in% 3:347 # FALSE
> 1:2 %in% 3:348 # FALSE
> 1:2 %in% 3:349 # FALSE
> 1:2 %in% 3:350 # FALSE
> 1:2 %in% 3:351 # FALSE
> 1:2 %in% 3:352 # FALSE
> 1:2 %in% 3:353 # FALSE
> 1:2 %in% 3:354 # FALSE
> 1:2 %in% 3:355 # FALSE
> 1:2 %in% 3:356 # FALSE
> 1:2 %in% 3:357 # FALSE
> 1:2 %in% 3:358 # FALSE
> 1:2 %in% 3:359 # FALSE
> 1:2 %in% 3:360 # FALSE
> 1:2 %in% 3:361 # FALSE
> 1:2 %in% 3:362 # FALSE
> 1:2 %in% 3:363 # FALSE
> 1:2 %in% 3:364 # FALSE
> 1:2 %in% 3:365 # FALSE
> 1:2 %in% 3:366 # FALSE
> 1:2 %in% 3:367 # FALSE
> 1:2 %in% 3:368 # FALSE
> 1:2 %in% 3:369 # FALSE
> 1:2 %in% 3:370 # FALSE
> 1:2 %in% 3:371 # FALSE
> 1:2 %in% 3:372 # FALSE
> 1:2 %in% 3:373 # FALSE
> 1:2 %in% 3:374 # FALSE
> 1:2 %in% 3:375 # FALSE
> 1:2 %in% 3:376 # FALSE
> 1:2 %in% 3:377 # FALSE
> 1:2 %in% 3:378 # FALSE
> 1:2 %in% 3:379 # FALSE
> 1:2 %in% 3:380 # FALSE
> 1:2 %in% 3:381 # FALSE
> 1:2 %in% 3:382 # FALSE
> 1:2 %in% 3:383 # FALSE
> 1:2 %in% 3:384 # FALSE
> 1:2 %in% 3:385 # FALSE
> 1:2 %in% 3:386 # FALSE
> 1:2 %in% 3:387 # FALSE
> 1:2 %in% 3:388 # FALSE
> 1:2 %in% 3:389 # FALSE
> 1:2 %in% 3:390 # FALSE
> 1:2 %in% 3:391 # FALSE
> 1:2 %in% 3:392 # FALSE
> 1:2 %in% 3:393 # FALSE
> 1:2 %in% 3:394 # FALSE
> 1:2 %in% 3:395 # FALSE
> 1:2 %in% 3:396 # FALSE
> 1:2 %in% 3:397 # FALSE
> 1:2 %in% 3:398 # FALSE
> 1:2 %in% 3:399 # FALSE
> 1:2 %in% 3:400 # FALSE
> 1:2 %in% 3:401 # FALSE
> 1:2 %in% 3:402 # FALSE
> 1:2 %in% 3:403 # FALSE
> 1:2 %in% 3:404 # FALSE
> 1:2 %in% 3:405 # FALSE
> 1:2 %in% 3:406 # FALSE
> 1:2 %in% 3:407 # FALSE
> 1:2 %in% 3:408 # FALSE
> 1:2 %in% 3:409 # FALSE
> 1:2 %in% 3:410 # FALSE
> 1:2 %in% 3:411 # FALSE
> 1:2 %in% 3:412 # FALSE
> 1:2 %in% 3:413 # FALSE
> 1:2 %in% 3:414 # FALSE
> 1:2 %in% 3:415 # FALSE
> 1:2 %in% 3:416 # FALSE
> 1:2 %in% 3:417 # FALSE
> 1:2 %in% 3:418 # FALSE
> 1:2 %in% 3:419 # FALSE
> 1:2 %in% 3:420 # FALSE
> 1:2 %in% 3:421 # FALSE
> 1:2 %in% 3:422 # FALSE
> 1:2 %in% 3:423 # FALSE
> 1:2 %in% 3:424 # FALSE
> 1:2 %in% 3:425 # FALSE
> 1:2 %in% 3:426 # FALSE
> 1:2 %in% 3:427 # FALSE
> 1:2 %in% 3:428 # FALSE
> 1:2 %in% 3:429 # FALSE
> 1:2 %in% 3:430 # FALSE
> 1:2 %in% 3:431 # FALSE
> 1:2 %in% 3:432 # FALSE
> 1:2 %in% 3:433 # FALSE
> 1:2 %in% 3:434 # FALSE
> 1:2 %in% 3:435 # FALSE
> 1:2 %in% 3:436 # FALSE
> 1:2 %in% 3:437 # FALSE
> 1:2 %in% 3:438 # FALSE
> 1:2 %in% 3:439 # FALSE
> 1:2 %in% 3:440 # FALSE
> 1:2 %in% 3:441 # FALSE
> 1:2 %in% 3:442 # FALSE
> 1:2 %in% 3:443 # FALSE
> 1:2 %in% 3:444 # FALSE
> 1:2 %in% 3:445 # FALSE
> 1:2 %in% 3:446 # FALSE
> 1:2 %in% 3:447 # FALSE
> 1:2 %in% 3:448 # FALSE
> 1:2 %in% 3:449 # FALSE
> 1:2 %in% 3:450 # FALSE
> 1:2 %in% 3:451 # FALSE
> 1:2 %in% 3:452 # FALSE
> 1:2 %in% 3:453 # FALSE
> 1:2 %in% 3:454 # FALSE
> 1:2 %in% 3:455 # FALSE
> 1:2 %in% 3:456 # FALSE
> 1:2 %in% 3:457 # FALSE
> 1:2 %in% 3:458 # FALSE
> 1:2 %in% 3:459 # FALSE
> 1:2 %in% 3:460 # FALSE
> 1:2 %in% 3:461 # FALSE
> 1:2 %in% 3:462 # FALSE
> 1:2 %in% 3:463 # FALSE
> 1:2 %in% 3:464 # FALSE
&gt
```


%>%

magrittr

Ceci n'est pas un pipe.

Data i øvelsen til næste gang: pakkedownloads på The Comprehensive R Archive Network (CRAN) d. 8. juli 2014

```
> cran
# A tibble: 225,468 x 11
   X      date      time    size r_version r_arch    r_os    package version country ip_id
  <int>   <chr>   <chr>   <int>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr> <int>
1     1 2014-07-08 00:54:41  80589    3.1.0 x86_64 mingw32  htmltools 0.2.4    US      1
2     2 2014-07-08 00:59:53 321767    3.1.0 x86_64 mingw32  tseries 0.10-32  US      2
3     3 2014-07-08 00:47:13  748063    3.1.0 x86_64 linux-gnu party 1.0-15   US      3
4     4 2014-07-08 00:48:05  606104    3.1.0 x86_64 linux-gnu Hmisc 3.14-4   US      3
5     5 2014-07-08 00:46:50   79825    3.0.2 x86_64 linux-gnu digest 0.6.4    CA      4
6     6 2014-07-08 00:48:04   77681    3.1.0 x86_64 linux-gnu randomForest 4.6-7   US      3
7     7 2014-07-08 00:48:35  393754    3.1.0 x86_64 linux-gnu plyr 1.8.1    US      3
8     8 2014-07-08 00:47:30   28216    3.0.2 x86_64 linux-gnu whisker 0.3-2    US      5
9     9 2014-07-08 00:54:58    5928      <NA>   <NA>   <NA>   Rcpp 0.10.4   CN      6
10    10 2014-07-08 00:15:35 2206029    3.0.2 x86_64 linux-gnu hflights 0.1     US      7
# ... with 225,458 more rows
```

Vi ønsker at gøre flg.:

- ① gruppér observationerne efter pakke
- ② udregn nogle summary stats for hver pakke
- ③ reducer til pakker downloadet i mere end 60 lande
- ④ sortér efter land og pakkestørrelse

Klodset løsning I: Gem hvert skridt i et nyt objekt (→ rod)

```
by_package <- group_by(cran, package)

pack_sum <- summarize(by_package,
                      count = n(),
                      unique = n_distinct(ip_id),
                      countries = n_distinct(country),
                      avg_bytes = mean(size))

top_countries <- filter(pack_sum, countries > 60)

result1 <- arrange(top_countries, desc(countries), avg_bytes)
```

Klodset løsning II: Skriv funktionerne indlejret (nested) i hinanden (→ totalt ulæseligt)

```
result2 <-  
  arrange(  
    filter(  
      summarize(  
        group_by(cran,  
                  package  
        ),  
        count = n(),  
        unique = n_distinct(ip_id),  
        countries = n_distinct(country),  
        avg_bytes = mean(size)  
      ),  
      countries > 60  
    ),  
    desc(countries),  
    avg_bytes
```

Smart løsning: brug pipede funktioner (→ enkelt, letlæseligt)

```
result3 <-  
  cran %>%  
  group_by(package) %>%  
  summarize(count = n(),  
             unique = n_distinct(ip_id),  
             countries = n_distinct(country),  
             avg_bytes = mean(size)  
  ) %>%  
  filter(countries > 60) %>%  
  arrange(desc(countries), avg_bytes)
```

bemærk: den originale data frame 'pipes' ned gennem funktionerne, så angives kun én gang

Øvelse: databehandling med dplyr

- ① load swirl med `library(swirl)`
- ② hent kurset "Getting and Cleaning Data"
 - bør fungere med `install_course("Getting and Cleaning Data")`
 - ellers download filen og kørs `install_course()`
- ③ løs øvelse 1 (dplyr verber)
- ④ løs øvelse 2 (piping fra 69 pct.)

Næste gang:

- OLS brush-up
- AP 3.1+3.2
- læs Newman (kun mhp. metoden), derefter Solt
- lektie: færdiggør modul 1+2 i “Getting and Cleaning Data”

Tak for i dag!