2: Tidy data

Videregående kvantitative metoder i studiet af politisk adfærd

Frederik Hjorth
adjunkt, Ph.D.
fh@ifs.ku.dk
fghjorth.github.io
@fghjorth

Institut for Statskundskab
Københavns Universitet

13. september 2018

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)`

DataCamp: Introduction to the Tidyverse

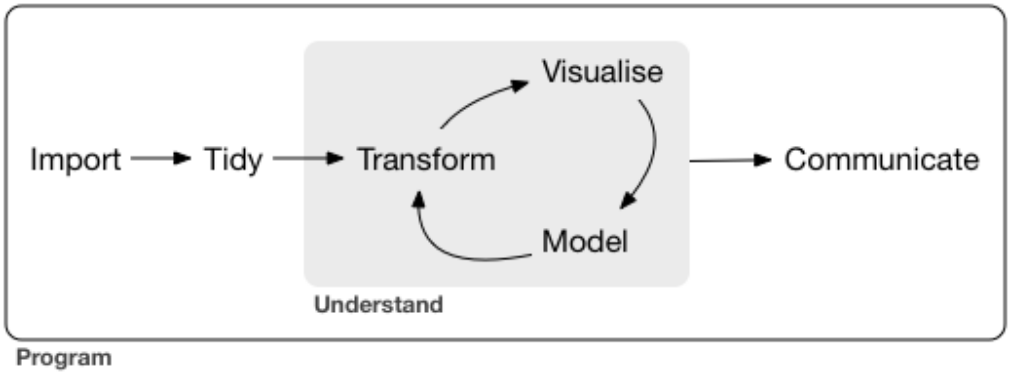
- 1: Data wrangling
  - gapminder-datasættet
  - 'filter'
  - 'arrange'
  - 'mutate'
- 3: Grouping and summarizing
  - 'summarize'
  - 'group_by'

Tænkeren bag 'tidy data': Hadley Wickham

i alt forfatter til $\approx$ 63 R-pakker (a.k.a. 'hadleyverse' eller 'tidyverse')

Den nye generation af R-udviklere:

Et typisk data science workflow:



Centrale Hadley-pakker til databehandling:

- ggplot2 (visualisering)
- dplyr (databehandling)
- tidyr (omformning)
- readr (import af csv o. lign.)
- purrr (funktioner og loops)
- tibble (smartere version af data frames)
- stringr (behandling af character-objekter)
- forcats (faktorer)

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

Andre nyttige hadleyverse-pakker:

- haven (import af data fra Stata, SPSS etc.)
- readxl (Excel-filer)
- rvest (web scraping)
- httr (API'er)
- lubridate (dato-objekter)

Den hyppigste, mest tidskrævende form for tidying: 'nørkling' med data

```
head(salary.selected)
```

```
##                      name          department_name total.earnings
## 1          Abadi,Kidani A     Assessing Department     $46,591.98
## 2        Abasciano,Joseph Boston Police Department     $97,579.88
## 3  Abban,Christopher John   Boston Fire Department    $124,623.25
## 4           Abbasi,Sophia            Green Academy     $18,249.83
## 5 Abbate-Vaughn,Jorgelina     BPS Ellis Elementary     $85,660.28
## 6        Abberton,James P  Public Works Department     $50,337.63
```

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!

Endnu et eksempel på messy ctr. tidy:

TABLE 1 "Messy" data on video game sales and review scores. Sales data from VGChartz; review data from Metacritic. Data set sourced from Rush Kirubi, via Kaggle (bit.ly/2KNZgM9).

Name	Publisher	Global sales (million units)	Critic	User
Wii Sports	Nintendo	82.53	Score = 76, Count = 51	Score = 8, Count = 322
Mario Kart Wii	Nintendo	35.52	Score = 82, Count = 73	Score = 8.3, Count = 709
Wii Sports Resort	Nintendo	32.77	Score = 80, Count = 73	Score = 8, Count = 192
New Super Mario Bros.	Nintendo	29.80	Score = 89, Count = 65	Score = 8.5, Count = 431
Wii Play	Nintendo	28.92	Score = 58, Count = 41	Score = 6.6, Count = 129

TABLE 2 "Tidy" data on video game sales and review scores.

Name	Publisher	Global sales (million units)	Rater	Score
Wii Sports	Nintendo	82.53	Critic	76
Wii Sports	Nintendo	82.53	User	80
Mario Kart Wii	Nintendo	35.52	Critic	82
Mario Kart Wii	Nintendo	35.52	User	83
Wii Sports Resort	Nintendo	32.77	Critic	80
Wii Sports Resort	Nintendo	32.77	User	80
New Super Mario Bros.	Nintendo	29.80	Critic	89
New Super Mario Bros.	Nintendo	29.80	User	85
Wii Play	Nintendo	28.92	Critic	58
Wii Play	Nintendo	28.92	User	66

Kilde: `https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1740-9713.2018.01169.x`

De fire verber i databehandling:

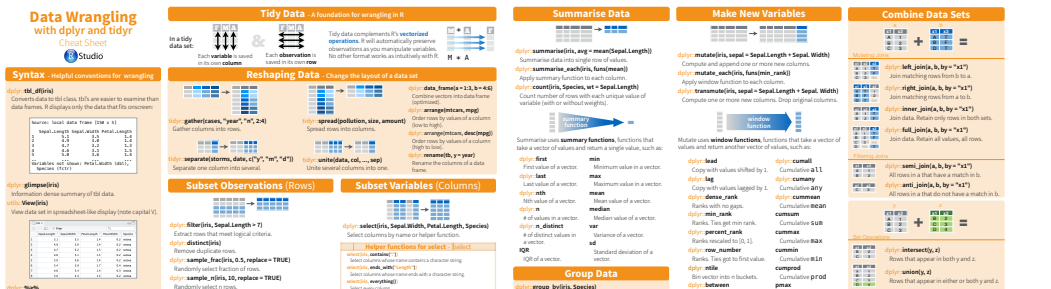
1. *filter*: udvælg observationer
2. *transform*: tilføje eller omkode variable
3. *aggregate*: opsummere værdier til én eller færre
4. *sort*: ændre observationers rækkefølge

$\rightarrow$ vi kan foretage (næsten) al databehandling med `dplyr`

Implementering af de fire verber i `dplyr`:

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

Uvurderligt opslagsværk til databehandling: *data wrangling cheat sheet*



ligger også på fagets hjemmeside under *extras*


%>%
magrittr
Ceci n'est pas un pipe.


1963
child mortality
30%
20%
10%
0%
babies born per woman
1
2
3
4
5
6
7
8

Data: datasættet i pakken 'gapminder'

```
> gapminder
# A tibble: 1,704 x 6
   country     continent  year lifeExp      pop gdpPercap
   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
 1 Afghanistan Asia       1952    28.8  8425333       779
 2 Afghanistan Asia       1957    30.3  9240934       821
 3 Afghanistan Asia       1962    32.0 10267083       853
 4 Afghanistan Asia       1967    34.0 11537966       836
 5 Afghanistan Asia       1972    36.1 13079460       740
 6 Afghanistan Asia       1977    38.4 14880372       786
 7 Afghanistan Asia       1982    39.9 12881816       978
 8 Afghanistan Asia       1987    40.8 13867957       852
 9 Afghanistan Asia       1992    41.7 16317921       649
10 Afghanistan Asia       1997    41.8 22227415       635
# ... with 1,694 more rows
> 
```

https://youtu.be/Z8t4k0Q8e8Y?t=54s

Vi ønsker at gøre flg.:

1. gruppér observationerne efter land
2. udregn gennemsnitslevetid og medianindbyggertal over hele perioden for hvert land
3. reducér til lande m. mere end 10M indbyggere
4. sortér efter land (omvendt alfabetisk) og gennemsnitslevetid

Klodset løsning I: Gem hvert skridt i et nyt objekt ($\rightarrow$ rod)

```
by_country <- group_by(gapminder, country)

country_sum <- summarize(by_country,avglifeexp=mean(lifeExp),
                                    medianpop=median(pop))

big_countries <- filter(country_sum, medianpop > 10000000)

result1 <- arrange(big_countries, desc(country), avglifeexp)
```

Klodset løsning II: Skriv funktionerne indlejret (nested) i hinanden ($\rightarrow$ totalt ulæseligt)

```
result2 <-
  arrange(
    filter(
      summarize(
        group_by(gapminder,
                 country
        ),
        avglifeexp=mean(lifeExp),
        medianpop=median(pop)
      ),
      medianpop > 10000000
    ),
    desc(country),
    avglifeexp
  )
```

Smart løsning: brug pipede funktioner ($\rightarrow$ enkelt, letlæseligt)

```
result3 <- gapminder %>%
group_by(country) %>%
summarize(avglifeexp=mean(lifeExp),
medianpop=median(pop)) %>%
filter(medianpop > 10000000) %>%
arrange(desc(country), avglifeexp)
```

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

»This directory contains data on nearly 3 million tweets sent from Twitter handles connected to the Internet Research Agency, a Russian "troll factory" and a defendant in an indictment filed by the Justice Department in February 2018, as part of special counsel Robert Mueller's Russia investigation. (...) The basis for the Twitter handles included in this data are the November 2017 and June 2018 lists of Internet Research Agency-connected handles that Twitter provided to Congress. This data set contains every tweet sent from each of the 2,752 handles on the November 2017 list since May 10, 2015.«

Kilde: `https://github.com/fivethirtyeight/russian-troll-tweets`

1. hent et af datasættene fra `https://github.com/fivethirtyeight/russian-troll-tweets`
2. hvad er de hyppigste og næsthyppigste sprog?
3. i hvilken region har trolde flest følgere?
4. hvilket sprog retweetes oftest?
5. hvor ofte nævnes hhv. Trump og Clinton i tweetsne?

Næste gang: Data fra online-kilder

- screen scraping
- API'er
- pensum: MRMN kap. 9+14
- vigtigt i kap. 9: 9.1.10+
- case: Hjorth (ananas i egen juice $\rightsquigarrow$ eksempel på data fra online-kilder, læs kursorisk)
- ekstra hjemmearbejde: `lav en twitter API key`

Tak for i dag!