Political Data Science

Lektion 3:

R Workshop II: Import, tidy, transform

Undervist af Jesper Svejgaard, foråret 2018
Institut for Statskundskab, Københavns Universitet
github.com/jespersvejgaard/PDS

I dag

1. Opsamling fra sidst
2. Highlights fra pensum
    - tibbles
    - import
    - tidyr
    - relationel data
3. Workshop
4. Opsamling og næste gang

Overblik

1. Intro til kurset og R
2. R Workshop I: Explore
3. R Workshop II: Import, tidy, transform
4. R Workshop III: Programmering & Git
5. Web scraping & API
6. Tekst som data
7. Visualisering
8. GIS & spatiale data
9. Estimation & prædiktion
10. Superviseret læring I
11. Superviseret læring II
12. Usuperviseret læring
13. Refleksioner om data science
14. Opsamling og eksamen

1. Opsamling fra sidst

Sidste gang

- Funktioner fra sidst
  - ggplot()
  - filter()
  - arrange()
  - select()
  - mutate()
  - summarize()
  - group_by()

- Gennemgang af opgaver (`02_script.R`)

2. Highlights fra pensum

Tibbles

- Tibbles = tweakede dataframes
- Jeg omtaler dem begge som data frames
- Funktionerne i the tidyverse outputter tibbles
- Forskelle:
  - Printing: tibbles printer 10 rækker og fitter kolonner til skærmbredde
  - Subsetting: returnerer tibble + ingen partial matching

Tibbles

Forskel 1: printing

```
# Dataframe
flights %>% as.data.frame()

# Tibble
flights %>% as.tibble()
```

Tibbles

Forskel 2: subsetting og partial matching

```
# Partial matching
flights %>% as.data.frame() %>% .$yea
```

```
##     [1] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##    [14] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##    [27] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##    [40] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##    [53] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##    [66] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##    [79] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##    [92] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##   [105] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##   [118] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##   [131] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##   [144] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##   [157] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##   [170] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##   [183] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
##   [196] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013
```

Tibbles

Forskel 2: subsetting og partial matching

```
# Ingen partial matching
flights %>% as.tibble() %>% .$yea
```

```
## Warning: Unknown or uninitialised column: 'yea'.
```

```
## NULL
```

Data import

- Indlæsning af flade filer med `readr`
- Pakken `readr`:
  - read_csv()
  - read_csv2()
  - read_delim()
  - m.fl.
- Hurtigere, mere transparent og mere konsistent end read.csv()

```
# Eksempel
read_delim(sti, skip = n, comment = "#", delim = "|", na = ".")
```

Data import

Hvad går galt her?

```
csv <- "vare, pris, pris_usd
        tv, 2.300, $385
        ur, 1.300, $220"

read_csv(csv)
```

```
## # A tibble: 2 x 3
##    vare  pris pris_usd
##   <chr> <dbl>    <chr>
## 1    tv   2.3     $385
## 2    ur   1.3     $220
```

Data import

```
# Import med specifikationer
read_csv(csv,
         locale = locale(grouping_mark = "."),
         col_types = cols(vare = col_character(),
                          pris = col_number(),
                          pris_usd = col_number()))
```

```
## # A tibble: 2 x 3
##    vare  pris pris_usd
##   <chr> <dbl>    <dbl>
## 1    tv  2300      385
## 2    ur  1300      220
```

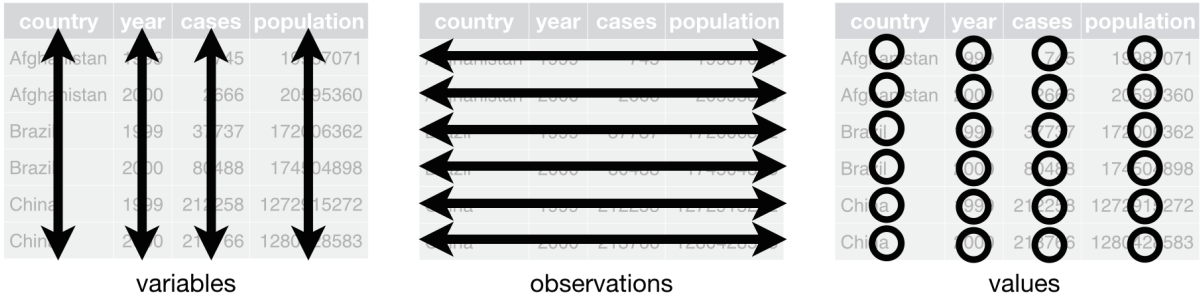
Data import

Pointen er:

- Selv .csv-filer kan se forskellige ud og give problemer
- Man bør være opmærksom når man importerer sine data
- Ikke meningen at I skal huske koden i hovedet - Google og StackOverflow er stadig jeres venner!

Tidy data

- Som vi ved:
  - Hver variabel har sin egen kolonne
  - Hver observation har sin egen række
  - Hver værdi har sin egen celle



Untidy data 1

Hvad er problemet her - og hvad kan vi gøre ved det?

```
untidy1
```

```
## # A tibble: 5 x 3
##         land  `2010`  `2015`
##        <chr>   <dbl>   <dbl>
## 1       Kina 1336681 1367486
## 2     Indien 1173109 1251696
## 3        USA  309348  321369
## 4 Indonesien  243423  255994
## 5  Brasilien  195835  204260
```

gather()

```
gather(untidy1, `2010`:`2015`, key = "år", value = "population")
```

```
## # A tibble: 10 x 3
##          land    år population
##         <chr> <chr>      <dbl>
##  1       Kina  2010    1336681
##  2     Indien  2010    1173109
##  3        USA  2010     309348
##  4 Indonesien  2010     243423
##  5  Brasilien  2010     195835
##  6       Kina  2015    1367486
##  7     Indien  2015    1251696
##  8        USA  2015     321369
##  9 Indonesien  2015     255994
## 10  Brasilien  2015     204260
```

gather()

Værdier i kolonnerne => `gather()`

Untidy data 2

Hvad er problemet her - og hvad kan vi gøre ved det?

```
untidy2
```

```
## # A tibble: 8 x 4
##    land    år observationer      antal
##   <chr> <dbl>         <chr>      <dbl>
## 1  Kina  2010    population 1336681000
## 2  Kina  2010      turister   56900000
## 3  Kina  2015    population 1367486000
## 4  Kina  2015      turister   54000000
## 5   USA  2010    population  309348000
## 6   USA  2010      turister   75000000
## 7   USA  2015    population  321369000
## 8   USA  2015      turister   73500000
```

spread()

```
spread(untidy2, key = "observationer", value = "antal")
```

```
## # A tibble: 4 x 4
##    land    år population turister
## * <chr> <dbl>      <dbl>    <dbl>
## 1  Kina  2010 1336681000 56900000
## 2  Kina  2015 1367486000 54000000
## 3   USA  2010  309348000 75000000
## 4   USA  2015  321369000 73500000
```

spread()

Når flere variable er i samme kolonne => `spread()`

country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

table2

Untidy data 3

Hvad gør vi nu?

```
untidy3
```

```
## # A tibble: 5 x 2
##         land      population
##        <chr>           <chr>
## 1       Kina 1336681/1367486
## 2     Indien 1173109/1251696
## 3        USA   309348/321369
## 4 Indonesien   243423/255994
## 5  Brasilien   195835/204260
```

separate()

- `separate()` deler én kolonne op i flere

```
separate(untidy3, population, into = c("2010", "2015"), sep = "/")
```

```
## # A tibble: 5 x 3
##         land  `2010`  `2015`
## *      <chr>   <chr>   <chr>
## 1       Kina 1336681 1367486
## 2     Indien 1173109 1251696
## 3        USA  309348  321369
## 4 Indonesien  243423  255994
## 5  Brasilien  195835  204260
```

- `unite()` gør det modatte - samler flere kolonner i én

Relationel data

- Når tabeller har en relation til hinanden
- Relationerne udnyttes med mutation joins, filtering joins og set operations

Keys

- Relationerne bygger på keys
- Primary keys = kolonne/kolonner som unikt identificerer én observation
- Foreign keys = kolonne/kolonner i en anden tabel, som kan kobles til en primary key

Keys

Hvad kan være primary key i `flights`?

```
## # A tibble: 336,776 x 8
##     year month   day dep_time dep_delay arr_time carrier tailnum
##    <int> <int> <int>    <int>     <dbl>    <int>   <chr>   <chr>
##  1  2013     1     1      517         2      830      UA  N14228
##  2  2013     1     1      533         4      850      UA  N24211
##  3  2013     1     1      542         2      923      AA  N619AA
##  4  2013     1     1      544        -1     1004      B6  N804JB
##  5  2013     1     1      554        -6      812      DL  N668DN
##  6  2013     1     1      554        -4      740      UA  N39463
##  7  2013     1     1      555        -5      913      B6  N516JB
##  8  2013     1     1      557        -3      709      EV  N829AS
##  9  2013     1     1      557        -3      838      B6  N593JB
## 10  2013     1     1      558        -2      753      AA  N3ALAA
## # ... with 336,766 more rows
```

Keys

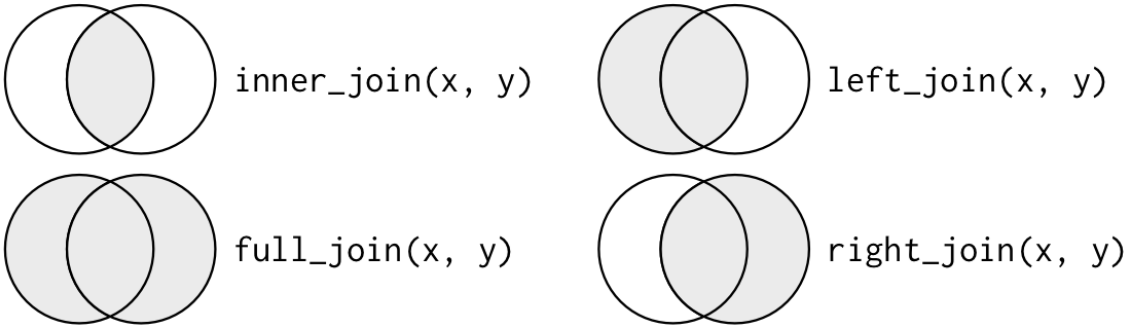
Hvad med kombinationen: `year`, `month`, `day`, `tailnum`?

… Næh …

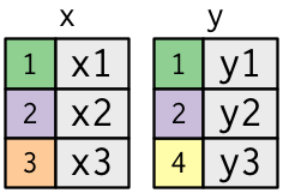
```
flights %>% count(year, month, day, tailnum) %>% filter(n > 1)
```

```
## # A tibble: 64,928 x 5
##     year month   day tailnum     n
##    <int> <int> <int>   <chr> <int>
##  1  2013     1     1  N0EGMQ     2
##  2  2013     1     1  N11189     2
##  3  2013     1     1  N11536     2
##  4  2013     1     1  N11544     3
##  5  2013     1     1  N11551     2
##  6  2013     1     1  N12540     2
##  7  2013     1     1  N12567     2
##  8  2013     1     1  N13123     2
##  9  2013     1     1  N13538     3
## 10  2013     1     1  N13566     3
## # ... with 64,918 more rows
```

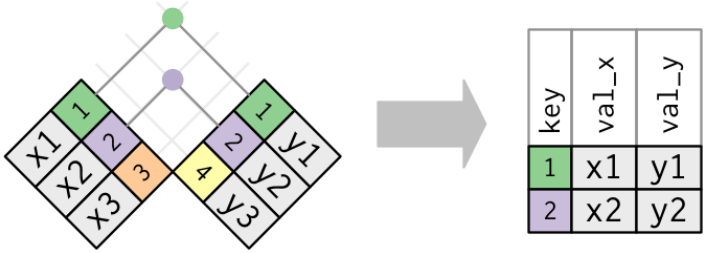
Joins



Joins

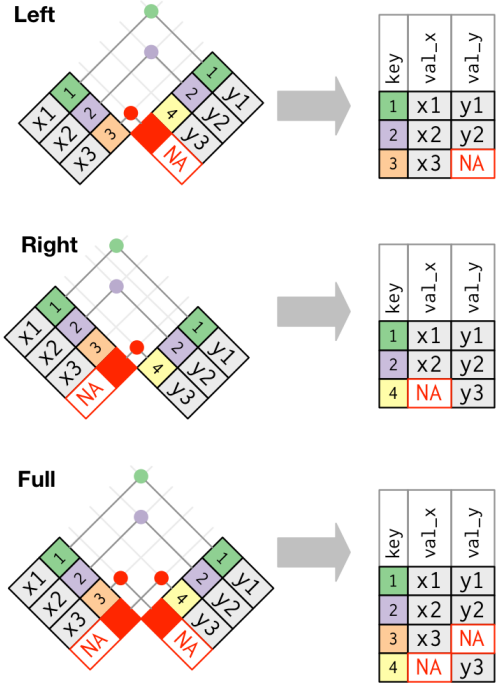


Inner joins



key	val_x	val_y
1	x1	y1
2	x2	y2

Outer joins


Left
x1
x2
x3
y1
y2
y3
NA
key val_x val_y
1 x1 y1
2 x2 y2
3 x3 NA
Right
x1
x2
x3
y1
y2
y3
NA
key val_x val_y
1 x1 y1
2 x2 y2
4 NA y3
Full
x1
x2
x3
y1
y2
y3
NA
NA
key val_x val_y
1 x1 y1
2 x2 y2
3 x3 NA
4 NA y3

left_join()

`flights_small`

```
## # A tibble: 2 x 6
##    year month   day dep_time dep_delay carrier
##   <int> <int> <int>    <int>     <dbl>   <chr>
## 1  2013     1     1      517         2      UA
## 2  2013     1     1      533         4      UA
```

`airlines`

```
## # A tibble: 4 x 2
##   carrier                   name
##     <chr>                  <chr>
## 1      9E      Endeavor Air Inc.
## 2      AA American Airlines Inc.
## 3      AS   Alaska Airlines Inc.
## 4      B6        JetBlue Airways
```

left_join()

```
left_join(flights_small, airlines, by = "carrier")
```

```
## # A tibble: 336,776 x 7
##     year month   day dep_time dep_delay carrier                    name
##    <int> <int> <int>    <int>     <dbl>   <chr>                   <chr>
##  1  2013     1     1      517         2      UA    United Air Lines Inc.
##  2  2013     1     1      533         4      UA    United Air Lines Inc.
##  3  2013     1     1      542         2      AA   American Airlines Inc.
##  4  2013     1     1      544        -1      B6          JetBlue Airways
##  5  2013     1     1      554        -6      DL     Delta Air Lines Inc.
##  6  2013     1     1      554        -4      UA    United Air Lines Inc.
##  7  2013     1     1      555        -5      B6          JetBlue Airways
##  8  2013     1     1      557        -3      EV ExpressJet Airlines Inc.
##  9  2013     1     1      557        -3      B6          JetBlue Airways
## 10  2013     1     1      558        -2      AA   American Airlines Inc.
## # ... with 336,766 more rows
```

Joins

Mutating joins

- `left_join()` = beholder alle observationer i x
- `right_join()` = beholder alle observationer i y
- `full_join()` = beholder alle observationer, som findes i x ELLER y
- `inner_join()` = beholder alle observationer, som findes i x OG y

Filtering joins

- `semi_join()` = beholder kun de observationer i x, som også findes i y
- `anti_join()` = dropper de observationer i x, som har et match i y

Set operations

Set opreations

- `intersect()` = beholder kun de observationer, som findes i x OG y
- `union()` = beholder alle observationer, som findes i x ELLER y
- `setdiff()` = beholder kun de observationer i x, som ikke findes i y

3. Workshop

Opgaver

- Find opgaverne i `03_opgaver.pdf` på GitHub

4. Opsamling og næste gang

Pointer fra i dag

- tibbles og deres fordele
- data import og opmærksomhedspunkter
- `tidyr`: `gather()`, `spread()`, `separate()` og `unite()`
- relationelle data og keys
- mutating joins, filtering joins og set operations

Vigtigt: Inden næste gang!

- Lave en bruger på GitHub
- Installere GitHub Desktop

Næste gang

- Indhold:
  - R workshop III: Programmering & Git
- Pensum:
  - R4DS: kap 17 - 19 + 21 om pipes, funktioner og loops
  - van Strien (2016) om version control
- DataCamp:
  - Intermediate R
- Supplerende:
  - R4DS: kap 20