

# Political Data Science

## Lektion 1: Introduktion

Undervist af Jesper Svejgaard, foråret 2018  
Institut for Statskundskab, Københavns Universitet  
[github.com/jespersvejgaard/PDS](https://github.com/jespersvejgaard/PDS)

Hej & velkommen!

# I dag

1. Præsentation
2. Faget Political Data Science
3. Formalia
4. Praktisk
5. Introduktion til R
6. Opgaver
7. Vigtigste pointer fra i dag
8. Næste gang

# 1. Præsentation

# Hvem er jeg?

- Jeg er **Jesper Svejgaard**, cand.scient.pol fra Institut for Statskundskab, KU
- ... og jeg har **glædet** mig!
- Tidligere ansat i Den Sociale Kapitalfond, Finansministeriet, m.fl.
- Nu ekstern lektor, selvstændig... og technical advisor?!



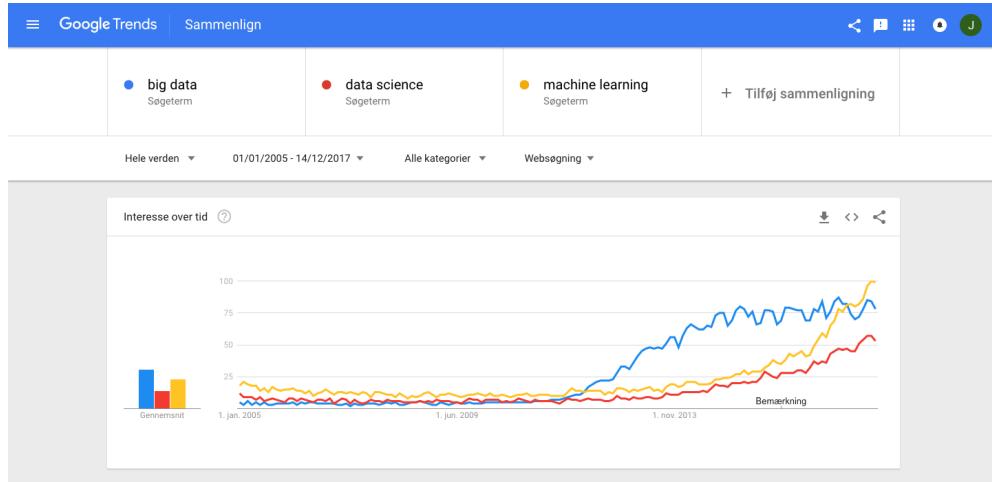
# Hvem er I?

- Navn
- Erfaring med R
- Random fact om jer selv :)

## 2. Faget Political Data Science

# Fagets titel & relevans

Et tørt fag med en smart titel? **Ikke kun!**



Data Scientist: The Sexiest Job of the 21st Century, jf. [Harvard Business Review](#)

Popsmart eller ej, det er en **trend** og en **efterspørgsel**, vi gerne vil tappe ind i!

# Et håndværksfag

- Værktøjer & selve håndværket
- Vi skal have olie på fingrene - i hver lektion!
- Man bliver aldrig "færdig"
- Min rolle: At **facilitere** jeres læring

# 3. Praktisk

# Fagets hjemmeside

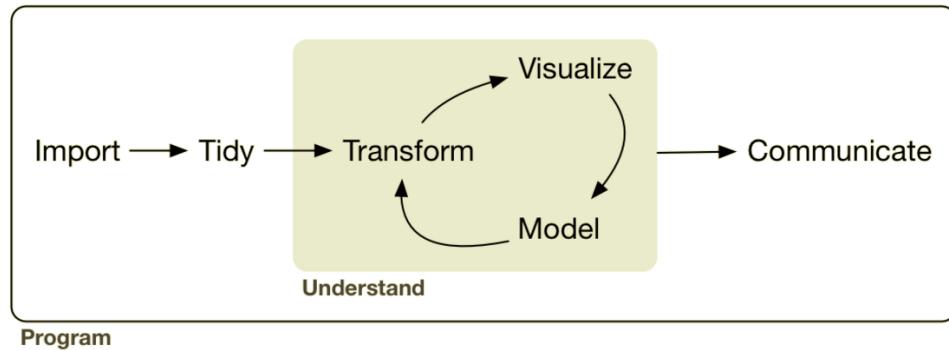
- Faget har en hjemmeside, <https://github.com/jespersvejgaard/PDS>, som ligger på [GitHub](#).
- Slides, data, scripts, opgaver mm.
- Fagets autoritative lektionsplan

# Software mm.

- R
- R Studio
- DataCamp
- Er der nogen, der ikke har fået det installeret og oprettet?

# Fagets opbygning og indhold

Lidt som Hadley Wickham's data-proces:



# Faglige forudsætninger

- Kun metodiske
- Derfor: Antagelse om 0 kendskab til R
- Overlap og gentagelse for nogen
- **Mit tip:** Se det som en kærkommen mulighed for repetition af the basics!

# DataCamp

- Havde været pensum hvis muligt
- Dækker væsentlige dele af pensum, dog ikke 1:1
- Ikke obligatorisk, men aldeles nyttigt - man kan ikke læse dig til at blive datamagiker!
- Gratis ifm. akademia (så benyt nuligheden!)
- Data science track
- Challenge: **Bobler** til den mest ihærdige på holdet!

# 4. Formalia

# Fagets pensum

- Læseplanen på [GitHub](#) er den autoritative
- Obligatorisk vs. DataCamp vs. supplerende litteratur
- Lærebøger, artikler og online-materiale - alt tilgængeligt online og gratis!
- Ca. 900 sider
- Justeringer undervejs forventes

# Målbeskrivelse

Kursets målsætning er at klæde den studerende på til at kunne:

- Importere, håndtere, transformere og visualisere data i R
- Forklare væsensforskellene mellem kausalestimation og prædiktion
- Formulere og designe et prædiktionsproblem
- Analyseret selvvalgt politologisk emne ved at anvende fagets metoder
- Reflektere over fordele og ulemper ved fagets metoder

# Eksamenskrift

Formelle krav:

- krav om mindst 75 % tilstedeværelse i undervisningen
- seminaropgave på 10 - 20 normalsider
- skrives individuelt
- må ikke være en tidligere afleveret opgave
- afleveringsfrist: TBC

Typer af eksamensopgaver:

- fri opgave
- replikationsstudium
- specialeforstudium

Bedømmelseskriterier:

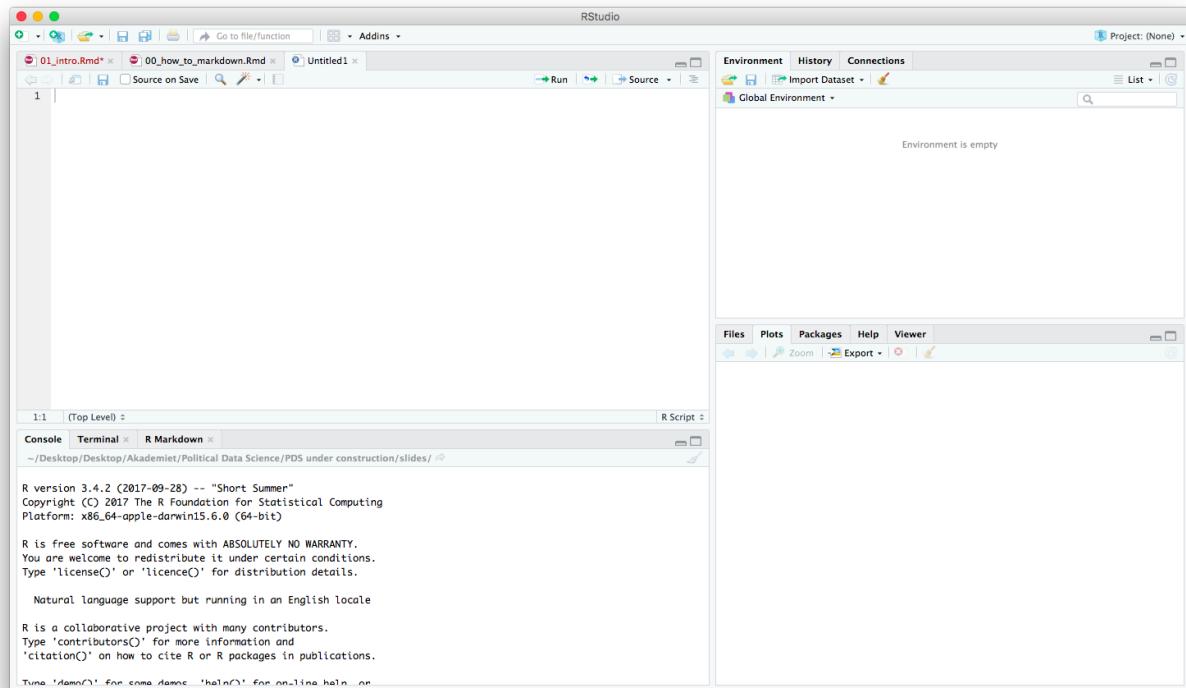
- ikke nitty-gritty referering af pensum
- fokus på anvendelse af fagets metoder til besvarelse af en politologisk problemstilling

# 5. Introduktion til R

# Hvad er R?

- R er et programmerings-sprog, som er særligt godt til statistik og visualisering
- R er open-source (nice!)
- Vi arbejder i R Studio, som er en IDE, et , dvs. et program.

# Sådan "ser R ud"



# Programmering i R

- Hadley Wickham

# Når koden driller

(og det gør den)

- Skriv ? foran koden, fx ?sum
- R Studio markerer fejl i syntaks

```
20 # importerer data fra sti
✖ 21 turnout <- import(sti)✖
22
23 unexpected token ')'
```

- R Studio giver fejlmeddelser

```
> ggplot(data = seat, aes(x = year, y = RV_andel)) +
+   geom_bar(stat = "identity")
Error in ggplot(data = seat, aes(x = year, y = RV_andel)) :
  object 'seat' not found
> |
```

- Google + Stackoverflow =

# Googling Stackoverflow



# Installation og indlæsning af pakker

- Essentiel del af det at arbejde med R
- Pakker indeholder funktioner, dokumentation og sample data

```
## Installerer pakken tidyverse  
install.packages("tidyverse")
```

```
## Loader pakken tidyverse  
library(tidyverse)
```

# R kan regne

```
## Plus og minus
```

```
5 + 6 - 4
```

```
## [1] 7
```

```
## Gange og divider
```

```
(4 + 6 + 2) / 2
```

```
## [1] 6
```

```
## Eksponenter mm.
```

```
2^2 / sqrt(16)
```

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

# R er et objekt-orienteret sprog

```
## Definerer x og y
x <- 5
y <- 6
```

```
## Printer x
x
```

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

```
## Er x og y det samme?
x == y
```

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

# Klasser af objekter

```
## Definerer objekter af forskellige klasser
x <- "Political Data Science"
z <- c(1, 2, 3, 5, 7, 11)
q <- c(2, 3, 5, "æble", 7, "pære")
p <- TRUE
```

Hvilke klasser har objekterne her?

# Klasser af objekter

```
# x <- "Political Data Science"  
class(x)
```

```
## [1] "character"
```

```
# z <- c(1, 2, 3, 5, 7, 11)  
class(z)
```

```
## [1] "numeric"
```

```
# q <- c(2, 3, 5, "æble", 7, "pære")  
class(q)
```

```
## [1] "character"
```

```
# p <- TRUE  
class(p)
```

```
## [1] "logical"
```

# Vektorer

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}_{n \times 1}$$

$$y = (y_1 \quad y_2 \quad \cdots \quad y_m)_{1 \times m}$$

# Vektorer i R

- Vektorer kan kun indeholde elementer af den samme klasse
- Vi gemmer vektorer med funktionen `c()`, der står for concatenate/combine

```
## Gemmer en vektor med tallene fra 1 til 10
x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
z <- 1:10
y <- seq(from = 1, to = 10, by = 1)

## Ganger vektoren x med 2
x * 2
```

```
## [1]  2  4  6  8 10 12 14 16 18 20
```

```
## Lægger tre vektorer sammen
x + z + y
```

```
## [1]  3  6  9 12 15 18 21 24 27 30
```

# Matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}_{n \times m} = (a_{ij})_{n \times m}$$

# Data frames

- Data frame = data i den form, som I kender fra Excel, hvor der er rækker og kolonner
- I R kan man gemme og arbejde med flere data frames på samme tid
- Data frames kan indeholde elementer af forskellig type ( $\neq$  matricer)
- Kolonnerne har kolonnenavne ( $\neq$  matricer)
- Data siges at være **tidy**, når variablene er i kolonnerne og enhederne i rækkerne

ID	Fødeår	Navn
1	1952	Mogens
2	1958	Margrethe
3	1949	Magnus
...	...	...

# Data frames

```
## Laver en df
df <- data.frame(
  "ID" = c(1, 2, 3, 4),
  "Navn" = c("Mogens", "Margrethe", "Magnus", "Magda"),
  "Fødeår" = c(1953, 1958, 1949, 1944),
  "Øjenfarve" = c("Blå", "Brun", "Grøn", "Brun"))
```

```
## Tjekker df
df
```

```
##   ID      Navn Fødeår Øjenfarve
## 1  1     Mogens  1953      Blå
## 2  2 Margrethe  1958      Brun
## 3  3     Magnus  1949     Grøn
## 4  4     Magda  1944      Brun
```

```
## Tjekker klassen af df
class(df)
```

```
## [1] "data.frame"
```

# Selektion af værdier, rækker og kolonner

Vi kan både bruge component selectoren `$` og matriceangivelse `[r,k]` der er en del af base R

```
df$Øjenfarve
```

```
## [1] Blå  Brun Grøn Brun  
## Levels: Blå Brun Grøn
```

```
df[2,2]
```

```
## [1] Margrethe  
## Levels: Magda Magnus Margrethe Mogens
```

```
df[2, ]
```

```
##   ID      Navn Fødeår Øjenfarve  
## 2  2 Margrethe    1958      Brun
```

```
df[,2]
```

```
## [1] Mogens Margrethe Magnus Magda  
## Levels: Magda Magnus Margrethe Mogens
```

# Selektion af værdier, rækker og kolonner I

```
print(df)
```

```
##   ID      Navn Fødeår Øjenfarve
## 1 1    Mogens  1953     Blå
## 2 2 Margrethe 1958     Brun
## 3 3    Magnus 1949    Grøn
## 4 4     Magda 1944     Brun
```

Hvad giver de følgende udtryk?

```
df[1,3]
df[4, ]
df[,2]
```

# Selektion af værdier, rækker og kolonner II

Hvad giver de følgende udtryk?

```
df[1,3]
```

```
## [1] 1953
```

```
df[4, ]
```

```
##   ID Navn Fødeår Øjenfarve
## 4  4 Magda    1944      Brun
```

```
df[,2]
```

```
## [1] Mogens Margrethe Magnus Magda
## Levels: Magda Magnus Margrethe Mogens
```

# Operationer I

```
sum(x)
```

```
## [1] 55
```

```
sum(x)/length(x)
```

```
## [1] 5.5
```

```
mean(x)
```

```
## [1] 5.5
```

```
sd(x)
```

```
## [1] 3.02765
```

# Operationer II

## logiske operatorer

- Lig med: ==
- Større end: >
- Mindre end: <
- Større end eller lig med: >=
- Mindre end eller lig med: <=
- Forskellig fra: !=

# Indlæsning af data I

Når man arbejder i R, arbejder man altid ud fra en sti (et directory)

```
# Hvilken sti?  
getwd()
```

```
# Endre wd  
setwd("/Users/jespersvejgaard/Desktop/PDS")
```

# Indlæsning af data II

Der findes forskellige pakker til import af forskellige typer data:

- `readxl` (Excel)
- `haven` (Stata, SPSS, SAS)
- `readr` (flade filer såsom .txt, .csv m.fl.)
- `rio` (spiser det hele)

Base R har funktioner som `read.csv` og `read.table`

Go-to pakken i tidyverse er `readr`

# Indlæsning af data III

```
# Definerer sti (online)
sti_online <- "https://raw.githubusercontent.com/jespersvejgaard/PDS/master/data/seats.csv"
sti_lokal <- "/Users/jespersvejgaard/Desktop/PDS/data/seats.csv"
```

```
# Indlæser data
df <- read_csv(sti_online)
```

```
# Tjekker df ud
glimpse(df)
```

```
## Observations: 23
## Variables: 22
## $ year <int> 1953, 1957, 1960, 1964, 1966, 1968, 1971, 1973, 1975, 19...
## $ s <int> 74, 70, 76, 76, 69, 62, 70, 46, 53, 65, 68, 59, 56, 54, ...
## $ rv <int> 14, 14, 11, 10, 13, 27, 27, 20, 13, 6, 10, 9, 10, 11, 10...
## $ k <int> 30, 30, 32, 36, 34, 37, 31, 16, 10, 15, 22, 26, 42, 38, ...
## $ cd <int> NA, NA, NA, NA, NA, NA, 14, 4, 11, 6, 15, 8, 9, 9, 9...
## $ rfb <int> 6, 9, 0, 0, 0, 0, 5, 0, 6, 5, 0, 0, 0, NA, 0, NA, NA, ...
## $ sf <int> NA, NA, 11, 10, 20, 11, 17, 11, 9, 7, 11, 21, 21, 27, 24...
## $ dkp <int> 8, 6, 0, 0, 0, 0, 6, 7, 7, 0, 0, 0, 0, 0, NA, NA, ...
## $ df <int> NA, ...
## $ fk <int> NA, 4, 0...
## $ lc <int> NA, NA, NA, NA, 4, 0, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ kd <int> NA, NA, NA, NA, NA, 0, 7, 9, 6, 5, 4, 5, 4, 4, 4, 0, ...
## $ sp <int> 1, 1, 1, 0, NA, 0, 0, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ u <int> 0, 0, 6, 5, 0, 0, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ v <int> 42, 45, 38, 38, 35, 34, 30, 22, 42, 21, 22, 20, 22, 19, ...
```

# Base R og the tidyverse

- Base R indeholder grundlæggende funktionalitet såsom `df[2,3]` og `df$ID`
- Tidyverse er en samling af pakker til R, som deler logik, og som man nærmest bruger i alle analyser
- Centrale pakker er bl.a. `dplyr`, `ggplot2`, `tidyR`, `readr` og `magrittr`
- PDS er skruet sammen omkring R4DS og the tidyverse

# 6. Opgaver

# Opgaver

1. Lav en vektor med årstallene 2007, 2011, 2015 [`x <- c(...)`]
2. Lav en vektor med LA's mandater i de 3 valgår, henholdsvis 5, 9, 13
3. Kombiner de to vektorer til en dataframe [`data.frame()`]
4. Hvor mange mandater fik LA i gennemsnit ved de tre valg? [`mean()`]
5. Hvilket directory arbejder du fra? [`getwd()`]
6. Ændr dit directory til selvvalgt directory [`setwd()`]
7. Installér og load pakken "tidyverse"
8. Indlæs datasættet `seats.csv` fra  
<https://raw.githubusercontent.com/jespersvejgaard/PDS/master/data/seats.csv>
9. Hvor mange observationer er der i datasættet? Hvor mange variable?
10. Hvilke år dækker datasættet?
11. Hvor mange stemmer har S fået i gennemsnit?
12. Hvad med DF? ... Hvorfor virker `mean()` ikke? [`mean(... , na.rm = TRUE)`]
13. Hvor stor en andel af manderne har RV fået ved valgene?
14. Hvad er standardafvigelsen i K's antal mandater?
15. Hvilke år har K fået flere end 30 mandater?

# 7. Vigtigste pointer fra i dag

# Vigtigste pointer fra i dag

- Introduktion til kurset
- R som et objektorienteret sprog
- Vektorer, matricer og dataframes
- Beregninger i R
- Operationer i R
- Indlæsning af data i R

# 8. Næste gang

# Næste gang

- Indhold:
  - R workshop I: Explore
- Pensum:
  - R4DS kap 2 - 6
  - CS: Transformation
- DataCamp:
  - Introduction to the Tidyverse
- Supplerende:
  - Zhang (2017)
  - Wickham (2014)
  - Risdal (2016)
- Sørg for at være med i pensum inden lektionen :)
- Hvis I har nogle spørgsmål, sving mig en mail
- Jeg modtager meget gerne feedback og kommentarerer på undervisningen