

Introduktion til R I

6. februar 2023

Kristian G. Kjellmann (kgk@socsci.aau.dk)
&
Rolf L. Lund (rolfll@socsci.aau.dk)

Institut for Sociologi og Socialt Arbejde



AALBORG UNIVERSITY
DENMARK

Dagens program



1. Hvad er R?
2. Brug af RStudio
3. R-sproget
4. Objekter i R
5. Funktioner i R
6. Indlæsning af funktioner fra andre pakker
7. Introduktion til tabeldata (dataframes)
8. Udforskning af tabeldata i R
9. Deskriptive mål i R

Dagens læringsmål



I kan arbejde med R i RStudio

I kan arbejde med simple objekter og funktioner i R

I kan skrive et simpelt R script

I ved hvad tabeldata er

I kan indlæse et tabeldatasæt i R

I kan udforske tabeldata som data frames i R

Hvem er vi?



Rolf

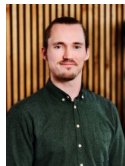


Lektor ved sociologi

Roder med forskellige kvantitative metoder

Brugt R siden 2021

Kristian



Ph.d. studerende ved sociologi

Roder med forskellige kvantitative metoder

Brugt R siden 2016

Hvad er R?



R er et gratis analyseprogram med sit eget kommandosprog/kodesprog.

Programmet egner sig især til kvantitative analyser og visualiseringer af kvantitative data.

R kan arbejde med mange forskellige dataformater. Da programmet er “open source”, findes ufatteligt mange udvidelser til programmet, der tilføjer funktioner.



Hvad er RStudio?



R i sig selv er meget begrænset. RStudio tilføjer en brugerflade ovenpå R, der gør det rarere at arbejde med.

Man arbejder typisk i RStudio, når man bruger R.

Fordele ved RStudio

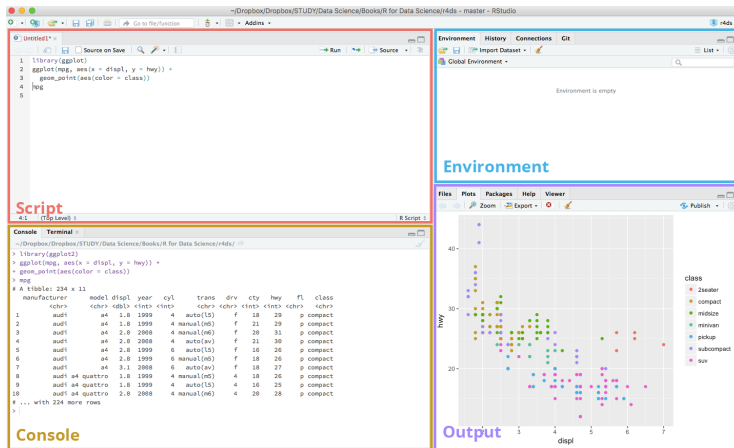
Giver overblik over projektfiler

Hjælper til at tilrette og fuldende kommandoer i script

Overblik over aktivt miljø

Hjælper med at importere data

Hvad er RStudio?

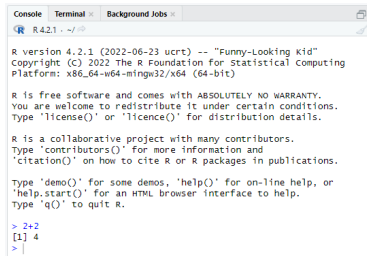


Source: Cecilia Lee

R fungerer ved at man skriver kommandoer i R sproget, som R derefter “fortolker”.

“Fortolkning” i R er blot et spørgsmål om R forstår, hvad du forsøger at gøre. Hvis R forstår det du beder om, gør R den ting. Hvis R ikke forstår det, får man en fejl.

Al fortolkning i R foregår gennem R's konsol. I konsollen skriver man enten R kode direkte eller også sender man kode dertil fra et script (en tekstfil med R-kode).



```
R version 4.2.1 (2022-06-23 ucrt) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 2+2
[1] 4
> |
```


R som en veltrænet hund

Naiv

Forstår kommandoer sagt på en bestemt måde

Gør hvad du beder om

Kan lære nye ting



Operator	Function
+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Exponentiation
%%	Modulo
/%/%	Integer Division

Script-filer er tekst filer med kode, som R kan forstå (“fortolke”).

En script-fil kan forstås som en “analyseopskrift”, der indeholder alle kommandoer nødvendige for at foretage en analyse. Det tillader også, at man nemt kan køre kommandoer igen.

Man bør altid skrive de kommandoer, som er nødvendige for at lave analysen, ind i et script.

Brug konsolvinduet til at finde frem til den rigtige kommando.

kan bruges til kommentarer (ignoreres når man kører koden)

BEMÆRK! Der er ingen fortryd-knap i R! Når koden er kørt, er ændringen sket. Den eneste måde at “fortryde” er ved at genskabe det, som man har lavet, ved at køre tidligere kode igen. Netop derfor er scripts vigtige.

R fungerer ved at lagre værdier og information i “objekter”. Disse objekter kan derefter bruges i forskellige funktioner. Funktioner kan være alt fra at udregne et gennemsnit, lave en figur, gemme et datasæt osv.

Forsimpelt sagt: Et objekt er en eller anden lagret information, mens en funktion er noget, som kan bearbejde eller gøre noget ved informationen i et objekt.

At arbejde med R involverer kontinuerligt at definere objekter. Objekter er blot et navn til at kalde lagret information frem igen.

Objekter kan være mange ting:

- et ord
- et tal
- en talrække
- et datasæt
- en matematisk formel
- et resultat
- en filsti
- en graf
- og så videre...

Når et objekt er defineret, er det tilgængeligt i det aktuelle “miljø”. I R skal et miljø forstås som en samling af objekter og funktioner, som er til rådighed (defineret, “native” eller indlæst).



bolden er tallet 42

`bolden <- 42`



```
print(bold)
```

```
Error in print(bold): object 'bold' not found
```



```
print(bolden)
```

```
[1] 42
```


Funktioner er kommandoer brugt til at transformere objekter på en eller anden måde og give et output.

Det, som man sætter i funktionen, kaldes et “argument” eller “input”. Antallet af argumenter varierer mellem funktioner.

Funktioner har alle den samme opbygning:

`funktionsnavn(arg1, arg2, arg3)`. (funktionsnavn med argumenterne i parentes adskilt med kommaer).

Nogle argumenter er krævede, mens andre er valgfrie.

Funktion:

Mikrobølgeovn

Krævet argument:

Det der skal tilberedes

Valgfrie argumenter:

Mikrobølgeovnens indstillinger



```
microwave(squash, watts = 750, time = 2)
```

Funktioner ændrer aldrig et objekt. Når man bruger en funktion, beder man R om at se et output, men ikke om at ændre noget (hvordan ser min squash ud, når den har været i mikroovnen?).

Hvis man vil ændre et objekt, skal man derfor lagre outputtet i et nyt eller eksisterende objekt (jeg vil gerne erstatte min squash med den, som har været i mikroovnen).

R adskiller mellem objekter via deres “class”.

En “class” er R’s måde at holde styr på, hvilken type af information, som objektet indeholder.

En class sætter samtidig betingelserne for, hvad der kan lade sig gøre med objektet (fx at vi kan lave udregninger med tal, men at vi ikke kan det med tekst).

Classes (datatype)



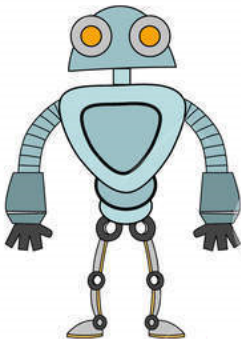
Fordi R er “open source”, bliver der konstant tilføjet nye funktioner til R. Funktioner, som andre har lavet, kan læses ind via “R pakker”, som kan gøres til del af ens “R bibliotek”.

R adskiller mellem installation og indlæsning. Dette for at undgå konflikter mellem pakker.

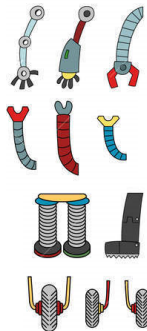
Man behøver kun at installere pakker én gang.

Pakker, som bruges i et script, indlæses i starten af scriptet.

Basis R



R pakker



Tidyverse er en samling af pakker til R, der letter arbejdet med at indlæse, håndtere og arbejde med data.

Pakkerne fra Tidyverse har den fordel, at de alle følger den samme designfilosofi og opbygning i deres funktioner.

Vi vil i disse R introduktioner primært bruge funktioner fra tidyverse til data- og variabelhåndtering.

Alle pakker fra tidyverse kan installeres og indlæses på én gang:

```
install.packages('tidyverse')
```

```
library(tidyverse)
```


R kan arbejde med data på mange forskellige måder.

For at R kan arbejde med data, skal data indlæses i en af R's *datastrukturer*.

Datasæt kan eksistere i mange forskellige former. Måden vi indlæser data i R, skal derfor hænge sammen med den form og det format, som data har “uden for R”.

Ofte vil datasæt findes i en eller anden form for *tabelformat*.

Data i tabelformat er struktureret i rækker og kolonner.

Rækker vil typisk bestå af *observationer* (én række per person, land, dyreart eller hvadend datasæt indeholder information om).

Kolonner vil typisk afspejle *informationer vedrørende observationerne* (fx fødselsår, kommune man bor i, beskæftigelse).

Hver celle indeholder en specifik information for en specifik observation (fødselsår for en person).

Indlæsning af data - tabeldata



idno	yrbrn	gndr	eduyrs	wkhtot	health
5816	1974	Male	35	37	Good
7251	1975	Female	13	34	Fair
7887	1958	Male	25	39	Fair
9607	1964	Female	13	34	Good
11688	1952	Female	2	37	Very bad
12355	1963	Male	14	37	Fair

Et udbredt filformat til at gemme data i tabelformat er *csv* (comma-separated values).

CSV-filer er ikke andet end rene tekstfiler, som følger nogle simple formatteringsregler:

- Hver række er en observation

- Hver værdi for en observation er adskilt med et komma (svarende til kolonner)

- Kolonnenavne fremgår af første linje (hvis der er kolonnenavne)

Indlæsning af data - csv-filer



```
idno,yrbrn,gndr,eduyrs,wkhtot,health  
5816,1974,Male,35,37,Good  
7251,1975,Female,13,34,Fair  
7887,1958,Male,25,39,Fair  
9607,1964,Female,13,34,Good  
11688,1952,Female,2,37,Very bad  
12355,1963,Male,14,37,Fair
```

En “data frame” er en datastruktur i R, som bruges til at håndtere data i tabelformat (data i rækker og kolonner).

For R er en data frame bare endnu et objekt af en bestemt type (class).

Fordi data frames blot er objekter, kan man arbejde med så mange data frames ad gangen, som man har lyst til.

Af samme grund er man nødt til at fortælle R, når man vil gøre noget med information, som befinder sig i en bestemt data frame.

Vectors (enkeltvariable)



En basal datastruktur i R er en vector.

En vector er en række af værdier af den samme type (fx en række tal, en række ord osv.).

En vector svarer til en enkelt variabel/kolonne i en dataframe, og man arbejder med vectors på samme måde, som man arbejder med enkelte kolonner (samme funktioner kan bruges).

Vectors dukker op i mange forskellige sammenhænge i R, da de bruges hver gang, at man skal angive en samling af flere værdier.

Vectors (enkeltvariable)



For R er værdier (tal, tekst, osv.) *altid* vectors.

Selv objekter, der blot består af ét tal, behandler R som en vector (en vector bestående af én talværdi).

Derfor virker kommandoer og funktioner i R ens, uanset om de bruges på enkelte tal, enkelte vectors eller enkelte kolonner i en data frame.

Data vil ofte indeholde missing-værdier. Missing-værdier angiver ikke-gyldige værdier; fx et manglende svar, ugyldigt svar, information der ikke kunne skaffes eller lignende.

Missing-værdier bruges til at give en værdi uden at give en værdi (cellerne skal indeholde noget). På den måde er man ikke nødt til at fjerne hele rækker fra datasættet, selvom der mangler oplysninger.

I R angives missing-værdier med `NA` og er hverken høje eller lave.