Introduktion til R II

7. februar 2023

Kristian G. Kjelmann (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 datahåndtering?
2. Subsetting i R
3. Tilføje nye variable i R
4. Rekodning i R
5. Håndtering af missingværdier i R
6. Tips til god kodepraksis

Dagens læringsmål

I ved hvad datahåndtering indebærer

I får kendskab til typiske datahåndteringsudfordringer

I kan foretage relevante datahåndteringsoperationer i R (subsetting, nye variable, rekodning, håndtering af missing)

I kan skrive et R script, der foretager relevant datahåndtering af et datasæt

I kan skrive et R script, som andre kan læse og forstå

Hvad er datahåndtering?

Når man arbejder med data, er man næsten altid nødt til at foretage visse ændringer i data, før at de er egnet til analyse. Dette kaldes "datahåndtering".

Datahåndtering inkluderer blandt andet:

- Udvælge specifikke observationer og variable (kaldes også subsetting)
- Danne nye variables
- Rekode værdier

Subsetting

"Subsetting" henviser generelt til den del af datahåndtering, der har at gøre med at udvælge de relevante dele af data. Hvad der er relevant skal altid ses ift. den analyse, som man foretager.

Subsetting inkluderer blandt andet:

- Begrænse data til specifikke variable/kolonner
- Begrænse data til en bestemt tidsperiode, geografisk område eller andet
- Udvælge observationer i data ud fra bestemte betingelser (filtrering)

Filtrering er den mest centrale del af subsetting, da det er i dette arbejde, at man sørger for, at data stemmer overens med den population, som man undersøger.

Subsetting


CAROUSEL
THE
MINIMUM
HEIGHT
FOR THIS RIDE
IS
130CM
UNLESS
ACCOMPANIED BY
AN
ADULT!

Brug af betingelser i R

Subsetting og særligt filtrering er i sidste ende et spørgsmål om at sætte regler for sine data: hvilke betingelser skal observationer opfylde for at blive inkluderet i analysen?

Samme logik gør sig gældende i måden, som vi filtrerer data i R, da vi i R kan spørge, om data opfylder en bestemt betingelse eller ej.

Brug af betingelser i R - logiske værdier

En lang række kommandoer og funktioner i R vil returnere *logiske værdier*.

Logiske værdier kan kun være `TRUE` eller `FALSE` (eller missing/`NA`).

Bruger man fx disse operatorer i R, returneres altid en logisk værdi:

Operator	Betydning
>	Større end
>=	Større end eller lig med
<	Mindre end
<=	Mindre end eller lig med
==	Lig med
!=	Ikke lig med

Variabelændringer

En væsentlig del af datahåndtering er at foretage forskellige variabelændringer - enten i form af at rekode værdier i en variabel eller ved at danne nye variable, som gør brug af information fra andre variable.

Eksempler på variabelændringer:

- Danne sammensatte mål (fx mål for samlet metal sundhed eller socioøkonomisk status)
- Sammenslå kategorier (fx at behandle 'meget uenig' og 'lidt uenig' som 'uenig')
- Danne kategoriseringer (fx kategorier for alder, indkomst eller uddannelsesniveau)

Variabelændringer

Gode data er meget detaljerede og nuancerede og derfor meget komplekse.

En væsentlig del af at arbejde med kvantitative metoder indebærer at reducere kompleksitet i data på sådan en måde, at der kan foretages meningsfulde konklusioner uden at datas integritet forværres.

Rekodninger

Variabelændringer involverer ofte ændringer af værdierne i variablen. Dette refereres til som *rekodning*.

Vi kan overordnet adskille mellem tre typer af rekodning:

- Simpel rekodning (fx omregning af eksisterende variabel)
- Ændring af kategorier (fx ved at sammenslå flere kategorier til én)
- Omkodning fra numerisk til kategorisk (fx ved at lave variable for kategoriseringer af alder, indkomst eller andet)

Afhængig af rekodningen, kan rekodning enten være *aritmetisk* (baseret på udregning), *manuel* eller baseret på *betingelser*.

Hvornår laver man en ny variabel?

Når man laver datahåndtering, bør man altid bestræbe sig efter ikke at fjerne information fra datasættet.

Lav ny variabel	**Behold eksisterende variabel**
Værdier regnes om	Fejl rettes i variablen (fx stavefejl)
Kategorier slås sammen	Datatype ændres (tal som tekst)
Rekodning fra numerisk til kategorisk	Værdier kodes til missing
Rangorden i kategorier ændres	

Er man i tvivl *bør man altid lave en ny variabel*. På den måde er der ikke information, som går tabt.

Missingværdier

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.

Missing-værdier kan både være givet på forhånd (fx at oplysning mangler) eller være noget, som man selv er nødt til at angive (værdier skal behandles som missing - frasorteres analyse).

Missingværdier - Kod til missing

Der kan være flere grunde til, at man er nødt til at kode værdier til missing:

1. Det er ikke givet at missing-værdier er kodet som missing på forhånd i et datasæt. Hvordan missing-værdier kodes varierer mellem programmer. Derfor bruger man ofte specifikke talværdier (fx 777777 eller 888888) til at indikere missing-værdier.

2. Afhængig af analysen kan man have behov for at ignorere visse observationer, fx for at undgå at outliers skævvrider resultatet.

Begge situationer involverer *rekodning*, hvor man erstatter de pågældende værdier med missing (enten manuelt eller ud fra betingelser).

Missingværdier - Fjern missing

Missing-værdier er ikke gyldige og kan ikke direkte indgå i beregninger. Når det kommer til analyse, håndteres missing typisk på en af overordnede måder:

1. Observationer med missing i relevante variable fjernes (listwise deletion)
2. Missing-værdier estimeres ud fra øvrige data (imputation)

Vi forholder os her i dag kun til at fjerne missing.

Håndtering af datatyper i R

R adskiller mellem objekter via deres “class”.

Enkelte variable(/vectors) kan *kun* indeholde værdier med samme class (class sættes for vector - ikke for enkelte værdi).

Ved tvetydighed (fx variable med både tekst og tal), vil R sætte variablen til den class, som alle værdier kan passe ind i (typisk tekst, da tekst ikke kan konverteres direkte til tal).

Dette problem kan også opstå, hvis der gemmer sig skjulte tegn (fx mellemrum) eller enkelte bogstaver i variablen, når datasæt indlæses.

En simpel løsning på dette er at tvinge variablen om til en anden class (“class coercion”).

Tips til god kodepraksis

God datahåndteringspraksis hænger sammen med god *kodepraksis*.

God kodepraksis er overordnet et spørgsmål om at gøre koden læsbar og forståelig *i sig selv*.

Kort sagt: Hvis en anden R bruger ikke kan gennemskue hvad din kode gør, uden at køre koden, så er det et tegn på dårlig kodepraksis.

Tips til god kodepraksis

1. Brug kommentarer!
2. Brug sigende navne til objekter (fx `_df` for at angive at det er en data frame)
3. Behold kun kode der er relevant for at foretage analysen (fjern fx prints og linjer, der ikke har konsekvens for analysen)
4. Vær konsistent i navngivning af objekter (skift ikke mellem brug af `_`, `.` og brug af versaler)
5. Brug mellemrum og linjeskift for at gøre koden mere overskuelig (men vær konsistent!)

Tips til god kod kodepraksis - Kommentarer

God kode er kode med *masser* af kommentarer!

Kommentarer kan bruges til:

- Give kodeblokke overskrifter
- Skrive forklaringer
- Notere kritiske valg i koden (hvorfor fx et specifikt argument i funktion bruges)
- Skrive relevante henvisninger ind
- Skrive ting ind at være opmærksom på (fx hvis noget tager lang tid at køre)

Er du i tvivl? Skriv en kommentar!