Helle Sørensen (MATH, KU)
Efterår, 2018

Opgaver til Statistisk Dataanalyse 1

Opgave HS.1 (At arbejde med R og Studio)

Sørg for at have de **nyeste versioner** af R og RStudio installeret. Det er beskrevet på siden *Generel info* på Absalon hvorfra programmerne kan downloades.

1. Åbn RStudio. Gå til konsollen (Console, venstre eller nederste venste vindue) og skriv et par kommandoer efter prompten (>), fx dem nedenfor. Skriv kommandoerne en ad gangen og tryk *Enter* efter hver af dem.

   ```
   6*12
   sqrt(49)
   2^3
   x <- 7
   x + 7
   ```

   Bemærk at et nyt objekt, `x`, nu optræder i *Environment*-vinduet (øverste højre vindue) og kan bruges til nye beregninger.

Kommandoer skrevet ved prompten bliver ikke gemt til senere brug! Det er fint at bruge prompten når du „leger“ og til kommandoer der ikke skal bruges igen, men som hovedregel skal du lave dit arbejde i et R-program eller en Markdown-fil, som du bliver introduceret til senere.

Et R-program (engelsk: R script) er en fil med R-kommandoer, og filnavnet ender som regel med `.R`.

2. Åbn et nyt R-program via filmenuen: *File* → *New File* → *R script*
3. Skriv en kommando i filen, fx noget i stil med kommandoerne ovenfor. Klik *Run* eller tast *Ctrl* + *Enter*. Det sidste anbefales! Læg mærke til at kommandoen bliver kopieret til prompten og udført med det samme.
4. Skriv en ny kommando i en ny linie i R-programmet hvor du laver en variabel **y** med værdien 21.
5. Skriv en hashtag (#) før en af kommandoerne, og kør den igen. Der sker ingenting! Du kan derfor bruge hashtags til at skrive kommentarer til dig selv i dine R-programmer.
6. Lav en mappe på din computer som du kan bruge til R-arbejde på kurset. Gem R-programmet i mappen vha. filmenuen.
7. Luk RStudio ned. Vælg „Nej“ når R spørger om du vil gemme dit *workspace image*.
8. Åbn RStudio igen. Dit program fra før vil formentlig være åbent i editoren, Ellers kan du åbne det via filmenuen.

   Bemærk at *Environment*-vinduet er tomt: Dit **y** fra før er slettet, men du har jo gemt kommandoen, så du kan nemt lave det igen.

9. Prøv følgende kommandoer:

```
bp <- c(96,119,119,108,126,128,110,105,94)
bp
mean(bp)
```

   Dette er data fra opgave 1.2 i lærebogen.

Gem dine filer ofte. Det hænder — heldigvis sjældent — at du beder R om noget der er så mærkeligt, at programmet lukker ned, og det er ærgerligt at have mistet flere timers arbejde.

Opgave HS.2 (Installation af pakker, datasæt fra pakker)

Basisinstallationen af R indeholder et væld af funktioner og datasæt, men der findes også R-pakker med ekstra funktioner og datasæt. Nogle pakker er faktisk del af standardinstallationen, mens andre skal installeres separat.

R-pakker skal installeres og loades før de kan bruges; datasæt skal desuden loades før de kan bruges. Du kan se hvilke pakker der er installeret og loadet hvis du klikker på *Packages* i nederste højre vindue. Hvis pakken optræder på listen, så er den installeret, Hvis den desuden et „hakket af", så er den også loadet.

Der hører en R-pakke, *isdals*, til lærebogen; pakken indeholder de fleste datasæt fra bogen. Denne opgave går ud på at installere den og ganske kort se på et datasæt fra pakken.

1. Installér *isdals*-pakken, fx på følgende måde: Klik *Tools* → *Install packages*, og vælg CRAN i øverste boks, skriv `isdals` i midterste boks og tryk OK. Pakken bliver nu installeret og optræder i listen over pakker; check gerne *Packages*-menuen.

2. Load pakken med kommandoen

   ```
   library(isdals)
   ```

   Alternativt kan du klikke til venstre for pakkens navn i listen over pakker. Pakken er nu „hakket af" i listen, hvilket betyder at du har adgang til funktioner og data fra pakken.

3. Pakken indeholder blandt andet et datasæt der hedder **ricestraw**. Dette datasæt kan loades med kommandoen

   ```
   data(ricestraw)
   ```

   Bemærk at datasættet nu optræder i *Environments* (øverste højre vindue). Det betyder at du nu kan bruge datasættet. Prøv kommandoen

   ```
   ricestraw
   ```

Bemærk at en pakke kun skal installeres en gang, mens pakken skal loades i hver R-session hvor du skal bruge datasæt eller funktioner fra pakken. Datasættet skal tilsvarende loades i hver session hvor du skal bruge det.

Opgave HS.3 (Datasæt og variable)

Vi skal arbejde meget med R-datasæt i løbet af kurset, så det er vigtigt at forstå strukturen af dem og kunne bruge dem. Denne opgave bruger datasættet **ricestraw** i *isdals*-pakken. Se opgave HS.2 for installation og brug af pakken.

Datasæt er organiseret så rækker svarer til observationer (eller datalinier) og søjler svarer til variable — ganske som du ville organisere dine data i et Excelark.

1. Brug kommandoen `ricestraw`. Hvor mange observationer/datalinier er der i datasættet? Hvilke variable er der i datasættet? Prøv kommandoen `?ricestraw`, og læs i hjælpefilen hvad variablene beskriver.
2. Prøv kommandoen `weight`, og læs fejlmeddelelsen.
3. R kender ikke variablen `weight`. Det er fordi du ikke har fortalt R at skal finde den i datasættet. Prøv i stedet kommandoen `ricestraw$weight`. Man kan altså bruge $ hvis man skal have fat i en variabel i et datasæt.
4. Prøv så kommandoen `hist(weight)` — som ikke virker. Justér kommandoen så du får tegnet et histogram.
5. Funktionen `with` er et nyttigt alternativt til `$`-syntaksen. Prøv kommandoen

   ```
   with(ricestraw, hist(weight))
   ```

   Syntaksen er altså `(navn-på-datasæt, kommando)`, og effekten er at den givne kommando udføres med brug af det givne datasæt.
6. Lav et scatterplot (punktplot) hvor `time` er på x-aksen og `weight` er y-aksen. Du kan enten bruge $-syntaksen eller `with`.
7. Mange funktioner har et `data`-argument, som angiver hvilket datasæt R kan trække variable fra. Man kan fx skrive

   ```
   lm(weight ~ time, data=ricestraw)
   ```

   Prøv kommandoen. Hvordan skal tallene i outputtet fortolkes? Frembring samme output på to andre måder: dels vha. `$`-syntaksen og dels vha. `with`.

Bogen bruger ofte funktionen `attach`. Ideen er at man med en `attach` kommando tillader R at kigge i datasættet uden at specificere det direkte. Det virker smart, men er i virkeligheden lidt farligt: Hvis man har flere variable med samme navn, er det meget vanskeligt at holde styr på hvilken en man bruger. Derfor anbefaler jeg at du ikke bruger `attach`.

Opgave HS.4 (Omskrivning af opgave 3.2)

1. Læs teksten i starten af opgave 3.2 fra lærebogen. Forklar hvorfor datasættet lægger op til en ensidet ANOVA hvis man vil undersøge om behandlingerne virker.
2. Data ligger som datasættet **tartar** i *isdals*-pakken. Load datasættet som forklaret i opgave HS.2, og skriv datasættet ud på skærmen (bare for at se at du har adgang til det).
3. Brug kommandoen

```
boxplot(index ~ treat, data=tartar)
```

Forklar hvad figuren viser.

4. Brug kommandoen

```
lm(index ~ treat -1, data=tartar)
```

Sammenlign outputtet med tabellen i opgaveteksten i bogen. Hvad angiver tallene?

5. Brug kommandoen

```
lm(index ~ treat, data=tartar)
```

Kan du gennemskue hvad tallene angiver?