Statistisk Dataanalyse 2 - Case 2 - 2017

Anders Tolver

14 Sep 2017

Case 2: Reproducérbare statistiske analyser

Motivation

Når man i praksis arbejder med statistik som værktøj vil der være en stor (tidsmæssig) gevinst ved at forsøge at effektivisere sine arbejdsprocesser. Videnskabelig forskning foregår oftest som et samarbejde mellem flere personer med vidt forskellige kompetencer, og arbejdet med indsamling af data, rensning af data, analyse af data og præsentation af resultater strækker sig typisk over måneder eller år. For en god og effektiv arbejdsproces er det hensigtsmæssigt, hvis det er let at

- genkalde sig, hvad man lavede for måneder siden
- kommunikere resultater på tværs af personer, som er involveret i projektet
- overlevere dele af (eller eventuelt hele) projektet med dataanalysen til andre personer
- lave analyserne om igen, hvis man ønsker at ændre noget på figurer eller i den statistiske analysemetode
- lave analyser eller figurer om, hvis man opdater fejl i data på et sent tidspunkt i arbejdsprocessen

Formålet med denne case er at give jer mulighed for at reflektere over ovenstående punkter og derved træne jeres evne til at lave reproducerbare statistiske analyser. Samtidig regner vi på et datasæt, hvor du får mulighed for at træne din evne til at lave flersidet variansanalyse.

Opbygning af case 2

Arbejdet med dagens case består af tre faser

- **Fase 1:** På baggrund af en beskrivelse af en videnskabelig problemstilling og et tilhørende datasæt, skal I forsøge at skrive et R-program med tilstrækkelig mange kommentarer til, at programmet kan læses, forstås og køres af en anden person end jer selv. Der er en række bundne ting, som skal være med i jeres R-program (-se nedenfor). Den reelle udfordring ved denne del af opgaven er, at I ikke vil have de konkrete data til rådighed.
- **Fase 2:** Til denne del af casen får I udleveret et test-datasæt, der har samme struktur som det *rigtige* datasæt. Tanken er at test-datasættet kan bruges til at teste jeres R-program fra **Fase 1** og rette oplagte fejl. Datafilen **case2test.txt** gøres automatisk tilgængelig i Absalon kl. 13:45.
- **Fase 3:** Til den sidste del af casen får I udleveret de rigtige data, og formålet er at køre den statistiske analyse ved hjælp af det R-program, I har udviklet i løbet af de to første faser. Dette arbejde indebærer blandt andet, at kommentarerne i R-programmet skal opdateres, så man får præsenteret resultaterne svarende til en statistisk analyse af de rigtige data. Der er to ekstra udfordringer ved denne fase af casen, som først afsløres i forbindelse med udleveringen af de rigtige data. Datafilen **case2full.txt** gøres automatisk tilgængelig i Absalon kl. 14:30.

Formatet af løsningen

Der er som udgangspunkt to mulige formater for, hvordan I kan løse casen (-se nedenfor)

- I kan lave R-programmet som et kommenteret R-script (ikke anbefalet løsning)
- I kan lave et fuldautomatisk dokument i R-markdown (anbefalet løsning)

Kommenteret R-script

Opgaven kan løses ved at lave et R-script, hvor man via kommentarer skriver, hvilke analyser der skal laves samt hvilke konklusioner der skal drages på baggrund af analyserne.

Eksempel på arbejdsgang ved denne løsning

- Åben et nyt R-script
- Gem filen som "case2besvar.R"
- Skriv kommentarer
- Skriv R-kommandoer
- Forklar via kommentarer hvilke spørgsmål der skal besvares vha. R-kommandoerne du har kørt

Strukturen af R-scriptet kunne være som følger

```
### besvarelse af case 2

### indlæsning af data til case 2

data <- read.table(file = "case2full.txt", header = T)
dim(data)
### datasættet indeholder ??? observationer og ??? variable
names(data)
### datasættet indeholder en faktor 'var' (erstat 'var' med rigtigt variabelnavn også i R-kode)
table(data$var)
### faktoren 'var' er balanceret / ikke- balanceret og har ??? niveauer
```

R markdown

R-markdown formatet er per konstruktion mere velegnet til at lave et samlet dokument med kommentarer, R-kode og R-output, som let kan strikkes sammen (klik på "Knit HTML") til et noget flottere layout i form af et HTML-dokument.

Den primære forskel i forhold til R-koden i den grå boks ovenfor er, at man nu blot skriver alle kommentarlinjer (der starter med ### i eksemplet ovenfor) som almindelig tekst, og indlejrer (eng: embed'er) R-koden i såkaldte *code chunks*.

For at komme igang med denne løsning skal du gøre følgende.

- Installere R-pakken "R markdown" på din computer
- Åbne et "R Markdown" dokument fra den relevante menu i R-studio (-og slette irrelevante kommentarer i den skabelon, der dukker op)
- Gemme filen som "case2besvar.Rmd"
- Du kan f.x. indsætte code chunks (=grå bokse til R-kode) ved at vælge "Code -> Insert Chunk" i R-studio
- Klik på "Knit HTML" for at strikke dokumentet sammen til en html-fil, der indeholder både R-kode og output
- Gå tilbage i R-markdown filen og supplér med de kommentarer, der er nødvendige for at kunne forstå R-koden og R-output.

Hvis du (stadig) har problemer med at åbne en R markown fil så er den letteste løsning utvivlsomt, at du geninstallerer den nyeste version af R (og opdaterer til en tilstrækkelig ny version af R studio). Husk i den forbindelse at (gen-)installere relevante R-pakker herunder **R markdown** pakken.

Fase 1: ca. 13:00-13:45

Data til denne opgave består af målinger af omgangstider på løbeture i Parc Montsouris beliggende i den sydlige del af Paris. Formålet med dataindsamlingen var at undersøge, om der er fysiologisk belæg for at hævde, at nogle mennesker ikke er egnede til at løbe om morgenen. Derfor er der for hver tidsmåling registreret, om løbeturen fandt sted om morgenen (før kl. 9) eller senere på dagen (efter kl. 9).

I datasættet findes desuden oplysninger om, hvorvidt der blev løbet med høj eller lav intensitet på hver enkelt omgang. Intensiteten er defineret ud fra målinger af den gennemsnitlige puls (-og selve pulsmålingen er ikke registreret i datasættet). Endelig er der en formodning om, at der kan være en udmatningseffekt, som har indflydelse på omgangstiderne. Derfor er der for hver omgangstid i datasættet også angivet, hvor mange omgange der tidligere var løbet på samme løbetur.

Jeres endelige produkt skal så vidt muligt indeholde punkterne 1.-6. nedenfor.

1. Forsøgets formål

Skriv en ultra kort tekst (kopier evt. noget fra opgaveformuleringen).

2. Data

En beskrivelse af datastrukturen. Hvilke variable (med tilhørende navne) indgår i data, hvilke variable er numeriske og hvilke er faktorer. Lav desuden mindst en tabel til opsummering af forsøgsdesignet, f.x. ved at undersøge/optælle hvor mange observationer der er for hver værdi af udvalgte variable.

Nyttige R-kommandoer kan være

```
summary(var1)
table(var2)
table(var2,var3)
```

hvor *var1, var2 ,var3* refererer til variable i datasættet.

3. Figur

Jeres løsning skal også indeholde R-kode til at lave mindst en figur i tilknytning til datasættet. Der er intet krav til layout af figuren, og den behøver heller ikke være meget relevant i forhold til den problemstilling som datasættet skal belyse.

Nyttige R-kommandoer kan være

```
hist(var1)
plot(var1)
boxplot(var1 ~ var2)
```

4. Statistisk model

Her skal du forklare, hvilken statistisk model du gerne vil bruge til at analysere data. Vælg imellem en ensidet variansanalyse, en tosidet variansanalyse eller en tresidet variansanalayse. Du skal opskrive/forklare hvilken model du bruger, og du skal opskrive R-kode til at fitte modellen ved hjælp af lm-funktionen i R. De modige studerende bør vælge en model, der indeholder mindst to forklarende variable (faktorer), men hvis du føler dig udfordret nok på andre fronter i den pågældende case, så kan du blot vælge en ensidet variansanalysemodel.

Nyttige R-kommandoer kan være

```
lm(var1 ~ var2)
lm(var1 ~ var2 * var3)
lm(var1 ~ var2 + var3)
```

Den ambitiøse studerende kan supplere med R-kode til at lave grafisk modelkontrol (jf. forelæsning d. 14/9-2017). Husk også at lave nogle kommentarer som viser, hvad du har tænkt dig at konkludere på baggrund af figuren/figurerne.

5. Test af hypotese

Opstil på baggrund af din model under 4. en reduceret model (hypotese), og angiv R-koden til at fitte modellen med lm-funktionen.

Lav desuden et test af hypotesen i R.

Ambitiøse studerende kan lave en hel række af succesive modelreduktioner.

Nyttige R-kommandoer kan være

```
m0 <- lm(var1 ~ var2)
m1 <- lm(var1 ~ 1)
anova(m1, m0)
```

6. Konklusion

Skriv et oplæg til en konklusion på den statistiske analyse, hvor detaljerne kan fyldes ud, når R-programmet er kørt på de rigtige data.

Afsnittet bør indeholde kommentarer omkring:

- hvilken statistisk model der bedst beskriver data
- parameterestimater fra den endelige statistiske model herunder en forklaring (i ord) af, hvad de forskellige estimater udtrykker, og hvordan de skal fortolkes.

Nogle nyttige R-kommandoer

```
### en af følgende
m0 <- lm(var1 ~ var2)
m0 <- lm(var1 ~ var2-1)
m0 <- lm(var1 ~ relevel(var2, ref = '1')) ### erstat '1' med passende værdi for 'var2'
### ... efterfulgt af
summary(m0)
confint(m0)
```

Fase 2: ca. 13:45-14:30

Indlæs test-datasættet **case2test.txt** via Absalon. Datasættets variabelnavne og typen af variable (numeriske eller faktorer) er nøjagtig som i det rigtige datasæt, der vil blive udleveret senere.

Brug tiden på at køre R-kommandoerne i dit R-program på test-datasættet, og ret eventuelle fejl, så du er sikker på, at programmet kan køres.

Ret desuden i dine kommentarer til R-programmet, så du er helt sikker på, hvilke resultater der skal kommenteres på, når du senere kommer til at køre analysen på de rigtige data.

Fase 3: ca. 14:30-15:00

Hent det rigtige datasæt **case2full.txt** i Absalon og kør dit R-program.

Hvis du arbejder i et R-script skal du opdatere kommentarerne i dokumentet, så det bliver klart, hvilke konklusioner du drager på baggrund af den statistiske analyse. Hvis du arbejder i R-markdown skal du køre dit R-program ved at klikke på "Knit HTML". Herefter kan du kigge på html-filen og gå tilbage i din R-markdown fil med henblik på at opdatere dokumentet med de kommentarer der skal til, for at det endelig produkt kommer til at indeholde en diskussion af resultaterne fra den statistiske analyse.

Opsummering: ca. 15:15-16

Efter en kort pause samler vi op på jeres erfaringer.

- Hvad fungerede godt?
- Hvad var den største udfordring?
- Hvilke ting kunne fungere endnu bedre, og er der tricks som I mangler for at kunne gøre tingene endnu mere reproducerbare?

I bedste fald tager vi et par *frivillige* og kigger på deres endelige produkt.

Nogle nyttige links

Hvis du allerede har vænnet dig til at arbejde i R markdown, så kan du prøve at gøre lidt ud af, at få lavet din automatisk generede rapport i et format / layout, der er til at holde ud at kigge på. Søg f.eks. inspiration i R Markdown Cheat Sheet.

Prøv at eksperimentere med forskellige **Chuck Options**, der har betydning for, hvordan R kode og output bliver inkluderet, når du *knitr* dit R markdown dokument sammen. Se evt. R Markdown på StatData2 og prøv at eksperimentere med at skrive følgende options i starten af dine **Code Chunks**

- `{r echo = FALSE}`
- `{r echo = TRUE}`
- `{r include = FALSE}`
- `{r eval = FALSE}`
- `{r results = "hide"}`