R i 02402: Introduktion til Statistik

Per Bruun Brockhoff
DTU Informatik, DK-2800 Lyngby

17. januar 2012

Indhold

1 Anvendelse af R på Databar-systemet på DTU

1.1 Adgang

En beskrivelse af databar systemet på DTU kan findes på `http://www.gbar.dtu.dk`. Adgang til G-baren forudsætter et login (studienummer) og et password, hvilket alle studerende ved DTU får udleveret ved optagelse. Login foretages

- via en tynd klient (terminal på DTU)
- via login over internet - benyt ThinLinc

Se `http://www.gbar.dtu.dk/introguide/introguide_dk.pdf`) for en nærmere beskrivelse af dette.

1.2 R

Programmet R er et open source statistikprogram, der i vid udstrækning er en kopi af det kommercielle program Splus. Det introduceres kort i Appendix C i lærebogen til 02402. Det kører på gbaren, men er nemt og hurtigt at hente ned til sin egen computer, hvad enten man bruger Windows, Mac eller Linux: `http://www.r-project.org`. Det anbefales at hente programmet til sin egen computer, idet man jo IKKE har adgang til gbaren ved eksamen, hvor man godt må have programmet med på sin egen labtop. Der findes samme sted adskillige tilgængelige introduktioner til programmet.
Programmet fungerer i sin grundform vha. af et kommando-vindue (R Console), hvor man ved prompten kan køre kommandoer/funktionskald. I gbaren kan programmet således køre "interaktivt"alene via sådan en konsol. I windows versionen er konsollen automatisk pakket ind i en brugergrænseflade, hvor konsollen i opstarten er det eneste aktive vindue, men med nogle forskellige overordnede menuer. Denne version af at køre programmet findes ligeledes på gbaren.
Man kan med fordel altid starte en script-editor ('File'→'New Script') inden for R, hvorfra det er nemt at 'submitte' kommandoer samtidig med, at man ikke "mister dem igen".
I kurset 02441 Anvendt statistik og statistisk programmel, som kører som et 3-ugers kursus i Januar hvert år får man mulighed for at arbejde videre med brugen af programmet R til praktisk statistisk dataanalyse - en direkte projekt-orienteret overbygning på introduktionskurset 02402, se `http://www.imm.dtu.dk/courses/02441`), hvor man også kan finde henvisninger til godt (online) lærebogsmateriale. Også i den nye version af kurset 02418 (erstatter det tidligere 02416) får man videreudviklet sine R-kompetencer ift. at lave mere avancerede statistiske dataanalyser.

1.2.1 R Commander

På trods af at R således kører via menuer, er der i denne grundform IKKE egentlige menuer til at udføre selve de statistiske analyser, som kendes fra de fleste standard kommercielle statistikprogrammer. Vi vil derfor i tillæg til dette bruge en menubaseret overbygning til at lave statistik i R. Der findes forskellige af sådanne overbygninger, vi bruger pakken ("package") "Rcmdr"

også kaldet "R Commander". Denne er klar til brug i databarens R version - for at starte: ved prompten skriv:

```
> library(Rcmdr)
```

Man kan nemt installere "Rcmdr" på sin egen computer også ved inden for programmet R at gøre følgende: (internetforbindelse forudsættes)

1. Klik 'Packages' →'Install Packages(s)'
2. Vælg "Mirror Sit" - f.eks. "Denmark"
3. Find og vælg pakken "Rcmdr" på listen - så pakken bliver "installere"(dvs. kopieret til computeren)
4. Derefter, kør: library(Rcmdr) ved prompten. Man "loader" programmet på denne vis. (Dette skal skrives hver gang man starter R op for også at starte R Commander) (Første gang spørger den sandsynligvis til nogle pakker, der mangler - man svarer blot ja til at den skal installere det nødvendige)

NB, vedrørende installationen og første load R Commander: For visse platforme er der forskellige detaljer, der måske kan drille lidt: For Windows 7 og Vista: Hvis du installerer R, som computeren nok vil foreslå, under "Pogramme Files", så skal du køre R "som administrator" for at kunne installere pakker. Enten højreklik på R ved opstart og vælg dette ELLER vælg ved installeringen at installere R helt uden for dette, f.eks. blot direkte i roden. For Mac-brugere er der et par ekstra udfordringer: Man skal sikre sig at "X Windows" er tilgængelig OG at "Tcl/Tk for X Windows" er installeret inden man går i gang. Uanset platform, se følgende hjemmeside for detaljer og for Mac-brugere direkte links til det nødvendige:
(`http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/installation-notes.html`).

Bemærk, at R Commander giver et ekstra program-vindue, der inkluderer hhv. en script-editor og et outputvindue. Når man laver grafik vil graferne poppe op i seperate vinduer - evt. inden for det oprindelige R-vindue. Bemærk også det ret væsentligt: Man får for alle menubaserede valg ikke blot resultatet af disse valg, men også de R-scripts, som dette svarer til. Man kan således nemt springe imellem script-baseret og menubaseret brug af programmet.

1.2.2 Import af data

Data importeres til R ved f.eks. at bruge "read.table" funktionen, se Appendiks C. Eller via menuerne i R Commander: 'Data' →'Import Data'.
Forskellige valg er filformat er mulige - default setting for "text-files" virker umiddelbart for de .TXT-filer, som hører til bogen (www.pearsonhughered.com/datasets) eller mere direkte via (såfremt browseren kører på en maskine på DTU)
`http://www.imm.dtu.dk/courses/02402/Bookdata8ED`
(7ed: `http://www.imm.dtu.dk/courses/02402/Bookdata`). Ved import af data fra et regneark kan det nogen gange være en fordel at eksportere data fra regnearket til komma separeret (csv) format inden data importeres til R. (Der findes pakker, der bedre og mere direkte kan håndtere f.eks. Excell-filer, men det springes over her (pakken `RODBC`))

1.2.3 Brug af programmet

1.2.4 Lagring af tekst og grafik

Tekst fra vinduer i programmet R kan kopieres til andre programmer på sædvanlig vis:

- Marker teksten ved at holde venstre tast på musen nede og træk pointeren over den ønskede tekst.
- Skift til det andet program (f.eks. StarOffice), placer pointeren det ønskede sted og tryk på musens midterste tast.

Alt tekst i 'Commands Window' eller 'Report Window' kan gemmes i en tekstfil ved at gøre vinduet aktivt og vælge 'File' $\rightarrow$ 'Save As ...'.
Grafik kan gemmes i en grafikfil ved at gøre grafikvinduet aktivt og vælge 'File' $\rightarrow$ 'Save As ...'. Der er mulighed for at vælge blandt en række grafik formater (JPEG er default).

2 R i 02402

2.1 Pensum

R indgår i kursets pensum svarende til de afsnit i denne note, der i forelæsningsplanen/pensumlisten henvises til som "grundig" læsning ("g"). Denne pensumsdel er det, der tjekkes i "Testopgaverne" under hvert hovedafsnit, som er stillet som en del af øvelserne i løbet kursusforløbet. Disse opgaver, og således også eksamen, kræver IKKE at man sidder og har adgang til programmet, MEN kræver en forståelse af forskellige aspekter af det som programmet producerer. Man vil typisk kun kunne opnå denne forståelse/ dette kendskab til programmet, såfremt man i løbet af kurset træner sig lidt i selve programmet. Der bliver ved kursusstart arrangeret en decideret computerøvelse, hvor man således under vejledning kan arbejde med de stillede opgaver. Derudover må man selv arbejde med det. Installerer man R på sin egen labtop, kan man med fordel lade programmet erstatte lommeregneren i kurset, hvilket man så kan udnytte til eksamen. Ud over de stillede testopgaver, så er der i noten her en anvisning til hvilke af øvelsesopgaverne man kan løse med programmet. Det skal understreges, at programmets evne til at beregne ting for brugeren IKKE betyder at forståelsen for detaljerne i beregningerne kan glemmes - forståelsen er en vigtig del af pensum, som jo ligger i alle de sider i lærebogen, som udgør pensum.

2.2 Introducerende R-øvelse

Man kan, som beskrevet, bruge R på to forskellige måder: 1) Som et Menubaseret dataanalyseprogram, 2) Som en interaktiv regnemaskine med en lang række indbyggede statistiske funktioner og procedurer. Vi skal i denne øvelse mest bruge metode 1), MEN det anbefales også at prøve metode 2), hvor det foreslås! (Skulle de tekniske vanskeligheder ved installeringen og loading af R Commander pakken, se ovenfor, blive uoverstigelige på selve dagen, så er man stadig godt kørende ift. resten af kurset, hvis man gennemfører øvelsen alene baseret på metode 2).)

1. Start R

2. Download via Campusnet fildeling i kursus 02402 Excell-filen: karakterer2004.csv, der indholder 10 kolonner (variable) og 1555 observationer (rækker), som svarer til 1555 skoler:

Nummer	Variabelnavn	Forklaring (mundtlige karakterer sommer 2004)
Variabel 1	Skole	Skole-navn
Variabel 2	Type	Skoletype
Variabel 3	Type2	Skoletype
Variabel 4	Amt	Amts-navn
Variabel 5	Kommune	Kommune-navn
Variabel 6	Dansk.Eks	Dansk 9. kl. eksamensgennemsnit for skolen
Variabel 7	Dansk.Aars	Dansk 9. kl. årskaraktersgennemsnit for skolen
Variabel 8	Mat.Eks	Matematik 9. kl. eksamensgennemsnit for skolen
Variabel 9	Mat.Aars	Matematik 9. kl. årskaraktereksamensgennemsnit for skolen
Variabel 10	Antal	Antal elever i den pågældende årgang på skolen

3. Importer datamaterialet til R: Data,Import Data, From text file... I menuvinduet, der popper op, vælg (selv) et (R-)navn for datasættet, ændr "Field Separator" til "Other" og "Specify"semikolontegnet: ";". Og som "Decimal-Point Character" vælg "Comma" - klik OK til sidst.

4. Se på rådata:("View Data Set"). Kan du finde din egen skole?

5. Udfyld følgende skema over summary skole-statistics: (Vi kigger på skole-tallene UDEN at tage hensyn til at der er forskelligt antal elever på skolerne)

   - Enten: Brug menupunktet "Statistics", "Summaries", og så enten "Actice data set" eller "Numerical Summaries...". Marker relevante variable, klik OK.
   - Eller: Brug funktionerne listet side 529 i Appendix C. (I så fald husk først at skrive **attach(karakterer2004)**)

	Dansk.Eks	Dansk.Aars	Mat.Eks	Mat.Aars
Gennemsnit				
Median				
Varians				
Spredning				
øvre kvartil Q_3				
Nedre kvartil Q_1				

6. Hvilken "historie" foræller dette?

7. Sammenlign med histogrammerne for hver af de fire fordelinger.
   - Enten: Brug menupunktet Graph, Histogram
   - Eller: Brug funktionen **hist()**
8. Lav boxplots for hver af de fire fordelinger.
   - Enten: Brug menupunktet Graph, Boxplot
   - Eller: Brug funktionen **boxplot()**
9. Prøv at visualisere antallet af skoler af hver type. (Bar graph og/eller pie graph) (Menupunkt Graph)
10. Prøv at sammenligne Matematik eksamenskarakterfordelingerne for skoletyper (variabel: Type).(Brug menupunktet Graph, Boxplot, vælg Type som "plot by groups" variabel)
11. Prøv at undersøge om der er sammenhæng mellem karaktererne! (Graph, scatterplot, vælg x og y-variable)
12. Prøv selv andre plot-metoder - f.eks. "scatterplot matrix", prøv forskellige options, f.eks. "identify outliers with mouse" i boxplot
13. Prøv at ændre noget i de givne scripts og kør dem igen....

3 Diskrete fordelinger, uge 2

3.1 Beskrivelse

Kommandoer skrives ud for prompten ">".
Kommandoen `3:7`, genererer de hele tal fra 3 til 7 i en vektor og `7:3` genererer dem i omvendt rækkefølge:

```
> 3:7
[1] 3 4 5 6 7
> 7:3
[1] 7 6 5 4 3
```

Kommandoen `prod(x)` multiplicerer alle tallene i vektoren `x`:[1]

```
> prod(2:3)
[1] 6
```

Der betragtes følgende fordelinger:

R	Betegnelse
`binom`	Binomialfordelingen
`pois`	Poissonfordelingen

[1] Tilsvarende adderer `sum(x)` alle tallene i vektoren `x`

Den hypergeometriske fordeling findes også i R (`hyper`) men har en anden form (parametrisering) end i lærebogen og betragtes ikke i denne øvelse. Som beskrevet i lærebogens appendiks C findes der for hver fordeling 4 funktioner i R, hvis navne fremkommer ved at tilføje et af 4 bogstaver til navnet i tabellen:

`d` Tæthedsfunktion $f(x)$ (probability distribution).

`p` Fordelingsfunktion $F(x)$ (cumulative distribution function).

`r` Tilfældige tal fra den anførte fordeling. (Bruges og uddybes i Kapitel 9 i noten)

`q` Fraktil (quantile) i fordeling.

3.1.1 Binomialfordelingen:

- $b(x; n, p)$ på side 86 (7ed: 107) i lærebogen fås i R som `dbinom(x, n, p)`.
- $B(x; n, p)$ på side 87 (7ed: 107) i lærebogen fås i R som `pbinom(x, n, p)`.

3.1.2 Poissonfordelingen:

- $f(x; \lambda)$ på side 104 (7ed: 127) i lærebogen fås i R som `dpois(x, lambda)`.
- $F(x; \lambda)$ på side 105 (7ed: 128) i lærebogen fås i R som `ppois(x, lambda)`.

3.2 Træning vha. ugens øvelsesopgaver

- Løs opgave 4.15 både vha. `dbinom` og `pbinom`.
- Løs opgave 4.19 både vha. `dbinom` og `pbinom`.
- Løs opgave 4.57 vha. R.
- Løs opgave 4.59 vha. R. Prøv at bruge både `dpois` og `ppois` ved løsning af spørgsmål (a).
- Løs evt. de supperende opgaver 4.2,4.16 og 4.21 vha. R.

3.3 Testopgaver

3.3.1 Opgave

Lad X betegne en stokastisk variabel. R-kommandoen `dbinom(4,10,0.6)` køres med resultatet 0.1114767. Fremover vil dette vises som følger:

```
> dbinom(4,10,0.6)
[1] 0.1114767
```

Hvilken fordeling anvendes og hvad angiver tallet 0.1114767?

3.3.2 Opgave

Lad X betegne den stokastiske variabel fra før. Fra R fås to resultater:

```
> pbinom(4,10,0.6)
[1] 0.1662386
> pbinom(5,10,0.6)
[1] 0.3668967
```

Angiv sandsynlighederne $P(X \leq 5)$, $P(X < 5)$, $P(X > 4)$ og $P(X = 5)$.

3.3.3 Opgave

Lad X betegne en stokastisk variabel. Fra R fås :

```
> dpois(4,3)
[1] 0.1680314
```

Hvilken fordeling anvendes og hvad angiver tallet 0.1680314?

3.3.4 Opgave

Lad X betegne den stokastiske variabel fra før. Fra R fås to resultater:

```
> ppois(4,3)
[1] 0.8152632
> ppois(5,3)
[1] 0.916082
```

Angiv sandsynlighederne $P(X \leq 5)$, $P(X < 5)$, $P(X > 4)$ og $P(X = 5)$.

4 Kontinuerte fordelinger, normalfordelingen, uge 3

4.1 Beskrivelse

Orienter dig i starten af lærebogens appendiks C, specielt 'Probability Distributions' og 'Normal Probability Calculations' på side 530 (7ed: 611) (dog ikke `qqnorm`). Nedenfor er en række fordelinger listet:

R	Betegnelse
`norm`	Normalfordelingen
`unif`	Den uniforme fordeling
`lnorm`	Log-normalfordelingen
`exp`	Exponentialfordelingen

Som beskrevet i lærebogens appendiks C findes der for hver fordeling 4 funktioner i R, hvis navne fremkommer ved at tilføje et af 4 bogstaver til navnet i tabellen:

`d` Tæthedsfunktion $f(x)$ (probability density function).

`p` Fordelingsfunktion $F(x)$ (cumulative distribution function).

`r` Tilfældige tal fra den anførte fordeling.(Bruges og uddybes i Kapitel 9 i noten)

`q` Fraktil (quantile) i fordelingen.

4.1.1 Normalfordelingen:

- $f(x;\mu,\sigma^2)$ på side 125 (7ed: 154) i lærebogen fås i R som `dnorm(x, μ, σ)`.
- Fordelingsfunktionen for en normalfordeling med middel μ og varians σ^2 fås som `pnorm(x, μ, σ)`. Dvs. $F(z)$ på side 126 (7ed: 154) fås som `pnorm(z, 0, 1)`[2]
- Antag at Z er en standard normalfordelt stokastisk variabel. Den værdi af z for hvilken $P(Z \leq z) = p$ fås som `qnorm(p)`. Denne værdi kaldes p-fraktilen i den standardiserede normalfordeling.

Bemærk at R bruger σ og ikke σ^2.

4.2 Træning vha. ugens øvelsesopgaver

- Løs opgave 5.19 vha. `pnorm`.
- Løs opgave 5.21 vha. `qnorm`.
- Løs opgave 5.27 vha. R.
- Løs opgave 5.33 vha. R.
- Løs opgave 5.114 vha. R. (7Ed: 5.113)

4.3 Testopgaver

4.3.1 Opgave

Følgende 3 R kommandoer og resultater haves:

```
> pnorm(2)
[1] 0.9772499
> pnorm(2,1,1)
[1] 0.8413447
> pnorm(2,1,2)
[1] 0.6914625
```

Angiv hvilke fordelinger og sandsynligheder, der er tale om i hvert tilfælde. (Gerne ved en skitse)

4.3.2 Opgave

Hvad bliver resultatet af R kommandoen `qnorm(pnorm(2))`?

[2] Eller blot `pnorm(z)` idet R som default bruger den standardiserede normalfordeling.

4.3.3 Opgave

Følgende 2 R kommandoer og resultater haves:

```
> qnorm(0.975)
[1] 1.959964
> qnorm(0.975,1,1)
[1] 2.959964
> qnorm(0.975,1,2)
[1] 4.919928
```

Angiv hvilke tal, der er tale om i hvert tilfælde. (Brug gerne skitse)

5 Kontinuerte fordelinger, uge 4

5.1 Beskrivelse

Orienter dig i starten af lærebogens appendiks C, specielt 'Probability Distributions' og 'Normal Probability Calculations' på side 530 (7ed: 611).

5.1.1 Log-normal-fordelingen:

- $f(x)$ nederst side 136 (7ed: 166) i lærebogen fås i R som `dlnorm(x, α, β)`.
- Sandsynligheden i eksemplet side 137 (7ed: 167) i lærebogen fås i R som `plnorm(8.2,2,0.1)-plnorm(6.1,2,0.1)`.
- Samme sandsynlighed fås i R ligeledes som `pnorm(log(8.2),2,0.1)-pnorm(log(6.1),2,0.1)`.
- Og endelig som `pnorm((log(8.2)-2)/0.1)-pnorm((log(6.1)-2)/0.1)`.
- Bemærk, at i R hedder den naturlige logaritmefunktion `log`
- Bemærk også, at den beregnede sandsynlighed i lærebogen er en smule anderledes. Det skyldes, at man i bogen afrunder de tal, der indsættes i standardnormal-funktionen inden disse slås op i tabellen. Den i R beregnede sandsynlighed er således mere korrekt end den i bogen angivne.

5.1.2 Den uniforme fordeling:

- $f(x)$ på side 135 (7ed: 165) i lærebogen fås i R som `dunif(x, α, β)`.

5.1.3 Eksponentialfordelingen:

- $f(x)$ side 140 (7ed: 170) i lærebogen fås ligeledes i R som `dexp(x, 1/β)`.

5.1.4 Normalfordelingsplot

Som beskrevet i appendix C side 530 (7ed: 611) kan man bruge R-funktionen `qqnorm`. Den anvender en metode, der er en smule anderledes end den i bogen beskrevne. I opgavebessvarelsen til opgave 5.120 vil man kunne finde en præcis beskrivelse af denne variant af konstruktionen af plottet. R ombytter ligeledes x- og y-aksen i plottet ift. bogen. Har man importeret de data, der bruges i opgave 5.120 (”2-66.TXT”) OG ”attachet” dem, Så opnås plottet simpelt hen ved at skrive `qqnorm(speed)`.

5.2 Træning vha. ugens øvelsesopgaver

- Løs opgave 5.46 vha. `punif`.
- Løs opgave 5.51 vha. `plnorm`.
- Løs opgave 5.58 vha. `pexp`.
- Løs opgave 5.38 vha. `pnorm`.
- Løs opgave 5.111 vha. `punif`. (7Ed: 5.110)
- Løs opgave 5.120 vha. `qqnorm`. (7Ed: 5.119)

5.3 Testopgaver

5.3.1 Opgave

Angiv formel for og/eller skitser betydningen af følgende R kommando og resultat:

```
> punif(0.4)
[1] 0.4
```

5.3.2 Opgave

Angiv formel for og/eller skitser betydningen af følgende R kommando og resultat:

```
> dexp(2,0.5)
[1] 0.1839397
> pexp(2,0.5)
[1] 0.6321206
```

5.3.3 Opgave

Angiv formel for og/eller skitser betydningen af følgende R kommando og resultat:

```
> qlnorm(0.5)
[1] 1
```

6 Samplingfordelinger, uge 5 og 8

6.1 Beskrivelse

Orienter dig i starten af lærebogens appendiks C, specielt 'Sampling Distributions' side 530 (7ed: 612). De sampling fordelinger, der introduceres i kapitel 6 i lærebogen er:

R	Betegnelse
`t`	t-fordelingen
`chisq`	χ^2-fordelingen
`f`	F-fordelingen

Som tidligere beskrevet i lærebogens appendiks C findes der helt tilsvarende til alle andre fordelinger i R fire funktioner i R, hvis navne fremkommer ved at tilføje et af 4 bogstaver til navnet i tabellen:

`d` Tæthedsfunktion $f(x)$ (probability density function).

`p` Fordelingsfunktion $F(x)$ (cumulative distribution function).

`r` Tilfældige tal fra den anførte fordeling. (Bruges og uddybes i Kapitel 9 i noten)

`q` Fraktil (quantile) i fordelingen.

6.1.1 t-fordelingen

- Tallene i Tabel 4, side 516 (7ed: 587) i lærebogen er givet ved funktionen `qt(1-`α`,`ν`)` (giver værdierne i tabellen) eller tilsvarende `1-pt(x,`ν`)` (giver α-værdierne), hvor x angiver værdierne i tabellen.
- Sandsynligheden i eksemplet side 188 (7ed: 218) i lærebogen for at ligge under -3.19 kan i R fås direkte som `pt(-3.19,19)` og tilsvarende kan sandsynligheden for at ligge over fås som: `1-pt(3.19,19)`.

6.1.2 χ^2-fordelingen: (uge 8)

- Tallene i Tabel 5, side 517 (7ed: 588) i lærebogen er givet ved funktionen `qchisq(1-`α`,`ν`)` (giver værdierne i tabellen) eller tilsvarende `1-pchisq(x,`ν`)` (giver α-værdierne), hvor x angiver værdierne i tabellen.
- Sandsynligheden i eksemplet side 190 (7ed: 219-220) i lærebogen kan i R fås som `1-pchisq(30.2,19)`

6.1.3 F-fordelingen:(uge 8)

- Tallene i Tabel 6, side 518-519 (7ed: 589-590) i lærebogen er givet ved funktionen `qf(1-`α`,`ν_1`,`ν_2`)` (giver værdierne i tabellen) eller tilsvarende `1-pf(x,`ν_1`,`ν_2`)` (giver α-værdierne), hvor x angiver værdierne i tabellen.

- Sandsynligheden 0.95 i eksemplet side 191 (7ed: 221) i lærebogen kan i R fås som `1-pf(0.36,10,20)` eller som `pf(2.77,20,10)`.
- Man kunne finde værdien i tabellen, der giver de 0.95 ved `qf(1-0.95,10,20)` eller `1/qf(0.95,20,10)`.

6.2 Træning vha. ugens øvelsesopgaver

6.3 Testopgaver

6.3.1 Opgave

Angiv formel for og/eller skitser betydningen af følgende R kommando og resultat:

```
> qt(0.975,17)
[1] 2.109816
> qt(0.975,1000)
[1] 1.962339
```

6.3.2 Opgave

Angiv formel for og/eller skitser betydningen af følgende R kommando og resultat:

```
> pt(2.75,17)
[1] 0.993166
```

7 Hypotese-test og konfidensintervaller for et og to gennemsnit, Kap. 7+8, uge 6-7

7.1 Beskrivelse

Orienter dig i starten af lærebogens appendiks C, specielt 'Confidence Intervals and Tests of Means' på side 531 (7ed: 612). Som beskrevet i appendix C, side 531 (7ed: 612) kan man bruge R-funktionen `t.test` til både et gennemsnit, to gennemsnit samt den parrede situation. Funktionen beregner både hypotese-test og konfidensinterval. Som navnet indikerer, giver dette altså KUN mulighed for at lave test og intervaller baseret på t-fordelingen, IKKE z-test. Dette afspejler, at man som regel er i denne situation i alle virkelige anvendelser af disse ting. Skulle man have tilstrækkelig store n til at Z-test er OK, så fås dette jo automatisk, idet t-test'ene jo så giver resultater, der er stort set lig med z-testene.

Kaldes funktionen med et enkelt sæt af tal, f.eks. som `t.test(x)`, hvor x således indeholder en række tal, vil funktionen automatisk agere som i sektion 7.2 og 7.6 i bogen. Som default vælges to-sidet test og niveau $\alpha = 5\%$. ønsker man et ensidet test og/eller et andet test-niveau anføres dette i kaldet til funktionen, f.eks.: `t.test(x,alt=''greater'',conf.level=0.90)`. Bemærk, at konfidensniveauet $= 1 - \alpha$.

Kaldes funktionen med to sæt af tal, f.eks. som `t.test(x1,x2)`, hvor x1 således indeholder en række tal og x2 en anden række tal, vil funktionen automatisk agere som i sektion 8 i bogen, altså betragte de to sæt af tal som to uafhængige stikprøver. Som default vælges to-sidet test og

niveau $\alpha = 5\%$. ønsker man et ensidet test og/eller et andet test-niveau anføres dette i kaldet til funktionen, f.eks.: `t.test(x1,x2,alt=''less'',conf.level=0.90)`.

Er der tale om to parrede stikprøver, kaldes funktionen på samme måde, MEN der tilføjes en option til kaldet: `t.test(x1,x2,paired=T)`. Dette giver således præcis det samme som at kalde funktionen med det enkelte sæt af tal, der udgøres af differenserne: `t.test(x1-x2)`. Der gælder de samme ting vedr. ensidet/tosidet og test-niveau.

Når funktionen kaldes med et ensidet alternativ (`alt=''greater''` eller `alt=''less''`, så angiver den et andet konfidens-interval end ellers. Dette er et såkaldt ensidet konfidens-interval, som vi IKKE berører i kurset!

7.1.1 One-sample t-test/konfidensinterval

Man kan opnå t-fordelingsversionen af resultaterne i eksemplet nederst side 210 ved at:

1. Importere "C2nanoheight.TXT" (vha. file-menu). Kald det (f.eks.) `nano`.
2. Attach dette data-sæt: `attach(nano)`. (Ved menubaseret analyse behøver man ikke "attache")
3. Brug funktionen: `t.test(height.nm.,conf.level=0.99)`.

Bemærk, at bogen jo angiver "large sample" versionen af konfidensintervallet, som bruger normal-fordelingen i stedet for t-fordelingen. Begge dele kan retfærdiggøres. Bemærk også, at man får et tosidet t-test for hypotesen om at $\mu = 0$ skrevet ud uanset om man har nogen interesse overhovedet i dette test! Det er jo f.eks. IKKE noget man gider kigge på i dette tilfælde!

7.1.2 Two-sample t-test/konfidensinterval

Man kan opnå resultaterne i eksemplet side 254-255 (7ed: 266-267) ved først at:

1. Importere "C8alloy.TXT" (vha. file-menu). Kald det (f.eks.) `C8alloy`.

I dette eksempel er data lagret på en typisk (og fornuftig) måde, der dog vanskeliggør brugen af funktionen `t.test`, som beskrevet i bogen, en anelse: Samtlige $58 + 27 = 85$ strength-værdier for de to alloys ligger i en enkelt variabel: `strength`, samtidig med at der findes en anden variabel `alloy`, der identificerer hver enkelt observation som enten alloy 1 eller alloy 2. Variablen `alloy` indeholder altså 85 1- og 2-taller. Man kan nu konstruere to nye variable `x1` og `x2`, der indeholder hver sit sæt af tal ved:

```
x1=strength[alloy==1]
x2=strength[alloy==2]
```

hvorefter man kan opnå resultaterne side 255 (7ed: 267) ved at kalde funktionen som beskrevet ovenfor: `t.test(x1,x2)`.

Man kan alternativt bruge data som de er og så via menuerne klikke sig frem. Før man kan det, skal man fortælle R, at alloy-variablen er en "gruppe-variabel", altså en variabel, der opdeler materialet i grupper - dette kalder man sædvanligvis en "faktor". Dette gøres enten ved kommandoen: `C8alloy$alloy=factor(C8alloy$alloy)` eller ved i menuerne:

'Data'→'Manage Data in active Data set'→'Convert numeric variables to factors' og vælg `alloy` (bestem selv navnene, f.eks. 1 og 2). Nu kan man så lave den statistiske analyse ved menuerne: 'Statistics'→'Means'→'Independent Samples t-test..'. Bemærk, at man ligeledes kan udføre one-sample beregninger via menuen. (Bemærk, at hvis variable ikke er på numerisk form, blive de automatisk opfattet af R som "faktorer")

Har man først "alloy" som en faktor kan man faktisk også køre `t.test` funktionen direkte på følgende form: `strength alloy,data=C8alloy`, hvor man bruger R's måde at opskrive modeller på: Brugen af "tilde-tegnet" ("~") betyder at strength udtrykkes som en funktion af alloy. (Man vil se i scriptvinduet ved brug af menuerne, at det faktisk er denne funktion, som menuen baserer sig på)

7.1.3 Parret t-test/konfidensinterval:

Ingen yderligere beskrivelse.

7.2 Træning vha. ugens øvelsesopgaver

For de fleste af opgaverne kan man naturligvis bruge fordelingerne, som øvet tidligere, i stedet for at slå op i tabellerne (det være sig z- eller t-fordelingen). For at anvende `t.test` (og/eller menuerne) skal man have rådata tilgængelig - det har man kun i visse af opgaverne:

- Løs opgave 7.61 (Data fra exercise 2.41: Importer "2-41.TXT"). (7Ed: 7.42)
- Løs opgave 7.63 og 7.64 (Data kan nemt indtastes: `x=c(14.5,14.2,14.4,14.3,14.6)`). (7Ed: 7.48 og 7.49)
- Løs opgave 8.21 (Importer "8-21.TXT"). (7Ed: 7.72)
- løs evt. opgave 8.10 og 8.11. (Data kan relativt nemt indtastes). (7Ed: 7.68 og 7.69)

7.3 Testopgaver

7.3.1 Opgave

Følgende R kommandoer og resultat haves:

```
> x=c(10,13,16,19,17,15,20,23,15,16)
> t.test(x,mu=20,conf.level=0.99)

        One-sample t-Test

data:  x
t = -3.1125, df = 9, p-value = 0.0125
alternative hypothesis:  mean is not equal to 20
99 percent confidence interval:
 12.64116 20.15884
sample estimates:
 mean of x
      16.4
```

Opskriv hypotese, alternativ, α og n svarende til dette output. Hvad er estimatet for standard error for gennemsnittet? Hvad er den maximale fejl med 99% konfidens? (For at svare på det sidste kan (dele af) følgende R-information bruges:)

```
> qt(0.995,9)
[1] 3.249836
> qt(0.975,9)
[1] 2.262157
> qt(0.95,9)
[1] 1.833113
```

7.3.2 Opgave

Følgende R kommandoer og resultat haves:

```
> x1=c(10,13,16,19,17,15,20,23,15,16)
> x2=c(13,16,20,25,18,16,27,30,17,19)
> t.test(x1,x2,alt=''less'',conf.level=0.95,var.equal = TRUE)
         Two Sample t-test

data:  x1 and x2
t = -1.779, df = 18, p-value = 0.04606
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
        -Inf -0.09349972
sample estimates:
mean of x mean of y
     16.4      20.1
```

Opskriv hypotese, alternativ, α, n_1 og n_2 svarende til dette output. Hvad er estimatet for standard error for forskellen på gennemsnittene? Hvilken R-kommando ville du bruge for at finde den kritiske værdi for det anvendte hypotesetest?

7.3.3 Opgave

Følgende R kommandoer og resultat haves:

```
> x1=c(10,13,16,19,17,15,20,23,15,16)
> x2=c(13,16,20,25,18,16,27,30,17,19)
> t.test(x1,x2,paired=T,alt=''less'',conf.level=0.95)

        Paired t-test

data:  x1 and x2
t = -5.1698, df = 9, p-value = 0.0002937
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf -2.388047
sample estimates:
mean of the differences
                   -3.7
```

Opskriv hypotese, alternativ, α, n_1 og n_2 svarende til dette output. Hvad er estimatet for standard error for forskellen på gennemsnittene? Hvilken R-kommando ville du bruge for at finde den kritiske værdi for det anvendte hypotesetest?

8 Hypotese-test og konfidensintervaller for andele, Kap. 10, uge 9

8.1 Beskrivelse

Som beskrevet i appendix C, side 531 (7ed: 612) kan man bruge to R-funktioner: `prop.test` og `chisq.test` (der findes flere relevante, men dem vil vi ikke gennemgå her).

8.1.1 Konfidensinterval for en andel, sec.10.1

Man kan opnå et 95% konfidensinterval, som i eksemplet side 280 (7ed: 295), ved at køre `prop.test(36,100)`. Resultatet bliver en smule anderledes end i bogen. Det skyldes dels, at R som default bruger en såkaldt kontinuitetskorrektion i stil med det vi så ifb. med at approximere binomialfordelingen vha. normalfordelingen, side 132 (7ed: 160). Den kan man "slå fra" ved at skrive:
`prop.test(36,100,correct=F)`. Resultatet vil stadig være en lille smule anderledes end i bogen, idet R anvender endnu en korrektion, der får intervallet til at ligne det eksakte interval, som man kan aflæse i Tabel 9. Denne detalje vil vi IKKE gennemgå her.

8.1.2 Hypotesetest for en andel, sec.10.2

Man kan opnå resulater som i eksemplet side 299 ved at køre
`prop.test(48,60,p=0.7,correct=F,alternative=''greater'')`
Bemærk, at man IKKE får en Z-test størrelse, men i stedet i χ^2-test størrelse. Der gælder dog at $Z^2 = \chi^2$
Når funktionen kaldes med et ensidet alternativ (`alt=''greater''` eller `alt=''less''`), så angiver den et andet konfidens-interval end ellers. Dette er et såkaldt ensidet konfidens-interval, som vi IKKE berører i kurset!

8.1.3 Hypotesetest for to eller flere andele, sec.10.3

Man kan opnå resulater som i eksemplet side 286-287 (7ed: 302) (eksemplet anvendt på side 531 (7ed: 612)) ved at køre

```
crumbled=c(41,27,22)
intact=c(79,53,78)
prop.test(crumbled,crumbled+intact)
```

Man kan alternativt bruge funktionen `chisq.test` og køre

```
chisq.test(matrix(c(crumbled,intact),ncol=2))
```

Bemærk at R notationen her er lidt anderledes end den R-notation, der er angivet i lærebogen side 531 (7ed: 612).

8.1.4 Analyse af $r \times c$ tabeller, sec.10.4

Man kan opnå resultaterne i eksemplet side 295 (7ed: 310) ved på tilsvarende vis at køre:

```
poor=c(23,60,29)
ave=c(28,79,60)
vgood=c(9,49,63)
chisq.test(matrix(c(poor,ave,vgood),ncol=3))
```

Har man data på "rå" form, som f.eks. de karakterdata, der anvendtes i introduktionsøvelsen, kan man via menuerne få lavet krydstabuleringer og χ^2-test for potentielle sammenhænge: 'Statistics'→'Contingency Tables'→'Two-way table'

8.2 Træning vha. ugens øvelsesopgaver

For de fleste af opgaverne kan man naturligvis bruge fordelingerne, som øvet tidligere, i stedet for at slå op i tabellerne (det være sig z- eller χ^2-fordelingen). I følgende opgaver kan de to gennemgåede R-funktioner anvendes:

- Løs opgave 10.1 (7Ed: 9.1)
- Løs opgave 10.28 (7Ed: 9.28)
- Løs opgave 10.29 (7Ed: 9.29)
- Løs opgave 10.40 (7Ed: 9.40)
- Løs opgave 10.41 (7Ed: 9.41)

8.3 Testopgaver

Ingen testopgaver idet denne del kun læses "orienterende".

9 Statistik ved hjælp af simulering, uge 10

9.1 Introduktion

En af de helt store gevinster inden for statistik og modellering af tilfældige systemer ved computerteknologiens indtog de seneste årtier er muligheden for at kunne simulere tilfældige systemer på computeren. Det giver grundlæggende set mulighed for at kunne beregne ting, som ellers fra et matematisk analytisk synspunkt ville være umulige at finde. Og selv i tilfælde, hvor den højtuddannede matematiker/fysiker måske kunne finde løsninger, så giver simuleringsværktøjet et generelt og simpelt beregningsredskab til alle os, der ikke har denne teoretiske indsigt. Man kan blive helt høj, når man pludselig indser hvor helt ufattelig nemt, med blot en ganske lille indsigt i programmering, det er at beregne ting, som man ellers aldrig i sit liv ville kunne komme i nærheden af at finde ud af.

Den direkte anledning til at gå i den retning her i vores kursus er det "hul", der i de situationer, der er dækket i Kapitel 7 og 8 (7Ed: Kap. 7). Situationerne, som dækkes af bogen i disse

kapitler er givet i oversigtsform i Tabel 8.1, side 267 (7Ed: Table 7.1). Kort sagt handler det altså her om statistik i forbindelse med et enkelt eller to gennemsnit. Kigger man lidt nærmere på, hvad tabellen giver os værktøjer til, så fremgår det, at så længe der er tale om store ($n \geq 30$) stikprøver, så har vi værktøjer til det vi ønsker, idet vi kan lave hypotesetest og konfidensintervaller ved brug af normalfordelingen, der som følge af central grænseværdi sætning (Kapitel 6), er en god tilnærmelse til de relevante stikprøvefordelinger. Når man er i situationer med små stikprøver, så er der i Tabel 8.1 (og kapitel 7-8) angivet den EKSTRA antagelse, at populationerne, hvor data stammer fra SKAL være normalfordelinger. Altså i praksis skal man prøve at sikre sig at de data man analyserer opfører sig som en normalfordeling: symmetrisk og klokkeformet histogram. I Kapitel 5 lærte vi også, at man kan lave et normalfordelingsplot for at undersøge denne antagelse i praksis, og evt. transformere data for at få dem til at blive så normalfordelt som muligt. Problemet med små stikprøver er dog, at det selv med disse kontrolværktøjer kan være svært at vide om den underliggende fordeling virkelig er "normal" eller ej. Og i mange tilfælde vil antagelsen om normalitet jo simpelt hen være åbenlys uholdbar. For eksempel, når den responsskala vi arbejder på er langt fra at være kvantitativ og kontinuert - det kunne f.eks. være en skala som "lille", "mellem" og "stor" - kodet som 1, 2 og 3. Vi har brug for et værktøj, der kan lave statistikken for os UDEN denne antagelse om, at normalfordelingen er den rigtige model for de data, vi observerer og arbejder med.

I bogen dækkes dette hul med Kapitel 14: Nonparametric Tests. Og i tidligere versioner af dette kursus (til og med 2010) har dele af dette kapitel været indholdet af denne uge 10 forelæsning. Der behandles her de traditionelle såkaldte ikke-parametriske statistiske test. Kort fortalt er det en samling af metoder, der gør brug af data på en mere grov måde, typisk ved at fokusere på rangen (engelsk: the rank) af observationerne i stedet for selve værdien af observationerne. Så i en parret t-test situation ville man f.eks. blot tælle hvor mange gange den ene er større end den anden i stedet for at beregne forskellen i gennemsnittene. På den måde kan man lave statistiske tests uden at bruge antagelsen om en underliggende normalfordeling. Der findes en lang række af sådanne ikke-parametriske test for forskellige situationer. Historisk set, før computeralderen, var det den eneste mulighed man rigtig havde for i praksis at håndtere denne situation. Disse tests er alle karakteriseret ved at de beregningsmæssigt er givet ved relativ simple beregningsformler, som man i tidligere tider nemt kunne håndtere.

De simuleringsbaserede metoder, som vi nu i stedet vil præsentere, har adskillige fordele frem for de tradionelle ikke-parametriske metoder:

- Konfidensintervaller er meget nemmere at opnå
- De er meget nemmere at anvende i mere komplicerede situationer
- De afspejler i højere grad dagens virkelighed - de anvendes simpelt hen nu i rigtig mange sammenhænge

9.2 Hvad er simulering egentlig?

Basalt set kan en computer naturligvis ikke lave et resultat, som er tilfældigt. En computer kan give et output som funktion af et input. (Pseudo)tilfældige tal fra en computer bliver genereret ud fra en tilfældighedsgenerator, som er en specialdesignet algoritme, der når den først er startet

kan lave tallet x_{i+1} ud fra tallet x_i. Algoritmen er så konstrueret på en måde, så når man kigger på en sekvens af disse tal, så kan man i praksis ikke se forskel på disse og så en sekvens af rigtige tilfældige tal. Dog skal algoritmen have et start-input, kaldet "seed". Det genererer computeren typisk ved hjælp af det indbyggede ur. Som regel kan man klare sig fint uden at skulle bekymre sig om dette, idet programmet selv finder ud af at håndtere det på en hensigtsmæssig måde. Kun hvis man ønsker at kunne genskabe præcis de samme resultater, har man brug for at gemme og selv sætte seed-værdier - det findes der så ligeledes `R`-funktioner til. For detaljer omkring dette og de i `R` brugte tilfældighedsgeneratorer, skriv `?Random`.

Vi har allerede set, at `R` kan generere tilfældige tal fra en hvilken som helst af de fordelinger, der er implementeret i programmet. I forhold til de fordelinger, som vi har mødt i kurset vil følgende funktioner således være relevante:

`rbinom`	Binomialfordelingen
`rpois`	Poissonfordelingen
`rhyper`	Den hypergeometriske fordeling
`rnorm`	Normalfordelingen
`rlnorm`	Lognormalfordelingen
`rexp`	Eksponentialfordelingen
`runif`	Den uniforme(lige) fordeling
`rt`	t-fordelingen
`rchisq`	χ^2-fordelingen
`rf`	F-fordelingen

Faktisk vil en grundlæggende tilfældighedsgenerator typisk generere (pseudo)tilfældige tal mellem 0 og 1, cf. Sektion 5.14, side 167-168 i lærebogen (8th Ed), i den forstand at tallene i praksis følger den lige (uniforme) fordeling på intervallet 0 til 1. Det betyder, som bekendt, at uanset hvilket delinterval man betragter, så vil antallet af observationer i delintervallet svare til bredden af delintervallet. Der findes så faktisk en nem måde, hvorpå man ud fra disse kan transformere til en hvilken som helst fordeling: (cf. Figure 5.30, side 168)

Hvis $U \sim \text{Uniform}(0,1)$ og F er en fordelingsfunktion for en eller anden sandsynlighedsfordeling, så vil $F^{-1}(U)$ følge fordelingen givet ved F

Husk at fordelingsfunktionen F i `R` findes i `p`-versionerne af fordelingerne, mens F^{-1} findes i `q`-versionerne. Men da nu `R` allerede har klaret dette for os, behøver vi egentlig ikke bruge dette, så længe vi kun har brug for fordelinger, som allerede er implementeret i `R`. Man kan bruge hjælpefunktionen for hver enkelt funktion, f.eks. `?rnorm` , for at tjekke præcis hvordan man angiver parametrene i de enkelte fordelinger. Syntaks følger 100% hvad der bruges i `p`, `d` og `q` versionerne af fordelingerne

9.2.1 Eksempel

Man kan genere 100 normalfordelte $N(2, 3^2)$ tal med `rnorm(100,mean=2,sd=3)`. Det samme ville man kunne opnå med `qnorm(runif(100),mean=2,sd=3)`.

9.3 Simulering som generelt beregningsværktøj

Helt grundlæggende er styrken ved simuleringen at man kan beregne vilkårlige funktioner af tilfældige variable og deres udfald, med andre ord man kan finde sandsynligheder for komplicerede udfald. Som sådan er værktøjet grundlæggende set ikke et statistisk værktøj, men et sandsynlighedsregnings-værktøj. Men idet statistikken netop drejer sig om at analysere og lære af konkrete data i lyset af visse sandsynligheder, så kan simlueringsværktøjet i høj grad anvendes i statistisk sammenhæng, som vi vil gøre særdeles konkret herunder. Lad os først eksemplificere styrken ved beregningsværktøjet.

9.3.1 Eksempel

En virksomhed producerer rektangulære plader. Længden af pladerne (i meter), X, antages at kunne beskrives med en normalfordeling $N(2, 0.1^2)$ og bredden af pladerne (i meter), Y, antages at kunne beskrives med en normalfordeling $N(3, 0.2^2)$. Der er altså tale om plader af størrelse 2×3 meter men med en hvis (lille) fejl i både længde og bredde. Antag at disse fejl er helt uafhængige af hinanden. Man er interesseret i arealet, som jo så er givet ved $A = XY$. Dette er en ikke-lineær funktion af X og Y, og faktisk betyder det, at med de værktøjer vi lærer i indeværende kursus, så kan vi IKKE finde ud af hvad middelarealet egentlig er, slet ikke hvad spredningen for arealet er fra plade til plade, og langt fra udtale os om sandsynlighederne for forskellige mulige udfald. Vi ville således ikke ane, hvordan vi skulle beregne f.eks. hvor ofte sådanne plader har et areal, der afviger mere end $0.1m^2$ fra de $6m^2$. Et udsagn der opsummerer al vores mangel på viden er: vi kender ikke fordelingen for A og ved ikke hvordan vi finder den. Med simulering er det lige ud af landevejen: Man kan finde alt relevant information om A ved bare at simulere X og Y rigtig mange gange, og fra dette beregne A lige så mange gange, og så observere/registrere hvad der sker med værdierne for A. Første trin er så givet ved:

```
k=10000 # Antal simulationer
X=rnorm(k,2,0.1)
Y=rnorm(k,3,0.2)
A=X*Y
```

R-objektet A indeholder nu 10000 observationer af A. Derefter findes middelværdi og spredning for A simpelt hen ved at beregne gennemsnit og spredning for de simulerede A-værdier:

```
mean(A)
[1] 5.999061
sd(A)
[1] 0.5030009
```

Og den ønskede sandsynlighed, $P(|A - 6| > 0.1) = 1 - P(5.9 \leq A \leq 6.1)$ findes ved at tælle hvor ofte hændelsen faktisk forekommer blandt de k udfald af A:

```
sum(abs(A-6)>0.1)/k
[1] 0.8462
```

Koden `abs(A-6)>0.1` laver en vektor med værdierne `TRUE` eller `FALSE` afhængig af om den absolutte værdi af $A - 6$ er større end 0.1 eller ej. Når man "summer" (adderer) disse sættes

`TRUE` automatisk til 1 og `FALSE` automatisk til 0, hvorved den ønskede optælling fås, der så deles med det samlede antal simulationer k. Husk, at hvis I gør dette selv vil i ikke få eksakt det samme resultat, idet seed-værdien i jeres konkrete kørsel vil være en anden end den, der er brugt her. Det er klart, at denne simulationsusikkerhed er noget man skal forholde sig til i praksis. Størrelsen af denne vil afhænge af situationen og af k. Man kan altid få en første fornemmelse af hvad den betyder i en konkret situation ved simpelt hen at gentage beregningen nogle gange, og se hvordan resultatet varierer. Faktisk kunne man så systematisere en sådan undersøgelse og gentage simulationen mange gange for at få styr på usikkerheden. Vi vil ikke formalisere dette her. Når det handler om en sandsynlighed, som i det sidste eksempel her, så kan man alternativt bruge standard binomial-statistik, som gennemgået i Kapitel 10. Med f.eks. k=100000 er usikkerheden for en beregnet proportion på omking 0.85 givet ved: $\sqrt{\frac{0.85(1-0.85)}{100000}} = 0.0011$. Eller med f.eks. k=$10000000$ er usikkerheden 0.00011. Resultatet med en sådan k blev 0.8414536, og fordi vi er lidt uheldig med afrundingsstedet, så kan vi i praksis sige noget i stil med at det EKSAKTE resultat afrundet til 3 decimaler er enten 0.841 eller 0.842. På den måde bliver en beregning som egentlig er baseret på simulering lige pludselig præcis i den forstand at afrundet til 2 decimaler, så er resultatet simpelt hen 0.84.

9.4 Fejlophobningslove

Inden for kemi og fysik taler man om målefejl og om hvorledes målefejl ophobes/akkumuleres hvis man evt. har flere målinger og/eller anvender disse målinger i efterfølgende formler/beregninger (Engelsk: propagation of error). For det første: Den grundlæggende måde hvorpå man "måler en målefej" - altså sætter tal på en målefejl er ved en standardafvigelse, også kaldet spredning. Spredningen udtrykker, som bekendt, den gennemsnitlige afvigelse fra middelværdien. Det er klart, at det kan forekomme, at et måleinstrument også gennemsnitlig set måler forkert. Det kalder man så "bias", men i det grundlæggende setup her antager vi, at instrumentet ingen bias har. En fejl-ophobnings problemstilling er altså omformuleret et spørgsmål om hvorledes spredningen for en funktion af nogle målinger afhænger af spredningerne for de enkelte målinger: Lad $X_1, \ldots, X_n$ være n målinger med spredninger(målefejl) $\sigma_1, \ldots, \sigma_n$. For alt der foregår i dette kursus antager vi, at disse målefejl er uafhængigie af hinanden. Der findes udvidelser af formlerne, der kan håndtere det modsatte, men dem må vi udelade. Vi skal altså i en generel formulering være i stand til at finde:

$$\sigma^2_{f(X_1,\ldots,X_n)} = \text{Var}(f(X_1, \ldots, X_n)) \tag{1}$$

Faktisk har vi allerede i kurset set den lineære fejlophobningslov, som er udtrykt i kassen på side 154 i Kapitel 5 i bogen:

$$\sigma^2_{f(X_1,\ldots,X_n)} = \sum_{i=1}^{n} a_i^2 \sigma_i^2, \text{ hvis } f(X_1, \ldots, X_n) = \sum_{i=1}^{n} a_i X_i$$

Der findes en mere generel ikke-lineær udvidelse af dette, som dog kun teoretisk set er et approximativt resultat, som involverer de partielle afledede af funktionen f mht. de n variable:

$$\sigma^2_{f(X_1,\ldots,X_n)} \approx \sum_{i=1}^{n} \left(\frac{\partial f}{\partial X_i} \right)^2 \sigma_i^2 \tag{2}$$

I praksis indsætter man så selve måleværdierne $X_1, \ldots, X_n$ i de partielt afledede. Dette er et ret stærk redskab til generelt set at finde (approximative) usikkerheder for komplicerede funktioner af mange målinger eller for den sags skyld: komplekse kombinationer af forskellige statistiske beregningsstørrelser. Når formlen anvendes til det sidste, kaldes metoden også i nogen sammenhænge for "delta-reglen" (der er matematisk set tale om en såkaldt 1. ordens (lineær) Taylor-approximation til den ikke-lineære funktion f). Når det bringes frem her, skyldes det naturligvis at man som alternativ til denne approximative formel, kan bruge simulering til formålet efter devicen:

Simuler k udfald af samtlige n målinger som $N(X_i, \sigma_i^2)$: $X_i^{(j)}, j = 1 \ldots, k$
Beregn spredningen direkte som den observerede spredning af de k værdier for f:

$$\sigma_{f(X_1,\ldots,X_n)} = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}(f_j - \bar{f})^2}$$

$$f_j = f(X_1^{(j)}, \ldots, X_n^{(j)})$$

9.4.1 Eksempel

Lad os fortsætte eksemplet med $A = XY$, og X og Y defineret som i eksemplet ovenfor. For at bruge den approximative fejlophobningslov, skal man således differentiere funktionen $f(x, y) = xy$ med hensyn til både x og y:

$$\frac{\partial f}{\partial x} = y, \quad \frac{\partial f}{\partial y} = x$$

Med to konkrete målinger for X og Y, f.eks. $x = 2.05m$ og $y = 2.99m$ ville fejlophobningsloven give følgende approximative varians for $A = 2.05 \times 2.99 = 6.13$:

$$\sigma_A^2 = y^2 \times 0.1^2 + x^2 \times 0.2^2 = 2.99^2 \times 0.1^2 + 2.05^2 \times 0.2^2 = 0.2575$$

Så med fejlophobningsloven ville vi kunne klare en del af udfordringen uden at simulere. Faktisk er vi ret tæt på ved hjælp af værktøjer givet i indeværende kursus at kunne finde den rigtige varians for $A = XY$. For ved hjælp af definitionen og følgende grundlæggende sammenhæng: (som ER en del af pensum)

$$\text{Var}(X) = \text{E}(X - \text{E}(X))^2 = \text{E}(X^2) - \text{E}(X)^2$$

Så kan man faktisk udlede variansen for A teoretisk, blot skal man i tillæg vide, at for uafhængige stokastiske variable, så gælder at $\text{E}(XY) = \text{E}(X)\text{E}(Y)$:

$$\begin{aligned}
\text{Var}(XY) &= \text{E}\left[(XY)^2\right] - \left[\text{E}(XY)\right]^2 \\
&= \text{E}(X^2)\text{E}(Y^2) - \text{E}(X)^2\text{E}(Y)^2 \\
&= \left[\text{Var}(X) + \text{E}(X)^2\right]\left[\text{Var}(Y) + \text{E}(Y)^2\right] - \text{E}(X)^2\text{E}(Y)^2 \\
&= \text{Var}(X)\text{Var}(Y) + \text{Var}(X)\text{E}(Y)^2 + \text{Var}(Y)\text{E}(X)^2 \\
&= 0.1^2 \times 0.2^2 + 0.1^2 \times 3^2 + 0.2^2 \times 2^2 \\
&= 0.0004 + 0.09 + 0.16 \\
&= 0.2504
\end{aligned}$$

Bemærk hvorledes den approximative fejlophobningslov faktisk svarer til de to sidste led i den rigtige varians, mens det første - produktet af de to varianser ignoreres. Heldigvis er dette led det mindste af de tre i dette tilfælde. Det behøver det dog ikke altid at være. En teoretisk udledning af tæthedsfunktionen for $A = XY$ ville kunne klares, hvis man tager kursus 02405 Sandsynlighedsregning.

9.5 Konfidensintervaller ved hjælp af simulering: bootstrapping

Generelt er et konfidensinterval for den ukendte parameter μ en måde at udtrykke usikkerheden ved hjælp af stikprøvefordelingen for $\hat{\mu} = \bar{x}$. Vi skal altså bruge en fordeling, der udtrykker hvorledes vores beregnede værdi ville variere fra stikprøve til stikprøve. Som anført, så har vi IKKE indtil videre nogen metode til dette HVIS vi kun har en lille stikprøve ($n < 30$), og data ikke antages at følge en normalfordeling. Der er i princippet to indgange til dette problem:

1. Find/identificer/antag en anden og mere rigtig fordeling for populationen("systemet")
2. Undlad at antage nogen fordeling overhovedet

Simuleringsmetoden bootstrapping, der i praksis går ud på at simulere mange stikprøver, findes i to versioner, der kan klare hver sin af disse to udfordringer:

1. Parametrisk bootstrap: Simuler gentagne stikprøver fra den antagede fordeling.
2. Ikke-parametrisk bootstrap: Simuler gentagne stikprøver direkte fra data.

Faktisk håndterer den parametriske bootstrap i tillæg den situation, hvor data måske nok kunne være normalfordelt, men det vi er interesseret i at beregne er noget helt andet end gennemsnittet, f.eks. variationskoefficienten. Det er et eksempel på en ikke-lineær funktion af data, som således ikke har en normalfordeling som stikprøvefordeling. Og den parametriske bootstrap er sådan set blot et eksempel på brug af simulering som et generelt beregningsværktøj, som gennemgået ovenfor. Begge metoder er således særdeles generelle og kan bruges i stort set alle sammenhænge. Vi vil dog nedenfor kun detaljere metoderne i vores en og to-stikprøve situationer med fokus på middelværdierne - og kun for den ikke-parametriske bootstrap, idet vi ellers ikke i kurset har så meget fokus på statistik ifbm. alternative fordelinger for kontinuert kvantitative data. Vi har mødt nogle stykker af sådanne alternative fordelinger, f.eks. log-normal-, uniform- og eksponentialfordelingerne, men har egentlig ikke lært at lave "klassisk" (small sample - små stikprøve) statistik for data, der kommer fra sådanne fordelinger. Den parametriske bootstrap er en måde at gøre dette UDEN at basere sig på teroretiske udledninger af tingene.

9.5.1 Ikke-parametrisk bootstrap for one-sample situationen

Vi har stikprøven (data) $x_1, \ldots, x_n$. $100(1-\alpha)\%$-konfidensintervallet for μ bestemt ved den ikke-parametriske bootstrap er defineret som:

Simuler k stikprøver af størrelse n ved at udtage tilfældigt blandt de tilgængelige data
(med tilbagelægning - stort k , e.g. $k > 1000$)
Beregn gennemsnittet i hver af de k stikprøver: $\bar{x}_1^*, \ldots, \bar{x}_k^*$
Beregn $100\alpha/2\%$- og $100(1-\alpha/2)\%$ fraktilerne for disse
Intervallet er: $\left[\text{fraktil}_{100\alpha/2\%}, \text{fraktil}_{100(1-\alpha/2)\%}\right]$

Der findes andre versioner af selve konstruktionen af konfidensintervallet end denne her, som dog er den mest direkte og følger et princip der nemt kan generaliseres til andre situationer.

Eksempel
I et studie undersøgte man kvinders cigaretforbrug før og efter fødsel. Man fik følgende observationer af antal cigaretter pr. dag:

før	efter	før	efter
8	5	13	15
24	11	15	19
7	0	11	12
20	15	22	0
6	0	15	6
20	20		

Dette er et typisk parret t-test setup, som behandlet i Kapitel 8.4, som altså håndteres ved at man finder de 11 differencer og således omformer det til en one-sample situation, som også behandlet i kapitel 7.2. Man får data ind i `R` og beregnet differencerne ved følgende kode:

```
x1=c(8,24,7,20,6,20,13,15,11,22,15)
x2=c(5,11,0,15,0,20,15,19,12,0,6)
dif=x1-x2
dif
 [1]  3 13  7  5  6  0 -2 -4 -1 22  9
```

Der findes en stikprøveudtagelses-funktion i `R` (som underliggende igen baserer sig på en uniform tilfældighedsgenerator): `sample`. F.eks. kan man få 5 gentagne stikprøver (MED tilbagelægning - `replace=TRUE`) ved:

```
> t(replicate(5,sample(dif,replace=TRUE)))
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
[1,]    6    0    6    3    5   -2    5    9   -2     7     6
[2,]   22   22    3    9   13    5   -2    0   13     3     3
[3,]    9    3    3    7    6    6    7   -2    5    -2     3
[4,]   13    5    9   22   13    9   13   13    5     6     6
[5,]    9   -2   -1    6    3   -4   -1   -4    9    -2     3
```

Forklaring: `r`eplicate er en funktion, der gentager kaldet til `sample` - i dette tilfælde 5 gange. Funktionen `t` transponerer simpelt hen matricen af tal, så den bliver 5×11 i stedet for 11×5 (blot anvendt for at vise tallene på lidt færre linier end ellers nødvendigt)
Man kan så køre følgende for at få et 95%-konfidensinterval for μ baseret på $k = 10.000$:

```
k=10000
mysamples=replicate(k,sample(dif,replace=TRUE))
mymeans=apply(mysamples,2,mean)
quantile(mymeans,c(0.025,0.975))
    2.5%    97.5%
1.363636 9.727273
```

Forklaring: `sample` funktionen kaldes 10.000 gange og resultaterne samles i en 11×10000 matrix. Dernæst beregnes i et enkelt kald de 10.000 gennemsnit, hvorefter de relevante fraktiler findes. Der findes faktisk en bootstrap-pakke til `R`, som simpelt hen hedder `bootstrap` som inkluderer en funktion, der hedder `bootstrap` . Installer først denne pakke (klik "Packages" $\rightarrow$"Install Packages" og find den på listen eller nemmere: skriv blot: `i`nstall.packages("bootstrap")). Dernæst load pakken:

```
library(bootstrap)
```

hvorefter beregningen kan udføres i et enkelt kald:

```
quantile(bootstrap(dif,k,mean)$thetastar,c(0.025,0.975))
    2.5%    97.5%
1.361364 9.818182
```

Denne funktion kan med fordel bruges, når man søger konfidensintervaller for mere komplicerede funktioner af data.

9.5.2 Two-sample situationen

Vi har nu stikprøverne $x_1, \ldots, x_{n_1}$ og $y_1, \ldots, y_{n_2}$ $100(1-\alpha)\%$-konfidensintervallet for $\mu_1 - \mu_2$ bestemt ved den ikke-parametriske bootstrap er defineret som:

Simuler k sæt af 2 stikprøver af størrelse n_1 og n_2 ved at udtage tilfældigt fra de respektive grupper
(med tilbagelægning - stort k , e.g. $k > 1000$)
Beregn forskellen i gennemsnittene for hver af de k stikprøvepar: $\bar{x}_1^* - \bar{y}_1^*, \ldots, \bar{x}_k^* - \bar{y}_k^*$
Beregn $100\alpha/2\%$- og $100(1-\alpha/2)\%$ fraktilerne for disse
Intervallet er: $\left[\text{fraktil}_{100\alpha/2\%}, \text{fraktil}_{100(1-\alpha/2)\%}\right]$

Eksempel
I et studie ville man undersøge, om børn der havde fået mælk fra flaske som barn havde dårligere eller bedre tænder end dem, der ikke havde fået mælk fra flaske. Fra 19 tilfældigt udvalgte børn registrerede man hvornår de havde haft deres første tilfælde af karies.

flaske	alder	flaske	alder	flaske	alder
nej	9	nej	10	ja	16
ja	14	nej	8	ja	14
ja	15	nej	6	ja	9
nej	10	ja	12	nej	12
nej	12	ja	13	ja	12
nej	6	nej	20		
ja	19	ja	13		

Man kan så køre følgende for at få et 95%-konfidensinterval for $\mu_1 - \mu_2$ baseret på $k = 10.000$:

```
x=c(9,10,12,6,10,8,6,20,12) # nej-gruppen
y=c(14,15,19,12,13,13,16,14,9,12) # ja-gruppen

k=10000 # Antal bootstrap-samples
xsamples=replicate(k,sample(x,replace=TRUE)) # Sampling af nej-gruppen
ysamples=replicate(k,sample(y,replace=TRUE)) # Sampling af ja-gruppen
mymeandifs=apply(xsamples,2,mean)-apply(ysamples,2,mean)  # Beregning af forskelle
quantile(mymeandifs,c(0.025,0.975)) # Fraktilerne
      2.5%      97.5%
-6.2222222 -0.1777778
```

9.6 Hypotesetest ved hjælp af simulering

Vi skal se to måder hvorpå vi kan lave hypotesetest ved hjælp af simulering.

9.6.1 Hypotesetest ved hjælp af bootstrap konfidensintervaller

Hypoteser, der kan formuleres ved en enkelt parameter - eller en direkte relation mellem to parametre, kan man teste ved hjælp af den sædvanlige sammenhæng mellem konfidensinterval og hypotesetest - her forsøgt generelt formuleret:

$$H_0 : \theta = \theta_0 \text{ accepteres} \Leftrightarrow \theta_0 \text{ ligger i konfidensintervallet for } \theta$$

Bruger man så det ikke-parametrisk bootstrap baserede konfidensinterval som kriterie, så har man således automatisk et simuleringsbaseret hypotesetest. En lille krølle er her, at vi i dette kursus ellers kun arbejder med 2-sidede konfidensintervaller, som så kan give et 2-sidet hypotesetest. Selvom vi ellers ikke opererer med 1-sidede konfidensintervaller, så kan man tilsvarende definere ensidede hypotese-test vha. bootstrappen på den oplagte måde, f.eks.:

$$H_0 : \theta = \theta_0 \text{ mod } H_1 : \theta > \theta_0 \text{ accepteres} \Leftrightarrow$$

$$\theta_0 > 100\alpha\%\text{-fraktilen for bootstrapværdierne for } \theta$$

9.6.2 One-sample setup, Eksempel

Vi fortsætter cigaretforbrugseksemplet. Man vil nu gerne påvise, at cigaretforbruget er faldet efter fødslen. Vi vil altså gerne udføre et en-sidet test for hypotesen $H_0 : \mu_1 - \mu_2 = 0$ mod $H_1 : \mu_1 - \mu_2 > 0$. Betragter man bootstrap konfidens-intervallet ovenfor kan vi konstatere så meget som at, idet 0 ligger uden for intervallet, at så forkaster vi hypotesen, når vi tester på niveau $\alpha = 0.025$. Man kan også finde P-værdien ved at observere, hvor 0 ligger i bootstrapsample fordelingen - eller med andre ord, hvor ofte det sker at en forskel bliver mindre end 0:

```
sum(mymeans<0)/k
[1] 0.0022
```

P-værdien er altså omkring 0.002, så en relativ klar indikation af at cigaretforbruget faktisk er faldet.

9.6.3 Hypotesetest ved hjælp af permutationstest

Der findes situationer, hvor man ikke nemt kan udtrykke den relevante hypotese på en meningsfuld skala, hvor et konfidensinterval kan løse opgaven for os. Det er det typiske, når man tester hypoteser, der involverer mere end 2 parametre. Vi har set et eksempel ifbm. $r \times c$-tabeller i kapitel 10, og vi skal se nogle flere i kapitel 12, hvor vi skal udvide netop to-gruppe situationerne fra kapitel 7 og 8, som vi fokuserer på her, til flergruppe-setups. Det går i al sin enkelhed ud på, at man "ryster posen" og ser hvad der sker. Mere præcist prøver man mange gange at trække lod om hvilke grupper de enkelte datapunkter skal tilhøre, og så beregner man teststørrelsen hver gang. Hvis den rent faktisk observerede teststørrelse, der måler gruppeforskellen, er usædvanlig

stor i forhold til de mange simulerede versioner, så forkaster man hypotesen. I praksis har man altså ved hver simulering præcis de samme observationer, som i det oprindelige datasæt, men blot anderledes fordelt på grupperne. Man siger også, at man har permuteret data på grupperne. Eller igen anderledes formuleret: man har samplet UDEN tilbagelægning.

9.6.4 Two-sample situationen

Vi har nu stikprøverne $x_1, \ldots, x_{n_1}$ og $y_1, \ldots, y_{n_2}$ med estimeret middelværdi $\hat{\mu_1} = \bar{x}$ og $\hat{\mu_2} = \bar{y}$
Et permutationstest for hypotesen $\mu_1 = \mu_2$ er defineret ved:

Simuler k sæt af 2 stikprøver af størrelse n_1 og n_2 ved at permutere de tilgængelige data (stort k , e.g. $k > 1000$)
Beregn forskellen i gennemsnittene for hver af de k stikprøvepar: $\bar{x}_1^* - \bar{y}_1^*, \ldots, \bar{x}_k^* - \bar{y}_k^*$
Find P-værdien ud fra positionen af $\bar{x} - \bar{y}$ i denne fordeling (2-sidet eller 1-sidet - på sædvanlig vis)

Eksempel Vi fortsætter eksemplet med tænderne. Vi ønsker at udføre et tosidet test for om $\mu_1 = \mu_2$. Følgende `R`-kode gennemfører beregningerne:

```
x=c(9,10,12,6,10,8,6,20,12) # nej-gruppen
y=c(14,15,19,12,13,13,16,14,9,12) # ja-gruppen

 k=100000
perms=replicate(k,sample(c(x,y)))
mymeandifs=apply(perms[1:9,],2,mean)-apply(perms[10:19,],2,mean)
sum(abs(mymeandifs)>abs(mean(x)-mean(y)))/k
[1] 0.05132
```

Forklaring: Først sættes et stort k, idet p-værdien viser sig at ligge lige omkring 5%. Dernæst kaldes `sample(c(x,y))`, som permuterer de 19 observationer. I `mymeandifs` beregnes for hver af de 100000 simpelt hen gennemsnittet mellem de første 9 og de sidste 10 tal. I sidste linie laves optællingen af hvor ofte disse simulerede forskelle er absolut set større end den rent faktisk observerede forskel i data.

Den opmærksomme læser vil måske have overvejet det lidt pudsige i det vi gør her: faktisk kan vi beregne præcis hvor mange forskellige permutationer, der egentlig findes. I eksemplet her er det antallet af forskellige stikprøver på 9 ud af 19, man kan tage. Det er præcis den såkaldte binomialkoefficient:

$$\text{Antal forskellige permutationer} = \begin{pmatrix} 19 \\ 9 \end{pmatrix} = \frac{19!}{(10!)(9!)} = 92378$$

I `R` kan tallet f.eks. findes som:

```
 factorial(19)/(factorial(10)*factorial(9))
[1] 92378
```

Der er faktisk kun i alt 92378 forskellige mulige udfald for forskellen på de to gennemsnit. En smule mere intelligent metode ville således være simpelt hen at liste dem alle sammen præcis en gang og så lave optællingen baseret på disse i stedet. Det vil give en såkaldt eksakt P-værdi for permutationstestet. Nogen gange er dette tal dog urealistisk stort, og dermed tyer man til at simulere fra disse i stedet. Og den `R`-tekniske implementering af testet er faktisk nemmere ved simuleringen her, så i dette kursus holder vi os til simuleringsversionen beskrevet i kassen herover.

9.7 Opgaver

Til dette pensum findes kun opgaver her. De fleste er stillet med udgangspunkt i, at man selv skal have `R` kørende for at kunne løse dem. Til eksamen vil opgaver inden for dette pensum bliver stillet på en måde, så man IKKE afkræves rent faktisk at anvende `R` i eksamenslokalet. Erfaringen og indsigten der opnås ved at løse de her stillede opgaver vil gøre en i stand til at løse evt. eksamensopgaver (i fællesskab med selve gennemgangen ovenfor, naturligvis)

9.7.1 Exercise

(Simulation as a computation tool). A system consists of three components A, B and C serially connected such that A is positioned before B again positioned before C. The system will function only so long as A, B and C all function. The lifetime in months of the three components are assumed to follow exponential distributions with means 2 months, 3 months and 5 months, respectively.

1. Generate, by simulation, a large number (at least 1000) of system lifetimes.
2. Estimate the mean system lifetime.
3. Estimate the standard deviation of system lifetimes.
4. Estimate the probability that the system fails within 1 month.
5. Estimate the median system lifetime
6. Estimate the 10th percentile of system lifetimes
7. What seems to be the distribution of system lifetimes? (histogram etc)

9.7.2 Exercise

(Non-linear error propagation). The pressure P, and the volume V of one mole of an ideal gas are related by the equation $PV = 8.31T$, when P is measured in kilopascals, T is measured in kelvins, and V is measured in liters.

1. Assume that P is measured to be 240.48kPa and V to be 9.987L with known measurement errors (given as standard deviations): 0.03kPa and 0.002L. Estimate T and find the uncertainty in the estimate.
2. Assume that P is measured to be 240.48kPa and T to be 289.12K with known measurement errors (given as standard deviations): 0.03kPa and 0.02K. Estimate V and find the uncertainty in the estimate.
3. Assume that V is measured V to be 9.987L and T to be 289.12K with known measurement errors (given as standard deviations): 0.002L and 0.02K. Estimate P and find the uncertainty in the estimate.
4. Try to answer one or more of these questions by simulation (assume that the errors are normally distributed)

9.7.3 Exercise

(Can be handled without using R) The following measurements were given for the cylindrical compressive strength (in MPa) for 11 prestressed concrete beams: 38.43, 38.43, 38.39, 38.83, 38.45, 38.35, 38.43, 38.31, 38.32, 38.48, 38.50. 1000 bootstrap samples (each sample hence consisting of 11 measurements) were generated from these data, and the 1000 bootstrap means were arranged on order. Refer to the smallest as $\bar{x}^*_{(1)}$, the second smallest as $\bar{x}^*_{(2)}$ and so on, with the largest being $\bar{x}^*_{(1000)}$. Assume that $\bar{x}^*_{(25)} = 38.3818$, $\bar{x}^*_{(26)} = 38.3818$, $\bar{x}^*_{(50)} = 38.3909$, $\bar{x}^*_{(51)} = 38.3918$, $\bar{x}^*_{(950)} = 38.5218$, $\bar{x}^*_{(951)} = 38.5236$, $\bar{x}^*_{(975)} = 38.5382$, and $\bar{x}^*_{(976)} = 38.5391$.

1. Consider why it may be questionable to use the t-distribution based confidence interval method for these data! (Look at the data, e.g. Plot the data - e.g. a box plot)

2. Compute a 95% bootstrap confidence interval for the mean compressive strength.

3. Compute a 90% bootstrap confidence interval for the mean compressive strength.

9.7.4 Exercise

Consider the data from the exercise above. These data are entered into R as:

```
x=c(38.43, 38.43, 38.39,38.83, 38.45,38.35,38.43,38.31,38.32,38.48,38.50)
```

Now generate 1000 bootstrap samples and compute the 1000 means.

1. What is the 2.5%, and 97.5% percentiles?

9.7.5 Exercise

A TV producer had 20 consumers evaluate the quality of two different TV flat screens - 10 consumers for each screen. A scale from 1 (worst) up to 5 (best) were used and the following results were obtained:

TV screen 1	TV screen 2
1	3
2	4
1	2
3	4
2	2
1	3
2	2
3	4
1	3
1	2

1. Carry out the test of the null hypothesis that TV screen2 has a quality of at most 2.5 versus the alternative of a larger than 2.5 quality. Use $\alpha = 0.01$. (Use the bootstrap approach)

2. Carry out the test of the hypothesis that the two TV screens have the same quality. Use $\alpha = 0.05$.

(a) By the bootstrap confidence interval method

(b) By the permutation test method.

3. How many different permuations are there really?

4. Compare the results with using t-test procedures!

10 Lineær regression, kap. 11, uge 11

10.1 Beskrivelse

Orienter dig i lærebogens appendiks C, specielt 'Regression' på side 513 (7ed: 613). Data relevante for denne beskrivelse kan downloades fra
`http://www2.imm.dtu.dk/courses/02402/Bookdata8ED`
eller hvis man arbejder med 7. edition af bogen:
`http://www2.imm.dtu.dk/courses/02402/Bookdata8`
hvorefter de kan importeres på sædvanlig vis. Vi bruger eksemplet fra side 303, 305, 306, 311, 312, 315 (7ed: 341, 347, 349, 351). Data kan downloades som:
Antag at data er gemt som C11evap:

```
> C11evap
   velocity evap
 1       20 0.18
 2       60 0.37
 3      100 0.35
 4      140 0.78
 5      180 0.56
 6      220 0.75
 7      260 1.18
 8      300 1.36
 9      340 1.17
10      380 1.65
```

Man kan plotte sammenhængen ved:

```
> attach(C11evap)
> plot(evap ~ velocity)
```

Den grundlæggende regressionsfunktion, som skal beskrives her er `lm`. Man fitter linien og gemmer resultatet af beregningerne ved følgende:

```
> fit.evap <- lm(evap ~ velocity)
```

og som beskrevet side 613 i lærebogen fås resultaterne opsummeret ved:

```
> summary(fit.evap)

Call:
lm(formula = evap ~ velocity)

Residuals:
```

```
     Min       1Q   Median       3Q      Max
-0.20103 -0.14671  0.05261  0.12318  0.17473

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0692424  0.1009737   0.686    0.512
velocity    0.0038288  0.0004378   8.746 2.29e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1591 on 8 degrees of freedom
Multiple R-squared: 0.9053,     Adjusted R-squared: 0.8935
F-statistic: 76.49 on 1 and 8 DF,  p-value: 2.286e-05
```

Man ser her parameterestimaterne, deres usikkerheder samt t-test for om de er nul (svarende til side boksene side 310-311 (7ed: 346)). Man ser også estimatet for s_e og R^2 (sammenlign med resultaterne i bogen).

Man kan lave lineær regression via menuen ved et bruge 'Statistics'→'Fit models'→'Linear regression'. Man kan også få regressionslinien tegnet ind i scatterplottet via 'graph'→'Scatterplot'.

10.2 Træning vha. ugens øvelsesopgaver

Løs opgaverne 11.4, 11.5 og evt. 11.6 vha. R.

10.3 Testopgaver

10.3.1 Opgave

Kører man en analyse af Matematik-eksamens karakterer som funktion af matematik årskarakterer får man: (cf. introduktionsøvelsen, afsnit 2)

```
> attach(karakterer2004)
> summary(lm(Mat.Eks~Mat.Aars))


Call:
lm(formula = Mat.Eks ~ Mat.Aars)

Residuals:
      Min        1Q    Median        3Q       Max
-2.450505 -0.266329  0.009203  0.281145  2.181145

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.4952     0.1750   14.26   <2e-16 ***
Mat.Aars      0.7194     0.0222   32.41   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4575 on 1553 degrees of freedom
Multiple R-squared: 0.4035,     Adjusted R-squared: 0.4031
F-statistic:  1051 on 1 and 1553 DF,  p-value: < 2.2e-16
```

Opskriv modellen og angiv estimat for linien. Er disse estimater signifikant forskellige fra 0? Hvad er korrelationen mellem de to sæt af karakterer? Hvad er konfidensintervallet for hældningskoefficienten? Og hvad er forøvrigt den øvre kvartil for eksamens-karaktererne?

11 Variansanalyse, Kap. 12.1 og 12.2, uge 12

11.1 Beskrivelse

Orienter dig i lærebogens appendiks C, specielt 'One-way Analysis of Variance (ANOVA)' på side 532 (7ed: 613). Data relevante for denne beskrivelse kan downloades fra
`http://www2.imm.dtu.dk/courses/02402/Bookdata8ED`
eller hvis man arbejder med 7. edition af bogen:
`http://www2.imm.dtu.dk/courses/02402/Bookdata`
hvorefter de kan importeres på sædvanlig vis. Data skal være struktureret som vist i nedenstående eksempel, hvor den ene kolonne er selve data, mens den anden kolonne angiver hvilken gruppe, hver observation tilhører:

```
 G       Materiale
 6.683     Guld
 6.681     Guld
 6.676     Guld
 6.678     Guld
 6.679     Guld
 6.661   Platin
 6.661   Platin
 6.667   Platin
 6.667   Platin
 6.664   Platin
 6.678     Glas
 6.671     Glas
 6.675     Glas
 6.672     Glas
 6.674     Glas
```

Data fra eksemplet side 363 (7ed: 408) kan importeres som
`http://www2.imm.dtu.dk/courses/02402/Bookdata8ED/C12tin.TXT`
eller hvis man bruger 7. edition:
`http://www2.imm.dtu.dk/courses/02402/Bookdata/C12tin.dat`
Antag at data er gemt som C12tin. Som beskrevet side 532 (7ed: 613) i lærebogen fås analysen også ved hjælp af funktionen `lm`:

```
> attach(C12tin)
> Lab <- factor(Lab)
> anova(lm(weight~Lab))

Analysis of Variance Table
```

```
Response: weight
          Df   Sum Sq   Mean Sq F value  Pr(>F)
Lab        3 0.013006 0.0043354  2.8097 0.05038 .
Residuals 44 0.067892 0.0015430
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Kommandolinie 2 `Lab <- factor(Lab)` sikrer at programmet tænker på laboratoriumkolonnen som en grupperingsfaktor og IKKE som en kvantitativ værdi. Laboratorierne er nemlig identificeret ved tallene 1, 2, 3 og 4 i dette tilfælde. Sammenligner man resultatet med bogens, ser man en lille afvigelse på F-størrelsen. Det skyldes at man i bogen bruger afrundede tal. Det får den pudsige konsekvens, at P-værdien jo faktisk er en smule over de 5% og ikke under, som anført i bogen. Men det viser jo således også, at der jo i realiteten ikke er nogen evidensmæssig forskel på en p-værdi på 4.9% og 5.1%!

Man kan lave variansanalysen via menuerne: (Når Lab-variablen er lavet til en faktor) 'Statistics'→'Means'→'
way ANOVA'.

11.1.1 Supplement: Generel variansanalyse ("Orienterende")

I ensidet variansanalyse kan den variabel, der beskriver grupperingen betragtes som en forklarende variable, der kan antage et lille antal værdier. Sådan en variable kaldes en kategorisk variabel. Variansanalyse kan også udføres i det tilfælde hvor der er flere forklarende kategoriske variable. Lærebogens afsnit 12.3 er et eksempel herpå.
Man kan lave denne mere generelle variansanalyse via menuerne: 'Statistics'→'Means'→'Multiway ANOVA'.

11.2 Træning vha. ugens øvelsesopgaver

- Løs opgave 12.10 vha. R
- Løs opgave 12.6 vha. R

Prøv at sætte data ind i de sædvanlige variansanalyse tabeller og forstå specielt frihedsgradstallene, teststørrelsen, samt p-værdien (brug evt. `pf(q,df1,df2)`).

11.3 Testopgaver

11.3.1 Opgave

Der kørtes to analyser af matematik årskarakterne. R kommandoerne og resultaterne ses i det følgende:

```
> anova(lm(Mat.Eks~Kommune))
> anova(lm(Mat.Eks~Amt))

Analysis of Variance Table

Response: Mat.Eks
```

```
Terms added sequentially (first to last)
            Df Sum of Sq   Mean Sq  F Value      Pr(F)
  Kommune  269  106.4311 0.3956547 1.159594 0.05405999
Residuals 1285  438.4435 0.3412012


Analysis of Variance Table

Response: Mat.Eks

Terms added sequentially (first to last)
            Df Sum of Sq  Mean Sq  F Value        Pr(F)
      Amt   15   16.2862 1.085744 3.161173 3.822564e-05
Residuals 1539  528.5885 0.343462
```

Opskriv hypoteser og angiv P-værdier for disse og fortolk resultaterne: Er der forskel på hhv. Kommuner og Amter hvad angår matematikeksamens karakterer? Hvor mange kommuner hhv. amter er med i undersøgelsen? Hvor meget varierer skoler inden for hhv. kommuner og amter?

12 Variansanalyse, Kap. 12.3, uge 12

12.1 Beskrivelse

I forhold til afsnittet ovenfor om ensidig variansanalyse skal data være på samme form dog med en ekstra kolonne svarende til "blok" informationen. For eksemplet side 373-375 (7ed: 420-422) kan data indlæses som:

```
example <- data.frame(y = c(13,7,9,3,6,6,3,1,11,5,15,5),
treatm = c(1,1,1,1,2,2,2,2,3,3,3,3),
block  = c(1,2,3,4,1,2,3,4,1,2,3,4))
```

som altså svarer til følgende struktur:

```
> example
    y treatm block
 1 13      1     1
 2  7      1     2
 3  9      1     3
 4  3      1     4
 5  6      2     1
 6  6      2     2
 7  3      2     3
 8  1      2     4
 9 11      3     1
10  5      3     2
11 15      3     3
12  5      3     4
```

Analysen køres nu helt som for den ensidige variansanalyse, blot med blok-faktoren tilføjet:

```
> attach(example)
> treatm <- factor(treatm)
> block <- factor(block)
> anova(lm(y~treatm+block))

Analysis of Variance Table

Response: y

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value     Pr(F)
   treatm  2        56 28.00000 3.230769 0.1116192
    block  3        90 30.00000 3.461538 0.0913831
Residuals  6        52  8.66667
```

Dette svarer til ANOVA-tabellen øverst side 422.

12.2 Træning vha. ugens øvelsesopgaver

- Løs opgave 12.47 vha. R. (7Ed: 12.50)

Data kan enten indtastes i et regneark og importeres på sædvanlig vis eller nedenstående kommandoer kan benyttes (brug "cut & paste" til at få dem ind i R):

```
dat.12.47 <- data.frame(conc.ppm = c(
                          23.8, 7.6, 15.4, 30.6, 4.2,
                          19.2, 6.8, 13.2, 22.5, 3.9,
                          20.9, 5.9, 14.0, 27.1, 3.0),
                        agency = rep(paste(''Agency'', 1:3), rep(5,3)),
                        site = rep(paste(''Site'', LETTERS[1:5]), 3))
```

12.3 Testopgaver

12.3.1 Opgave

Betragt følgende R kommando og resultat: (der er tale om målinger af brudstyrker (strength) for nogle tråde (thread) ifm. forskellige instrumenter (instrument) fra opgave 12.20)

```
> anova(lm(strength~thread+instrument))
Analysis of Variance Table

Response: strength

Terms added sequentially (first to last)
           Df Sum of Sq  Mean Sq  F Value     Pr(F)
    thread  4    70.173 17.54325 8.315979 0.0018781
instrument  3     0.330  0.11000 0.052143 0.9835259
 Residuals 12    25.315  2.10958
```

Opskriv hypoteser og angiv P-værdier for disse og fortolk resultaterne: Er der forskel på trådene og er der forskel på instrumenterne? Hvor mange slags tråde hhv. instrumenter er med i undersøgelsen? Hvad er standardafvigelsen for styrkemålingerne, når man ser bort fra systematiske forskelle mellem trådtyper og instrumenttyper?