LaTeX + R = Sant

Silje Synnøve Lyder Hermansen

November 14, 2019

Contents

1 LaTeXi knitr

1.1 Hva er LaTeX og hvorfor skal jeg bry meg?

1.1.1 LaTeX er et settespråk (typesetting system)

LaTeX er et settesystem ("typesetting system" evt. "document markup language"). Det vil si at det består av en serie med koder forå formattere tekst og bilder automatisk. Systemet er basert på TeX som først ble presentert av Donald Knuth i 1978: Det er et settespråk som har vært populært i mange år blant naturvitere fordi språket inneholder greske bokstaver og matematiske symboler som vanlige skriveprogram tidligere ikke inneholdt. LaTeX innebærer flere muligheter enn TeX: den inneholder flere teksttyper, jobber sømløst med referanseprogrammer (BibTeX) osv.

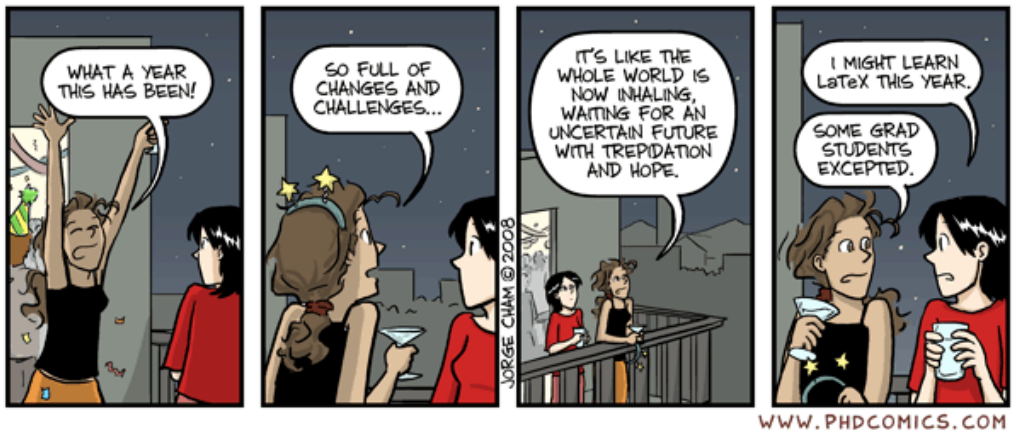
LaTeX er et språk, ikke et program! LaTeX inneholder koder som vi som skriver for å formattere teksten: Dette skjer ved at vi velger hvilken kategori teksten vår skal befinne seg i (tittel, undertittel, fotnote, kursiv osv.). I vanlige skriveprogrammer tilsvarer dette nedrekksmenyene med tilsvarende valgmuligheter. Forskjellen er at vi i LaTeX velger hva slags tekst vi jobber med (bok, artikkel, lysbilder, poster osv.) og hvilken del av teksten som faller i hvilken kategori, men LaTeX velger selve formatteringen for oss (fonter, marger, skriftstørrelse osv.). Dermed er LaTeX et utmerket valg for folk som er blottet for estetisk sans, men gjerne vi se bra ut likevel (det er her naturviterne kommer inn i bildet.)

Siden LaTeX kun er et språk, trenger vi et eget program som leser kodene og "oversetter" dem til formattering. Dette er hva programmet gjør når det "kompilerer" dokumentet og slår pdf av det. I prossessen lages en rekke dokumenter som vi i praksis ikke bruker (men *programmet* bruker dem, så ikke kast noe før du vet hva det er!): Selve teksten og kodene lagrer vi tradisjonelt i et .tex-dokument, som – ad omveier – blir til en pdf. Dermed vil pdf være formatet vi gir til leserene.

Programmer som leser LaTeX er:

- MikTeX (for Windows-maskiner): `http://miktex.org/`
- MacTeX (for Apple-maskiner): `https://tug.org/mactex/`

Vi kommer derimot aldri til å åpne eller kommunisere direkte med dette programmet, for det er *ikke* et tekstbehandlingsprogram. Det finnes egne tekstbehandlingsprogram laget kun for LaTeX (f.eks.: Overleaf for nettbasert



løsning, TeXworks for Windows, AquaMax for Apple), men vi kommer ikke til å bruke dem her. Vi kommer til å bruke RStudio via rammeverket Sweave/knitr.

1.1.2 LaTeXer praktisk og pent

Jeg kan komme på tre årsaker til at folk bruker LaTeX:

- **LaTeX er suveren når det gjelder matematiske symboler og formler:** Dette er årsaken til at LaTeX har vært så populært blant naturvitere. Per i dag har Word gode løsninger for å skrive formler, så med mindre man skriver veldig mange formler og utregninger i løpet av en dag (noe vi samfunnsvitere aldri gjør), ville ikke dette vært nok for meg, personlig.
- **LaTeX er pent** i betydningen at latex-dokumenter er homogene.[1]. Man behøver ikke bruke mye tid på formattering for at et dokument skal se bra ut. I tillegg håndterer LaTeX bildefiler godt, slik at jeg aldri har problemet vi regelmessig finner i Word, nemlig at bilder og figurer blir uklare.

[1] "Pent" innebærer også at LaTeX-dokumenter er lett gjenkjennelige. Dermed er det også i enkelte (smale) miljøer et statussymbol; det signalliserer kvalitet. De siste årene har programmer med åpen kildekode blitt svært brukervennlige, dermed betyr det også at stadig flere bruker LaTeX. Om jeg ikke hadde hatt andre behov enn dette, ville jeg brukt LyX: `http://www.lyx.org/` eller Overleaf `https://www.overleaf.com/`. Programmet skriver LaTeX-koden for meg, akkurat som nedtrekksløsningen i SPSS skriver syntaksen for meg

- **LaTeX og R jobber godt sammen!** Sweave er en løsning som "vever" R-koder (husk: R er gratisbroren til S) og LaTeX-koder mer eller mindre sømløst. Knitr er en serie med tilleggsfunksjoner som "strikker" sweave-filene.

Knitr er årsaken til at jeg ble "frelst": Sweave er en teknisk løsning for reproduserbar forskning: Jeg behandler data (koder/omkoder), analyserer og illustrerer data i samme dokumentet som hvor jeg skriver forskningen min. Det betyr ikke bare at andre kan "gå meg i sømmene"; jeg kan også sjekke selv hva jeg har gjort. Det er utrolig hvor fort man glemmer!

En annen fordel med knitr er at alle tabeller, figurer og referanser til tall oppdateres automatisk når datagrunnlaget og utregningene endrer seg. Å jobbe med LaTeX og R sammen innebærer en god del feilmeldinger og "debugging", men om man jobber mye med tall og tekst sammen, utgjør denne oppdateringsfunksjonen en enorm stordriftseffekt!

I det følgende starter jeg med å gå gjennom latex-delen av knitr, før vi innarbeider R-koder.

1.2 Begynn et `.Rnw`-dokument

1.2.1 Minste minimum av hva vi trenger

åpne "fil"-menyen i R-Studio, velg "Ny fil" og "R Sweave". Da får vi et dokument som allerede inneholder minimum av koder dere trenger. Dette formatet gjør at vi kan skrive b?de LaTeX-koder og R-koder i ett og samme dokument. Det er RStudio som tar seg av kommunikasjonen med hhv. MikTeX og R. Det eneste vi trenger å gjøre er å opplyse om hvilket språk vi snakker. "Default" er LaTeX.

Dokumentet krever minst disse elementene (de er inkludert når vi åpner et nytt dokument i RStudio). Alle LaTeX-koder starter med backslash (\) og en påfølgende kode som følger direkte etter streken. Om koden bare gjelder en liten tekstbit eller inneholder tilleggsargumenter, vil vi pakke dette inn i krøllete parenteser {}:

```
\documentclass{article}
\begin{document}
```

```
Her kan jeg skrive så mye tekst jeg vil!

\end{document}
```

1. *Alle LaTeX-dokumenter starter med en innledning ("preamble).* Den sier hva slags type dokument jeg ønsker (artikkel, bok el.). Det er minste minimum av informasjon. I tillegg kan jeg gi mange tillegsargumenter: hvilke pakker jeg vil bruke (mer om det senere), hvilket skriftspråk jeg bruker, dokumentets tittel, forfatter osv.

2. *Alle dokumenter må inneholde en kommando som forteller når dokumentet begynner, og når det slutter:* \begin{document} og \end{document}. Om vi bruker nedtrekksmenyen i R-Studio, kommer dette opp automatisk. Mellom begynnelsen og slutten skriver jeg selve dokumentet mitt: Her kan jeg skrive hva jeg vil, så lenge det ikke innebærer backslash eller krølleparantes; disse oppfattes per definisjon som koder. All tekst som følger etter at du har avsluttet dokumentet blir ikke lest. Det er praktisk, for eksempel når vi "debugger" LaTeX-koden.

3. *Når jeg ønsker å slå pdf av dokumentet*, må jeg først lagre dette der jeg ønsker å ha det: Filformatet vi jobber med er `.Rnw`. Siden holder det å trykke på "Compile PDF" øverst i tekstbehandlingsvinduet. Ut kommer en PDF.

Oppgave 1: Begynn med å opprette et dokument slik som er gjort i eksempelet. Sørg for at \og koden følger etterhverandre uten mellomrom.

1.2.2 En mer fancy innledning

Jeg kan gi LaTeX mye nyttig informasjon i innledningen. Ofte klipper og limer jeg innledningen min fra andre dokumenter, og gjør bare marginale endringer for hver gang.

Dokumentklasse Det første vi gjør er å velge dokumentklasse: for en typisk hjemmeoppgave vil `article` være utmerket. For en masteroppgave vil `book` være tingen. Det er også mulig å velge brev (`letter`), lage CV, poster, lysbildepresentasjon (`beamer`) osv.

```
\documentclass[a4paper, 12pt]{article}
```

Et praktisk tilleggsargument er skriftstørrelsen (vi kan velge mellom 11 og 12 punkter vanligvis) og papirstørrelsen. Det finnes selvsagt et hav av muligheter.

Tittelsiden Jeg kan legge til informasjon om tittel, forfatter og dato i innledningen (før jeg starter dokumentet). Dato vil alltid bli lagt til som default når du slår pdf. Det er veldig praktisk for å vite hvilken tekst-versjon dette er. Om du ikke ønsker dato, skriver du kommandoen med tom klamme: \date{}.

```
\title{LaTeX + R = Sant}
\author{Silje Synnøve Lyder Hermansen}
\date{\today}
```

Om du ønsker at pdf-en skal inneholde denne informasjonen, må du huske på å be latex lage tittelen *etter* at du har startet dokumentet:

```
\ begin{document}
\maketitle
```

LaTeX-pakker LaTeX er et åpent kildekode program og inneholder en haug med tilleggsfunksjoner skrevet av ymse forfattere, slik som R. Disse pakkene gjør det mulig å skrive formler, fargelegge tekst, skrive norske bokstaver, legge til hyperlinker i teksten osv. Noen av pakkene inneb?rer at dere kan bruke ekstra kommandoer senere i teksten, slik som i R. Andre pakker krever ingen ekstra kommandoer.

Vi lader inn pakkene vi ønsker å bruke i innledningen:

```
\usepackage[latin1]{inputenc}
```

En praktisk pakke for nordmenn er `inputenc` med tilleggsargument `[latin1]`. Den håndterer norske bokstaver: æ, ø og å.

```
\usepackage{amsmath}
```

Dette er den viktigste pakken deres: Den gjør det mulig å skrive matematiske symboler og formler.

```
\ usepackage[round,longnamesfirst]{natbib}
```

En nyttig pakke for masteroppgaven er `natbib`

```
\usepackage{url}
```

Denne pakken gjør det mulig å legge inn lenker i dokumentet ved hjelp av kommandoen \url{}.

1.3 Tekststruktur og bilder

1.3.1 Tekst og reserverte symboler

Man kan stort sett skrive vanlig i latex-dokumenter, men noen tegn er reserverte:

- \ er alltid reservert for kommunikasjon til LaTeX.
- Likeledes er % reservert for å kommentere ut tekst. LaTeX vil da slutte å lese teksten fram til neste gang vi skifter linje. % I Sweave blir teksten grønn. Da vet vi at vi ikke har slettet teksten, men den blir ikke med i det endelige dokumentet heller. Det er helt supert for tekstbiter dere er usikre på. Det er også en veldig hendig funksjon når man må "debugge" filen fordi den ikke vil kompilere.
- LaTeX fjerner doble mellomrom og enkle linjeskift. Om dere ønsker å skifte til et nytt avsnitt kan man enten skifte linje to ganger, eller be LaTeX gjøre dette: \newline{}. ønsker dere å skifte side er kommandoen tilsvarende: \newpage{}

1.3.2 Tekststruktur

Det finnes egne kategorier som sørger for tekststruktur. Vanlige koder vil være \textbf{} for **fet** tekst og \emph{} for *kursiv*.

Når dokumenttypen er "article", vil titler og undertitler best? av seksjoner, mens bøker også inneholder kapitler. RStudio gir oss muligheten til å bruke nedtrekksmenyer for de vanligste kategoriene: Trykk i så fall på "Format" og velg hva du vil ha.

```
\section{}
\subsection{}
\subsubsection{}
\paragraph{}
```

LaTeX teller for oss. Hvis vi ikke ønsker numererte seksjoner, kan vi legge en stjerne til mellom koden og teksten: \seksjon*{}.

Om vi ønsker en innholdsfortegnelse i starten av dokumentet, kan vi legge til dette etter at vi har begynt dokumentet:

```
\ begin{document}
\tableofcontents
```

Andre muligheter er kulepunkt-lister og numererte lister. Disse lages ved at man åpner et "miljø" ("environment") hvor man kan skrive egne koder. Man må alltid huske å lukke miljøet igjen, før man fortsetter skriveprosessen.

```
\begin{itemize}
\item Blablabla
\end{itemize}
```

1.3.3 Grafikker

En av de virkelig sterke sidene med LaTeX er hvordan den importerer filer.

Grafikker i pdf-format, jpg og liknende kan importeres ved å åpne et figur-miljø ("figure environment") hvor du fritt kan importere bilder. Når bildene ligger i samme mappe som Rnw-fila, holder det å kalle filnavnet. Hvis filene finnes på to forskjellige steder, må vi skrive filadressen.

```
\begin{figure}
\includegraphics{latex.jpg}
\end{figure}
```

LaTeX vil selv velge hvor den synes det er penest med en figur i teksten. Det er mulig å legge føringer med tilleggskommandoer: `[h]` (forsøk å legge bildet omtrent her), `[H]` (legg bildet HER!). Andre muligheter er `bt` (bunnen eller toppen) osv.

Vil dere sentrere bildet, kan dere det.

```
\begin{figure}[h]
\centering
\includegraphics{latex.jpg}
\end{figure}
```

Dere kan også legge til en billedtekst:

```
\begin{figure}
\includegraphics{latex.jpg}
\caption{Her kan jeg skrive billedteksten min}
\end{figure}
```

Størrelsen kan dere regulere med tilleggsargumenter til includegraphics. For eksempel kan vi definere bildet i forhold til bredden på teksten:

```
\includegraphics[width=0.5\textwidth]{latex.jpg}
```

1.4 Skrive matematikk i LaTeX

LaTeX har et eget "mattemiljø". Dette kan vi kalle fram på to måer:

- Vi kan pakke inn teksten vår i dollartegn $1+2$. Dette er fint når de matematiske symbolene er en del av den flytende teksten.
- Vi kan også åpne miljøet på tradisjonelt vis:

```
\begin{equation}
Her er matematikken.
\end{equation}
```

LaTeX numererer alle likninger. Hvis vi ønsker å fjerne likningene, kan vi bruke stjernetegnet igjen.

```
\begin{equation*}
\end{equation*}
```

Alle greske bokstaver har kode som er lik det greske navnet sitt. Små bokstaver skrives med små bokstaver (\sigma, σ), store bokstaver skrives med stor, så små bokstaver (\Sigma, Σ).

Andre nyttige tegn er:

- \frac{*teller*}{*nevner*}: $\frac{teller}{nevner}$
- \times: $\times$
- \in: $\in$
- \pm: $\pm$

En online-wysiwyg for formler Når jeg skal skrive store likninger, ønsker jeg ofte å se hva jeg skriver underveis uten å vente på at hele dokumentet mitt skal kompilere. Da bruker jeg en online LaTeX-kompilator. Den gir også en nedtrekksmeny som gjør det mulig å hente inn tegn som vi vet hvordan ser ut, men som vi ikke kan LaTeX-koden til: `https://www.codecogs.com/latex/eqneditor.php`.

2 R i knitr

Når vi ønsker å kommunisere med R, må vi informere RStudio om dette. Knitr kan snakke med R på to måter:

2.1 Store R-bolker

2.1.1 Opprette R-bolker og gi dem navn

All større kommunikasjon med R åpnes ved å opprette en "bolk" eller et "r-miljø". Dette kan du gjøre på to måter: 1) Med et "klikk" på det grønne ikonet øverst i høyre hjørne av notatblokken, eller 2) med to "større enn", to "mindre enn" og et "er lik"-tegn. Bolken avsluttes med en alfakrøll.

RStudio markerer at vi nå snakker til R ved å skifte farge til lysegrønn. Her kan dere gjøre alle tingene dere vanligvis gjør i R.

```
<<>>=

> 2+2

[1] 4

@
```

Dette er hva som gjør knitr til et utmerket verktøy for reproduserbar forskning: Her kan dere laste inn dataene deres, omkode dem, analysere dem og rapportere resultatene i hjemmeoppgaven. Dere kan gi hver R-bolk et eget navn. Dere kan navigere kjapt til de ulike R-bolkene ved å bruke nedtrekksmenyen nederst til venstre i RStudios notatblokk. Det gjør det lettere finne de ulike utregningene i dokumentet. Dokumentet vil ikke kompilere når dere har feil enten i LaTeX-koden eller R-koden. Når det er feil i R-koden vil RStudio fortelle dere hvilken R-blokk som skaper problemer:

```
<<label='Min forste R-bolk'>>=

@
```

2.1.2 Skal R-innholdet i boksen vises?

Dere kan velge om selve R-koden (`echo=TRUE`) og/eller svarene (`results='asis'`) skal vises til leseren; det er hva jeg gjør når jeg skriver seminarnotatene til R-kurset. Eventuelt kan jeg be om at de ikke rapporteres. Det vil være den relevante løsningen for hjemmeoppgaven (koden skal rapporteres i slutten av dokumentet):

```
<<results='hide', echo=FALSE>>=

@
```

Vi kan velge hva vi ønsker å gjøre for hver kode-blokk, men det tar tid, så det kan være lurt å starte dokumentet med å sette hvilke standardinstillinger du ønsker. For å gjøre dette, må du hente inn `knitr`-biblioteket.

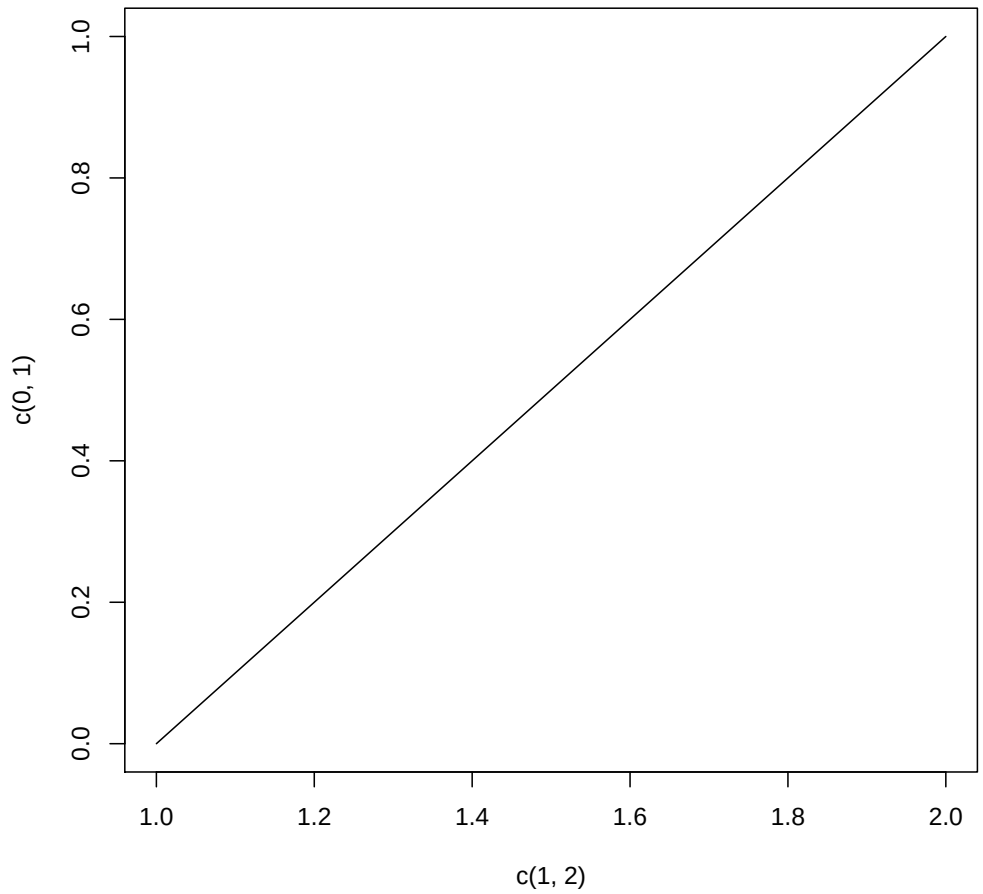
Her velger jeg at LaTeX ikke skal repetere alt jeg sier. Det betyr at koden blir evaluert, vi får svar og svarene blir rapportert, men ikke min egen kode.

```
knitr::opts_chunk$set(echo = TRUE)
```

2.1.3 Grafikker

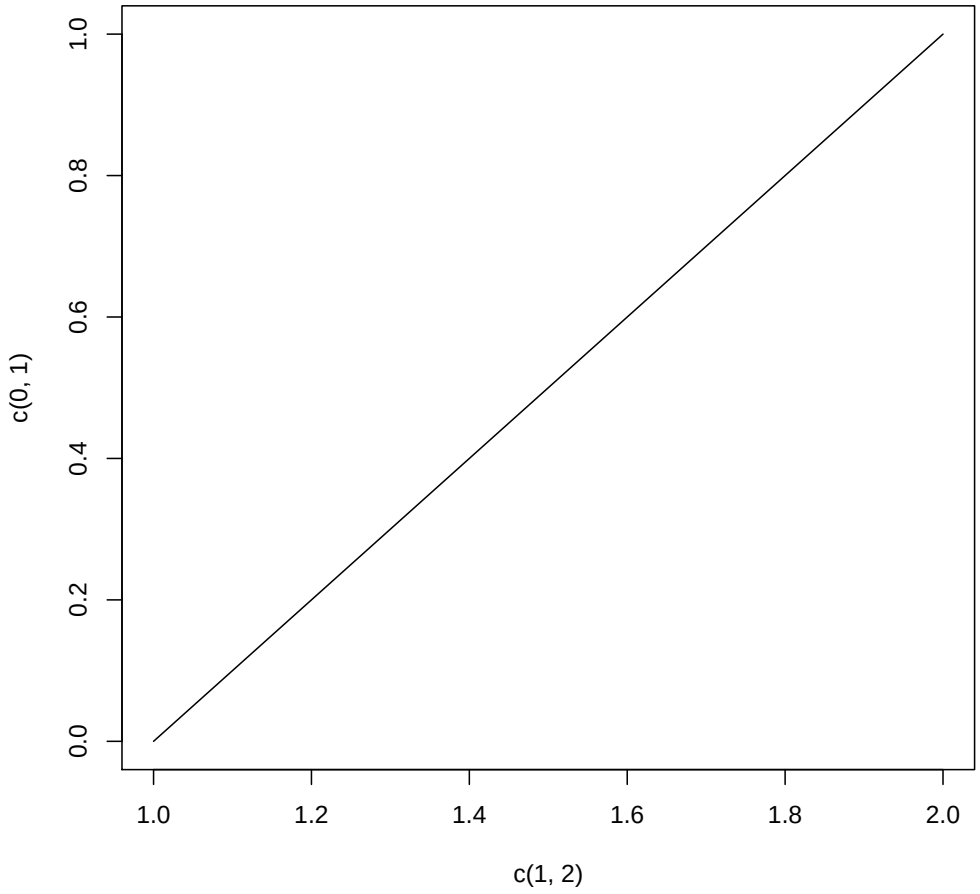
Med knitr kan man gjøre tilsvarende med grafikker:

```
plot(x=c(1,2),
     y=c(0,1),
     type="l")
```




```
<<results='asis', fig.keep='all'>>=
```

```
plot(x=c(1,2),
     y=c(0,1),
     type="l")
```



@

Personlig foretrekker jeg å eksportere, for så å importere:

```
pdf("MittForstePlott.pdf")
plot(x=c(1,2),
     y=c(0,1),
     type="l")
dev.off()
```

```
\begin{figure}
\includegraphics{MittForstePlott.pdf}
\end{figure}
```

2.2 Små R-referanser

En siste – genial – ting med Sweave er at man kan gj?re sm? referanser til R i teksten med kommandoen: \Sexpr{}. Inne i parantesen kan jeg putte R-objekter (f.eks: p-verdien), et indeksert objekt (f.eks: en spesiell rute i resultat-tabellen deres) eller utregninger. Det gjør at dere kan tolke resultatene deres skriftlig, omkode variablene og kjøre analysa på nytt, og teksten vil bli oppdatert!