R Notebooks og reproduserbarhet

Assignment 1 i MSB105 Data Science - innleveringsfrist 24.09.20

Katrine Hope Karl-Gunnar Severinsen

Innhold

1. Innledning

I denne oppgaven ønsker vi å se på reproduserbarhet og viktigheten av det. Vi vil i denne sammenhengen ta oss innom temaet om bruk av "R Notebook" i Rstudio.

Vi vil se på teori rundt reproduserbarhet og R Notebook, som vi også vil knytte opp til en analyse.

2. Teori

2.1 Replikerbarhet/reproduserbarhet

Det er ønskelig at vitenskapelige oppdagelser og fremskritt skal være robuste og pålitelige fordi vi ønsker å ha tillit til at resultatene er riktig og at undersøkelsene er gjort på en tilfredsstillende måte (Bollen et al., 2015). I følge McNutt (2014) er det viktig for at vitenskapen skal utvikle seg at funnene baserer seg på troverdige funn. Mange forskere mener at reproduserbarhet er en viktig fremgangsmåte for å kunne validere funnene sine, men dette har vist seg å være veldig vanskelig å gjennomføre (McNutt, 2014). Noe av det som kan være med å gjøre det vanskelig å reprodusere, men også replikere, er at tidligere studier kan mangle data eller koder, det kan være feil i programvaren som er sendt med, det kan mangle dokumentasjon, samt er det noen forskere som ikke ønsker å gi fra seg sine data og koder på studiene sine. National Science Foundation i USA viser til at for at vitenskap skal være robust og pålitelig må funnene være reproduserbar, replikerbart og generaliserbart (Bollen et al., 2015). I henhold til National Science Foundation blir de 3 påstandene definert som (Bollen et al., 2015):

1. **Reproduserbarhet**:
    - Å repodusere vil si å gjøre studiet på nytt, med de samme dataene og med den samme metodikken, men gjort av en uavhengig part.
2. **Replikerbarhet**:

- Å replikere vil si at et studie gjøres på nytt av en uavhengig part, gjerne med nye data og at resultater og konklusjon er lik som tidligere.

3. **Generaliserbarhet**:

    - Vil si at et studie kan gjøres på nytt med ny data og ny sammensetning, men komme frem til lignende konklusjoner.

Reproduksjon har potensiale til å være en minimumsstandard i følge Peng (2011), når full uavhengig replikasjon ikke er tilgjengelig. Dersom den nye studien kan bekrefte resultatene og konklusjonen fra et tidligere studie via replikasjon blir dette sett på som en vitenskapelig gullstandard (Jasny et al., 2011). Peng (2011) anser replikasjon som den ultimate standarden i likhet med Jasny et al. (2011).

2.2 Problemets omfang

Som forklart tidligere så kan tidligere studier mangle data, koder, fremgangsmåte, feil i programvare osv, samt er det flere forskere som ikke ønsker å gi fra seg all informasjon om sine studier. Dette gjør at det er nærmest umulig for forskere i senere tid å reprodusere en tidligere studie. Siden det er mer enn et problem vil det også være flere løsninger som må implementeres for at full reproduksjon skal være mulig.

Vi kan dele løsningene inn i tekniske løsninger og menneskelige løsninger. Den menneskelige komponenten i problemstillingen er at mange forskere ikke ønsker full åpenhet til data, koder, programvare osv i sine studier. Dette har ikke vært standard retningslinjer tidligere og det viser seg at dette kan fremdeles i dag være en utfordring, det trengs derfor klare retningslinjer om hva som bør anses som god forskning i henhold til kravene om reproduksjon og replikasjon. I den tekniske delen kan problemet ofte være at det mangler data, koder, fremgangsmåte eller at det er feil i programvare. Ved å integrere koder i selve artikkelen, selv om de ikke trenger nødvendigvis å være synlig, skal det være mulig for andre forskere å reprodusere og replikere studien.

Her er en oversikt over hva som skal sendes til tidsskriftene:

- Dokument med tekst.
- Kode til å lese inn dataen med.
- Kode til å kalkulere de ulike modellene.
- Kode for å teste modellene.
- Kode for å generere rapport av selve resultatene.

Hovedpunktet er at all data og koder vil bli sendt sammen med et fullstendig reproduserbart dokument.

2.2.1 Mulig løsning (teoretisk plan) I følge Gentleman og Lang (2007) er det viktig, kanskje også essensielt, å integrere beregninger og koder som brukes i dataanalyser, metodebeskrivelser og simuleringer. Dette kan gjøres via et kompendium i henhold til Gentleman og Lang (2007). Kompendium vil si en samling av de ulike elementene, som tekst, kode, data, metodikk og lignende, og dette skal settes sammen som en enhet for å kunne distribueres, håndteres og oppdateres. Dersom kompendiet er laget riktig skal forfattere enkelt kunne reprodusere resultatene.

Dynamiske dokumenter er en ordnet sammensetning av stykker ("chunks") med kode eller tekst. "Code chunk" kan brukes som et middel som gjengir utdata i dokumenter, eller bare for å vise kode for illustrasjon, siden "code chunks" utfører beregninger som trengs for å produsere riktig utdata i dokumentet, men også for å produsere mellomresultater brukt på tvers av forskjellige code chunks (Gentleman og Lang, 2007). "Text chunks" er ment å være formatert for lesing og beskriver som oftest problemet, koden, resultatene og tolkningen (Gentleman og Lang, 2007). Et optimalt kompendium vil derfor være et dynamisk dokument siden alle komponenter er til stedet for reproduksjon. I analysedelen av denne oppgaven vil vi vise i R Notebook i programmet Rstudio hvordan vi legger inn text- og code chunks.

2.2.2 Mulig løsning (R Notebooks): Tidligere var det R Markdown som ble brukt, men der fikk man ikke all tekst, koding og utdata (resultat) inn i samme dokument, som oftest ble dette delt opp i ulike vindu. R Notebook er den nyeste utgivelsen fra Rstudio på programutvikling. Rstudio er en IDE (Integrated Developer Enviroment) for alle R relaterte

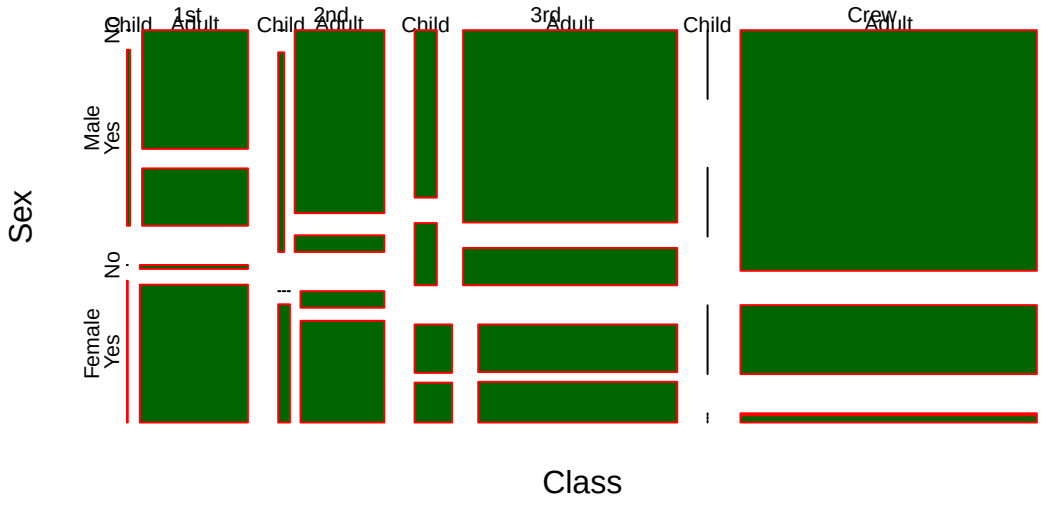
ting. Rstudio er et gratis programvare som du både kan laste ned lokalt på datamaskinen eller du kan jobbe over internett. Alle de vanlige platformene skal ha mulighet til å bruke R Studio som Mac, Window og Linux. R studio må knyttes sammen med andre programvarer og pakker for å kunne oppnå reproduserbarhet og replikerbarhet som er målet, se punkt 3.3 for mer info. RStudio kan knyttes opp mot github og det gjør at man har arbeidet sitt skylagret. En R Notebook er et R Markdown dokument med "code- og text chunks" som kan hente inn data og koder, utføre beregning ut i fra formlene som legges inn i "code chunks" og R Notebook vil vise oss resultatene direkte i samme dokument som vi arbeider i [xie2018].

3. Analyse

I dette kapittelet vil vi gå gjennom noen eksempler hvordan vi kan lage og vise frem data ved hjelp av *RStudio*. Vi vil også se på hva som kan gjøres for å vurdere viktigheten av å kunne presentere reproduserbare data, og om dette vil la seg gjøre på en enkel måte.

3.1 Titanic

Vi begynner først med å lage en oversikt over hvor mange som overlevde den skjebnesvangre jomfruturen til Titanic i 1912. Dette kan vi gjøre via datasettet *"Titanic"*, utviklet av R Core Team (2020).



Denne oversikten er gjerne litt rotete og forteller oss lite om nøyaktig hvor mange passasjerer som tilhører de forskjellige boksene. Den gir oss altså ikke stort mer enn en indikasjon på forholdet mellom død og overlevelse. Men vi kan tydelig se at det var generelt svært mange fra de høyere klassene, og da særlig blant barn og kvinner som overlevde.

Vi kan vise dette på følgende måte, som viser det totale antallet overlevende blant barn og voksne. Dette vil også fungere som et eksempel på en "chunk" som inneholder både kode og tekst.

```
## Vi summerer variablene hentet fra "Help"-funksjonen i RStudio.
apply(Titanic, c(3, 4), sum)
```

```
##         Survived
## Age        No Yes
##   Child    52  57
##   Adult  1438 654
```

```r
apply(Titanic, c(2, 3, 4), sum)
```

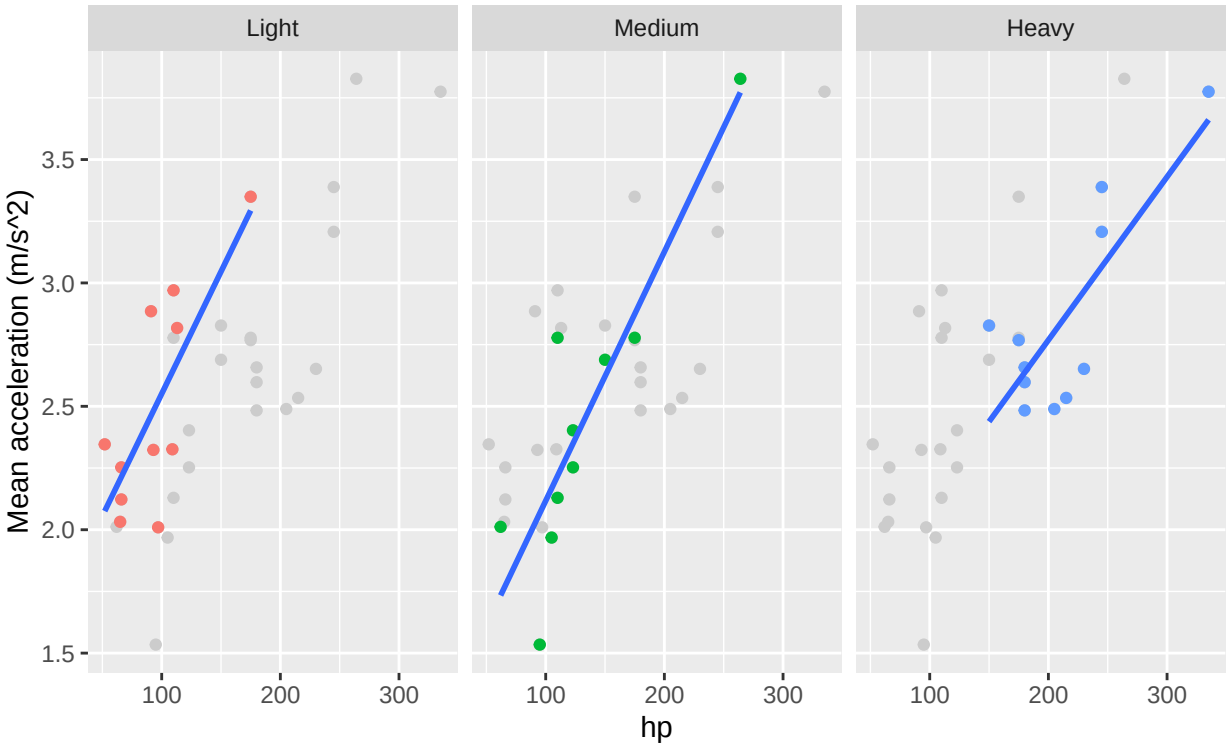
```
## , , Survived = No
##
##         Age
## Sex      Child Adult
##   Male      35  1329
##   Female    17   109
##
## , , Survived = Yes
##
##         Age
## Sex      Child Adult
##   Male      29   338
##   Female    28   316
```

3.2 Eksepel fra forelesning

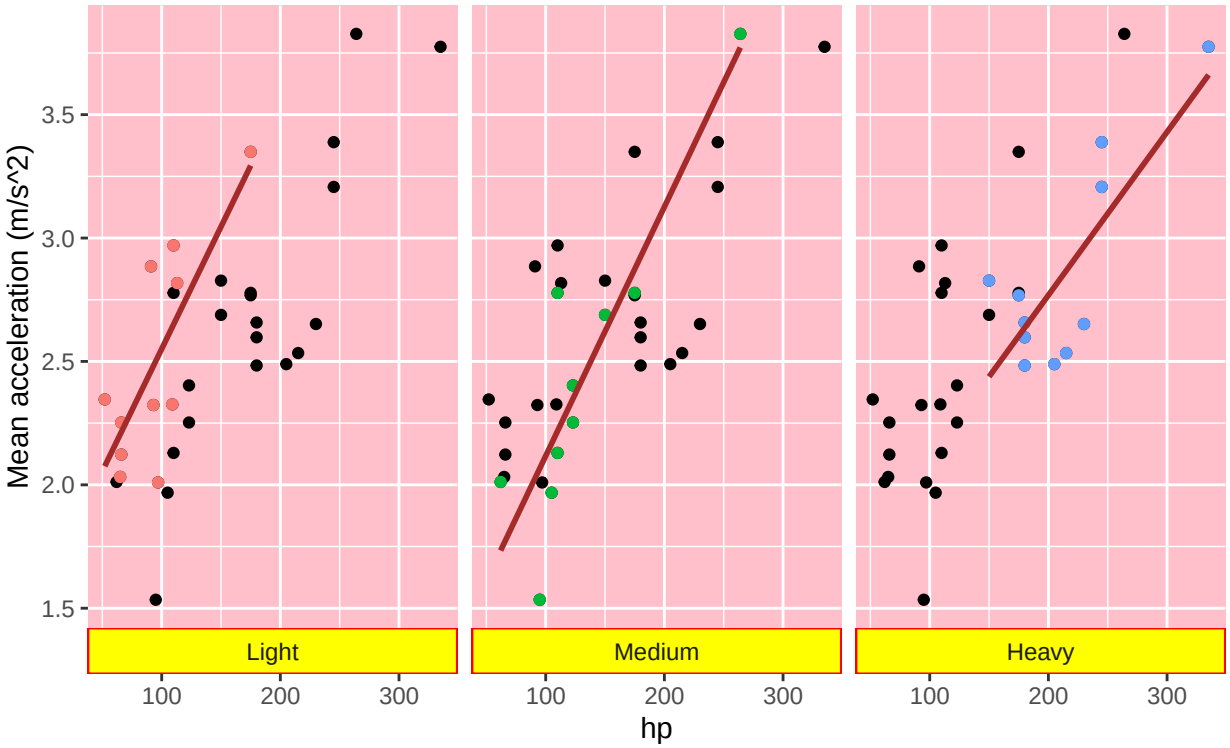
Vi kan også bruke pakken ggplot (utviklet av: Wickham, 2016) til å lage spennende og forklarende grafer dersom vi ønsker å forklare variabler i datasett.

De følgende grafene er da hentet fra forelesningsnotatene (Gjestland, 2020) og tar utgangspunkt i datasettet *mtcars*. Den første grafen er original, mens den andre lettere modifisert for å vise hvor lett det er å endre utseende og farger på grafen etter eget ønske.

```
## `geom_smooth()` using formula 'y ~ x'
```




```
## `geom_smooth()` using formula 'y ~ x'
```



3.3 Sessioninfo og koder

I dette kapittelet har vi vist eksempler på "forskningsresultater", der vi både viser og unnlater å vise hvordan vi kom frem til "sluttproduktet". Det skal være forholdsvis enkelt å reprodusere våre resultater. **Men** det forutsetter at andre brukere har installert akkurat nøyaktig de samme pakkene i *RStudio*, som vi har benyttet oss av. En løsning på denne utfordringen kan være å legge ved følgende "chunk". Den vil produsere en oversikt over nødvendig programvare, for å kunne reprodusere nøyaktig samme resultater som her.

```
sessionInfo()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19041)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Norwegian Bokmål_Norway.1252
## [2] LC_CTYPE=Norwegian Bokmål_Norway.1252
## [3] LC_MONETARY=Norwegian Bokmål_Norway.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=Norwegian Bokmål_Norway.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggpubr_0.4.0    forcats_0.5.0   stringr_1.4.0   dplyr_1.0.1
##  [5] purrr_0.3.4     readr_1.3.1     tidyr_1.1.1     tibble_3.0.3
##  [9] tidyverse_1.3.0 tinytex_0.25    ggplot2_3.3.2
```

```
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.5         lattice_0.20-41    lubridate_1.7.9    assertthat_0.2.1
##  [5] digest_0.6.25      R6_2.4.1           cellranger_1.1.0   backports_1.1.8
##  [9] reprex_0.3.0       evaluate_0.14      httr_1.4.2         pillar_1.4.6
## [13] rlang_0.4.7        curl_4.3           readxl_1.3.1       rstudioapi_0.11
## [17] data.table_1.13.0  car_3.0-9          blob_1.2.1         Matrix_1.2-18
## [21] rmarkdown_2.3      labeling_0.3       splines_4.0.2      foreign_0.8-80
## [25] munsell_0.5.0      broom_0.7.0        compiler_4.0.2     modelr_0.1.8
## [29] xfun_0.16          pkgconfig_2.0.3    mgcv_1.8-31        htmltools_0.5.0
## [33] tidyselect_1.1.0   rio_0.5.16         fansi_0.4.1        crayon_1.3.4
## [37] dbplyr_1.4.4       withr_2.2.0        grid_4.0.2         nlme_3.1-148
## [41] jsonlite_1.7.0     gtable_0.3.0       lifecycle_0.2.0    DBI_1.1.0
## [45] magrittr_1.5       scales_1.1.1       zip_2.1.1          carData_3.0-4
## [49] cli_2.0.2          stringi_1.4.6      farver_2.0.3       ggsignif_0.6.0
## [53] fs_1.5.0           xml2_1.3.2         ellipsis_0.3.1     generics_0.0.2
## [57] vctrs_0.3.2        openxlsx_4.1.5     tools_4.0.2        glue_1.4.1
## [61] hms_0.5.3          abind_1.4-5        yaml_2.2.1         colorspace_1.4-1
## [65] rstatix_0.6.0      rvest_0.3.6        knitr_1.29         haven_2.3.1
```

> *Det passer nok best å kjøre en slik kode helt til slutt i dokumentet, da det gjerne tar en del plass og kan virke forstyrrende inne i en artikkel (som her).*

Som vi var inne på i forrige kapittel, kan en god løsning være å samle alle kodene i et eget *kode-appendiks* i slutten av dokumentet. Dette er noe vi har valgt å gjøre for å vise frem muligheten. Vi benytter oss av følgende kode i en egen "chunk" i appendikset *"Koder"*:

`{r ref.label=knitr::all_labels(), echo = T, eval = F}`, som vi har hentet fra Xie (2018).

4. Konklusjon

Vi ser at ved å bruke R Notebook og lage et dynamisk dokument med både data, koder, fremgangsmåte, resultat og referanser, skal det være mulig å reprodusere, replikere og generalisere et studie. Det som kan gjøre det litt vanskelig er kompleksiteten på programmet, fordi selve kode- og data delen i RStudio er ikke det som er vanskeligst siden man kan skrive både formel og kode og få utdata i samme dokument, men det er alle programmer og pakker som skal snakke i lag som gjør arbeidet litt mer komplekst. Den viktigste komponenten vil uansett alltid være at forskeren ønsker å dele sin fulle utredelse slik at forskningen faktisk kan brukes av andre uavhengige parter, det er derfor viktig at det blir flere standard retningslinjer for hva som er minimumskrav for forskere ved utgivelse av undersøkelser.

Referanse

Bollen, K., Cacioppo, J. T., Krosnick, J. A., Olds, J. L., og Kaplan, R. M. (2015). *Social, Behavioral, and Economic Sciences Perspectives on Robust and Reliable Science* (Report of the Subcommittee on Replicability in Science Advisory Committee to the National Science Foundation Directorate for Social, Behavioral, and Economic Sciences). NSF.

Gentleman, R., og Lang, D. T. (2007). Statistical Analyses and Reproducible Research. *Journal of Computational and Graphical Statistics*, *16*(1), 1–23. https://doi.org/10.1198/106186007X178663

Gjestland, A. (2020). *R Notebook Showcasing Ggplot*. https://elastic-turing-41462a.netlify.app/presentation

Jasny, B. R., Chin, G., Chong, L., og Vignieri, S. (2011). Again, and Again, and Again. *Science*, *334*(6060), 1225–1225. https://doi.org/10.1126/science.334.6060.1225

McNutt, M. (2014). Reproducibility. *Science*, *343*(6168), 229–229. https://doi.org/10.1126/science.1250475

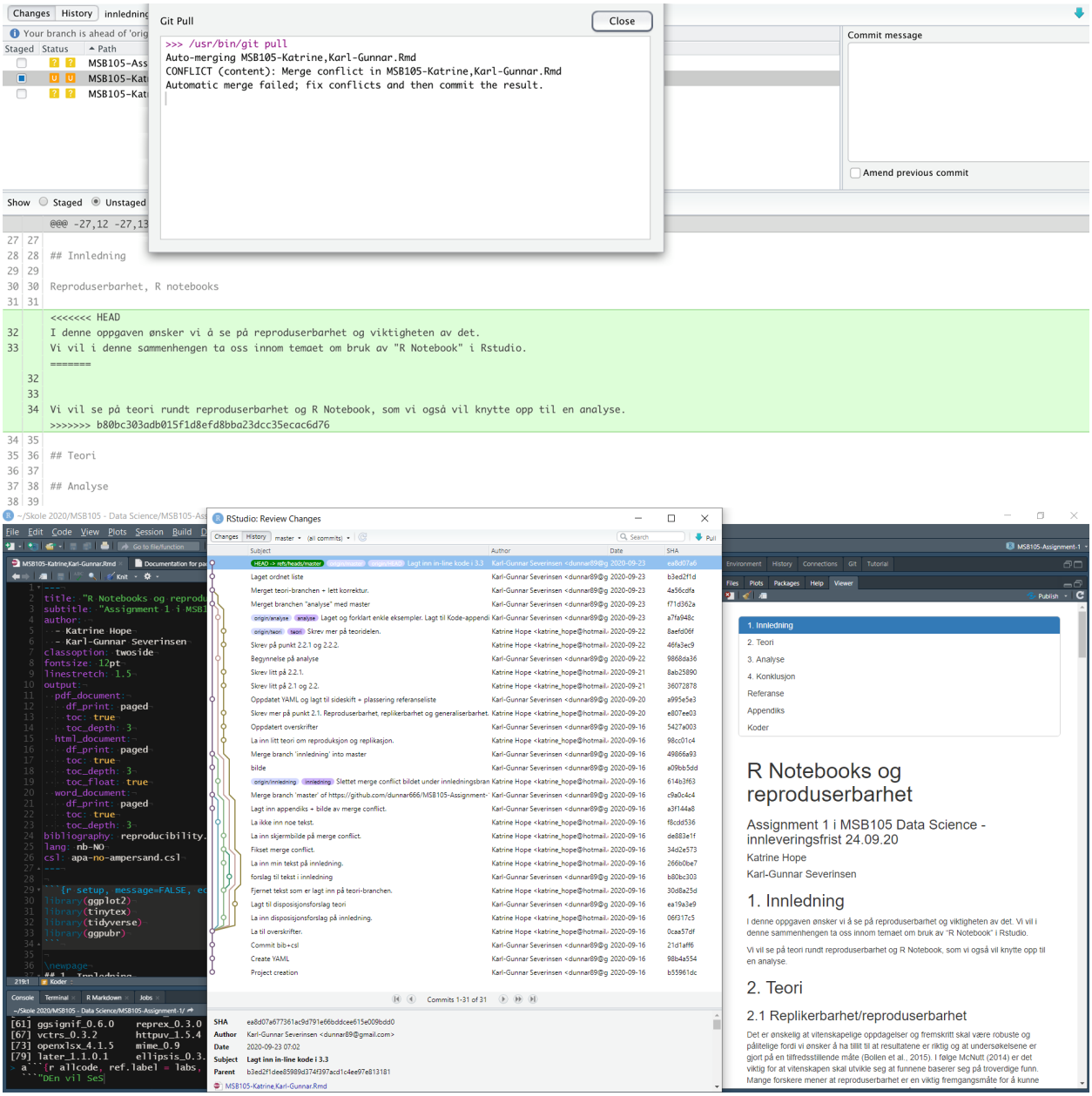
Peng, R. D. (2011). Reproducible Research in Computational Science. *Science*, *334*(6060), 1226–1227. https://doi.org/10.1126/science.1213847

R Core Team. (2020). *R: A Language and Environment for Statistical Computing* [Manual]. R Foundation for Statistical Computing.

Wickham, H. (2016). *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Xie, Y. (2018). *How to Put All Your Code in the Appendix in R Markdown*. https://yihui.org/en/2018/09/code-appendix/.

Appendiks


Changes
History
Git Pull
Close
>>> /usr/bin/git pull
Auto-merging MSB105-Katrine,Karl-Gunnar.Rmd
CONFLICT (content): Merge conflict in MSB105-Katrine,Karl-Gunnar.Rmd
Automatic merge failed; fix conflicts and then commit the result.
Commit message
Amend previous commit
Show
Staged
Unstaged
Innledning
Reproduserbarhet, R notebooks
<<<<<<< HEAD
I denne oppgaven ønsker vi å se på reproduserbarhet og viktigheten av det.
Vi vil i denne sammenhengen ta oss innom temaet om bruk av "R Notebook" i Rstudio.
=======
Vi vil se på teori rundt reproduserbarhet og R Notebook, som vi også vil knytte opp til en analyse.
>>>>>>> b80bc303adb015f1d8efd8bba23dcc35ecac6d76
Teori
Analyse
RStudio: Review Changes
R Notebooks og reproduserbarhet
Assignment 1 i MSB105 Data Science - innleveringsfrist 24.09.20
Katrine Hope
Karl-Gunnar Severinsen
1. Innledning
2. Teori
2.1 Replikerbarhet/reproduserbarhet

Koder

```
library(ggplot2)
library(tinytex)
library(tidyverse)
library(ggpubr)
mosaicplot(Titanic, main = "Overlevelse på Titanic", color = "darkgreen", border = "red"
## Vi summerer variablene hentet fra "Help"-funksjonen i RStudio.
apply(Titanic, c(3, 4), sum)
apply(Titanic, c(2, 3, 4), sum)
hp_acc <- data.frame(hp = mtcars$hp, acc = 1609.347/(2*mtcars$qsec^2))

ggplot(data = mtcars, mapping = aes(x = hp, y = 1609.347/(2*qsec^2))) +
  geom_point(data=hp_acc, mapping = aes(x = hp, y = acc), colour= "grey80") +
  facet_wrap(~cut_number(wt, n =3 , labels = c("Light", "Medium", "Heavy"))) +
  geom_point(aes(colour = cut_number(wt, n = 3, labels = c("Light", "Medium", "Heavy")))
             show.legend = FALSE) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(y="Mean acceleration (m/s^2)", colour = "Weight")
hp_acc <- data.frame(hp = mtcars$hp, acc = 1609.347/(2*mtcars$qsec^2))

ggplot(data = mtcars, mapping = aes(x = hp, y = 1609.347/(2*qsec^2))) +
  geom_point(data=hp_acc, mapping = aes(x = hp, y = acc), colour= "black") +
  facet_wrap(~cut_number(wt, n =3 , labels = c("Light", "Medium", "Heavy")), strip.posit
  theme(panel.background = element_rect(fill = "pink"), strip.background = element_rect
  geom_point(aes(colour = cut_number(wt, n = 3, labels = c("Light", "Medium", "Heavy")))
             show.legend = F) +
  geom_smooth(method = "lm", color = "brown", se = FALSE) +
  labs(y="Mean acceleration (m/s^2)", colour = "Weight")
sessionInfo()
```