cphbusiness

# Dataanalyse

## Tidy text☺

# Agenda

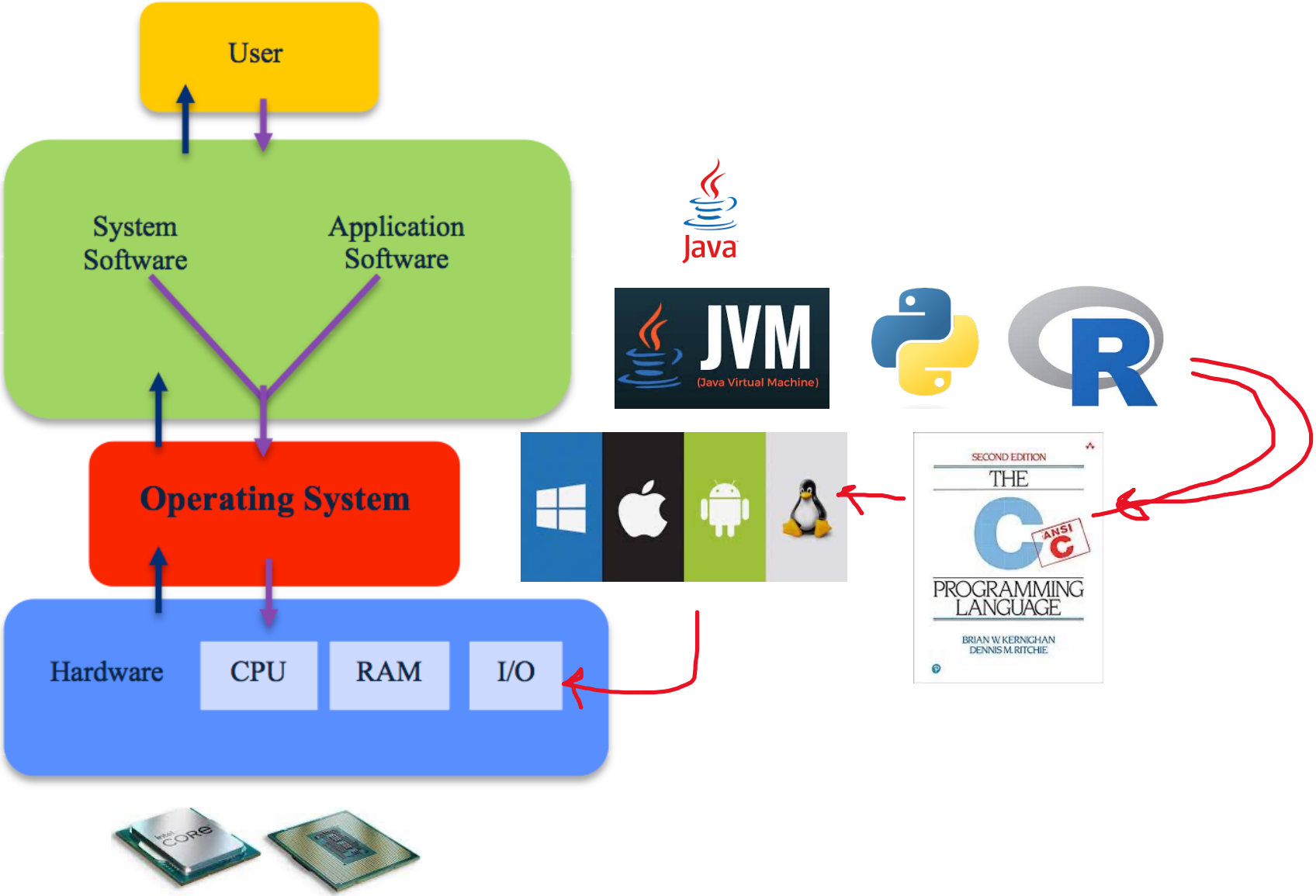
- Hardware issues
- Kolb’s læringsstile
- Heidi’s kode
- Sentida
- Tidy og Home vs EDC

cphbusiness

# Hardware issues ..

preeya@ubuntu: ~/Desktop/KanakPriyaSelvarani_4641619_Assn2

```
preeya@ubuntu:~$ cd ~/Desktop/KanakPriyaSelvarani_461619_Assn2/
bash: cd: /home/preeya/Desktop/KanakPriyaSelvarani_461619_Assn2/: No such file o
r directory
preeya@ubuntu:~$ cd ~/Desktop/KanakPriyaSelvarani_4641619_Assn2/
preeya@ubuntu:~/Desktop/KanakPriyaSelvarani_4641619_Assn2$ gcc CountryServer.c -
o CountryServer
In file included from CountryServer.c:1:0:
CountryServer.h:32:20: fatal error: iostream: No such file or directory
 #include <iostream>
                    ^
compilation terminated.
```


User
System Software
Application Software
Operating System
Hardware
CPU
RAM
I/O
Java
JVM
(Java Virtual Machine)
R
SECOND EDITION
THE
C
ANSI C
PROGRAMMING LANGUAGE
BRIAN W. KERNIGHAN
DENNIS M. RITCHIE

cphbusiness

```c
#include <stdio.h>

 int main(int argc, char** argv) {
  printf("h");
}
```

```
| => objdump -d  -S h.o

h.o:    file format mach-o 64-bit x86-64


Disassembly of section __TEXT,__text:

0000000000000000 <_main>:
;  int main(int argc, char** argv) {
       0: 55                            pushq   %rbp
       1: 48 89 e5                      movq    %rsp, %rbp
       4: 48 83 ec 20                   subq    $32, %rsp
       8: 89 7d fc                      movl    %edi, -4(%rbp)
       b: 48 89 75 f0                   movq    %rsi, -16(%rbp)
;   printf("h");
       f: 48 8d 3d 14 00 00 00          leaq    20(%rip), %rdi  # 2a <_main+0x2a>
      16: b0 00                         movb    $0, %al
      18: e8 00 00 00 00                callq   0x1d <_main+0x1d>
      1d: 31 c9                         xorl    %ecx, %ecx
      1f: 89 45 ec                      movl    %eax, -20(%rbp)
; }
      22: 89 c8                         movl    %ecx, %eax
      24: 48 83 c4 20                   addq    $32, %rsp
      28: 5d                            popq    %rbp
      29: c3                            retq
```

Mnemonic

Opcode

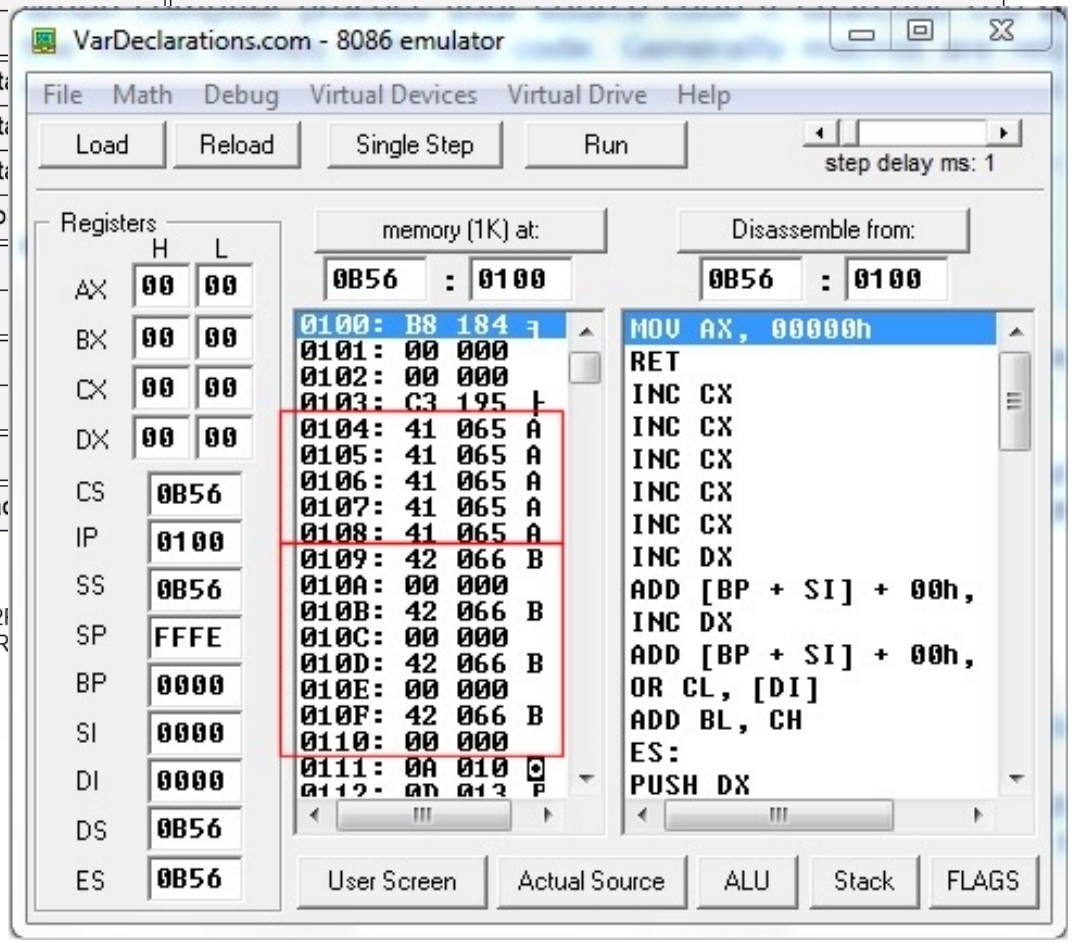
Operands

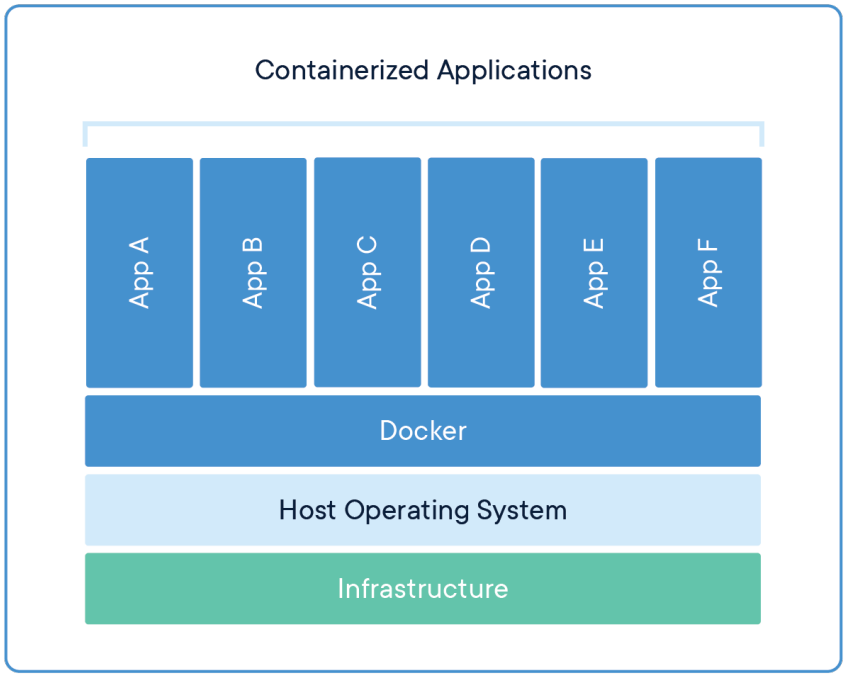
Description

| | | | |
|---|---|---|---|
| INC | $0 | <tar>[4R,12R,Mx] <tar-aop> | Increment contents of register or memory location[4,5] |
| DEC | $1 | <tar>[4R,12R,Mx] <tar-aop> | Decrement contents of register or memory location[4,5] |
| ADDC | $2 | <src>[CV,4R,Mx] <tar>[4R,Mx] <src-aop> <tar-aop> | Add contents of <src> and <tar> (with carry); store result in <tar>[1,2,3,4,5] |
| SUBB | $3 | <src>[CV,4R,Mx] <tar>[4R,Mx] <src-aop> <tar-aop> | Subtract contents of <src> from <tar> (with borrow); store result in <tar>[1,2,3,4,5] |
| ROLC | $4 | <tar>[4R,Mx] <tar-aop> | Rotate contents of <tar> left through the carry (C) status bit/flag[4,5] |
| RORC | $5 | <tar>[4R,Mx] <tar-aop> | |
| AND | $6 | <src>[CV,4R,Mx] <tar>[4R,Mx] <src-aop> <ta | |
| OR | $7 | <src>[CV,4R,Mx] <tar>[4R,Mx] <src-aop> <ta | |
| XOR | $8 | <src>[CV,4R,Mx] <tar>[4R,Mx] <src-aop> <ta | |
| CMP | $9 | <src1>[CV,4R,Mx] <src2>[4R,Mx] <src1-aop | |
| PUSH | $A | <src>[CV,4R,12R,Mx] <src-aop> | |
| POP | $B | <tar>[4R,12R,Mx] <tar-aop> | |
| JMP | $C | <0/1 #sb> <tar-aop> | |
| JSR | $D | <0/1 #sb> <tar-aop> | |
| NOP | $E | -- | |
| MOV | $F | <src>[CV,4R,12R,Mx] <tar>[4R,12R,Mx] <src-ac | |

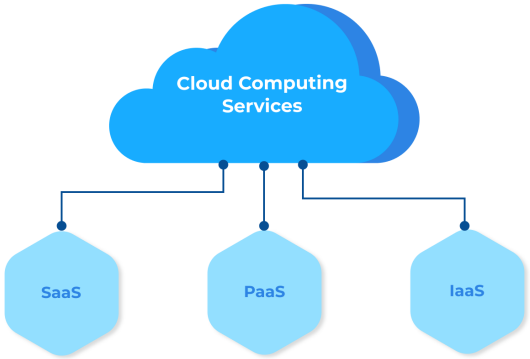
<src> = Source
<tar> = Target (destination)
<src-aop> = Additional operand (none if source is a 4R/12
<tar-aop> = Additional operand (none if target is a 4R/12R

0/1 = 1-bit logic 0 or 1 value
#sb = 3-bit value specifying status bit to test (0-7)






## Cloud Computing Services



cphbusiness

# Kolb

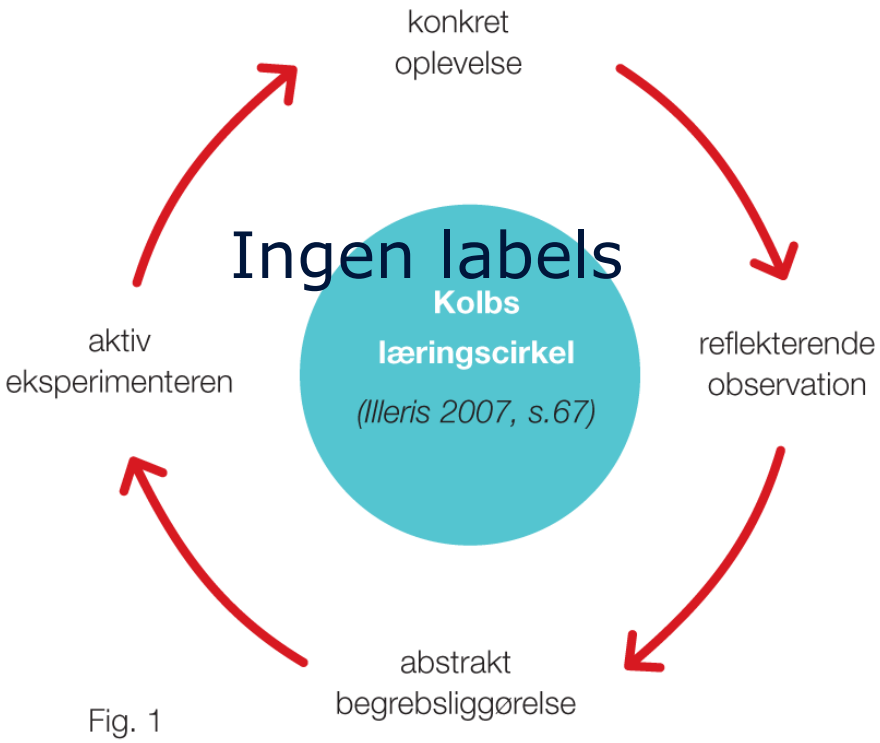
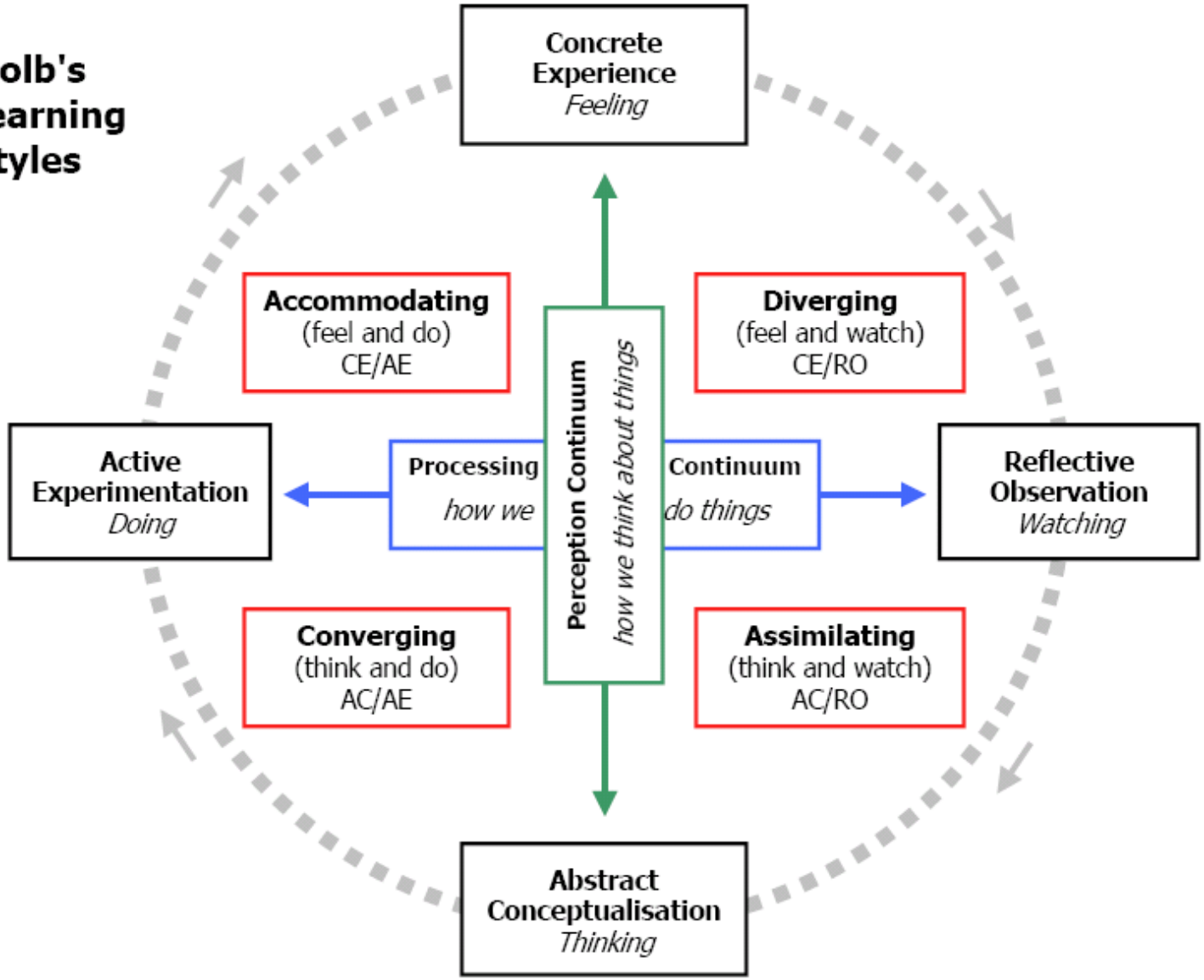



Fig. 1

# Kolb's learning styles

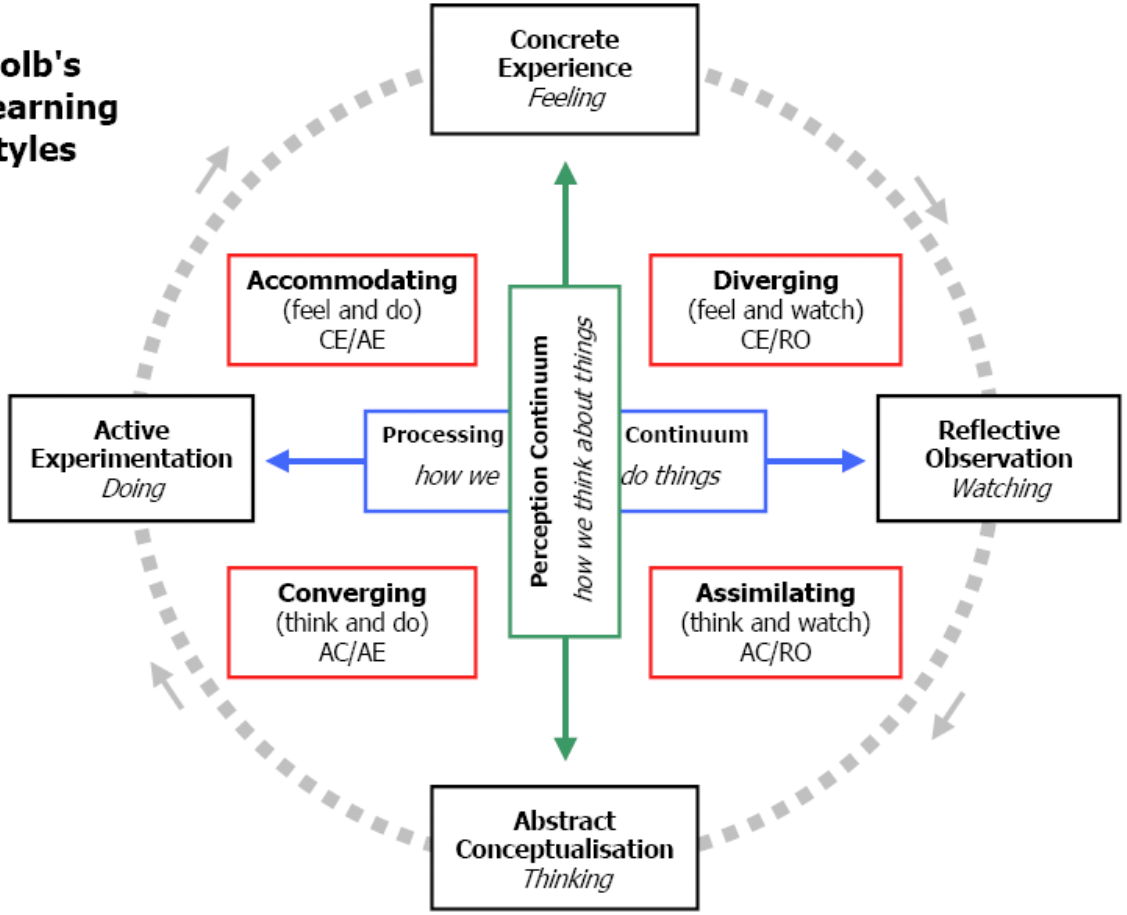


cphbusiness

# Heidi’s kode

cphbusiness

# Kolb's learning styles



```r
doScrape <- function(domain,mlim) {
  # dataframen til mine reviews
  reviews = data.frame()
  limit = doEstimate(domain)
  url <- paste0("https://dk.trustpilot.com/review/", domain)
```

```r
#Der laves et loop som går gennem alle siderne
for (i in (1:mlim)) {
```

```r

  #påbegynder scraping
  tmpurl=paste0(url,"?page=",i)
  #log_print(tmpurl)
  #
  #remDr$navigate(tmpurl)
  #tmpsource <- remDr$getPageSource()
  page <- read_html(tmpurl)
  Sys.sleep(3)

```

cphbusiness

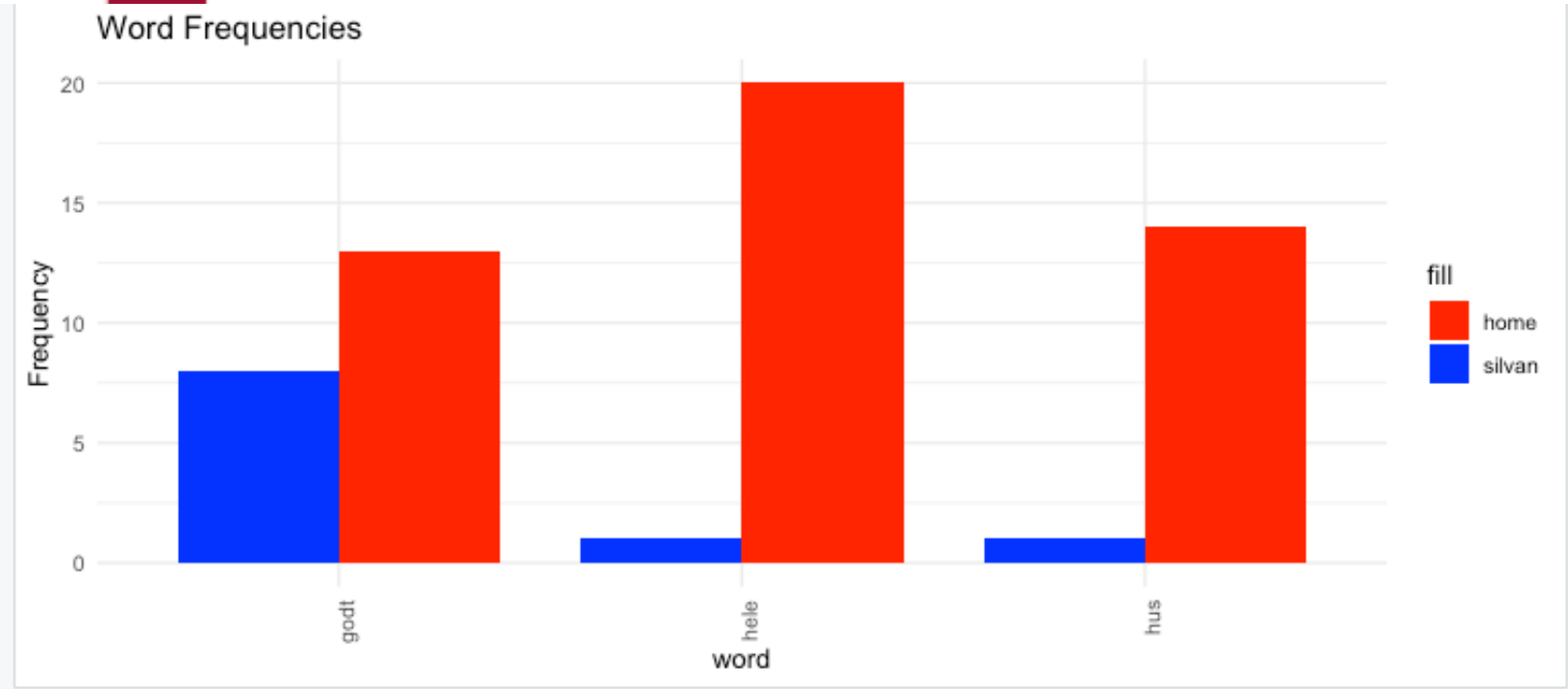
# Sentida

## Ingen labels

```r
library(Sentida)
library(dplyr)
library(ggplot2)


# indlæs Silvan - review

# indlæs Sentida pakken

# Find selv på en 1) positiv  2) negativ 3) neutral sætning og test scoren

# Undersøg hvordan sentida scorer på en dobbelt-negation
# "det er jo ikke sådan at vi ikke vil støtte Sudan, men ..."
# her er flere eksempler https://www.altinget.dk/artikel/pas-paa-dobbelthedernes-dumhed


# Beregn en sentida-score for alle reviews og find de mest positive.


#SPACYR
# I skal nu bruge spacyr til at finde alle navneord i et udvalgt review.

# INSTALLATION af spacy kan være besværlig.
# følg evt denne vejledning og spør!
# https://cran.r-project.org/web/packages/spacyr/readme/README.html
library(spacyr)
spacy_initialize(model = "da_core_news_sm")

# START test
txt="Otto bor i Lyngby med sin hund Vuf"
parsedtxt <- spacy_parse(txt, lemma = FALSE, entity = TRUE, nounphrase = TRUE)
edf = entity_extract(parsedtxt)

# Udfør entity_extract på alle reviews så der dannes en ny kolonne med en liste af locationer
```

cphbusiness

# Tidy og Home vs EDC



# Strukturer i tidy – ligner jeres almindelig opbygning

- Each variable is a column
- Each observation is a row
- Each type of observational unit is a table

# Tidy text med andre strukturer

- **String**: Text can, of course, be stored as strings, i.e., character vectors, within R, and often text data is first read into memory in this form.
- **Corpus**: These types of objects typically contain raw strings annotated with additional metadata and details.
- **Document-term matrix**: This is a sparse matrix describing a collection (i.e., a corpus) of documents with one row for each document and one column for each term. The value in the matrix is typically word count or tf-idf (see Chapter 3).

# Unnest token funktioner

# Token

```
text <- c("Because I could not stop for Death -",
          "He kindly stopped for me -",
          "The Carriage held but just Ourselves -",
          "and Immortality")

text
#> [1] "Because I could not stop for Death -"
#> [2] "He kindly stopped for me -"
#> [3] "The Carriage held but just Ourselves -"
#> [4] "and Immortality"
```

# Token

```
library(dplyr)
text_df <- tibble(line = 1:4, text = text)

text_df
#> # A tibble: 4 × 2
#>    line text
#>   <int> <chr>
#> 1     1 Because I could not stop for Death -
#> 2     2 He kindly stopped for me -
#> 3     3 The Carriage held but just Ourselves -
#> 4     4 and Immortality
```

# Unnest token

A token is a meaningful unit of text, most often a word, that we are interested in using for further analysis, and tokenization is the process of splitting text into tokens.

```
library(tidytext)

text_df %>%
  unnest_tokens(word, text)
#> # A tibble: 20 × 2
#>     line word
#>    <int> <chr>
#>  1     1 because
#>  2     1 i
#>  3     1 could
#>  4     1 not
#>  5     1 stop
#>  6     1 for
#>  7     1 death
#>  8     2 he
#>  9     2 kindly
#> 10     2 stopped
#> # … with 10 more rows
```

# Unnest token

## Usage

```
unnest_tokens_(tbl, output_col, input_col, token = "words", to_lower = TRUE, drop = TRUE, collapse = NULL, ...)

unnest_tokens(tbl, output, input, token = "words", to_lower = TRUE, drop = TRUE, collapse = NULL, ...)
```

## Arguments

| | |
|---|---|
| `tbl` | Data frame |
| `output_col` | Output column to be created |
| `input_col` | Input column that gets split |
| `token` | Unit for tokenizing, or a custom tokenizing function. Built-in options are "words" (default), "characters", "ngrams", "skip_ngrams", "sentences", "lines", "paragraphs", and "regex". If a function, should take a character vector and return a list of character vectors of the same length. |
| `to_lower` | Whether to turn column lowercase |
| `drop` | Whether original input column should get dropped. Ignored if the original input and new output column have the same name. |
| `collapse` | Whether to combine text with newlines first in case tokens (such as sentences or paragraphs) span multiple lines. If NULL, collapses when token method is "ngrams", "skip_ngrams", "sentences", "lines", "paragraphs", or "regex" |
| `...` | Extra arguments passed on to the tokenizer, such as `n` and `k` for "ngrams" and "skip_ngrams" or `pattern` for "regex" |
| `output` | Output column to be created as bare name |
| `input` | Input column that gets split as bare name |

# Fra ord til data til visualisering

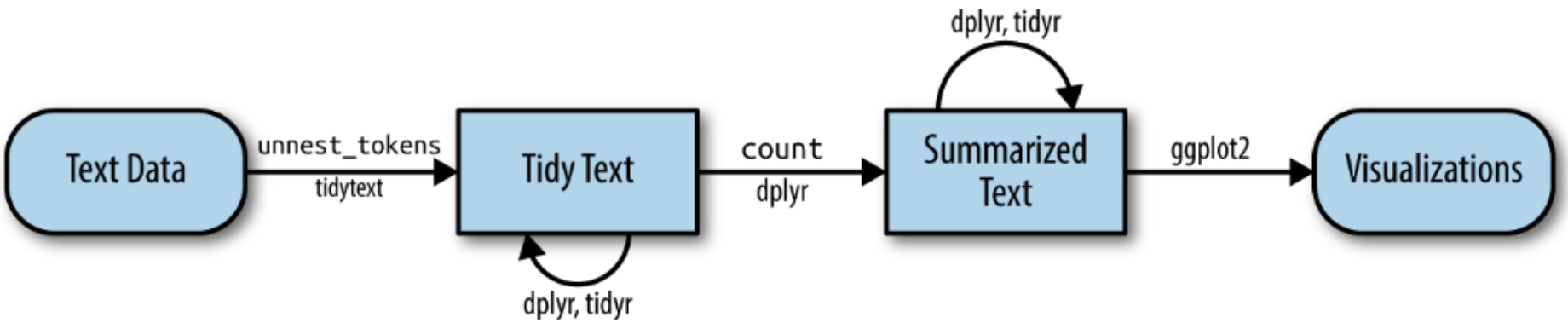



Figure 1.1: A flowchart of a typical text analysis using tidy data principles. This chapter shows how to summarize and visualize text using these tools.

cphbusiness

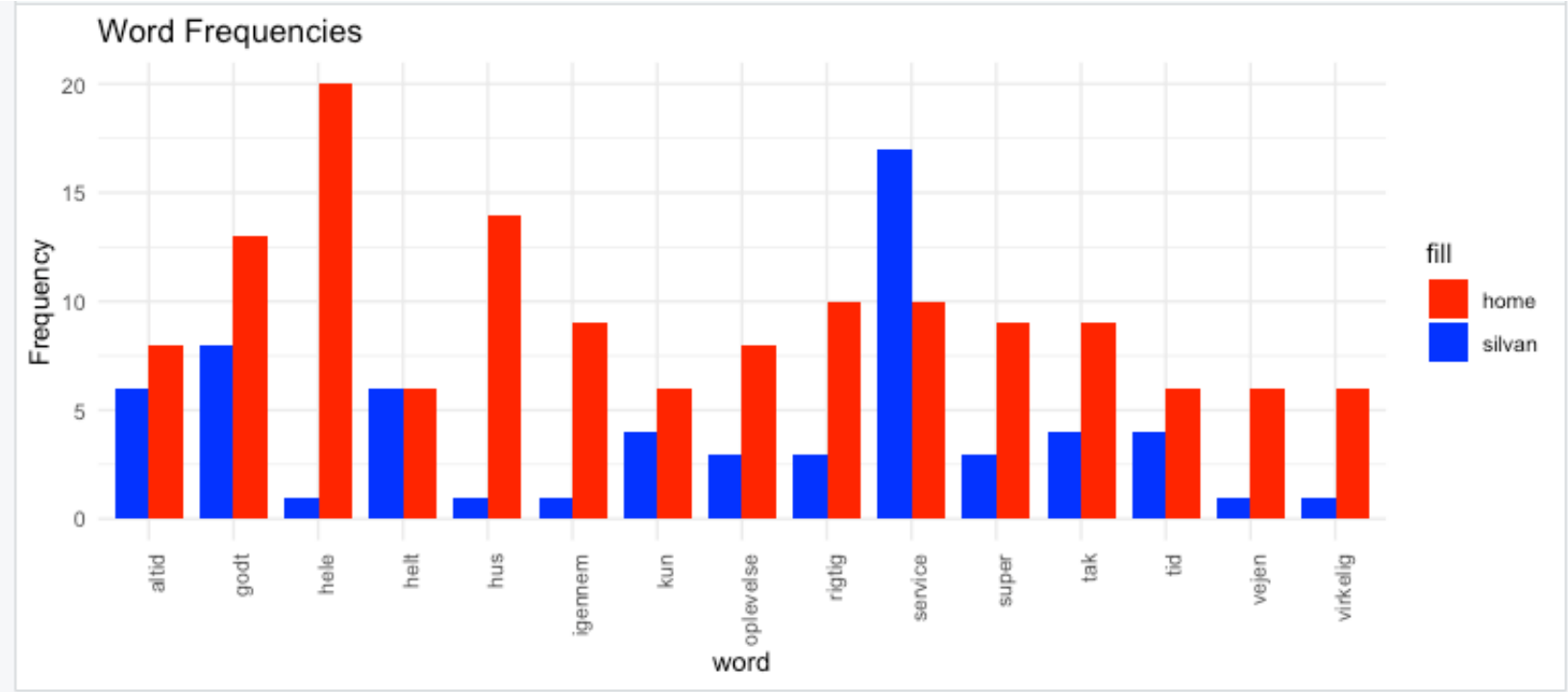
# Tidy og Silvaln

cphbusiness

## Opgaver

# Lav en sammenligning af highscores mellem home og edc

# Slut målet ..


Word Frequencies
Frequency
20
15
10
5
0
altid
godt
hele
helt
hus
igennem
kun
oplevelse
rigtig
service
super
tak
tid
vejen
virkelig
word
fill
home
silvan

# Vejen ..

Source on Save

```
4
5
6  #Byg en dataframe med review-indhold fra Silvan
7
8  #Byg en dataframe med review-indhold fra Home
9
10 #Find tokens for Silvan og Home
11
12 # bigrams For sjov :-)
13
14 #colnames(text)="content"
15
16
17 # Tæl ord for Home og Silvan
18
19
20 # Lav evt dine egne stopord
21
22
23 #Eller find dem på nettet
24
25
26 #Clean for stopord
27
28 #Gentag optælling
29
30 #Fjern ikke-ord
31
32
33 # Gør klar til join
34
35 # Join så vi kun har fælles ord
36
37 # kun ord af en vis frekvens vil vi plotte
38
39 #plot
40
41
```

cphbusiness

# Tidy og Jane Austin

Tekstanalyse

# En linje pr. række

```r
library(janeaustenr)
library(dplyr)
library(stringr)

original_books <- austen_books() %>%
  group_by(book) %>%
  mutate(linenumber = row_number(),
         chapter = cumsum(str_detect(text,
                                     regex("^chapter [\\divxlc]",
                                           ignore_case = TRUE)))) %>%
  ungroup()

original_books
#> # A tibble: 73,422 × 4
#>    text                    book                linenumber chapter
#>    <chr>                   <fct>                    <int>   <int>
#>  1 "SENSE AND SENSIBILITY" Sense & Sensibility          1       0
#>  2 ""                      Sense & Sensibility          2       0
#>  3 "by Jane Austen"        Sense & Sensibility          3       0
```

# Hvert ord for sig

```
library(tidytext)
tidy_books <- original_books %>%
  unnest_tokens(word, text)

tidy_books
#> # A tibble: 725,055 × 4
#>    book                linenumber chapter word
#>    <fct>                    <int>   <int> <chr>
#>  1 Sense & Sensibility          1       0 sense
#>  2 Sense & Sensibility          1       0 and
#>  3 Sense & Sensibility          1       0 sensibility
#>  4 Sense & Sensibility          3       0 by
#>  5 Sense & Sensibility          3       0 jane
#>  6 Sense & Sensibility          3       0 austen
```

# Fjerner stopord

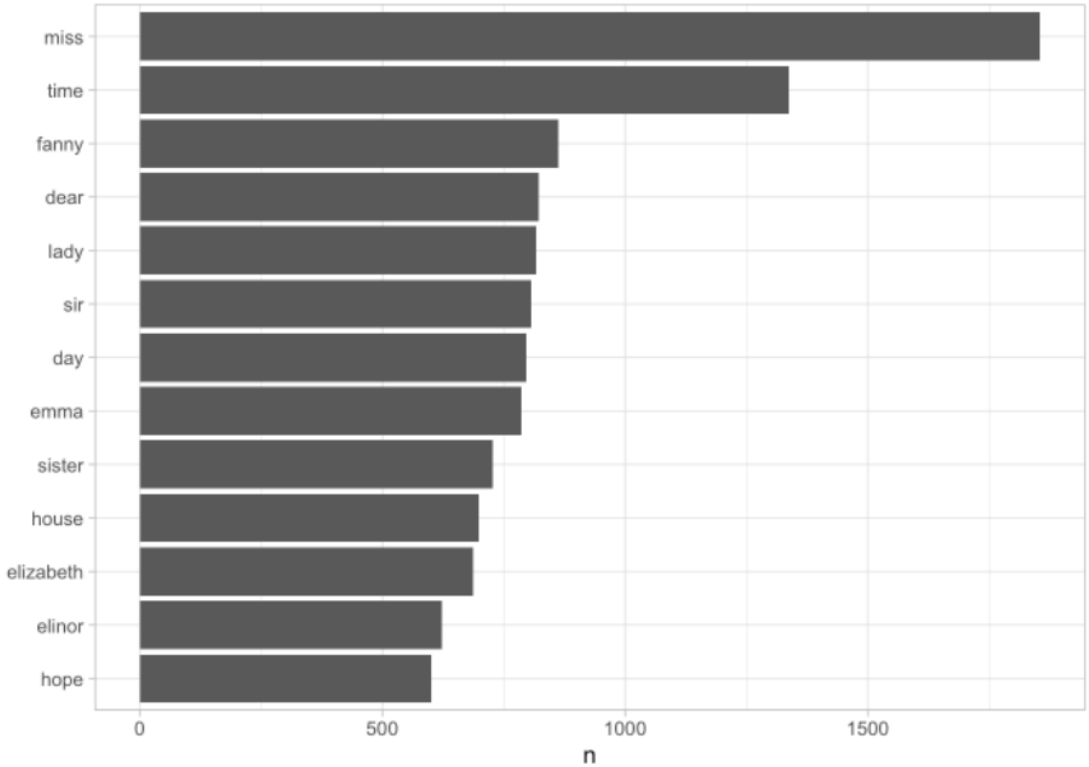
```
data(stop_words)


tidy_books <- tidy_books %>%
  anti_join(stop_words)
```

# Optælling af ord

```
tidy_books %>%
  count(word, sort = TRUE)
#> # A tibble: 13,914 × 2
#>    word       n
#>    <chr>  <int>
#>  1 miss    1855
#>  2 time    1337
#>  3 fanny    862
#>  4 dear     822
#>  5 lady     817
#>  6 sir      806
#>  7 day      797
#>  8 emma     787
#>  9 sister   727
#> 10 house    699
#> # … with 13,904 more rows
```

# Illustration

```
library(ggplot2)

tidy_books %>%
  count(word, sort = TRUE) %>%
  filter(n > 600) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  labs(y = NULL)
```



cphbusiness

# Opgaver

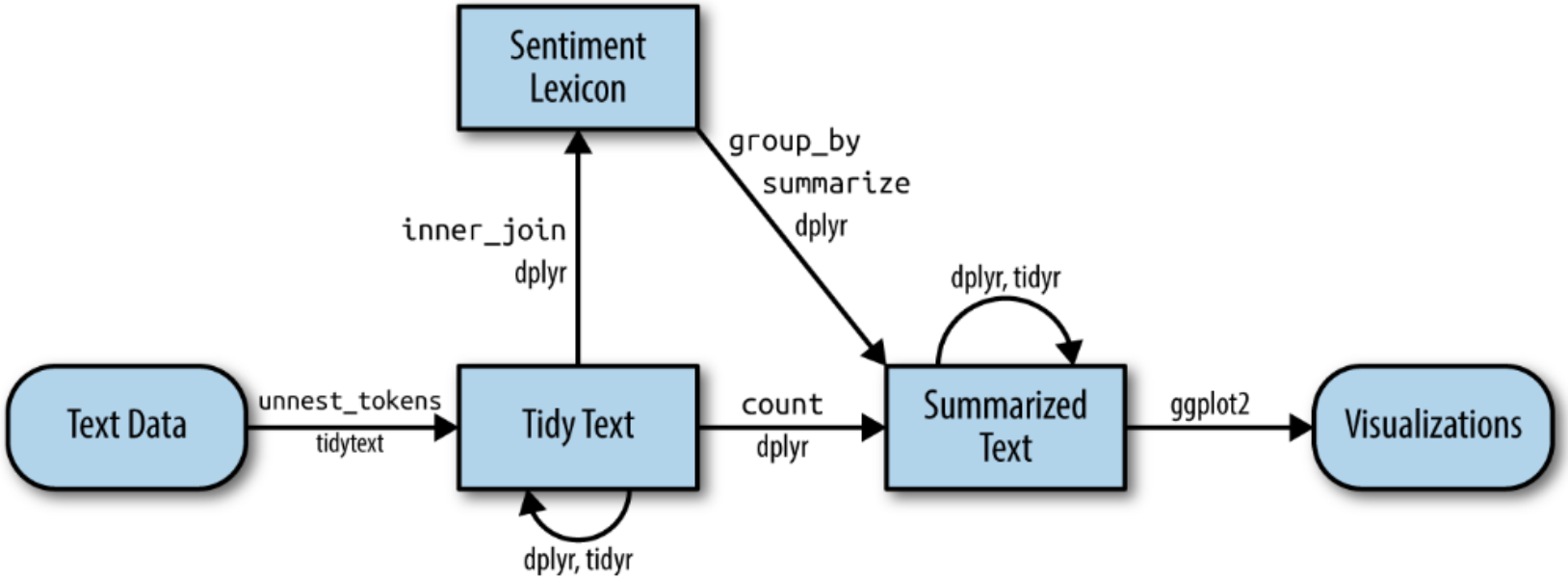
Lav en analyse af Tessas nytårstale

Lav en sammenligning af highscores mellem home og edc

# Tessas tale

- Struktur i dokument
- Datarens
- Data exploration
- Stop words
- Ordoptælling
- Analyse
- Konklusion (ud fra antagelse)

# Overblik


Sentiment Lexicon
group_by
summarize
dplyr
inner_join
dplyr
dplyr, tidyr
Text Data
unnest_tokens
tidytext
Tidy Text
count
dplyr
Summarized Text
ggplot2
Visualizations
dplyr, tidyr