

Opgaver lektion 3

Simon

19/11/2022

Inden opgaven læs følgende pakker!

```
library(tidyverse)
library(ggplot2)
```

Opgave 1:

Brug de følgende steps til at udregne BMI for Starwars karaktere. Baseret på deres hår farve:

Step 1

Vælg de 4 variable vi ønsker at bruge (Vi beholder “name”)

Fjern “NA” værdier med funktionen “drop_na()”

```
starwars %>%
  select(name, hair_color, mass, height)%>%
  drop_na()
```

```
## # A tibble: 54 x 4
##   name          hair_color    mass height
##   <chr>         <chr>      <dbl> <int>
## 1 Luke Skywalker blond          77    172
## 2 Darth Vader   none          136    202
## 3 Leia Organa   brown          49    150
## 4 Owen Lars     brown, grey   120    178
## 5 Beru Whitesun lars brown          75    165
## 6 Biggs Darklighter black          84    183
## 7 Obi-Wan Kenobi auburn, white  77    182
## 8 Anakin Skywalker blond          84    188
## 9 Chewbacca     brown        112    228
## 10 Han Solo      brown          80    180
## # ... with 44 more rows
```

Step 2

Brug mutate til at lave en ny kolonne der viser BMI:

Formlen for BMI er følgende: $BMI = \frac{mass(kg)}{height(M)^2}$

Da vi skal have height i meter skal vi også bruge mutate til at ændre denne kolonne:

Husk vi stadig skal bruge ovenstående steps

```
starwars %>%
  select(name, hair_color, mass, height)%>%
  drop_na() %>%
  mutate(height= height/100)
```

Tilføj nu selv en mutate funktion der laver BMI kolonnen, gem dette som et nyt datasæt kaldet “starwars_BMI”

Hvem har den højeste og mindste BMI? (Brug Arrange() funktionen)

Step 3

Vi kan nu udregne gennemsnit af BMI for hver hår farve:

Vi bruger en ny funktion kaldet “group_by()” Den opdeler kategoriske variable i hver deres kategori og fungerer derfor godt sammen med summarise funktionen:

```
starwars_BMI %>%
  group_by(hair_color) %>%
  summarise(mean(BMI))
```

```
## # A tibble: 9 x 2
##   hair_color      'mean(BMI)'
##   <chr>          <dbl>
## 1 auburn, white    23.2
## 2 black           22.8
## 3 blond           24.9
## 4 blonde          19.5
## 5 brown           24.5
## 6 brown, grey     37.9
## 7 grey            26.0
## 8 none            24.4
## 9 white           27.1
```

Så det ser ud til Starwars karaktere med brunt/gråt hår har den højeste BMI!

Step 5

Brug nu ggplot() funktionen til at lave et plot der viser “Mass” af x-aksen, og “height” af y-aksen. Brug punkter til at vise forholdet.

opgave 2 Når data er træls...

Jeg har hentet data på arbejdsløshedsraten direkte inde fra OECD. Her kræves der først en del arbejde med dataet før vi kan bruge det til noget...

```
library(readxl)
OECD_data <- read_excel("OECD data.xlsx")
glimpse(OECD_data)
```

```
## Rows: 48
## Columns: 8
## $ LOCATION      <chr> "DNK", "DNK", "DNK", "DNK", "DNK", "DNK", "DNK", "DNK", "~
## $ INDICATOR      <chr> "HUR", "HUR", "HUR", "HUR", "HUR", "HUR", "HUR", "HUR", "~
## $ SUBJECT        <chr> "TOT", "TOT", "TOT", "TOT", "TOT", "TOT", "TOT", "TOT", "~
## $ MEASURE        <chr> "PC_LF", "PC_LF", "PC_LF", "PC_LF", "PC_LF", "PC_LF", "PC_LF", "PC~
## $ FREQUENCY      <chr> "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A~
## $ TIME           <dbl> 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 201~
## $ Value          <dbl> 4.841667, 3.908333, 3.750000, 3.683333, 6.408333, 7.75000~
## $ 'Flag Codes'   <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
```

Alle navne er med stort, hvilket bliver lidt nederen i længden... Så jeg ændre til små bogstaver:

```
names(OECD_data) <- tolower(names(OECD_data))
```

```
OECD_data %>%
  count(location)
```

```
## # A tibble: 3 x 2
##   location     n
##   <chr>      <int>
## 1 DEU         16
## 2 DNK         16
## 3 USA         16
```

Step 1

Brug nu select funktionen til at vælge de tre kolonner: "location", "value" og "time".

```
OECD_data %>%
  select(location, value, time)
```

Vi vil nu gerne lave en kolonne med henholdsvis Danmarks, Tysklands, Italiens, USAs og OECDs arbejdsløshedsrate.

Dette kan gøres på flere måder:

1. Brug filter() funktionen til at vælge observationer kun for Danmark, gem dette i et dataset → Gør nu det samme for de andre lande → brug left_join() funktionen til at sammensætte de 5 individuelle dataset.

```
DK_dat=OECD_data %>%
  select(location, value, time) %>%
  filter(location == "DNK") %>%
  rename(dk_value = value)
```

```

DEU_dat=OECD_data %>%
  select(location, value, time) %>%
  filter(location == "DEU") %>%
  rename(deu_value = value)

USA_dat=OECD_data %>%
  select(location, value, time) %>%
  filter(location == "USA") %>%
  rename(usa_value = value)

option_1_data= DK_dat %>%
  left_join(DEU_dat, by = c("time")) %>%
  left_join(USA_dat, by = c("time")) %>%
  select(time,dk_value, deu_value, usa_value)

```

Denne metode tager en del tid og kode, nogle gange er det nødvendigt, men til lige præcis den her opgave kan vi bruge `pivot_wider()` funktionen som gør det meget nemmere:

2. Brug `pivot_wider()` funktionen

- Søg `pivot_wider()` funktionen op under “Help”
- brug “names_from” og “values_from” til at få dit nye dataset.
- Nedenfor kan i se hvordan det nye dataset skal se ud:

```
option_2_data
```

```

## # A tibble: 16 x 4
##   time   DNK   DEU   USA
##   <dbl> <dbl> <dbl> <dbl>
## 1  2005  4.84  11.3   5.07
## 2  2006  3.91  10.3   4.62
## 3  2007  3.75   8.54   4.62
## 4  2008  3.68   7.42   5.78
## 5  2009  6.41   7.22   9.27
## 6  2010  7.75   6.58   9.62
## 7  2011  7.76   5.52   8.95
## 8  2012  7.78   5.08   8.07
## 9  2013  7.38   4.95   7.38
## 10 2014  6.92   4.71   6.17
## 11 2015  6.27   4.37   5.29
## 12 2016  5.98   3.91   4.87
## 13 2017  5.82   3.57   4.35
## 14 2018  5.15   3.21   3.9
## 15 2019  5.03   2.98   3.67
## 16 2020  5.62   3.62   8.09

```

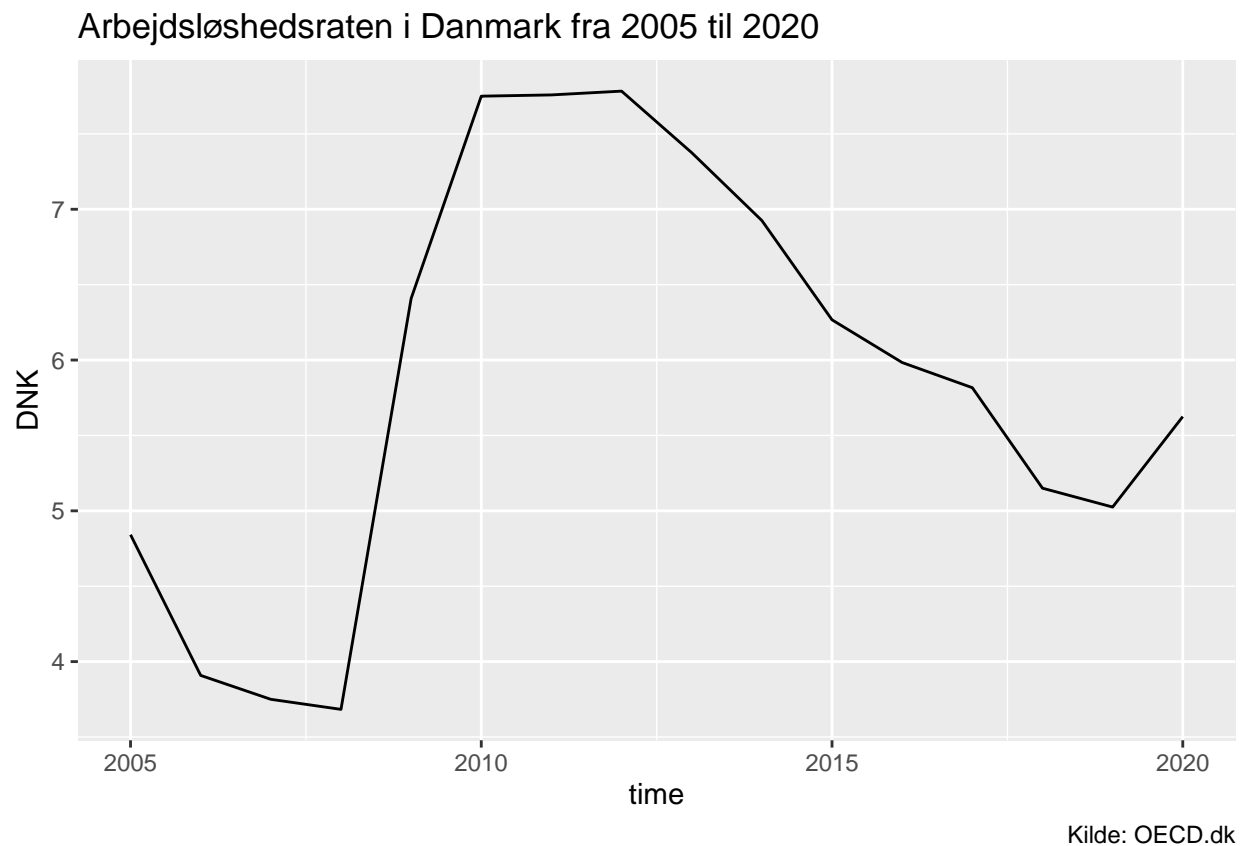
Step 3

Jeg kunne nu lave hver kolonne til en tidsserie og derved droppe time variablen som ses nedenfor:

```
data_ts= option_2_data %>%
  select(-time) %>%
  ts(start = c(2005), frequency = 1)
```

MEN så vil jeg ikke kunne bruge ggplot2, da dette ikke er lavet for tidsserier. Derfor beholder vi formatet “option_2_data” hvor jeg har en “time” variabel med. Vi kan nu lave forskellige plots:

```
ggplot(option_2_data, aes(x= time ,y = DNK)) + # angiver det datasæt, vi ønsker at plotte fra
# samt hvad der skal være på x og y akserne
geom_line() + # vi ønsker at lave et linjeplot
labs(title = "Arbejdsløshedsraten i Danmark fra 2005 til 2020",
caption = "Kilde: OECD.dk")
```



- Forsøg selv at lave flere plots ved hjælp af cheatsheet til ggplot2 som er citeret i slides.