Strava analysis

Over de data

Strava is een social platform waar je ritgegevens, hardloopgegevens etc in kan bijhouden. in deze opgave maken we gebruik van een beperkte dataset uit dit platform #Setup van document Inlezen libraries

```
library(tidyverse)
library(ggplot2)
library(dplyr)
library(lubridate)
library(scales)
```

Inlezen data - Pas de filenaam maar aan

```
act_data <- read_delim("dataset_mystravadata.csv",
    ";", escape_double = FALSE, trim_ws = TRUE)
```

Aanpak

Vraag: ik wil weten hoe mijn productiviteit (duur, kilometers) in de wintermaanden is geweest en of ik productiever ben geworden.

Stappenplan:

- Selectie van de benodigde kolommen: afstand, duur, activiteitstype, datum, fiets (gear)
- Bewerkingen uitvoeren
- Visualiseren

Selecteren

Ik heb niet alle kolommen nodig. via de documentatie van Strava kan je achterhalen welke velden je tot je beschikking hebt. In dit voorbeeld heb ik die select vast gemaakt met behulp van deze chunk, en t resultaat opgeslagen in een nieuwe csv.

```
winterdata <- act_data %>%
  select(distance,moving_time,start_date,gear_id,type)
```

Aanpassen/verrijken/opschonen - Transformeren

Jaar en maand extraheren

Uit de datum kunnen we het jaar en de maand halen mbv de `year` en de `month` functies. *head zorgt ervoor dat ik alleen de 'kop' van het resultaat zie*

```
head(year(winterdata$start_date))
```

```
## [1] 2021 2021 2021 2021 2021 2021
```

Toevoegen van nieuwe kolommen aan de dataset

Als je een kolom wil toevoegen of wijzigen, dan gebruik je vaak de `mutate` functie voor. in dit geval willen we een kolom 'Jaar' toevoegen, waarbij dus het jaar uit de start_date wordt gehaald. we wijzen het resultaat toe aan dezelfde dataset (overschrijven de dataset dus)

```
winterdata <- winterdata %>%
  mutate("Jaar" = year(start_date))
```

Datzelfde kan dus ook voor de maand

```
winterdata <- winterdata %>%
  mutate("Maand" = month(start_date))
```

Aanpassen distance

Als je goed kijkt, zie je dat de waardes in de distance kolom een beetje raar zijn. dat komt omdat er voorafgaand aan het inladen een komma verloren is gegaan. We moeten de waardes dus niet als meters interpreteren maar als decimeters! oftewel: delen door 10000 voor de waardes in km!

```
winterdata <- winterdata %>%
  mutate( distance  = (distance/10000))
```

Filteren

Wellicht zit er nog data in deze dataset die we niet nodig hebben. mbv een `filter` kunnen we die eruit halen. bijvoorbeeld: - Alleen data van type `Ride` of `VirtualRide` behouden mbv een `%in%` kun je op meerdere condities checken

```
winterdata <- winterdata %>%
  filter( type %in% c("Ride","VirtualRide"))
```

Analyseren

Door gegevens als distance of moving time op te tellen (count of sum) per groep (maand, jaar, gear_id) kunnen we een beeld vormen van de activiteiten - We combineren vaak de `group_by` en de `summarize`

```
 act_maand_jaar <- winterdata %>%
  group_by(Jaar, Maand) %>%
  summarize("afstand" = sum(distance))
```

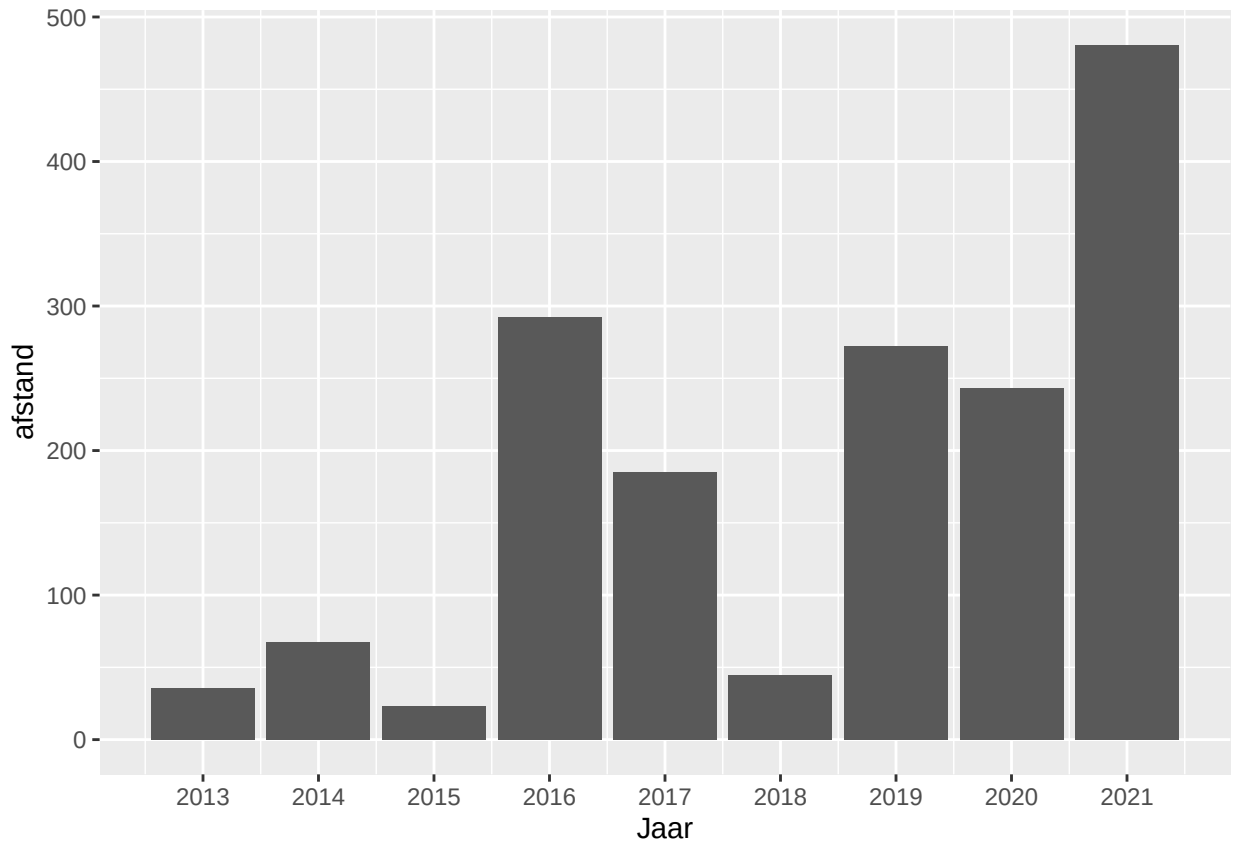
```
## `summarise()` has grouped output by 'Jaar'. You can override using the `.groups` argument.
```

```
act_maand_jaar
```

```
## # A tibble: 97 x 3
## # Groups:   Jaar [9]
##     Jaar Maand afstand
##    <dbl> <dbl>   <dbl>
##  1  2013     2    35.5
##  2  2013     4   342.
##  3  2013     5   128.
##  4  2013     6   216.
##  5  2013     7   535.
##  6  2013     8   493.
##  7  2013     9   331.
##  8  2013    10   144.
##  9  2013    11   204.
## 10  2013    12    46.2
## # ... with 87 more rows
```

Dit begint ergens op te lijken! laten we nu voor de maand februari eens kijken hoe die zich ontwikkeld heeft. Daarvoor gebruiken we een bar chart als visualisatie en een subset van deze data

```
ggplot(subset(act_maand_jaar,Maand == "2"), aes(x=Jaar, y=afstand))+
  geom_col()+
  scale_x_continuous(breaks=seq(2013, 2021, 1)) #fixed een raar probleem met de x as
```



Aanpassen & Nieuwe vragen Met deze voorbeelden moet je dmv aanpassen en kopieren van code chunks de volgende vragen kunnen gaan beantwoorden:

Vergelijk de afstand voor de wintermaanden december, januari en februari opgeteld per jaar

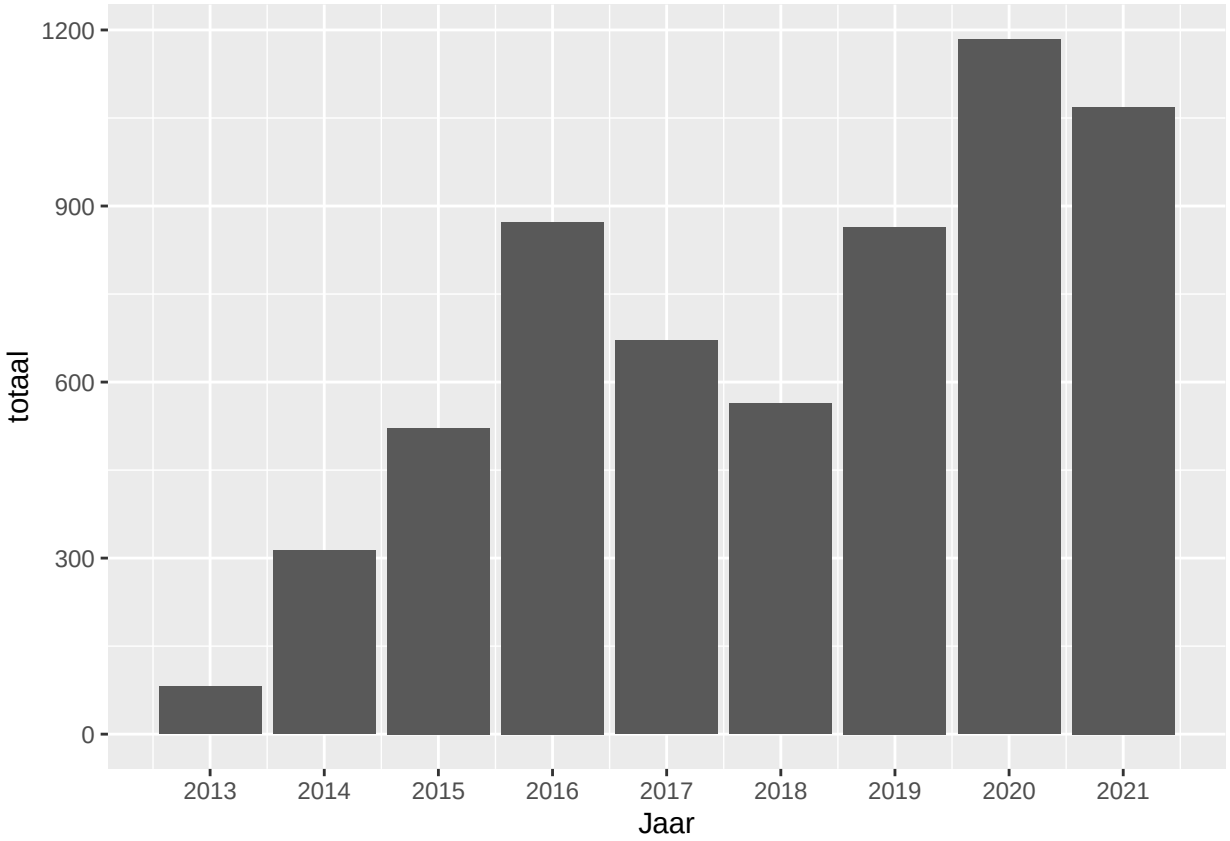
- Filter de dataset op de betreffende maanden

```
## # A tibble: 228 x 7
##    distance moving_time start_date          gear_id  type         Jaar Maand
##       <dbl>       <dbl> <dttm>              <chr>    <chr>       <dbl> <dbl>
##  1    53.1         6911 2021-02-28 12:13:45 b6734226 Ride         2021     2
##  2    54.2         7166 2021-02-26 08:18:43 b6734226 Ride         2021     2
##  3    41.2         5426 2021-02-24 14:20:02 b6734226 Ride         2021     2
##  4    43.4         5531 2021-02-23 13:37:52 b6734226 Ride         2021     2
##  5    28.5         6343 2021-02-21 08:14:13 b5957471 Ride         2021     2
##  6    53.2         9111 2021-02-20 12:06:03 b5957471 Ride         2021     2
##  7    28.9         5391 2021-02-19 12:14:34 b5957471 Ride         2021     2
##  8    21.3         3926 2021-02-18 15:41:02 b5957471 Ride         2021     2
##  9     2.46        3324 2021-02-15 14:18:00 b6734226 VirtualRide  2021     2
## 10    32.0         2721 2021-02-10 18:03:02 b6734226 VirtualRide  2021     2
## # ... with 218 more rows
```

- Groepeer en tel op per jaar (en dus niet meer per maand)

```
## # A tibble: 9 x 2
##    Jaar totaal
## * <dbl>  <dbl>
## 1  2013   81.8
## 2  2014  313.
## 3  2015  522.
## 4  2016  873.
## 5  2017  671.
## 6  2018  563.
## 7  2019  864.
## 8  2020 1184.
## 9  2021 1069.
```

- Visualiseer mbv een `ggplot('jouw_datasetnaam', aes(x=Jaar, y='jouw_som_per_jaar'))` en een `geom_col()`



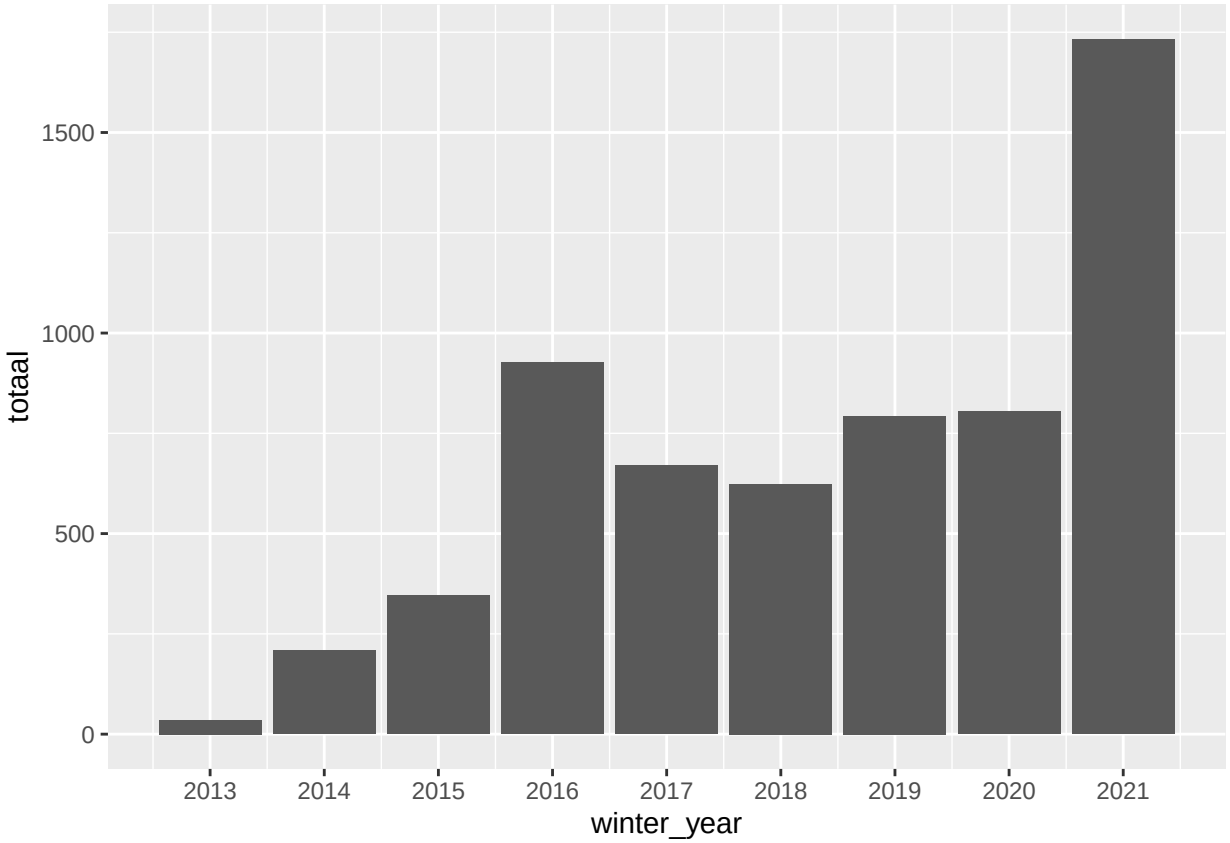
Let op: Eigenlijk zou je de winter van 2020/2021 moeten definieren als: december 2020 + januari 2021+ februari 2021, en niet december 2021 +jan 2021 +feb 2021. Met behulp van een nieuwe 'hulpkolom' kunnen we deze groepering aanpakken. Dat doen we als volgt: - We maken een nieuwe kolom "winter_year". - Als de waarde van Maand 12 is, dan wordt winter_year het jaar + 1 - Anders is winter_year gelijk aan het Jaar

```
winterdata_total <- winterdata_total %>%
  mutate("winter_year" =
           case_when(Maand == 12 ~ Jaar+1,
                     TRUE ~ Jaar))
```

En nu dus de groepering op jaar opnieuw:

```
## # A tibble: 9 x 2
##   winter_year totaal
## *       <dbl>  <dbl>
## 1        2013   35.5
## 2        2014  209.
## 3        2015  346.
## 4        2016  927.
## 5        2017  670.
## 6        2018  624.
## 7        2019  793.
## 8        2020  804.
## 9        2021 1732.
```

En de visualisatie opnieuw:



En voila, een heel ander resultaat!

Vergelijk de tijdsbesteding in de winter

- Je hebt nu moving_time nodig. deze is in seconden. wil je naar uren, deel dan door 3600
- kopieer en plak van eerdere statements en pas aan

Snelheid?

- Mbv distance en moving time kun je de snelheid bepalen.
- gear_id staat voor de 2 soorten fietsen: een (meerdere :)) racefietsen, een mtb, en crossers. Kun je achterhalen welke welke is?
- gebruik in een bestaande grafiek bv een `color = gear_id` in de het `aes` stuk
- Verschilt de snelheid op een binnen of buiten rit? (type = virtualride vs Ride) ...