

## Showcase af R og mine middelaldrende træningsdata :)

Jeg vil forsøge at vise mine R-færdigheder ved at analysere mine egne træningsdata. Jeg har brugt en række R-pakker for at rense, analysere og visualisere dataene. Jeg har lavet to versioner af den her portefølje - et med kode, og et uden :) Lad os dykke ned i det!

Først og fremmest importerer jeg data og gør dem klar til analyse. Det involverer at fjerne irrelevante kolonner og rette eventuelle fejl i dataene.

```
garmin_clean <- garmin_raw %>%
  janitor::clean_names() %>% # clean kolonne navne
  distinct() %>%

  # rename kolonner
  rename(
    aerobic_training_effect = aerobic_te,
    normalized_power = normalized_power_r_np_r,
    training_stress_score = training_stress_score_r,
    max_20min_power = max_avg_power_20_min,
    date_time = date) %>%

  # fjern række med filter()
  filter(
    row_number() != 375) %>% # fjern rækken.

  # væk med kolonner som ingen reel data har
  select(
    -favorite,
    -avg_vertical_ratio,
    -avg_vertical_oscillation,
    -avg_vertical_ratio,
    -avg_ground_contact_time,
    -avg_stride_length,
    -avg_stroke_rate,
    -avg_swolf,
    -flow,
    -total_reps,
    -decompression,
    -total_strokes,
    -grit,
    -best_lap_time,
    -number_of_laps) %>%

  # mutate kolonner
  mutate(
    # Datoer med lubridate
    date_time = ymd_hms(date_time), # convert to date-time
    date_onset = date(date_time), # extract date
    time_onset = as_hms(date_time), # extract time

    # distance
    distance = gsub("[^0-9.-]", "", distance), # remove non-numeric characters
    distance = as.numeric(distance), # convert to numeric
```

```

distance = distance / 100, # convert to kilometers
distance = ifelse(distance <= 1, NA, distance), # remove outlier

# Kalorier
calories = as.numeric(calories), # convert to numeric
calories = ifelse(calories > 20, calories / 100, calories), # convert to kilocalories
calories = ifelse(calories == 16.000, NA, calories), # bugged value

# Tid
time = as_hms(time), # convert to time

# Fart
avg_speed = gsub("[^0-9\\.]", "", avg_speed),
avg_speed = as.numeric(avg_speed), # convert to numeric
avg_speed = avg_speed / 10, # convert to km/h

# Hr
avg_hr = as.numeric(avg_hr))

```

Nu hvor dataene er rene, kan vi begynde at udforske dem. Der er helt sikkert flere variable som ville være praktiske at konvertere til andre data classes, men til denne forestilling kan det her gå :) PS: så er jeg gået lidt amok i farver til den her fremvisning. Jeg lover at holde det mere sobert fremadrettet :)

Table 1: Data summary

Name	garmin_clean
Number of rows	560
Number of columns	31
Column type frequency:	
character	23
Date	1
difftime	2
numeric	4
POSIXct	1
Group variables	None

### Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
activity_type	0	1	7	22	0	4	0
title	0	1	3	48	0	133	0
max_hr	0	1	2	3	0	65	0
aerobic_training_effect	0	1	2	3	0	21	0
max_speed	0	1	2	5	0	247	0
total_ascent	0	1	1	5	0	336	0
total_descent	0	1	1	5	0	338	0
avg_bike_cadence	0	1	2	2	0	30	0
max_bike_cadence	0	1	2	3	0	87	0
normalized_power	0	1	2	3	0	99	0
training_stress_score	0	1	3	5	0	249	0

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
max_20min_power	0	1	1	3	0	128	0
avg_power	0	1	2	3	0	103	0
max_power	0	1	2	5	0	250	0
min_temp	0	1	2	4	0	30	0
max_temp	0	1	3	4	0	32	0
avg_resp	0	1	2	2	0	12	0
min_resp	0	1	2	2	0	12	0
max_resp	0	1	2	2	0	19	0
moving_time	0	1	8	10	0	233	0
elapsed_time	0	1	8	10	0	305	0
min_elevation	0	1	1	4	0	167	0
max_elevation	0	1	1	3	0	170	0

### Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
date_onset	0	1	2013-06-09	2024-09-11	2017-03-29	531

### Variable type: difftime

skim_variable	n_missing	complete_rate	min	max	median	n_unique
time	0	1	607 secs	18618 secs	5850.5 secs	513
time_onset	0	1	20569 secs	77455 secs	46860.5 secs	553

### Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
distance	85	0.85	53.10	28.07	3.85	32.31	50.05	68.32	151.66
calories	9	0.98	3.79	2.65	0.38	1.62	2.70	5.76	9.97
avg_hr	143	0.74	143.03	12.30	47.00	137.00	144.00	151.00	166.00
avg_speed	78	0.86	26.62	4.32	0.00	25.63	27.40	28.58	42.10

### Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
date_time	0	1	2013-06-09 13:24:58	2024-09-11 10:14:26	2017-03-30 03:55:32	560

Først kan vi tage et kig på hvilke dage af ugen jeg oftest er ude at cykle.

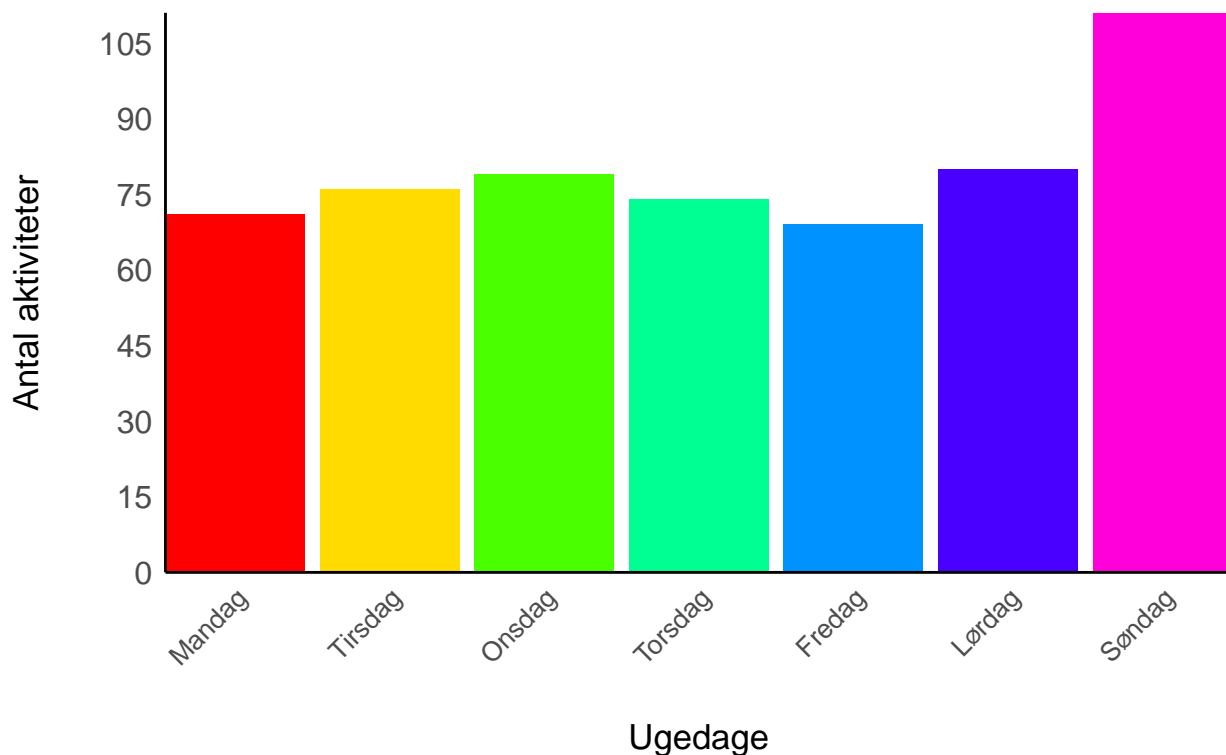
```
garmin_clean %>%
  mutate(
    weekday = wday(date_onset, label = TRUE, abbr = FALSE, week_start = 1),
    weekday = recode(weekday,
      "Monday" = "Mandag",
      "Tuesday" = "Tirsdag",
```

```

    "Wednesday" = "Onsdag",
    "Thursday" = "Torsdag",
    "Friday" = "Fredag",
    "Saturday" = "Lørdag",
    "Sunday" = "Søndag")) %>%
  count(weekday) %>%
  ggplot(mapping = aes(
    x = weekday,
    y = n,
    fill = weekday)) +
  geom_col(
    show.legend = FALSE) +
  scale_y_continuous(breaks = seq(0, 115, 15),
    expand = c(0, 0)) +
  scale_x_discrete(expand = c(0, 0)) +
  scale_fill_manual(values = rainbow(7)) +
  theme_modern() +
  labs(
    x = "Ugedage",
    y = "Antal aktiviteter",
    title = "Antal aktiviteter fordelt på ugedage") +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 10),
    plot.title = element_text(hjust = 0.5))

```

Antal aktiviteter fordelt på ugedage

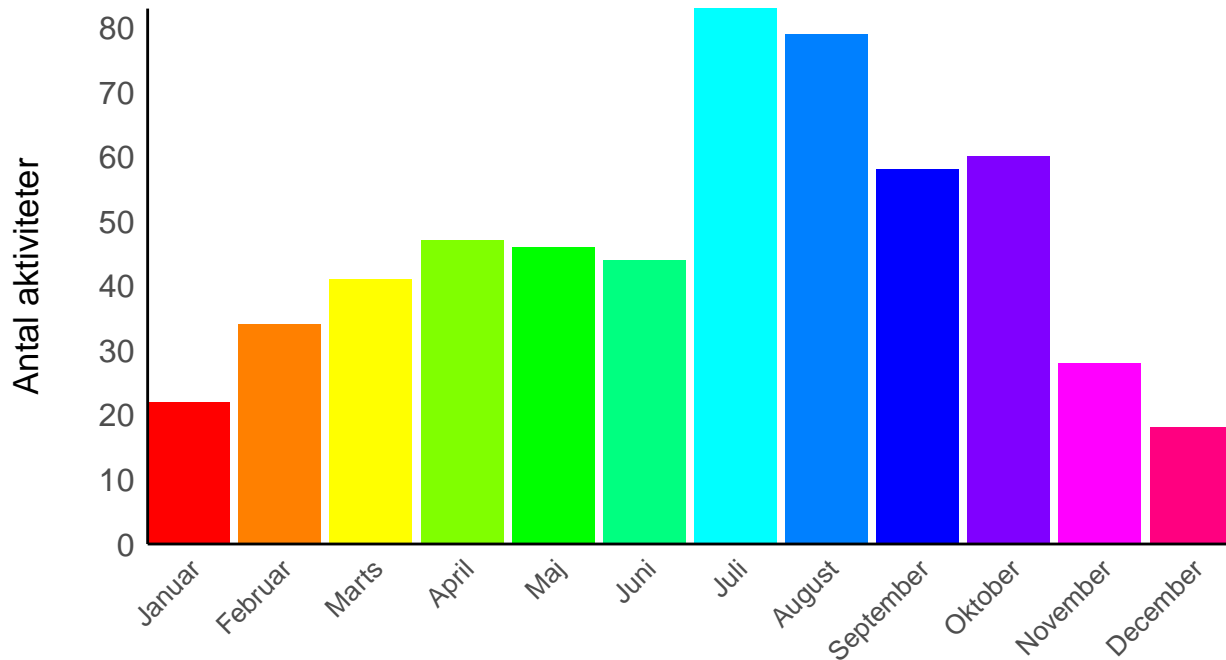


Det ville måske ikke være helt lyv, hvis man kaldte mig for en “Weekend warrior” :) Og som man kan se, så er søndage jo klart de bedste dage til en lang tur i sadlen.

Det ville være nærliggende at se på om jeg så også cykler hele året? Det kan vi gøre ved at gøre næsten det samme som med ugedagene ovenfor, men i stedet for uger, så opdele tiden i måneder.

```
garmin_clean %>%
  mutate(
    month = month(date_onset, label = TRUE, abbr = FALSE),
    month = recode(month,
      "January" = "Januar",
      "February" = "Februar",
      "March" = "Marts",
      "May" = "Maj",
      "June" = "Juni",
      "July" = "Juli",
      "October" = "Oktober")) %>%
  count(month) %>%
  ggplot(mapping = aes(
    x = month,
    y = n,
    fill = month)) +
  geom_col(
    show.legend = FALSE) +
  scale_y_continuous(breaks = seq(0, 90, 10),
    expand = c(0, 0)) +
  scale_x_discrete(expand = c(0, 0)) +
  scale_fill_manual(values = rainbow(12)) +
  theme_modern() +
  labs(
    x = "Måneder",
    y = "Antal aktiviteter",
    title = "Antal aktiviteter fordelt på måneder") +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 10),
    plot.title = element_text(hjust = 0.5))
```

## Antal aktiviteter fordelt på måneder



## Måneder

Der må man bare sige, jeg kunne nok godt stramme mig lidt an i vinterhalvåret. Ikke nok med at blive kaldt “Weekend warrior”, det ville være slemt nok, men hvis jeg blev kaldt “solskinsrytter” er det nok heller ikke helt forkert. Kulde er bare ikke mig!

Det helt store spørgsmål er vel om jeg så er blevet hurtigere eller langsommere på min cykel, og det har jeg undersøgt ved at opdele data i to grupper, før og efter min 30 års fødselsdag, og efterfølgende lavet en t-test.

```
garmin_clean %>%
  mutate(
    year_cat = case_when(
      year(date_onset) < "2018-01-20" ~ "Før 30 år",
      year(date_onset) >= "2018-01-20" ~ "Efter 30 år")) %>% # categorize years
  t.test(avg_speed ~ year_cat, data = .) %>% # students t-test
  print()
```

```
##
## Welch Two Sample t-test
##
## data: avg_speed by year_cat
## t = -3.7351, df = 453.16, p-value = 0.0002116
## alternative hypothesis: true difference in means between group Efter 30 år and group Før 30 år is not equal to 0
## 95 percent confidence interval:
## -1.9732740 -0.6126744
## sample estimates:
## mean in group Efter 30 år mean in group Før 30 år
## 25.72685 27.01982
```

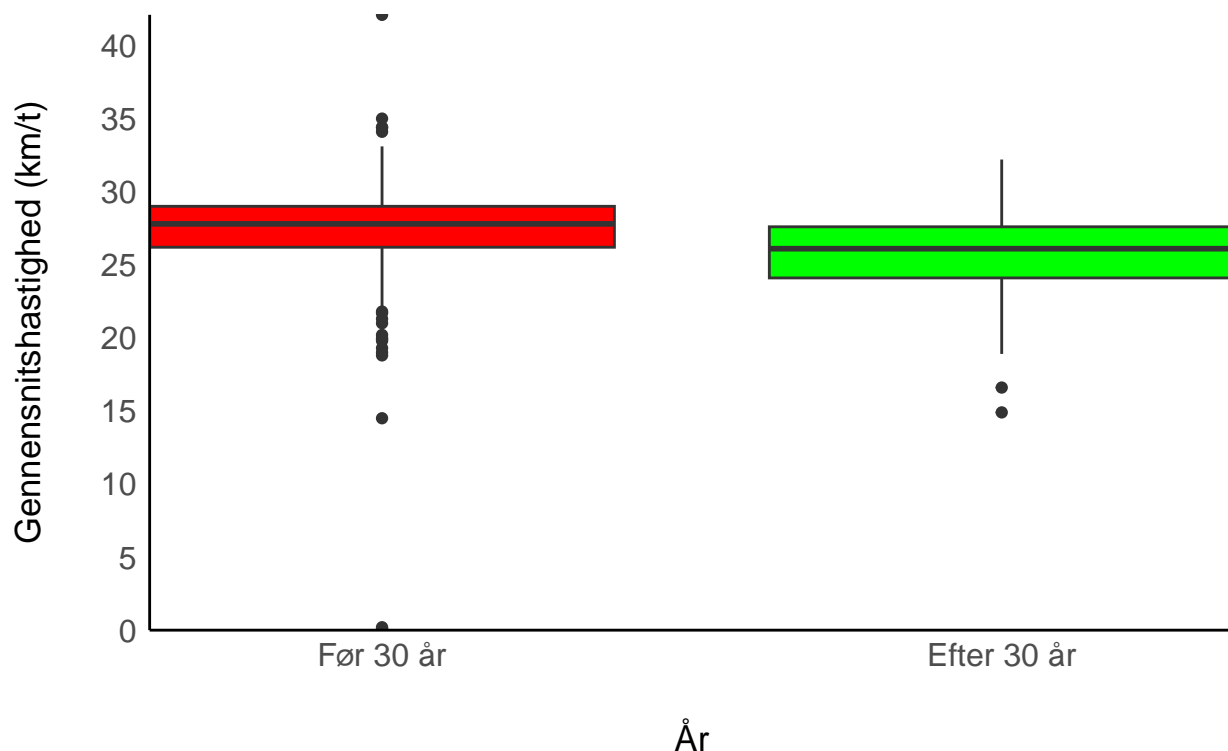
```
garmin_clean %>%
  mutate(
```

```

year_cat = case_when(
  year(date_onset) < "2018-01-20" ~ "Før 30 år",
  year(date_onset) >= "2018-01-20" ~ "Efter 30 år"), # categorize years
year_cat = fct_relevel(year_cat, "Før 30 år", "Efter 30 år")) %>%
ggplot(aes(
  x = year_cat,
  y = avg_speed,
  fill = year_cat)) +
geom_boxplot(show.legend = FALSE) + # add boxplot
scale_fill_manual(values = rainbow(3)) +
scale_y_continuous(breaks = seq(0, 45, 5),
  expand = c(0, 0)) +
scale_x_discrete(expand = c(0, 0)) +
theme_modern() +
labs (
  x = "År",
  y = "Gennemsnitshastighed (km/t)",
  title = "Gennemsnitshastigheden før/efter 30 års alderen",
  fill = "Years of cycling") +
theme(plot.title = element_text(hjust = 0.5))

```

## Gennemsnitshastigheden før/efter 30 års alderen



t-test viser at min gennemsnitshastighed før 30 år var 25.73 km/t og den efter 30 år var 27.01 km/t. forskellen i gennemsnitshastighed mellem aktiviteter før 30 år i forhold til aktiviteter efter 30 år var -1.29 km/t. 95% CI [-1.97, -0.61] og en  $p = < 0.001$  betyder at forskellen i gennemsnitshastighed er statistisk signifikant hvilket betyder at jeg desværre er blevet signifikant langsommere med alderen :(

Ok, så jeg bliver statistisk signifikant langsommere, men hvor meget langsommere bliver jeg så, for hvert år

jeg bliver ældre? Det har jeg undersøgt ved at lave en lineær regression mellem min gennemsnitsfart og tid, opdelt i år.

```
fart_model <- lm(avg_speed ~ year(date_onset), data = garmin_clean)
parameters(fart_model)
```

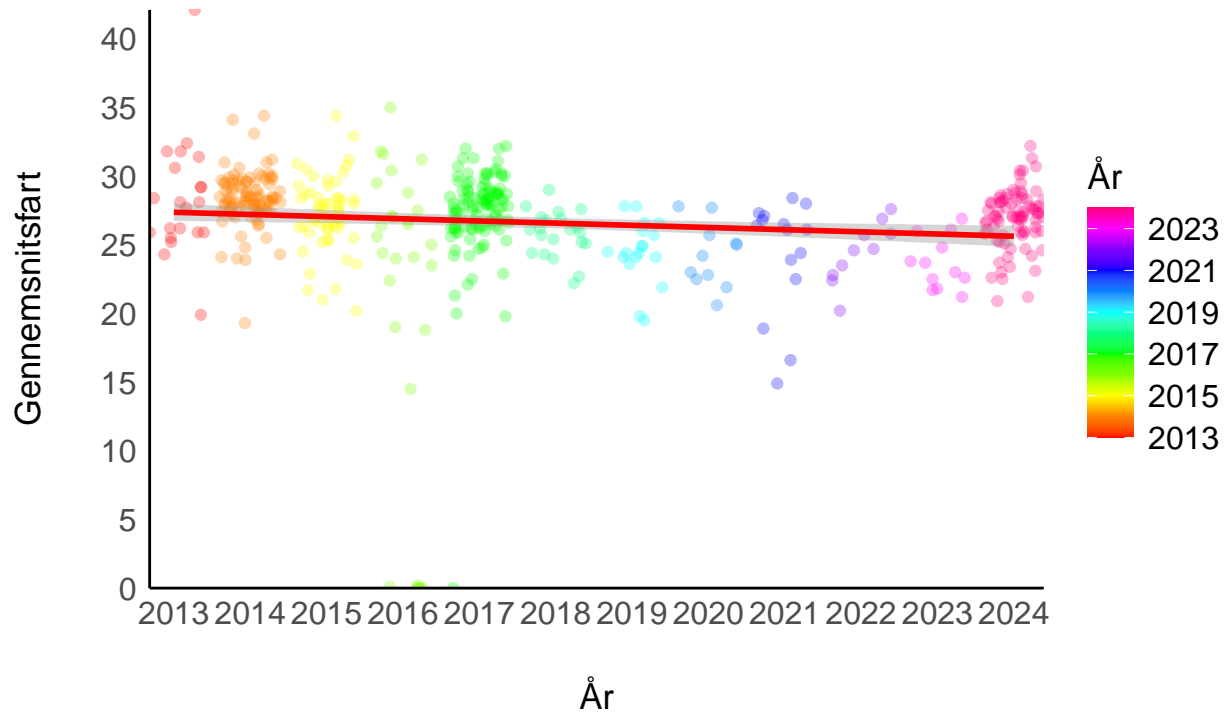
```
## Parameter          | Coefficient |      SE |          95% CI | t(480) |      p
## -----
## (Intercept)         |      346.23 | 108.09 | [133.85, 558.62] |   3.20 | 0.001
## year(date onset)    |      -0.16 |   0.05 | [ -0.26, -0.05] |  -2.96 | 0.003
##
## Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed
## using a Wald t-distribution approximation.
```

```
ggplot(data = garmin_clean,
       mapping = aes(
         x = year(date_onset),
         y = avg_speed,
         color = year(date_onset))) +
  geom_jitter(
    alpha = 0.3,
    width = 0.4,
    height = 0) +
  geom_smooth(method = lm, se = TRUE, color = "red") +
  scale_x_continuous(breaks = seq(2000, 2030, 1),
                    expand = c(0, 0)) +
  scale_y_continuous(breaks = seq(0, 40, 5),
                    expand = c(0, 0)) +
  scale_color_gradientn(colors = rainbow(12),
                      breaks = seq(2013, 2024, 2),
                      labels = seq(2013, 2024, 2)) +
  theme_modern() +
  labs(
    x = "År",
    y = "Gennemsnitsfart",
    title = "Lineær regression mellem gennemsnitsfart og tid",
    caption = "Garmin data mellem 2013-06-09 til 2024-09-11",
    color = "År") +
  theme(
    plot.title = element_text(hjust = 0.5),
    plot.caption = element_text(hjust = 0.5))
```

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



## Lineær regression mellem gennemsnitsfart og tid



Garmin data mellem 2013-06-09 til 2024-09-11

Som man kan se, så bliver jeg altså statistisk set 0.16 km/t langsommere for hvert år der går. Jeg forventer ikke at kune vende skuden, men jeg håber at kunne holde det nogenlunde stabilt :)

Denne analyse viser nogle af de mest interessante aspekter af min træningsdata og illustrerer mine R-færdigheder gennem dataimport, -rensning, og -visualisering. Alt sammen krydret med lidt humor omkring det faktum, at jeg måske ikke længere er helt så hurtig, som jeg engang var!