NYCflight13

## Data Transformation

### Flights that departed NYC in 2013 - DataFrame

library(tidyverse)  
library(nycflights13)

flights <- flights

Denne funksjonen viser oss hvor mange rekke (observasjoner) og hvor mange kolonner (variabler) vi har i datasettet:

dim(flights)

[1] 336776 19

Her kan vi spør om hva noe er. Her ser vi at det er en dataframe med utvidet funksjonalitet med en tibble:

# what type; data.frame, tibble?  
class(flights)

[1] "tbl\_df" "tbl" "data.frame"

### Bli kjent med datasettet - Head og Tail

Ofte det vi starter med er å se litt på starten (head) og slutten (tail) av filen for å få litt oversikt over hvordan ting ser ut:

* Her blir det skrevet ut de 10 første varablene

# vi legger alltid inn "paged.print: false" for å skrive det ut i "console"   
#| paged.print: false  
flights |>   
 head(n = 10)

# A tibble: 10 × 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
 1 2013 1 1 517 515 2 830 819  
 2 2013 1 1 533 529 4 850 830  
 3 2013 1 1 542 540 2 923 850  
 4 2013 1 1 544 545 -1 1004 1022  
 5 2013 1 1 554 600 -6 812 837  
 6 2013 1 1 554 558 -4 740 728  
 7 2013 1 1 555 600 -5 913 854  
 8 2013 1 1 557 600 -3 709 723  
 9 2013 1 1 557 600 -3 838 846  
10 2013 1 1 558 600 -2 753 745  
# ℹ 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

* Jeg vil se de 15 siste variablene:
* Her klartr vi ikke å bruke “tail” funksjonen, derfor måtte vi bruke “arrange”. Det kan se ut som at datoene ikke er sortert etter mnd og dag.

flights |>   
 arrange(desc(month), (desc(day)))

# A tibble: 336,776 × 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
 1 2013 12 31 13 2359 14 439 437  
 2 2013 12 31 18 2359 19 449 444  
 3 2013 12 31 26 2245 101 129 2353  
 4 2013 12 31 459 500 -1 655 651  
 5 2013 12 31 514 515 -1 814 812  
 6 2013 12 31 549 551 -2 925 900  
 7 2013 12 31 550 600 -10 725 745  
 8 2013 12 31 552 600 -8 811 826  
 9 2013 12 31 553 600 -7 741 754  
10 2013 12 31 554 550 4 1024 1027  
# ℹ 336,766 more rows  
# ℹ 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

### **Se på en delmengde - Filter**

for eksempel se på kun en dag. I utgangspunktet starter vi med å se nærmere på 1.januar. Bruker “print” komondo sånn det blir finere kode i chunken:

* filtrer ut -> jeg vil kun ha med videre “1.januar.2013”
* print ut de dataene og ta med kun de første 5 rekkene.
* Legge til vidde for å få plass til alt.

flights |>   
 filter(month == 1, day == 1) |>   
print(  
 n = 5,  
 width = 70  
 )

# A tibble: 842 × 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int>  
1 2013 1 1 517 515 2 830  
2 2013 1 1 533 529 4 850  
3 2013 1 1 542 540 2 923  
4 2013 1 1 544 545 -1 1004  
5 2013 1 1 554 600 -6 812  
# ℹ 837 more rows  
# ℹ 12 more variables: sched\_arr\_time <int>, arr\_delay <dbl>,  
# carrier <chr>, flight <int>, tailnum <chr>, origin <chr>,  
# dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
# minute <dbl>, time\_hour <dttm>

Om jeg ønsker å se på flere dager i januar, for eksempel 1. , 15. og 30. :

* %in% er en nyttig funksjon

print(  
 filter(flights, month == 1, day %in% c(1, 15, 30)),  
 n = 5,  
 width = 70  
 )

# A tibble: 2,636 × 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int>  
1 2013 1 1 517 515 2 830  
2 2013 1 1 533 529 4 850  
3 2013 1 1 542 540 2 923  
4 2013 1 1 544 545 -1 1004  
5 2013 1 1 554 600 -6 812  
# ℹ 2,631 more rows  
# ℹ 12 more variables: sched\_arr\_time <int>, arr\_delay <dbl>,  
# carrier <chr>, flight <int>, tailnum <chr>, origin <chr>,  
# dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
# minute <dbl>, time\_hour <dttm>

Om jeg vil se på alle dagene **utenom** 1. , 15. og 30. januar :

* så ved å bruke ! betyr det altså “ikke ta med”. Her betyr det da “ikke ta med dag 1, 15 og 30 i januar”

flights %>%   
 filter(month == 1, !(day %in% c(1, 15, 30))) %>%   
 print(width = 70)

# A tibble: 24,368 × 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int>  
 1 2013 1 2 42 2359 43 518  
 2 2013 1 2 126 2250 156 233  
 3 2013 1 2 458 500 -2 703  
 4 2013 1 2 512 515 -3 809  
 5 2013 1 2 535 540 -5 831  
 6 2013 1 2 536 529 7 840  
 7 2013 1 2 539 545 -6 959  
 8 2013 1 2 554 600 -6 845  
 9 2013 1 2 554 600 -6 841  
10 2013 1 2 554 600 -6 909  
# ℹ 24,358 more rows  
# ℹ 12 more variables: sched\_arr\_time <int>, arr\_delay <dbl>,  
# carrier <chr>, flight <int>, tailnum <chr>, origin <chr>,  
# dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
# minute <dbl>, time\_hour <dttm>

En annen klassisk komando er **“summary”** . Den tar for seg hver av de 19 variablene og gir oss et type sammendrag av hver variabel:

summary(flights)

year month day dep\_time sched\_dep\_time  
 Min. :2013 Min. : 1.000 Min. : 1.00 Min. : 1 Min. : 106   
 1st Qu.:2013 1st Qu.: 4.000 1st Qu.: 8.00 1st Qu.: 907 1st Qu.: 906   
 Median :2013 Median : 7.000 Median :16.00 Median :1401 Median :1359   
 Mean :2013 Mean : 6.549 Mean :15.71 Mean :1349 Mean :1344   
 3rd Qu.:2013 3rd Qu.:10.000 3rd Qu.:23.00 3rd Qu.:1744 3rd Qu.:1729   
 Max. :2013 Max. :12.000 Max. :31.00 Max. :2400 Max. :2359   
 NA's :8255   
 dep\_delay arr\_time sched\_arr\_time arr\_delay   
 Min. : -43.00 Min. : 1 Min. : 1 Min. : -86.000   
 1st Qu.: -5.00 1st Qu.:1104 1st Qu.:1124 1st Qu.: -17.000   
 Median : -2.00 Median :1535 Median :1556 Median : -5.000   
 Mean : 12.64 Mean :1502 Mean :1536 Mean : 6.895   
 3rd Qu.: 11.00 3rd Qu.:1940 3rd Qu.:1945 3rd Qu.: 14.000   
 Max. :1301.00 Max. :2400 Max. :2359 Max. :1272.000   
 NA's :8255 NA's :8713 NA's :9430   
 carrier flight tailnum origin   
 Length:336776 Min. : 1 Length:336776 Length:336776   
 Class :character 1st Qu.: 553 Class :character Class :character   
 Mode :character Median :1496 Mode :character Mode :character   
 Mean :1972   
 3rd Qu.:3465   
 Max. :8500   
   
 dest air\_time distance hour   
 Length:336776 Min. : 20.0 Min. : 17 Min. : 1.00   
 Class :character 1st Qu.: 82.0 1st Qu.: 502 1st Qu.: 9.00   
 Mode :character Median :129.0 Median : 872 Median :13.00   
 Mean :150.7 Mean :1040 Mean :13.18   
 3rd Qu.:192.0 3rd Qu.:1389 3rd Qu.:17.00   
 Max. :695.0 Max. :4983 Max. :23.00   
 NA's :9430   
 minute time\_hour   
 Min. : 0.00 Min. :2013-01-01 05:00:00.00   
 1st Qu.: 8.00 1st Qu.:2013-04-04 13:00:00.00   
 Median :29.00 Median :2013-07-03 10:00:00.00   
 Mean :26.23 Mean :2013-07-03 05:22:54.64   
 3rd Qu.:44.00 3rd Qu.:2013-10-01 07:00:00.00   
 Max. :59.00 Max. :2013-12-31 23:00:00.00

## Oppgaver - Filter

1. **How many flights were there in April and September (total)?**

* flights |>   
   filter(month %in% c(4, 9)) |>   
  # kan også legge inn "nrow" for å få bare svaret  
  nrow()
* [1] 55904
* kan også bruke denne metoden:
  + Her får vi ut alle antall flyvninger per måned.
* flights |>   
   group\_by(month) |>   
   summarise(antall\_flyvninger = n())
* # A tibble: 12 × 2  
   month antall\_flyvninger  
   <int> <int>  
   1 1 27004  
   2 2 24951  
   3 3 28834  
   4 4 28330  
   5 5 28796  
   6 6 28243  
   7 7 29425  
   8 8 29327  
   9 9 27574  
  10 10 28889  
  11 11 27268  
  12 12 28135

1. **How many flights were there in total the first two days of a month?**

* flights |>   
   filter(month == 1) |>   
   group\_by(day) |>   
   count() |>   
   #count teller opp antallet i en gruppe  
   print(n = 2)
* # A tibble: 31 × 2  
  # Groups: day [31]  
   day n  
   <int> <int>  
  1 1 842  
  2 2 943  
  # ℹ 29 more rows
* Denne viser hvor mange flyvninger som skjer de to første dagene hver måned i hele 2013??
* flights |>   
   filter(day %in% c(1, 2)) |>   
   nrow()
* [1] 21844
* Her ser vi hvilke dager som har mest flyvninger. Her er svaret 2.januar som hadde 943 flyvninger.
* flights |>   
   filter(month == 1) |>   
   group\_by(day) |>   
   count() |>  
   arrange(desc(n)) |>   
   print(n = 1)
* # A tibble: 31 × 2  
  # Groups: day [31]  
   day n  
   <int> <int>  
  1 2 943  
  # ℹ 30 more rows

1. **How may flights were there during 2013 that had an arrival delay greater than or equal to 15 min., but less than 30 min.?**
   * hvor mange flyvninger det var i 2013 som hadde en ankomstforsinkelse på **15 minutter eller mer**
   * **( >=15)**, men **mindre enn 30 minutter (<30):**

* flights |>   
   filter(arr\_delay >= 15, arr\_delay < 30) |>   
   nrow()
* [1] 27298

## NA verdi

**Dette handler om at det mangler en observersjon i vektoren. Kan bety “not available”**. Regel med NA er at hvis man har 1stk NA blir svaret NA.

Plukke ut “arr\_delay” -> da bruker vi “$” for å plukke ut variabelen

* Jeg ønsker å finne ut hvor mange **NA verdier** det er.
* Husk: True = 1 og False = 0
* = Det er totalt 9430 NA verdier i variabelen *som da er true, fordi false er null og vil derfor ikke vises.*

sum(is.na(flights$arr\_delay))

[1] 9430

## Select komandoen

Ta vekk “year” variablene for den er unødvendig å ha med da alt skjer i 2013. Overskriv “flights” med den nye versjonen som ikke har med “year”

# Dropper variabelen "year"   
flights <- flights |>   
 select(-year)

Nå vil jeg endre rekkefølge på variablene slik jeg kan sette øverst det jeg syns er mest interessant og så spessifisere jeg resten av rekken og til slutt hiver jeg på en “everything” for å liste opp resten jeg ikke nevnte i parantesen.

flights <- flights |>   
 select(month:day, carrier, tailnum, everything())

Endre navn på en variabel inne i select komandoen:

# Henter inn den gamle flights datasette som det ikke er blitt gjort noen endringer på:   
flights <- nycflights13::flights

Her har jeg **først droppet variablene** year da den er unødvendig å ha med da alt skjer i 2013. Videre har jeg **endret rekkefølgen på variablene**, samtidig som jeg har **endret to stk variabelnavn til norsk.**

flights <- flights |>   
 select(-year) |>   
 select(  
 month:day,   
 flyselskap = carrier,   
 halenummer = tailnum,  
 everything()  
 )  
# Kan også bruke "rename" funksjonen for å endre navn på variablene

Lage et nytt lite datasett med kunn dag, måned (mai) og arr\_delay:

my\_vars <- flights |>   
 select(day, month, arr\_delay) |>   
 filter(month == 5)   
 # Her er det jo egentlig unødvendig å ha med måned 5 (mai) da denne variablene vil stå likt hele veien, så kan derfor legge inn "select(-month) for å fjerne denne.

**Oppgave:** Gjennomsnittlig ankomstforsinkelse de ulike dagene i Mai?

#Gjennomsnittlig ankomstforsinkelse i desimalmunitt for dagene i Mai 2013  
my\_vars |>   
 group\_by(day) |>   
 summarise(gjennomsnittlig\_forsinkelse = mean(  
 arr\_delay,  
 na.rm = TRUE)  
 ) |>   
 print(n = 31)

# A tibble: 31 × 2  
 day gjennomsnittlig\_forsinkelse  
 <int> <dbl>  
 1 1 -8.38   
 2 2 -7.25   
 3 3 4.23   
 4 4 -8.54   
 5 5 -9.77   
 6 6 -7.58   
 7 7 -14.4   
 8 8 39.6   
 9 9 11.8   
10 10 9.02   
11 11 22.8   
12 12 -6.23   
13 13 -6.36   
14 14 -11.1   
15 15 -2.03   
16 16 4.20   
17 17 1.09   
18 18 -10.7   
19 19 19.5   
20 20 5.85   
21 21 4.08   
22 22 27.6   
23 23 62.0   
24 24 24.3   
25 25 0.0539  
26 26 -11.0   
27 27 -9.67   
28 28 1.10   
29 29 0.268   
30 30 -0.161   
31 31 -4.63

## Lage en ny vaiabel - Mutate

**“ends\_with” ->** Bruker den til å velge ut/plukke ut variabler som slutter med (…). Her plukker den ut 2 variabler som slutter på “delay” som i dette tilfellet er “*dep\_delay”* og *“arr\_delay”*

Her får vi ut en ny versjon av datasette. Det har kommet en ny variabel “**gain**” og **“speed”**.

* **Gain** - Hvor mye tid man har spart inne på flyplassen. Differeansen mellom avgansforsinkelse minus hvor mye det var forsinket når det kom inn. Dette er en ny variabel man har regnet ut ved hjelp av to eksisterende varaibler
* **Speed** - distansen delt på airtime. ganger med 60 for å få det omgjort til per. time. Altså hastighet pr time.

Her bruker vi %>% som pipe.

arrange(gain) -> for å få dataene økende

flights <- flights %>%  
 select(month:day, ends\_with("delay"), distance, air\_time) %>%  
 mutate(  
 gain = dep\_delay - arr\_delay,  
 speed = (distance/air\_time) \* 60  
 ) %>%  
 select(month:day, gain, everything()) %>%  
 arrange(gain)   
head(flights, n = 5)

# A tibble: 5 × 8  
 month day gain dep\_delay arr\_delay distance air\_time speed  
 <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 11 1 -196 -2 194 2475 336 442.  
2 4 18 -181 -2 179 1389 234 356.  
3 8 8 -165 180 345 2565 420 366.  
4 7 10 -161 16 177 2586 387 401.  
5 6 27 -157 43 200 544 120 272

## Transmute

Tar kun vare på den nye variabelen man lager, og dropper alle andre variabler.

Her vil kun “gain” og “speed” være det som kommer ut her videre i pipen. Alle andre variabler fra datasette er kastet vekk.

my\_data <- flights %>%  
 transmute(  
 gain = dep\_delay - arr\_delay,  
 speed = (distance/air\_time) \* 60  
 ) %>%  
 arrange(gain)  
head(my\_data, n = 3)

# A tibble: 3 × 2  
 gain speed  
 <dbl> <dbl>  
1 -196 442.  
2 -181 356.  
3 -165 366.