Data Verwerken
met dplyr en tidyr
Spiekbrief

RStudio

Data Opschonen - De basis voor verwerking in R

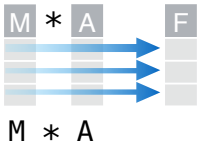
In een schone data set:

Elke **variabele** wordt in een **kolom** bewaard

&

Elke **observatie** wordt in een **rij** bewaard

Opgeschoonde data complementeren R's **gevectoriseerde operaties**. R verwerkt de waarnemingen automatisch terwijl je met de variabelen werkt. Geen ander format werkt zo intuïtief met R.



Syntax - Nuttige conventies over dataverwerking

dplyr::**tbl_df(iris)**

Converteert data naar de tbl klasse. tbl's zijn makkelijker te bekijken dan data frames. R laat alleen de data zien die op het scherm past:

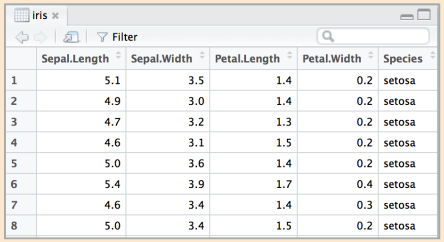
```
Source: local data frame [150 x 5]

   Sepal.Length Sepal.Width Petal.Length
1           5.1         3.5          1.4
2           4.9         3.0          1.4
3           4.7         3.2          1.3
4           4.6         3.1          1.5
5           5.0         3.6          1.4
..          ...         ...          ...
Variables not shown: Petal.Width (dbl),
  Species (fctr)
```

dplyr::**glimpse(iris)**

Gecomprimeerde samenvatting van tbl data.

utils::**View(iris)**

Bekijk de dataset in een spreadsheet vorm (Let op de hoofdletter V !).



	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

dplyr::**%>%**

Stuurt het object aan de linkerkant als eerste argument (of . argument) naar de functie aan de rechterkant.

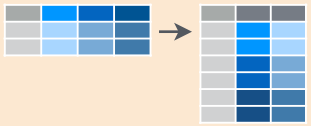
`x %>% f(y)` *is gelijk aan* `f(x, y)`

`y %>% f(x, ., z)` *is gelijk aan* `f(x, y, z )`

"Piping" met %>% maakt de code beter leesbaar, bijv.

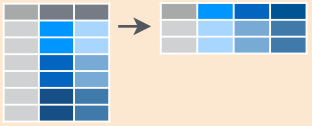
```
iris %>%
  group_by(Species) %>%
  summarise(avg = mean(Sepal.Width)) %>%
  arrange(avg)
```

Data Herorganiseren - Verander de organisatie van de data set



tidyr::**gather(cases, "year", "n", 2:4)**

Verzamel kolommen in rijen



tidyr::**spread(pollution, size, amount)**

Verspreid rijen over kolommen.

tidyr::**separate(storms, date, c("y", "m", "d"))**

Splits één kolom in verschillende kolommen

tidyr::**unite(data, col, ..., sep)**

Voeg verschillende kolommen samen

dplyr::**data_frame(a = 1:3, b = 4:6)**

Combineer vectoren in één dataframe

dplyr::**arrange(mtcars, mpg)**

Sorteer rijen volgens de waarden in een kolom (laag naar hoog).

dplyr::arrange(mtcars, **desc(mpg)**)

Sorteer rijen volgens de waarden in een kolom (hoog naar laag).

dplyr::**rename(tb, y = year)**

Hernoem de kolommen in een dataframe.

Deelverzameling Observaties (Rijen)

dplyr::**filter(iris, Sepal.Length > 7)**

Selecteert de rijen die aan een logisch criterium voldoen.

dplyr::**distinct(iris)**

Verwijdert dubbele rijen.

dplyr::**sample_frac(iris, 0.5, replace = TRUE)**

Selecteert een deel van de rijen op basis van toeval.

dplyr::**sample_n(iris, 10, replace = TRUE)**

Selecteert n rijen op basis van toeval.

dplyr::**slice(iris, 10:15)**

Selecteert rijen op basis van positie.

dplyr::**top_n(storms, 2, date)**

Selecteert en sorteert de top n waarnemingen (per groep als de data gegroepeerd is).

Logica in R - ?Comparison, ?base::Logic

<	Kleiner dan	!=	Niet gelijk aan
>	Groter dan	%in%	Lid van groep
==	Gelijk aan	is.na	Is NA
<=	Kleiner of gelijk aan	!is.na	Is niet NA
>=	Groter of gelijk aan	&,|,!,xor,any,all	Boolean operatoren

devtools::install_github("rstudio/EDAWR") voor data sets

Deelverzameling Variabelen (Kolommen)

dplyr::**select(iris, Sepal.Width, Petal.Length, Species)**

Selecteer kolommen op naam of met helper functie

Helper functies voor select - ?select

select(iris, contains("."))
Selecteert kolommen waarvan de naam een character string bevat

select(iris, ends_with("Length"))
Selecteert kolommen waarvan de naam eindigt met een character string.

select(iris, everything())
Selecteert alle kolommen.

select(iris, matches(".t."))
Selecteert de kolommen waarvan de naam overeenkomt met een expressie.

select(iris, num_range("x", 1:5))
Selecteert de kolommen genaamd x1, x2, x3, x4, x5.

select(iris, one_of(c("Species", "Genus")))
Selecteert de kolommen waarvan de namen onderdeel zijn van een groep van namen.

select(iris, starts_with("Sepal"))
Selecteert kolommen waarvan de naam begint met een character string

select(iris, Sepal.Length:Petal.Width)
Selecteert alle kolommen tussen Sepal.Length and Petal.Width (inclusief).

select(iris, -Species)
Selecteert alle kolommen behalve Species.

Meer informatie: browseVignettes(package = c("dplyr", "tidyr")) • dplyr 0.4.0• tidyr 0.2.0 • Updated: 1/15

Data Samenvatten

dplyr::**summarise(iris, avg = mean(Sepal.Length))**
Vat data samen in een enkele rij.

dplyr::**summarise_each(iris, funs(mean))**
Pas de `summary` functie toe op elke kolom.

dplyr::**count(iris, Species, wt = Sepal.Length)**
Tel het aantal rijen met elke uniek waarde van de variabele. (met of zonder gewichten).

Summarise gebruikt **summary functions**, deze functies gebruiken een vector met waarden en genereren een enkele waarde, bijv.:

dplyr::**first**
Eerste waarde in een vector.
dplyr::**last**
Laatste waarde in een vector.
dplyr::**nth**
N^{de} waarde van een vector.
dplyr::**n**
het aantal waardes in een vector.
dplyr::**n_distinct**
het aantal verschillende waardes in een vector.
IQR
Interkwartielafstand van een vector
min
De kleinste waarde in een vector.
max
De grootste waarde in een vector.
mean
De gemiddelde waarde in een vector.
median
De mediaan in een vector.
var
De variantie in een vector.
sd
De standaard deviatie in een vector.

Data Groeperen

dplyr::**group_by(iris, Species)**
Groepeer data in rijen met dezelfde waarde voor Species.

dplyr::**ungroup(iris)**
Verwijder de groepsinformatie van het data frame.

iris %>% group_by(Species) %>% summarise(…)
Bereken samenvatting voor elke groep.

Nieuwe Variabelen Maken

dplyr::**mutate(iris, sepal = Sepal.Length + Sepal. Width)**
Bereken en voeg één of meer kolommen toe

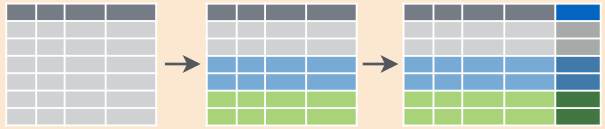
dplyr::**mutate_each(iris, funs(min_rank))**
Pas een window function toe op iedere kolom

dplyr::**transmute(iris, sepal = Sepal.Length + Sepal. Width)**
Bereken één of meer nieuwe kolommen. Verwijder de originele kolommen.

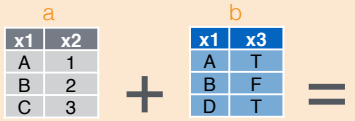
Mutate gebruikt **window functions**, deze functies gebruiken een vector met waarden en genereren een nieuwe vector van waarde bijv.:

dplyr::**lead**
Maakt een kopie van de vector met alle waarden 1 positie naar links verschoven
dplyr::**lag**
Maakt een kopie van de vector met alle waarden 1 positie naar rechts verschoven.
dplyr::**dense_rank**
Aansluitende rangen
dplyr::**min_rank**
Rangen. Gelijken krijgen de laagste rang.
dplyr::**percent_rank**
Rangen geschaald van [0, 1].
dplyr::**row_number**
Rangen. Gelijken krijgen de eerste waarde
dplyr::**ntile**
Verdeel de vector over n klassen
dplyr::**between**
Zijn er waarden tussen a and b?
dplyr::**cume_dist**
Cumulatieve distributie.
dplyr::**cumall**
Cumulatieve versie van `all`
dplyr::**cumany**
Cumulatieve versie van `any`
dplyr::**cummean**
Cumulatieve versie van `mean`
cumsum
Cumulatieve versie van `sum`
cummax
Cumulatieve versie van `max`
cummin
Cumulatieve versie van `min`
cumprod
Cumulatieve versie van `prod`
pmax
Maximum per vector element
pmin
Minimum per vector element

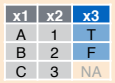
iris %>% group_by(Species) %>% mutate(…)
Bereken nieuwe variabelen per groep



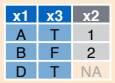
Data Sets Combineren



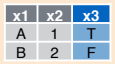
Samenvoegingen muteren



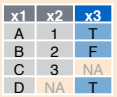
dplyr::**left_join(a, b, by = "x1")**
Voeg overeenkomende rijen samen van b naar a.



dplyr::**right_join(a, b, by = "x1")**
Voeg overeenkomende rijen samen van a naar b

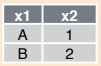


dplyr::**inner_join(a, b, by = "x1")**
Voeg data samen. Behoud alleen de rijen.

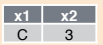


dplyr::**full_join(a, b, by = "x1")**
Voeg data samen. Behoud alle rijen en waarden.

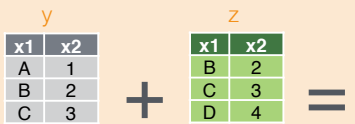
Samenvoegingen selecteren



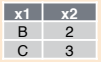
dplyr::**semi_join(a, b, by = "x1")**
Alle rijen die ook in b aanwezig zijn



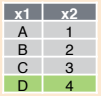
dplyr::**anti_join(a, b, by = "x1")**
Alle rijen die niet in b aanwezig zijn



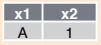
Set Operaties



dplyr::**intersect(y, z)**
Rijen die zowel in y als z voorkomen

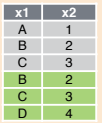


dplyr::**union(y, z)**
Rijen die voorkomen in y of z of beiden

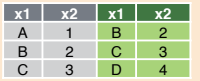


dplyr::**setdiff(y, z)**
Rijen die voorkomen in y maar niet in z

Koppelen



dplyr::**bind_rows(y, z)**
Koppel z aan y als nieuwe rijen.



dplyr::**bind_cols(y, z)**
Koppel z aan y als nieuwe kolommen.
NB rijen houden hun positie!

devtools::install_github("rstudio/EDAWR") voor data sets

Meer weten? browseVignettes(package = c("dplyr", "tidyr")) • dplyr 0.4.0• tidyr 0.2.0 • Updated: 1/15