Intro to Data Wrangling & Scraping Using R

Mochamad Kautzar Ichramsyah
https://www.linkedin.com/in/kautzarichramsyah/
ICW-HDDA-X 2020

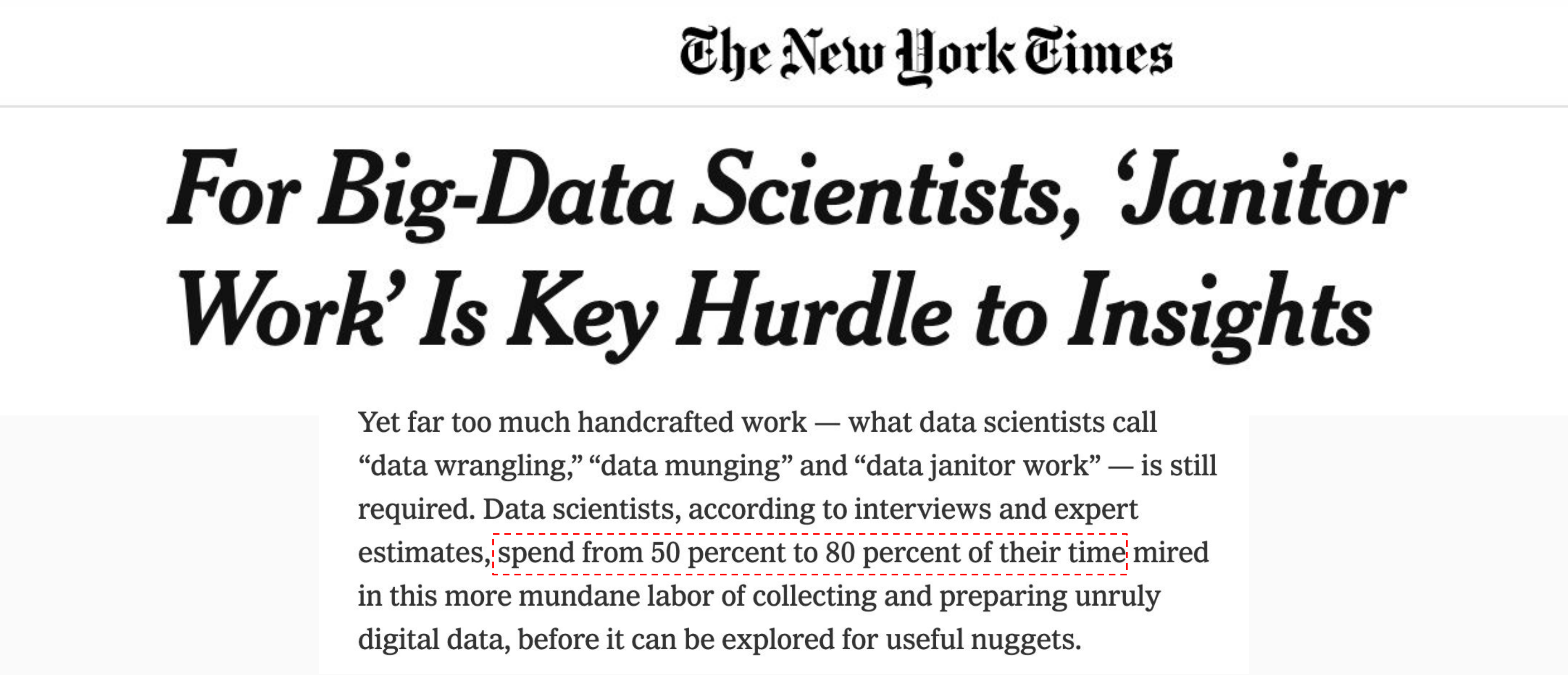
About Data Wrangling

About Data Wrangling (1)

“Anything you ***need*** to do ***before*** doing data ***analysis***.”

- Spot variables and observations
- Derive new variables and observations
- Reshape into best format
- Join multiple datasets
- Group-wise summarize

About Data Wrangling (2)



The New York Times

For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights

Yet far too much handcrafted work — what data scientists call "data wrangling," "data munging" and "data janitor work" — is still required. Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.

https://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html

Data Wrangling Using R

Data Wrangling Using R (1)

https://cran.r-project.org/

https://rstudio.com/products/rstudio/download/

Data Wrangling Using R (2)

Two ***packages*** to help us
doing magic with the structure of data.

tidyr
dplyr

```
install.packages(c('tidyr', 'dplyr'))
library(tidyr)
library(dplyr)
```

Data Wrangling Using R (3)

Another packages we may need.

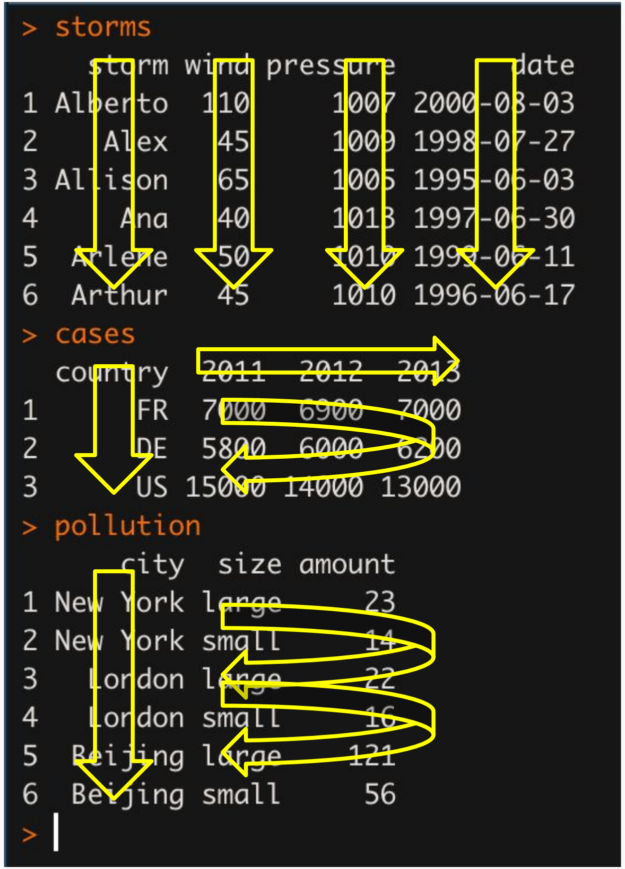
devtools

EDAWR

```
install.packages('devtools')
library(devtools)
install_github('rstudio/EDAWR')
library(EDAWR)
```

```
?storms
?cases
?pollution
?tb
```

Data Wrangling Using R (4)



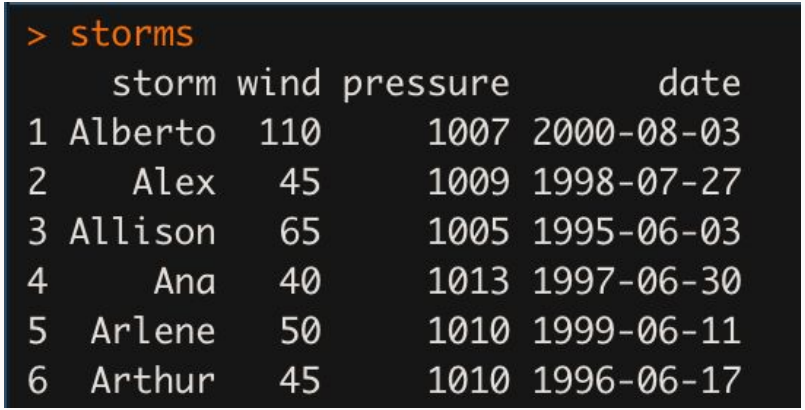
```
> storms
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2    Alex   45     1009 1998-07-27
3 Allison   65     1005 1995-06-03
4     Ana   40     1013 1997-06-30
5  Arlene   50     1010 1999-06-11
6  Arthur   45     1010 1996-06-17
> cases
  country  2011  2012  2013
1      FR  7000  6900  7000
2      DE  5800  6000  6200
3      US 15000 14000 13000
> pollution
      city  size amount
1 New York large     23
2 New York small     14
3   London large     22
4   London small     16
5  Beijing large    121
6  Beijing small     56
> |
```

storms:
storm_name, wind_speed, air_pressure, date
storms$storm
storms$wind
storms$pressure
storms$date

cases:
country, year, count
cases$country
names(cases)[-1]
unlist(cases[1:3, 2:4])

pollution:
city, large particle amount, small particle amount
pollution$city[1, 3, 5]
pollution$amount[1, 3, 5]
pollution$amount[2, 4, 6]

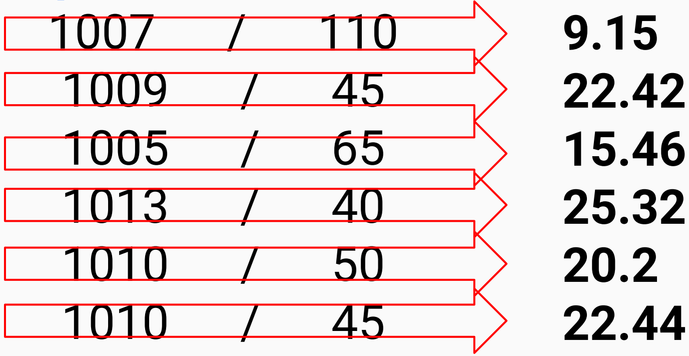
Data Wrangling Using R (5)



```
> storms
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2    Alex   45     1009 1998-07-27
3 Allison   65     1005 1995-06-03
4     Ana   40     1013 1997-06-30
5  Arlene   50     1010 1999-06-11
6  Arthur   45     1010 1996-06-17
```

$$ratio = \frac{pressure}{wind}$$

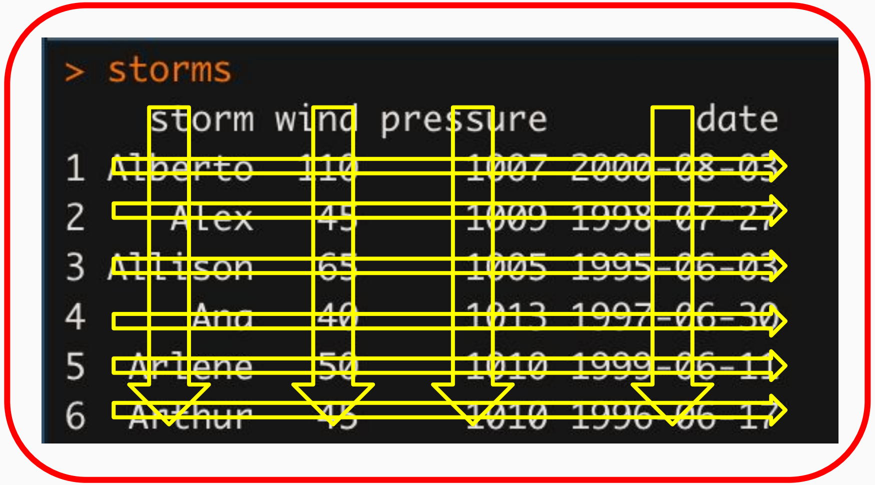
storms$pressure / storms$wind



```
1007 / 110   9.15
1009 / 45    22.42
1005 / 65    15.46
1013 / 40    25.32
1010 / 50    20.2
1010 / 45    22.44
```

Tidying Data Using tidyr

Tidying Data Using tidyr (1)



```
> storms
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2    Alex   45     1009 1998-07-27
3 Allison   65     1005 1995-06-03
4     Ana   40     1013 1997-06-30
5  Arlene   50     1010 1999-06-11
6  Arthur   45     1010 1996-06-17
```

1. Each **variable** saved in its own **column**
2. Each **observation** saved in its own **row**
3. Each type of observation saved in **single table**

Tidying Data Using tidyr (2)

Tidy Data!

Easy to access
Preserves observations

Tidying Data Using tidyr (3)

tidyr

Package **to reshape** layout of tables.

Two main functions:

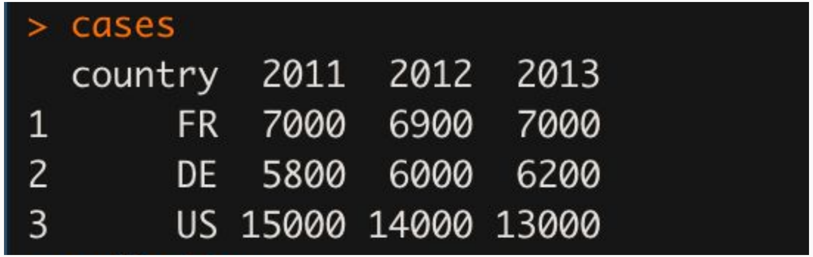
gather()

spread()

```
# install.packages('tidyr')
library('tidyr')
?gather
?spread
```

Tidying Data Using tidyr (4)

30 seconds to guess, raise your hand please!

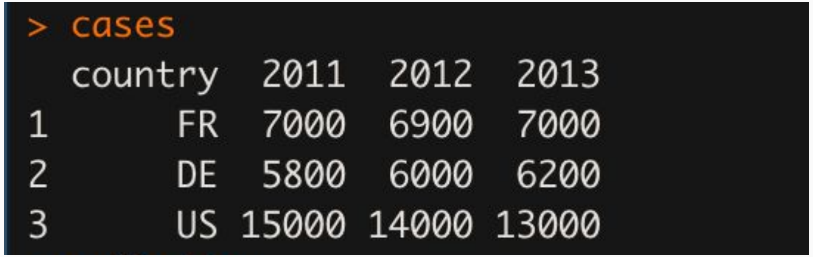


```
> cases
  country  2011  2012  2013
1      FR  7000  6900  7000
2      DE  5800  6000  6200
3      US 15000 14000 13000
```

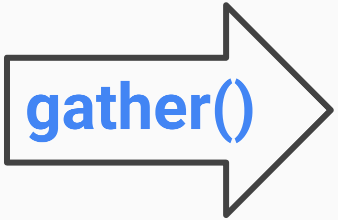
If dataset ***cases*** has been tidied up with 3 variables:
country, *year*, and *count*,
how the data would look like?

Tidying Data Using tidyr (5)

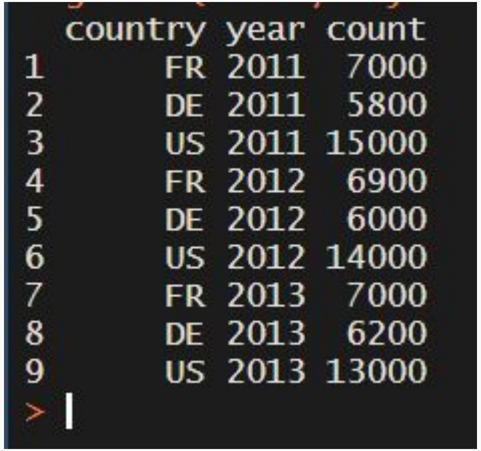
key = 'year' (former column names)
value = 'count' (former cells)



```
> cases
  country  2011  2012  2013
1      FR  7000  6900  7000
2      DE  5800  6000  6200
3      US 15000 14000 13000
```



gather()



```
  country year count
1      FR 2011  7000
2      DE 2011  5800
3      US 2011 15000
4      FR 2012  6900
5      DE 2012  6000
6      US 2012 14000
7      FR 2013  7000
8      DE 2013  6200
9      US 2013 13000
> |
```

Tidying Data Using tidyr (6)

Collapses multiple columns into two columns.

```
gather(cases, 'year', 'count', 2:4)
```

Function to reshape the data frame.

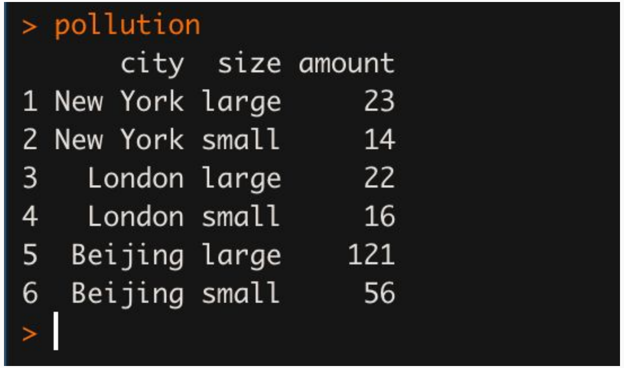
Data frame to reshape.

Name of the new key column. String.

Name of the new value column. String.

Names or numeric indexes of columns to collapse.

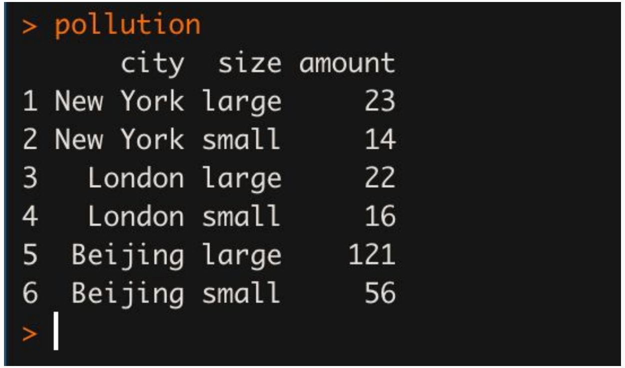
30 seconds to guess, raise your hand please!



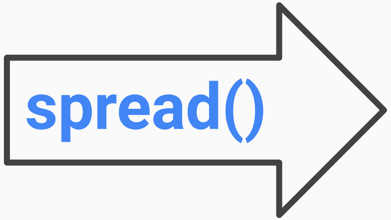
```
> pollution
      city  size amount
1 New York large     23
2 New York small     14
3   London large     22
4   London small     16
5  Beijing large    121
6  Beijing small     56
> 
```

If dataset ***pollution*** has been tidied up with 3 variables:
city, *large*, and *small*,
how the data would look like?

Tidying Data Using tidyr (8)

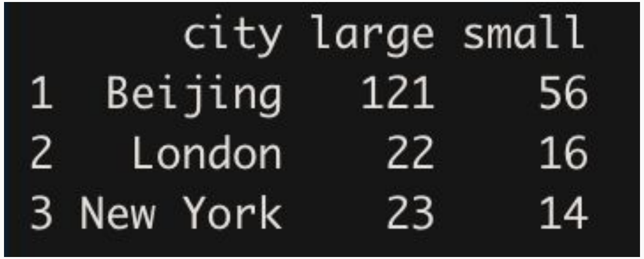


```
> pollution
      city  size amount
1 New York large     23
2 New York small     14
3   London large     22
4   London small     16
5  Beijing large    121
6  Beijing small     56
> 
```



spread()

key = size (former column1 names)
value = amount (former column2 names)



```
      city large small
1  Beijing   121    56
2   London    22    16
3 New York    23    14
```

Tidying Data Using tidyr (9)

Generates multiple columns from two columns.

spread(pollution, size, amount)

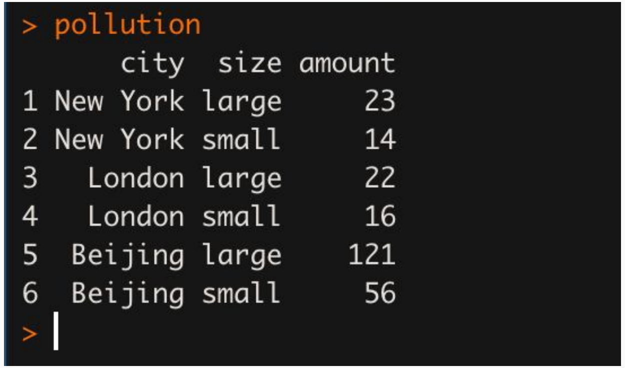
Function to reshape the data frame.

Data frame to reshape.

Column to use for keys, create new column names.

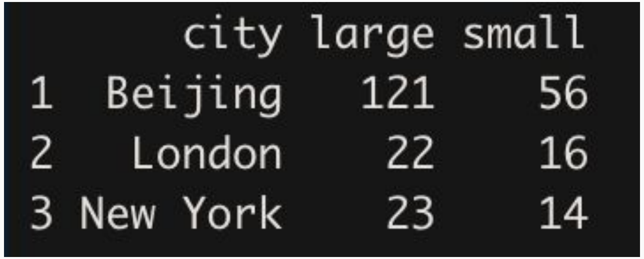
Column to use for values, create new column cells.

Tidying Data Using tidyr (8)



```
> pollution
      city  size amount
1 New York large     23
2 New York small     14
3   London large     22
4   London small     16
5  Beijing large    121
6  Beijing small     56
> 
```

spread()



```
      city large small
1  Beijing   121    56
2   London    22    16
3 New York    23    14
```

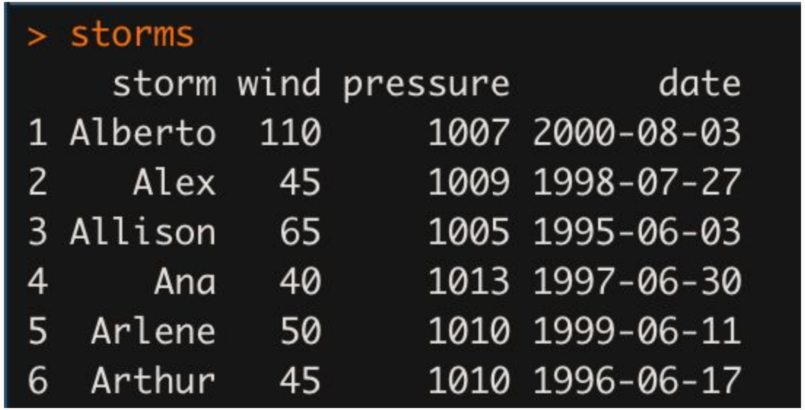
x <- spread(pollution, size, amount)

gather()

gather(x, 'size', 'amount', 2:3)

Tidying Data Using tidyr (9)

Do you know we still have **three more variables hidden** in storms?



```
> storms
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2    Alex   45     1009 1998-07-27
3 Allison   65     1005 1995-06-03
4     Ana   40     1013 1997-06-30
5  Arlene   50     1010 1999-06-11
6  Arthur   45     1010 1996-06-17
```

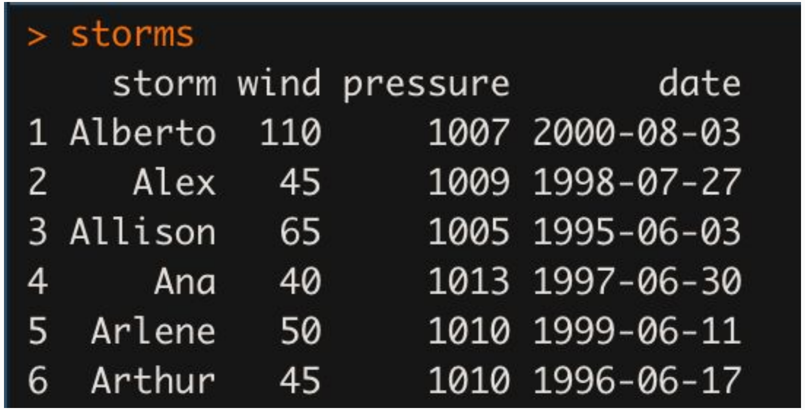
- year
- month
- day

Tidying Data Using tidyr (10)

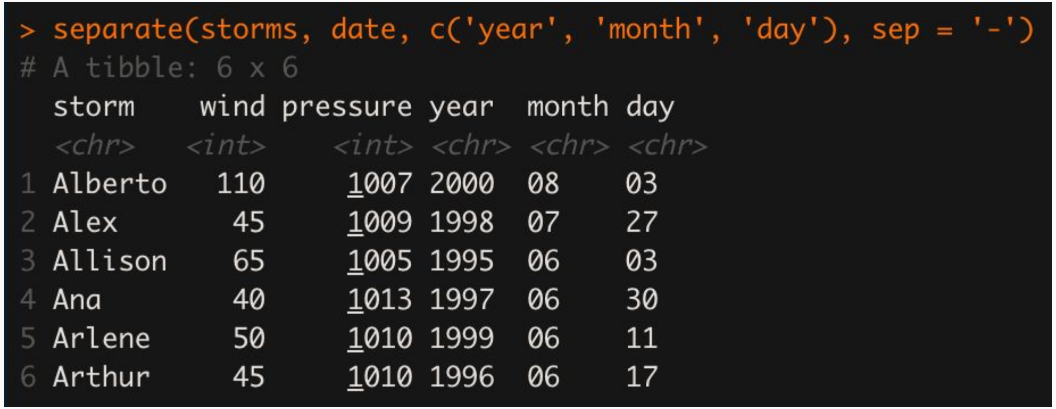
separate()

Splits a column by a character string operator.

separate(storms, date, c('year', 'month', 'day'), sep = '-')



```
> storms
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2    Alex   45     1009 1998-07-27
3 Allison   65     1005 1995-06-03
4     Ana   40     1013 1997-06-30
5  Arlene   50     1010 1999-06-11
6  Arthur   45     1010 1996-06-17
```



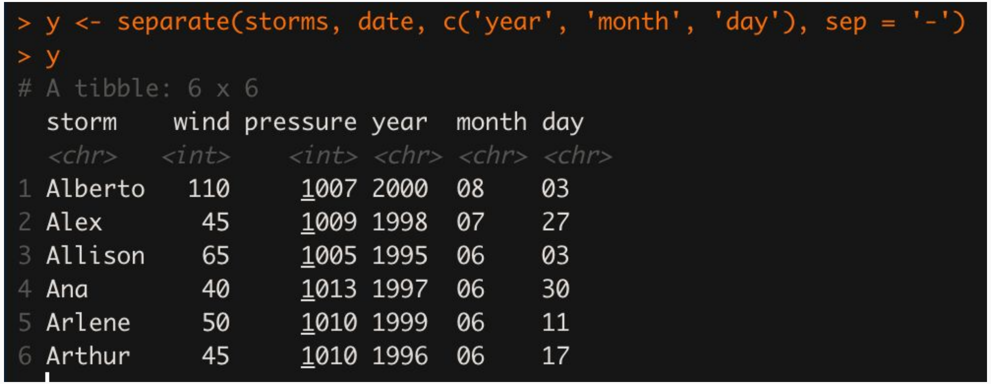
```
> separate(storms, date, c('year', 'month', 'day'), sep = '-')
# A tibble: 6 x 6
  storm    wind pressure year  month day
  <chr>   <int>    <int> <chr> <chr> <chr>
1 Alberto   110     1007 2000  08    03
2 Alex       45     1009 1998  07    27
3 Allison    65     1005 1995  06    03
4 Ana        40     1013 1997  06    30
5 Arlene     50     1010 1999  06    11
6 Arthur     45     1010 1996  06    17
```

Tidying Data Using tidyr (10)

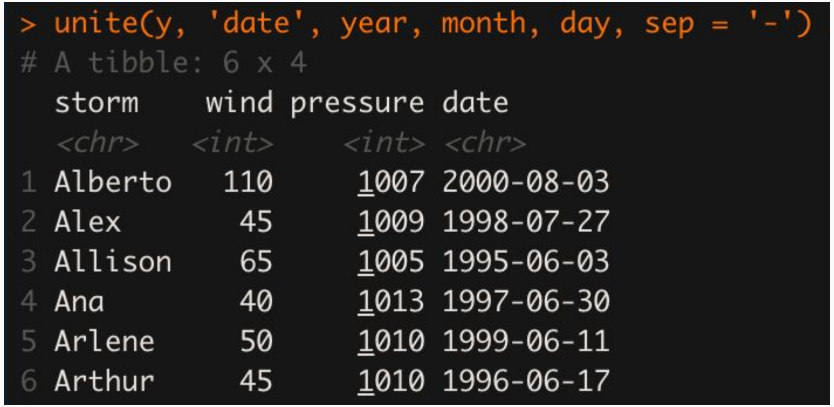
unite()

Unites columns into a single column.

unite(y, 'date', year, month, day, sep = '-')



```
> y <- separate(storms, date, c('year', 'month', 'day'), sep = '-')
> y
# A tibble: 6 x 6
  storm    wind pressure year  month day
  <chr>   <int>    <int> <chr> <chr> <chr>
1 Alberto   110     1007 2000  08    03
2 Alex       45     1009 1998  07    27
3 Allison    65     1005 1995  06    03
4 Ana        40     1013 1997  06    30
5 Arlene     50     1010 1999  06    11
6 Arthur     45     1010 1996  06    17
```



```
> unite(y, 'date', year, month, day, sep = '-')
# A tibble: 6 x 4
  storm    wind pressure date
  <chr>   <int>    <int> <chr>
1 Alberto   110     1007 2000-08-03
2 Alex       45     1009 1998-07-27
3 Allison    65     1005 1995-06-03
4 Ana        40     1013 1997-06-30
5 Arlene     50     1010 1999-06-11
6 Arthur     45     1010 1996-06-17
```

Tidying Data Using tidyr (11)

Recap

tidyr Package to reshape layout of data sets

gather() Make observations from variables

spread() Make variables from observations

separate() Split single column to many columns

unite() Merge many columns to single column

Manipulate Data Using dplyr

Manipulate Data Using dplyr (1)

dplyr

Package **to transform** tabular data.

```
# install.packages('dplyr')
library('dplyr')
?select
?filter
?arrange
?mutate
?summarise
?group_by
```

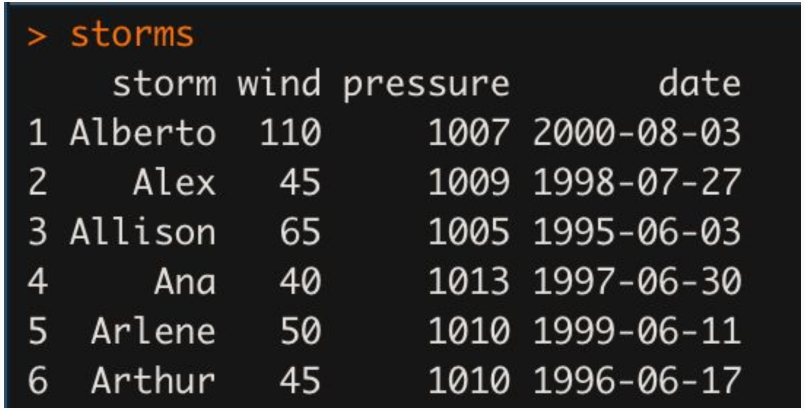
Manipulate Data Using dplyr (2)

How to access information?

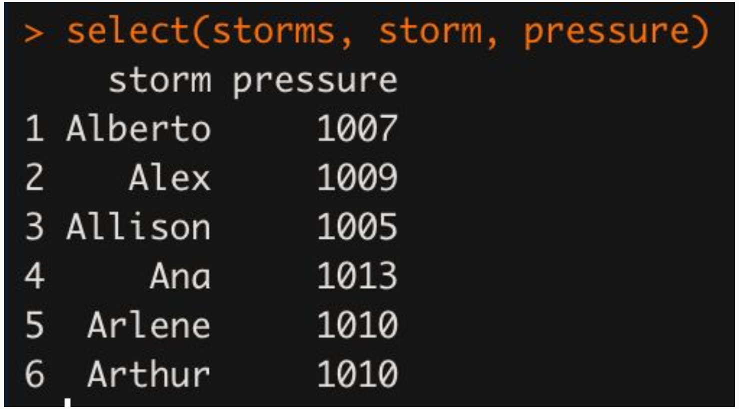
1. Extract existing variables.	select()
2. Extract existing observations.	filter()
3. Derive new variables (from existing variables)	mutate()
4. Change the unit of analysis	summarise()

Manipulate Data Using dplyr (3)

select()



```
> storms
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2    Alex   45     1009 1998-07-27
3 Allison   65     1005 1995-06-03
4     Ana   40     1013 1997-06-30
5  Arlene   50     1010 1999-06-11
6  Arthur   45     1010 1996-06-17
```

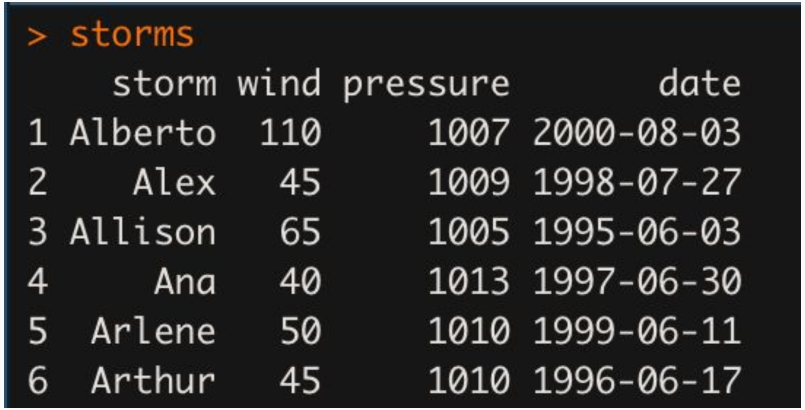


```
> select(storms, storm, pressure)
    storm pressure
1 Alberto     1007
2    Alex     1009
3 Allison     1005
4     Ana     1013
5  Arlene     1010
6  Arthur     1010
```

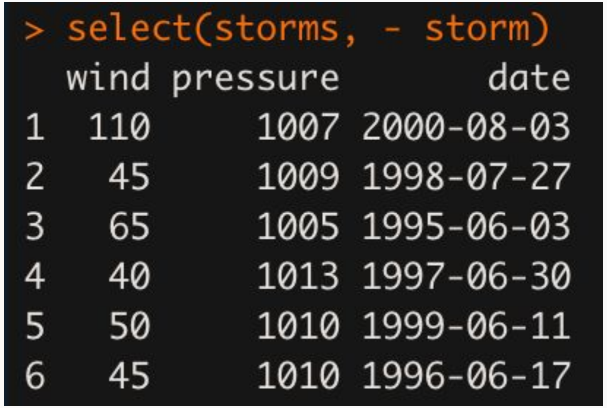
select(storms, storm, pressure)

Manipulate Data Using dplyr (4)

select()



```
> storms
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2    Alex   45     1009 1998-07-27
3 Allison   65     1005 1995-06-03
4     Ana   40     1013 1997-06-30
5  Arlene   50     1010 1999-06-11
6  Arthur   45     1010 1996-06-17
```



```
> select(storms, - storm)
  wind pressure       date
1  110     1007 2000-08-03
2   45     1009 1998-07-27
3   65     1005 1995-06-03
4   40     1013 1997-06-30
5   50     1010 1999-06-11
6   45     1010 1996-06-17
```

select(storms, - storm)
select(storms, wind:date)

Manipulate Data Using dplyr (5)

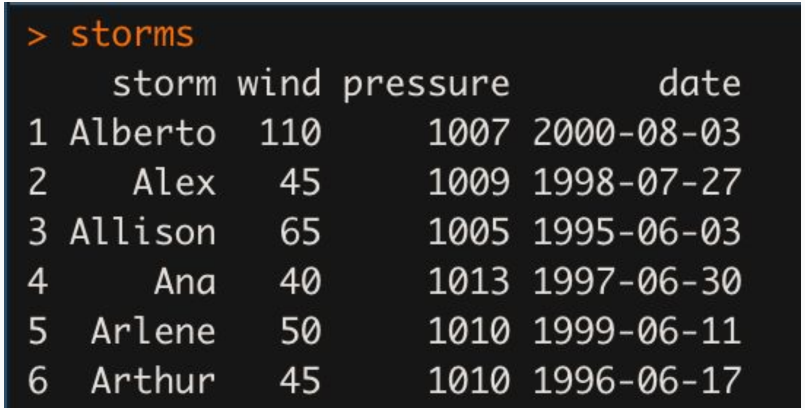
Useful select() functions

*** Blue colored functions come in dplyr**

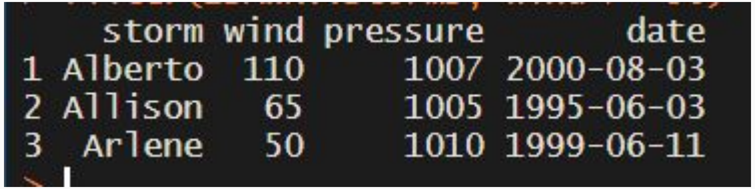
-	Select everything but
:	Select range
contains()	Select columns whose name contains a character string
ends_with()	Select columns whose name ends with a string
everything()	Select every column
matches()	Select columns whose name matches a regular expression
num_range()	Select columns named X1, X2, X3, X4, X5
one_of()	Select columns names are in group of names
starts_with()	Select columns whose name starts with a string

Manipulate Data Using dplyr (6)

filter()



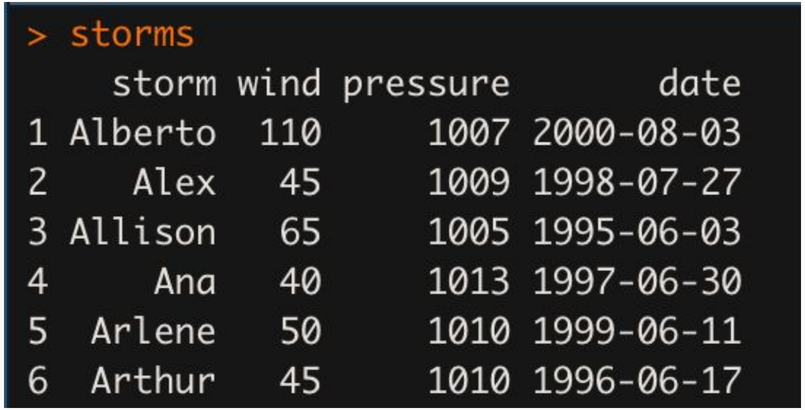
```
> storms
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2    Alex   45     1009 1998-07-27
3 Allison   65     1005 1995-06-03
4     Ana   40     1013 1997-06-30
5  Arlene   50     1010 1999-06-11
6  Arthur   45     1010 1996-06-17
```



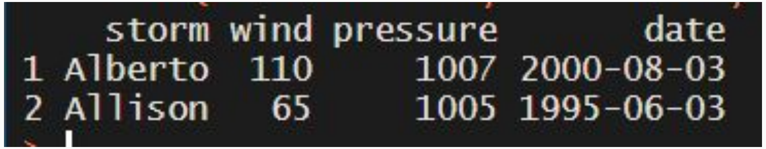
```
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2 Allison   65     1005 1995-06-03
3  Arlene   50     1010 1999-06-11
>
```

filter(storms, **wind >= 50**)

filter()



```
> storms
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2    Alex   45     1009 1998-07-27
3 Allison   65     1005 1995-06-03
4     Ana   40     1013 1997-06-30
5  Arlene   50     1010 1999-06-11
6  Arthur   45     1010 1996-06-17
```



```
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2 Allison   65     1005 1995-06-03
```

filter(storms, wind >= 50, storm %in% c('Alberto', 'Alex', 'Allison'))

Manipulate Data Using dplyr (8)

Logical Tests in R

?Comparison

<	Less than
>	Greater than
==	Equal to
<=	Less than or equal to
>=	Greater than or equal to
!=	Not equal to
%in%	Group membership
is.na	Is NA (Not Available)
!is.na	Is not NA (Not Available)

Manipulate Data Using dplyr (9)

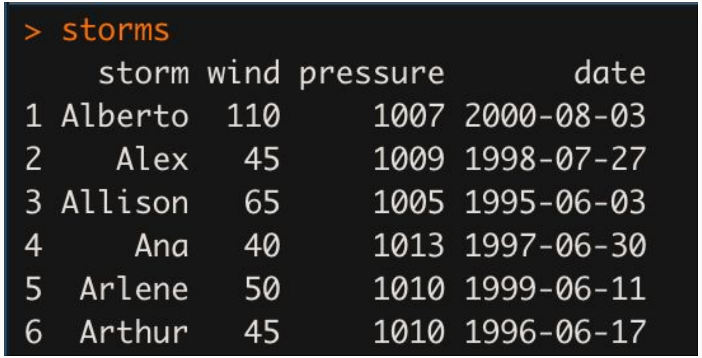
Logical Tests in R

?base::Logic

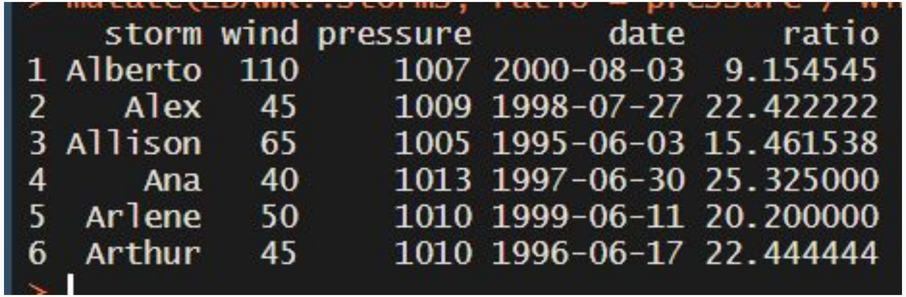
&	Boolean AND
|	Boolean OR
xor	Exactly OR
!	Not
any	Any true
all	All true

Manipulate Data Using dplyr (10)

mutate()



```
> storms
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2    Alex   45     1009 1998-07-27
3 Allison   65     1005 1995-06-03
4     Ana   40     1013 1997-06-30
5  Arlene   50     1010 1999-06-11
6  Arthur   45     1010 1996-06-17
```

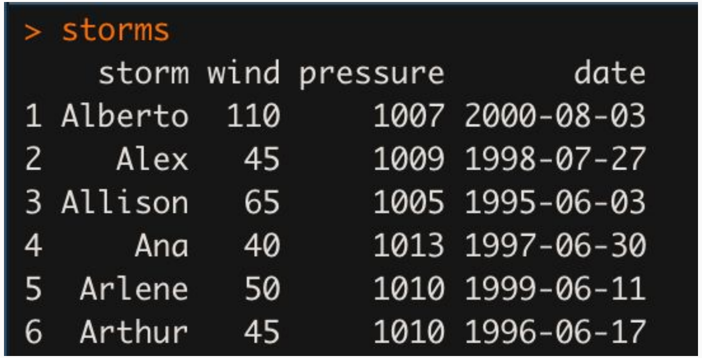


```
> mutate(EDAWR::storms, ratio = pressure / wi
    storm wind pressure       date     ratio
1 Alberto  110     1007 2000-08-03  9.154545
2    Alex   45     1009 1998-07-27 22.422222
3 Allison   65     1005 1995-06-03 15.461538
4     Ana   40     1013 1997-06-30 25.325000
5  Arlene   50     1010 1999-06-11 20.200000
6  Arthur   45     1010 1996-06-17 22.444444
>
```

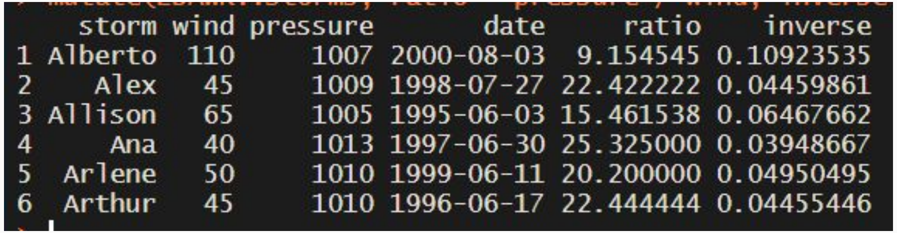
mutate(storms, **ratio = pressure / wind**)

Manipulate Data Using dplyr (11)

mutate()



```
> storms
     storm wind pressure       date
1  Alberto  110     1007 2000-08-03
2     Alex   45     1009 1998-07-27
3  Allison   65     1005 1995-06-03
4      Ana   40     1013 1997-06-30
5   Arlene   50     1010 1999-06-11
6   Arthur   45     1010 1996-06-17
```



```
     storm wind pressure       date     ratio    inverse
1  Alberto  110     1007 2000-08-03  9.154545 0.10923535
2     Alex   45     1009 1998-07-27 22.422222 0.04459861
3  Allison   65     1005 1995-06-03 15.461538 0.06467662
4      Ana   40     1013 1997-06-30 25.325000 0.03948667
5   Arlene   50     1010 1999-06-11 20.200000 0.04950495
6   Arthur   45     1010 1996-06-17 22.444444 0.04455446
```

mutate(storms, ratio = pressure / wind, inverse = ratio ^ -1)

Manipulate Data Using dplyr (12)

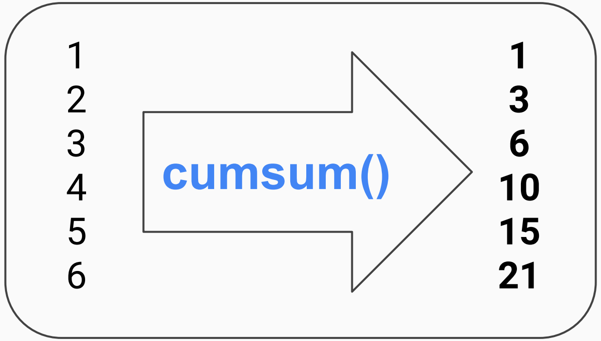
Useful mutate() functions

***All take a vector of values and return a vector of values **Blue colored functions come in dplyr**

pmin(), pmax()	Element-wise min and max
cummin(), cummax()	Cumulative min and max
cumsum(), cumprod()	Cumulative sum and product
between()	Are values between a and b?
cume_dist()	Cumulative distribution of values
cumall(), cumany()	Cumulative all and any
cummean()	Cumulative mean
lead(), lag()	Copy with values one position
ntile()	Bin vector into n buckets
dense_rank(), min_rank(), percent_rank(),row_number()	Various ranking methods

Manipulate Data Using dplyr (13)

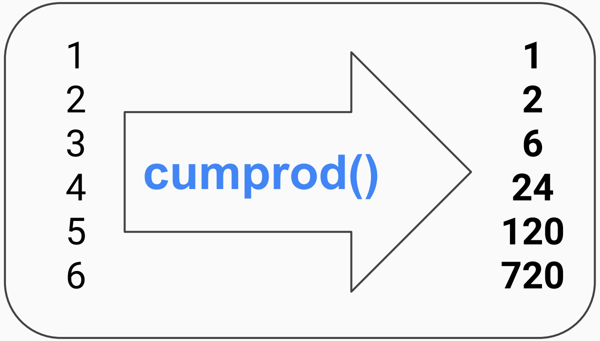
“Window” Functions



1
2
3
4
5
6

cumsum()

1
3
6
10
15
21



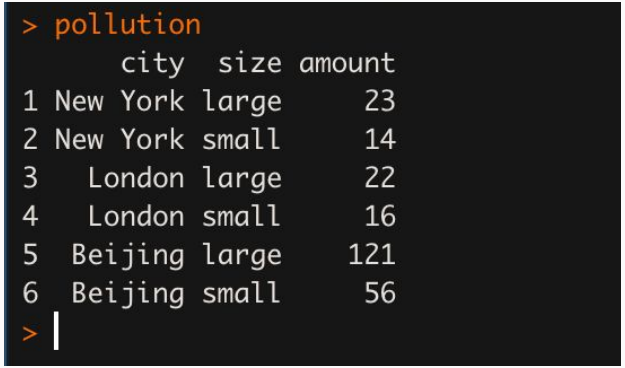
1
2
3
4
5
6

cumprod()

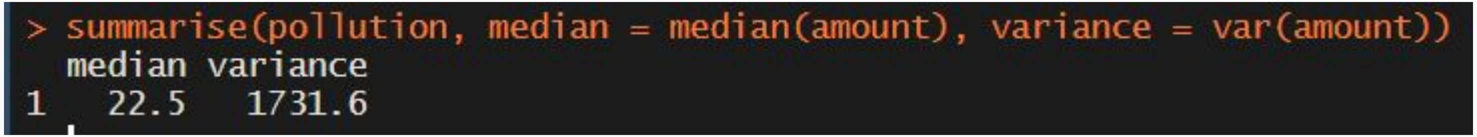
1
2
6
24
120
720

Manipulate Data Using dplyr (14)

summarise()



```
> pollution
      city  size amount
1 New York large     23
2 New York small     14
3   London large     22
4   London small     16
5  Beijing large    121
6  Beijing small     56
> 
```

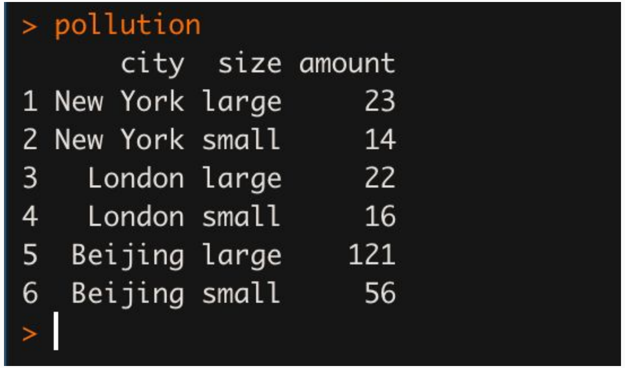


```
> summarise(pollution, median = median(amount), variance = var(amount))
  median variance
1   22.5   1731.6
```

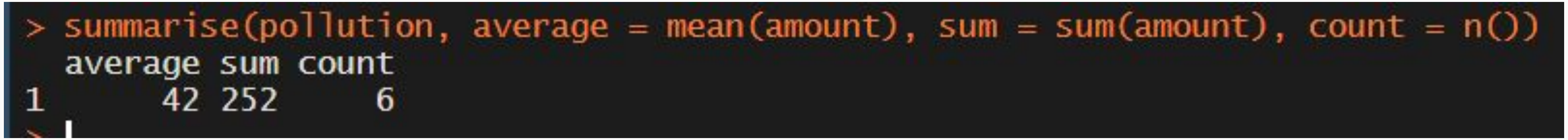
summarise(pollution, median = median(amount), variance = var(amount))

Manipulate Data Using dplyr (15)

summarise()



```
> pollution
      city  size amount
1 New York large     23
2 New York small     14
3   London large     22
4   London small     16
5  Beijing large    121
6  Beijing small     56
> 
```



```
> summarise(pollution, average = mean(amount), sum = sum(amount), count = n())
  average sum count
1      42 252     6
>
```

summarise(pollution, average = mean(amount), sum = sum(amount), count = n())

Manipulate Data Using dplyr (16)

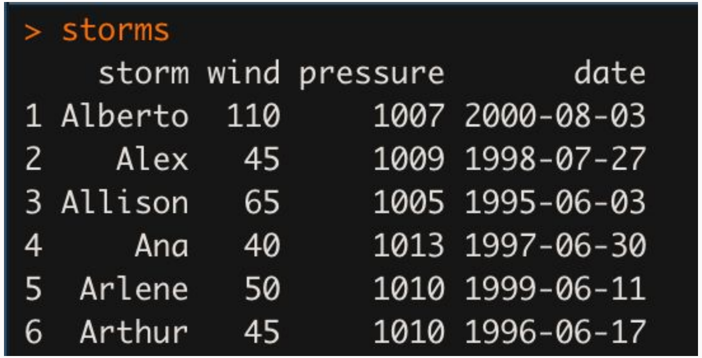
Useful summary() functions

***All take a vector of values and return a single value **Blue colored functions come in dplyr**

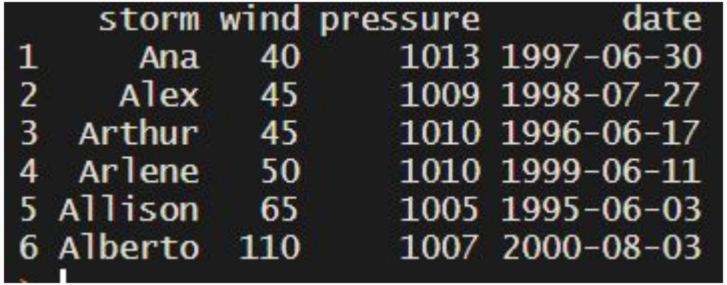
min(), max()	Minimum and maximum values
mean(), median()	Mean and median values
sum()	Sum of values
var(), sd()	Variance and standard deviation of a vector
first(), last()	First and last value in a vector
nth()	N-th value in a vector
n()	The number of values in a vector
n_distinct()	The number of unique values in a vector

Manipulate Data Using dplyr (17)

arrange()



```
> storms
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2    Alex   45     1009 1998-07-27
3 Allison   65     1005 1995-06-03
4     Ana   40     1013 1997-06-30
5  Arlene   50     1010 1999-06-11
6  Arthur   45     1010 1996-06-17
```

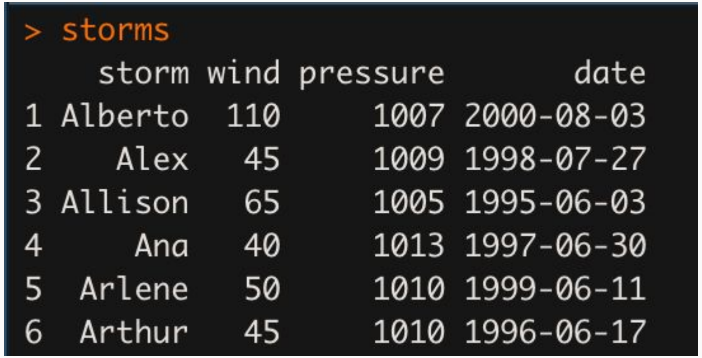


```
    storm wind pressure       date
1     Ana   40     1013 1997-06-30
2    Alex   45     1009 1998-07-27
3  Arthur   45     1010 1996-06-17
4  Arlene   50     1010 1999-06-11
5 Allison   65     1005 1995-06-03
6 Alberto  110     1007 2000-08-03
```

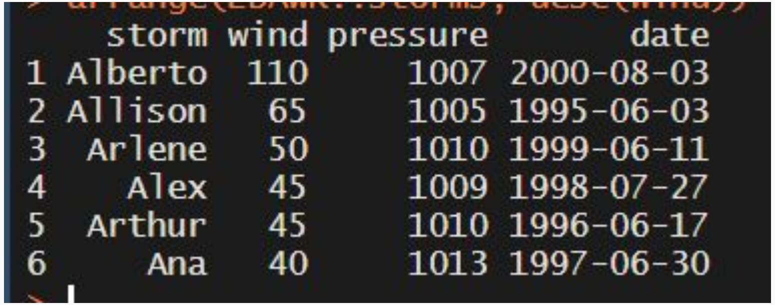
arrange(storms, **wind**)

Manipulate Data Using dplyr (18)

arrange()



```
> storms
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2    Alex   45     1009 1998-07-27
3 Allison   65     1005 1995-06-03
4     Ana   40     1013 1997-06-30
5  Arlene   50     1010 1999-06-11
6  Arthur   45     1010 1996-06-17
```

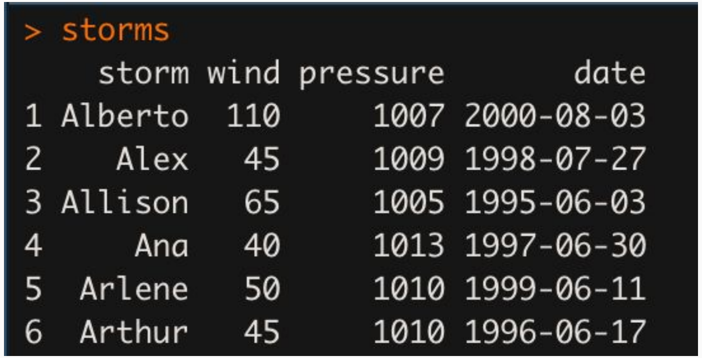


```
> arrange(EDAWR::storms, desc(wind))
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2 Allison   65     1005 1995-06-03
3  Arlene   50     1010 1999-06-11
4    Alex   45     1009 1998-07-27
5  Arthur   45     1010 1996-06-17
6     Ana   40     1013 1997-06-30
> |
```

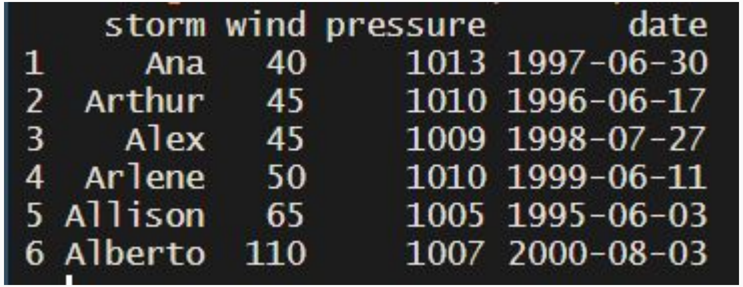
arrange(storms, **desc(wind)**)

Manipulate Data Using dplyr (19)

arrange()



```
> storms
    storm wind pressure       date
1 Alberto  110     1007 2000-08-03
2    Alex   45     1009 1998-07-27
3 Allison   65     1005 1995-06-03
4     Ana   40     1013 1997-06-30
5  Arlene   50     1010 1999-06-11
6  Arthur   45     1010 1996-06-17
```



```
    storm wind pressure       date
1     Ana   40     1013 1997-06-30
2  Arthur   45     1010 1996-06-17
3    Alex   45     1009 1998-07-27
4  Arlene   50     1010 1999-06-11
5 Allison   65     1005 1995-06-03
6 Alberto  110     1007 2000-08-03
```

arrange(storms, wind, date)

Manipulate Data Using dplyr (20)

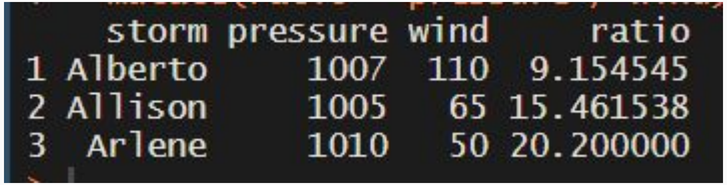
%>% pipe operator

```
select(storms, storm, pressure)
storms %>% select(storm, pressure)

filter(storms, wind >= 50)
storms %>% filter(wind >= 50)

mutate(storms, ratio = pressure / wind)
storms %>% mutate(ratio = pressure / wind)

summarise(storms, median = median(amount))
storms %>% summarise(median = median(amount))
```

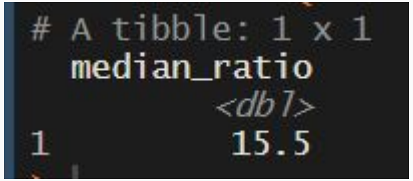
Manipulate Data Using dplyr (21)

%>% pipe operator

```
storms %>%
  select(storm, pressure, wind) %>%
  filter(wind >= 50) %>%
  mutate(ratio = pressure / wind) %>%
  summarise(median_ratio = median(ratio))
```



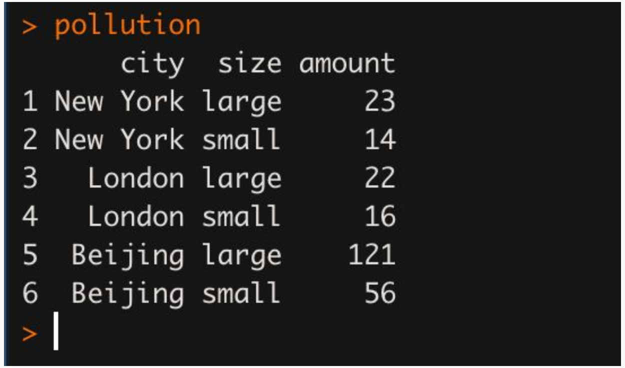
```
    storm pressure wind     ratio
1 Alberto     1007  110  9.154545
2 Allison     1005   65 15.461538
3  Arlene     1010   50 20.200000
```



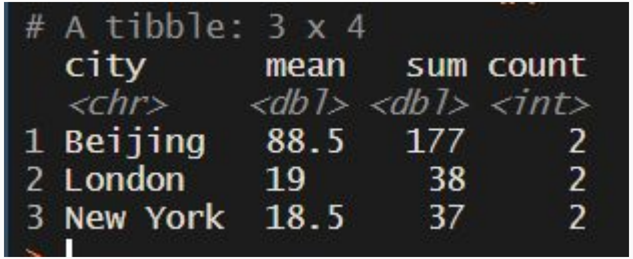
```
# A tibble: 1 x 1
  median_ratio
         <dbl>
1         15.5
```

Manipulate Data Using dplyr (23)

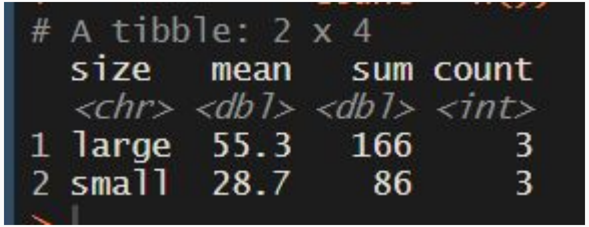
group_by() + summarise()



```
> pollution
      city  size amount
1 New York large     23
2 New York small     14
3   London large     22
4   London small     16
5  Beijing large    121
6  Beijing small     56
> |
```



```
# A tibble: 3 x 4
  city      mean   sum count
  <chr>    <dbl> <dbl> <int>
1 Beijing   88.5   177     2
2 London    19      38     2
3 New York  18.5    37     2
```



```
# A tibble: 2 x 4
  size   mean   sum count
  <chr> <dbl> <dbl> <int>
1 large  55.3   166     3
2 small  28.7    86     3
```

Manipulate Data Using dplyr (22)

%>% pipe operator

Shortcut Key

Cmd + Shift + M (Mac)

or

Ctrl + Shift + M (Windows)

Manipulate Data Using dplyr (24)

group_by() + summarise()

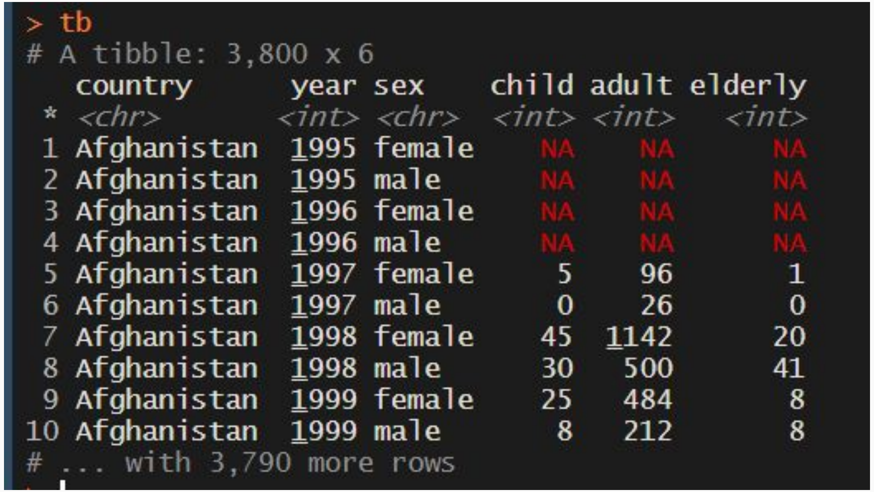
```
pollution %>%
  group_by(size) %>%
  summarise(
      mean = mean(amount),
      sum = sum(amount),
      count = n())
```

```
pollution %>%
  group_by(city) %>%
  summarise(
      mean = mean(amount),
      sum = sum(amount),
      count = n())
```

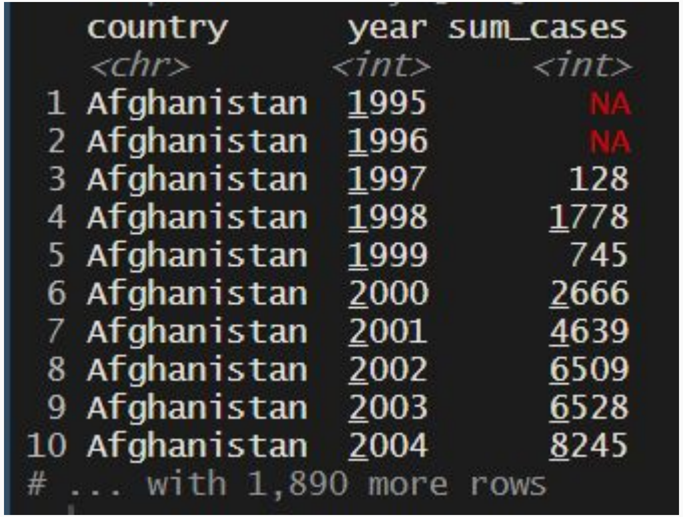
Please check about ungroup() function
(type ?ungroup in your R console)

Manipulate Data Using dplyr (25)

group_by() + summarise()



```
> tb
# A tibble: 3,800 x 6
   country      year sex    child adult elderly
 * <chr>       <int> <chr>  <int> <int>   <int>
 1 Afghanistan  1995 female    NA    NA      NA
 2 Afghanistan  1995 male      NA    NA      NA
 3 Afghanistan  1996 female    NA    NA      NA
 4 Afghanistan  1996 male      NA    NA      NA
 5 Afghanistan  1997 female     5    96       1
 6 Afghanistan  1997 male       0    26       0
 7 Afghanistan  1998 female    45  1142      20
 8 Afghanistan  1998 male      30   500      41
 9 Afghanistan  1999 female    25   484       8
10 Afghanistan  1999 male       8   212       8
# ... with 3,790 more rows
```



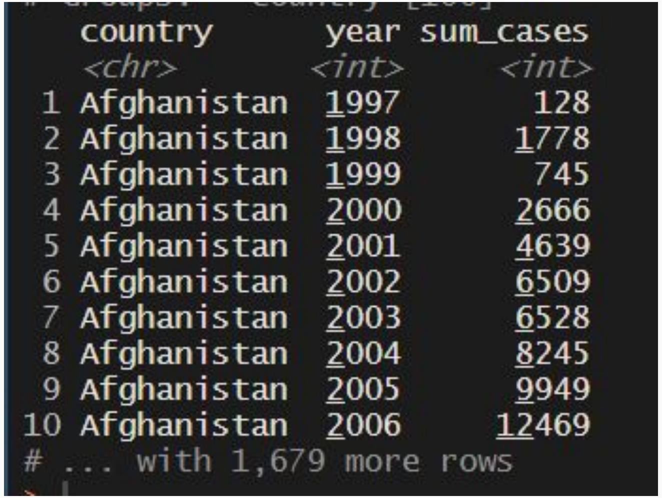
```
   country      year sum_cases
   <chr>       <int>     <int>
 1 Afghanistan  1995        NA
 2 Afghanistan  1996        NA
 3 Afghanistan  1997       128
 4 Afghanistan  1998      1778
 5 Afghanistan  1999       745
 6 Afghanistan  2000      2666
 7 Afghanistan  2001      4639
 8 Afghanistan  2002      6509
 9 Afghanistan  2003      6528
10 Afghanistan  2004      8245
# ... with 1,890 more rows
```

30 seconds to guess, raise your hand please!

Manipulate Data Using dplyr (26)

group_by() + summarise()

```
tb %>%
  group_by(country, year) %>%
  summarise(sum_cases = sum(child + adult + elderly)) %>%
  filter(!is.na(sum_cases))
```



```
   country      year sum_cases
   <chr>       <int>     <int>
 1 Afghanistan  1997       128
 2 Afghanistan  1998      1778
 3 Afghanistan  1999       745
 4 Afghanistan  2000      2666
 5 Afghanistan  2001      4639
 6 Afghanistan  2002      6509
 7 Afghanistan  2003      6528
 8 Afghanistan  2004      8245
 9 Afghanistan  2005      9949
10 Afghanistan  2006     12469
# ... with 1,679 more rows
```

Recap

dplyr Package to transform tabular data

select() filter() mutate() summarise() arrange() Main functions you should understand

%>% pipe operator Simplifying workflow using dplyr

group_by() It works like magic with summarise()

Joining Data (still using dplyr)

Joining Data (1)

bind_cols()

y

x1	x2
a	1
b	2
c	3

+

z

x1	x2
a	1
b	2
c	3

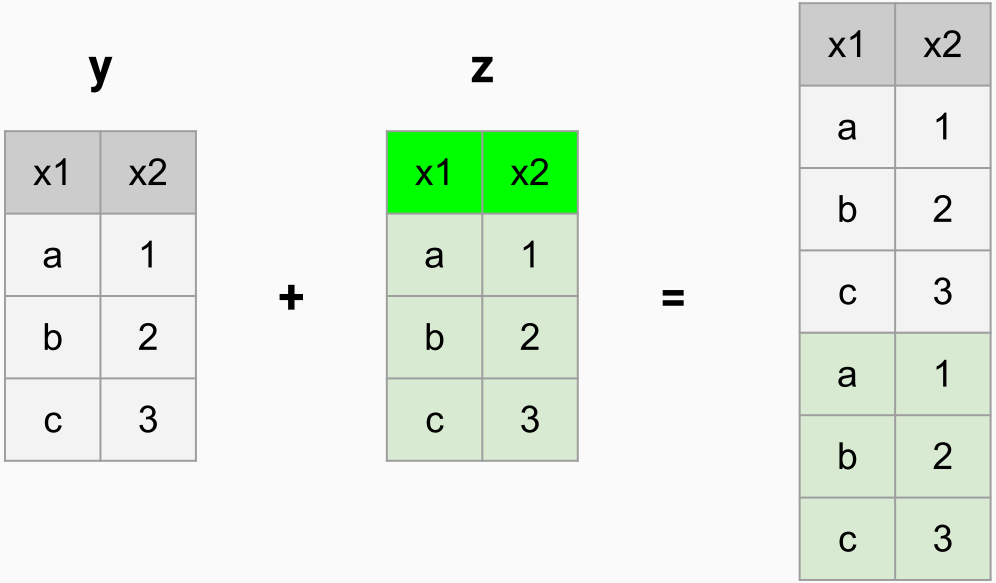
=

x1	x2	x1	x2
a	1	a	1
b	2	b	2
c	3	c	3

bind_cols(y, z)

Joining Data (2)

bind_rows()



y

x1	x2
a	1
b	2
c	3

+

z

x1	x2
a	1
b	2
c	3

=

x1	x2
a	1
b	2
c	3
a	1
b	2
c	3

bind_rows(y, z)

Joining Data (3)

union()

y

x1	x2
a	1
b	2
c	3

+

z

x1	x2
a	1
b	2
d	4

=

x1	x2
a	1
b	2
c	3
d	4

union(y, z)

Joining Data (4)

intersect()

y

x1	x2
a	1
b	2
c	3

+

z

x1	x2
b	2
c	3
d	4

=

x1	x2
b	2
c	3

intersect(y, z)

Joining Data (5)

setdiff()

y

x1	x2
a	1
b	2
c	3

+

z

x1	x2
b	2
c	3
d	4

=

x1	x2
a	1

setdiff(y, z)

Joining Data (6)

left_join()

y

x1	x2
a	1
b	2
c	3
d	4

+

z

x1	x3
b	!
c	@
d	#
e	$

=

x1	x2	x3
a	1	<NA>
b	2	!
c	3	@
d	4	#

```
left_join(y, z, by = 'x1')
```

Joining Data (7)

inner_join()

y

x1	x2
a	1
b	2
c	3
d	4

+

z

x1	x3
b	!
c	@
d	#
e	$

=

x1	x2	x3
b	2	!
c	3	@
d	4	#

```
inner_join(y, z, by = 'x1')
```

Joining Data (8)

Please check other join functions!

right_join

semi_join()

anti_join()

Recap: Best format for analysis

1. **Variables** in columns
2. **Observations** in rows
3. **Separate** all variables
4. **Unit of analysis** matches
5. **Single** table

Learn more at:

Data Wrangling with dplyr and tidyr Cheat Sheet

About Data Scraping

About Data Scraping (1)

“Data *extracting* from websites.”

Data Scraping Using R

Data Scraping Using R (1)

One of ***package*** can help us to do data scraping.

rvest

```
install.packages('rvest')
library(rvest)
```

Data Scraping Using R (2)

Another packages we may need.

selectr

xml2

jsonlite

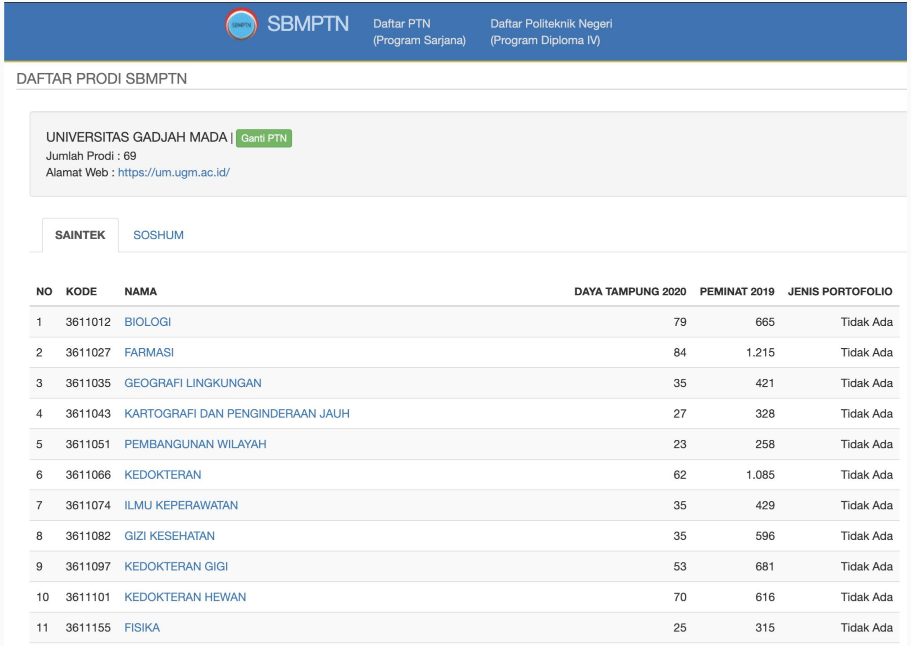
stringr

https://cran.r-project.org/web/packages/rvest/vignettes/selectorgadget.html

Data Scraping Using R (3)

We will try to scrap data on this page:

https://sidata-ptn.ltmpt.ac.id/ptn_sb.php?ptn=361



SBMPTN
Daftar PTN (Program Sarjana)
Daftar Politeknik Negeri (Program Diploma IV)

DAFTAR PRODI SBMPTN

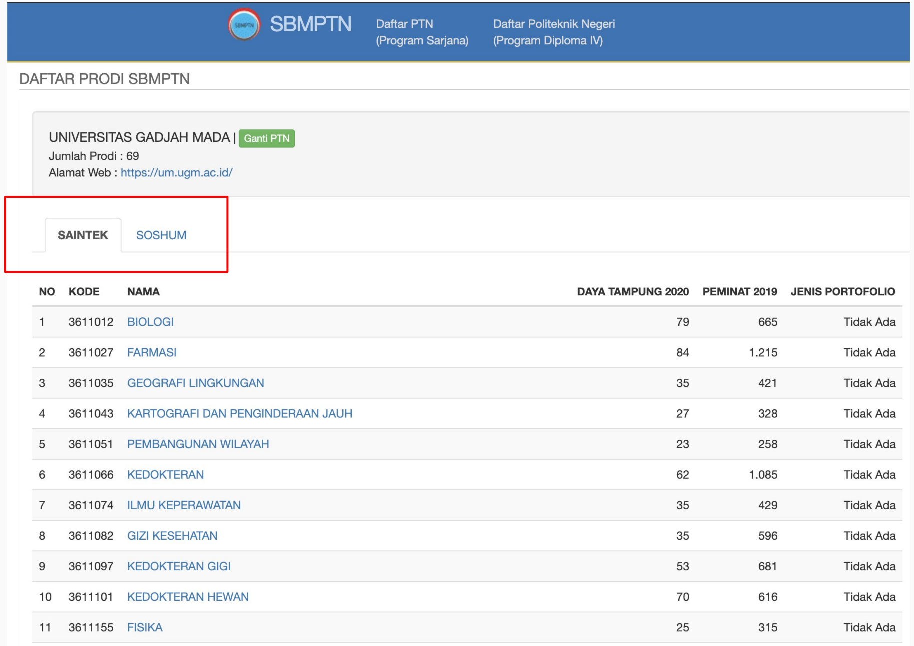
UNIVERSITAS GADJAH MADA | Ganti PTN
Jumlah Prodi : 69
Alamat Web : https://um.ugm.ac.id/

SAINTEK SOSHUM

NO	KODE	NAMA	DAYA TAMPUNG 2020	PEMINAT 2019	JENIS PORTOFOLIO
1	3611012	BIOLOGI	79	665	Tidak Ada
2	3611027	FARMASI	84	1.215	Tidak Ada
3	3611035	GEOGRAFI LINGKUNGAN	35	421	Tidak Ada
4	3611043	KARTOGRAFI DAN PENGINDERAAN JAUH	27	328	Tidak Ada
5	3611051	PEMBANGUNAN WILAYAH	23	258	Tidak Ada
6	3611066	KEDOKTERAN	62	1.085	Tidak Ada
7	3611074	ILMU KEPERAWATAN	35	429	Tidak Ada
8	3611082	GIZI KESEHATAN	35	596	Tidak Ada
9	3611097	KEDOKTERAN GIGI	53	681	Tidak Ada
10	3611101	KEDOKTERAN HEWAN	70	616	Tidak Ada
11	3611155	FISIKA	25	315	Tidak Ada

Data Scraping Using R (4)

```
url <- 'https://sidata-ptn.ltmpt.ac.id/ptn_sb.php?ptn=361'
webpage <- read_html(url)
```


SBMPTN
Daftar PTN (Program Sarjana)
Daftar Politeknik Negeri (Program Diploma IV)
Ganti PTN
SAINTEK
SOSHUM

DAFTAR PRODI SBMPTN

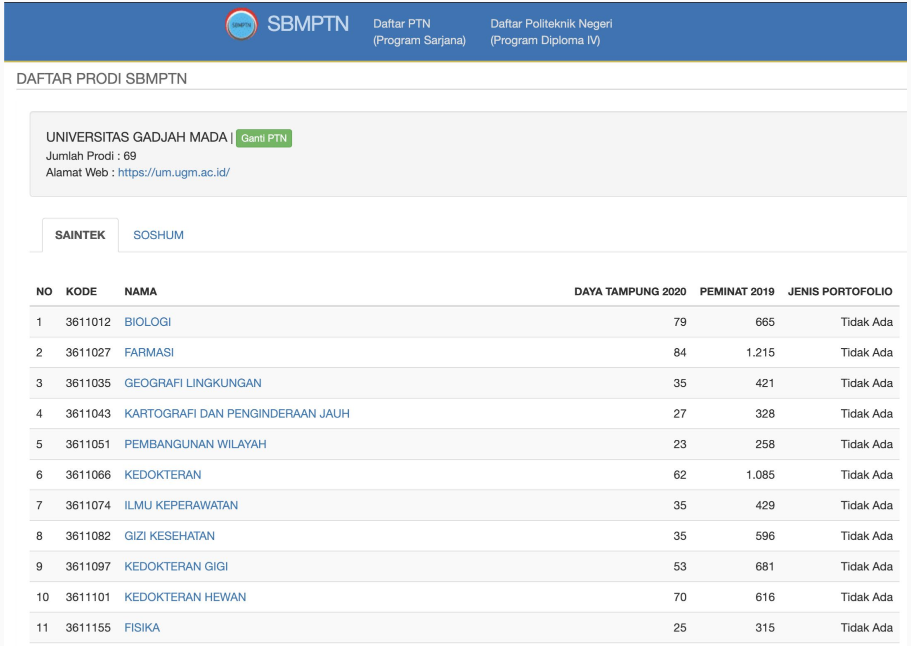
UNIVERSITAS GADJAH MADA |
Jumlah Prodi : 69
Alamat Web : https://um.ugm.ac.id/

NO	KODE	NAMA	DAYA TAMPUNG 2020	PEMINAT 2019	JENIS PORTOFOLIO
1	3611012	BIOLOGI	79	665	Tidak Ada
2	3611027	FARMASI	84	1.215	Tidak Ada
3	3611035	GEOGRAFI LINGKUNGAN	35	421	Tidak Ada
4	3611043	KARTOGRAFI DAN PENGINDERAAN JAUH	27	328	Tidak Ada
5	3611051	PEMBANGUNAN WILAYAH	23	258	Tidak Ada
6	3611066	KEDOKTERAN	62	1.085	Tidak Ada
7	3611074	ILMU KEPERAWATAN	35	429	Tidak Ada
8	3611082	GIZI KESEHATAN	35	596	Tidak Ada
9	3611097	KEDOKTERAN GIGI	53	681	Tidak Ada
10	3611101	KEDOKTERAN HEWAN	70	616	Tidak Ada
11	3611155	FISIKA	25	315	Tidak Ada

Data Scraping Using R (5)

Details we need from this page are:

1. SAINTEK part
2. NO
3. KODE
4. NAMA
5. DAYA TAMPUNG 2020
6. PEMINAT 2019



SBMPTN
Daftar PTN (Program Sarjana)
Daftar Politeknik Negeri (Program Diploma IV)

DAFTAR PRODI SBMPTN

UNIVERSITAS GADJAH MADA | Ganti PTN

Jumlah Prodi : 69

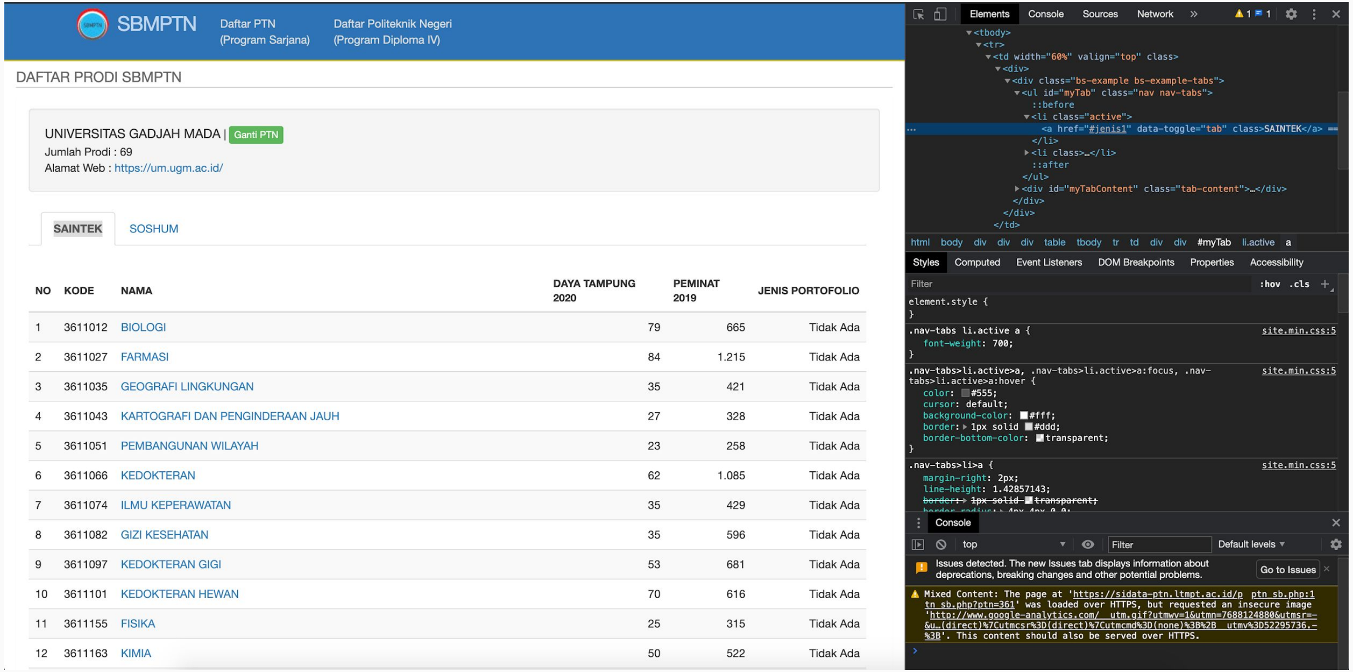
Alamat Web : https://um.ugm.ac.id/

SAINTEK SOSHUM

NO	KODE	NAMA	DAYA TAMPUNG 2020	PEMINAT 2019	JENIS PORTOFOLIO
1	3611012	BIOLOGI	79	665	Tidak Ada
2	3611027	FARMASI	84	1.215	Tidak Ada
3	3611035	GEOGRAFI LINGKUNGAN	35	421	Tidak Ada
4	3611043	KARTOGRAFI DAN PENGINDERAAN JAUH	27	328	Tidak Ada
5	3611051	PEMBANGUNAN WILAYAH	23	258	Tidak Ada
6	3611066	KEDOKTERAN	62	1.085	Tidak Ada
7	3611074	ILMU KEPERAWATAN	35	429	Tidak Ada
8	3611082	GIZI KESEHATAN	35	596	Tidak Ada
9	3611097	KEDOKTERAN GIGI	53	681	Tidak Ada
10	3611101	KEDOKTERAN HEWAN	70	616	Tidak Ada
11	3611155	FISIKA	25	315	Tidak Ada

Data Scraping Using R (6)

We are using Google Chrome web browser in this example.
Inspect Element the web page.



DAFTAR PRODI SBMPTN

UNIVERSITAS GADJAH MADA | Ganti PTN
Jumlah Prodi : 69
Alamat Web : https://um.ugm.ac.id/

SAINTEK SOSHUM

NO	KODE	NAMA	DAYA TAMPUNG 2020	PEMINAT 2019	JENIS PORTOFOLIO
1	3611012	BIOLOGI	79	665	Tidak Ada
2	3611027	FARMASI	84	1.215	Tidak Ada
3	3611035	GEOGRAFI LINGKUNGAN	35	421	Tidak Ada
4	3611043	KARTOGRAFI DAN PENGINDERAAN JAUH	27	328	Tidak Ada
5	3611051	PEMBANGUNAN WILAYAH	23	258	Tidak Ada
6	3611066	KEDOKTERAN	62	1.085	Tidak Ada
7	3611074	ILMU KEPERAWATAN	35	429	Tidak Ada
8	3611082	GIZI KESEHATAN	35	596	Tidak Ada
9	3611097	KEDOKTERAN GIGI	53	681	Tidak Ada
10	3611101	KEDOKTERAN HEWAN	70	616	Tidak Ada
11	3611155	FISIKA	25	315	Tidak Ada
12	3611163	KIMIA	50	522	Tidak Ada

```
▼<tbody>
  ▼<tr>
    ▼<td width="60%" valign="top" class>
      ▼<div>
        ▼<div class="bs-example bs-example-tabs">
          ▼<ul id="myTab" class="nav nav-tabs">
              ::before
            ▼<li class="active">
...             <a href="#jenis1" data-toggle="tab" class>SAINTEK</a> ==
              </li>
            ▶<li class>…</li>
              ::after
            </ul>
          ▶<div id="myTabContent" class="tab-content">…</div>
          </div>
        </div>
      </td>
```

html body div div div table tbody tr td div div #myTab li.active a

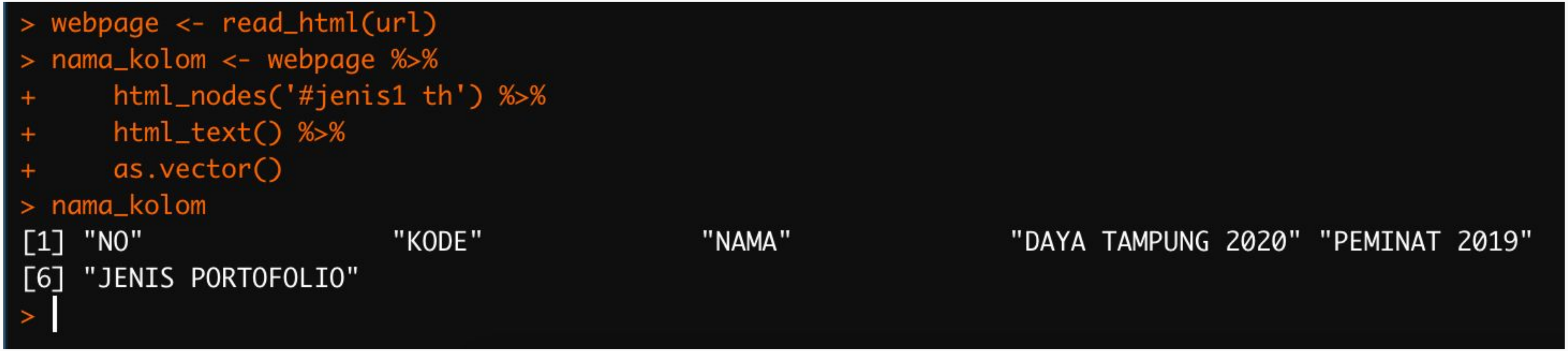
```
element.style {
}
.nav-tabs li.active a {                                      site.min.css:5
   font-weight: 700;
}
.nav-tabs>li.active>a, .nav-tabs>li.active>a:focus, .nav-    site.min.css:5
tabs>li.active>a:hover {
   color: #555;
   cursor: default;
   background-color: #fff;
   border: ▶ 1px solid #ddd;
   border-bottom-color: transparent;
}
.nav-tabs>li>a {                                             site.min.css:5
   margin-right: 2px;
   line-height: 1.42857143;
   ~~border: ▶ 1px solid transparent;~~
```

Issues detected. The new Issues tab displays information about deprecations, breaking changes and other potential problems.

```
⚠ Mixed Content: The page at 'https://sidata-ptn.ltmpt.ac.id/p  ptn_sb.php:1
  tn_sb.php?ptn=361' was loaded over HTTPS, but requested an insecure image
  'http://www.google-analytics.com/__utm.gif?utmwv=1&utmn=7688124880&utmsr=-
  &u…(direct)%7Cutmcsr%3D(direct)%7Cutmcmd%3D(none)%3B%2B__utmv%3D52295736.-
  %3B'. This content should also be served over HTTPS.
>
```

Data Scraping Using R (7)

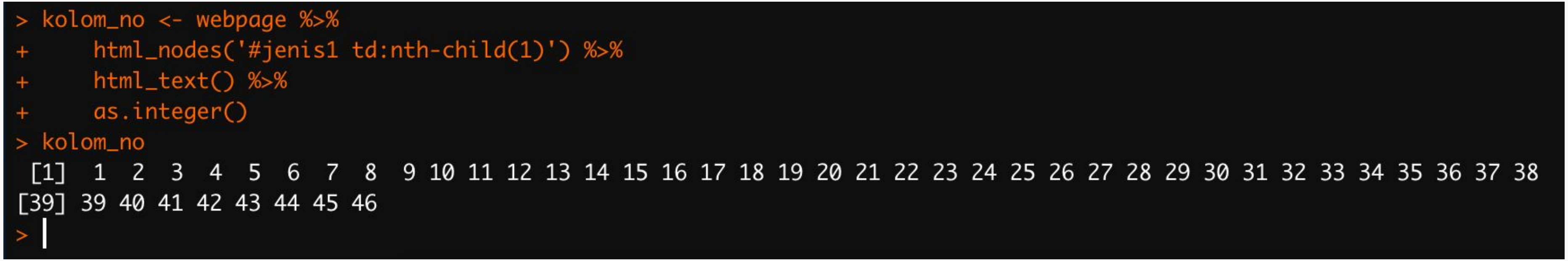
```
nama_kolom <- webpage %>%
  html_nodes('#jenis1 th') %>%
  html_text() %>%
  as.vector()
```



```
> webpage <- read_html(url)
> nama_kolom <- webpage %>%
+     html_nodes('#jenis1 th') %>%
+     html_text() %>%
+     as.vector()
> nama_kolom
[1] "NO"                "KODE"              "NAMA"              "DAYA TAMPUNG 2020" "PEMINAT 2019"     
[6] "JENIS PORTOFOLIO"
> |
```

Data Scraping Using R (8)

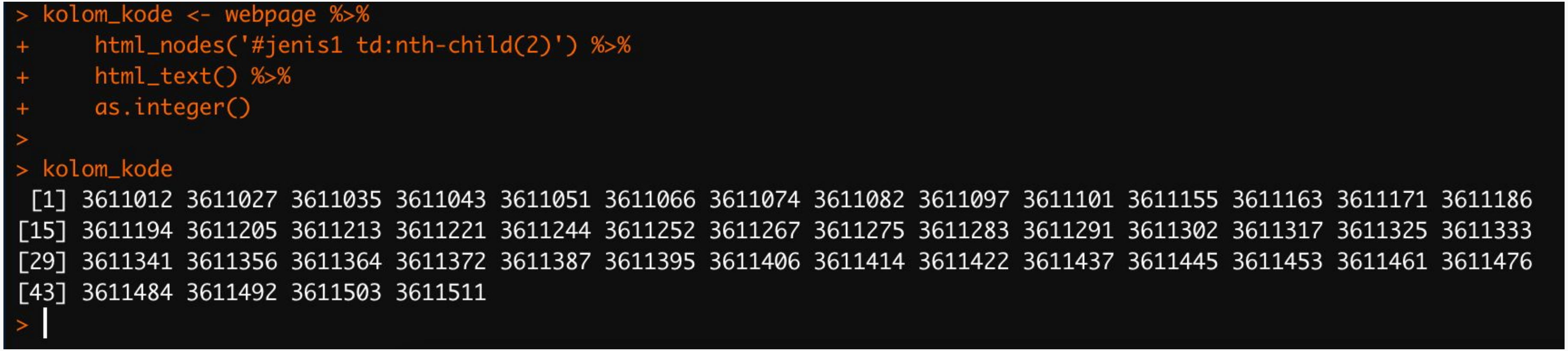
```
kolom_no <- webpage %>%
  html_nodes('#jenis1 td:nth-child(1)') %>%
  html_text() %>%
  as.integer()
```



```
> kolom_no <- webpage %>%
+     html_nodes('#jenis1 td:nth-child(1)') %>%
+     html_text() %>%
+     as.integer()
> kolom_no
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
[39] 39 40 41 42 43 44 45 46
> 
```

Data Scraping Using R (9)

```
kolom_kode <- webpage %>%
  html_nodes('#jenis1 td:nth-child(2)') %>%
  html_text() %>%
  as.integer()
```

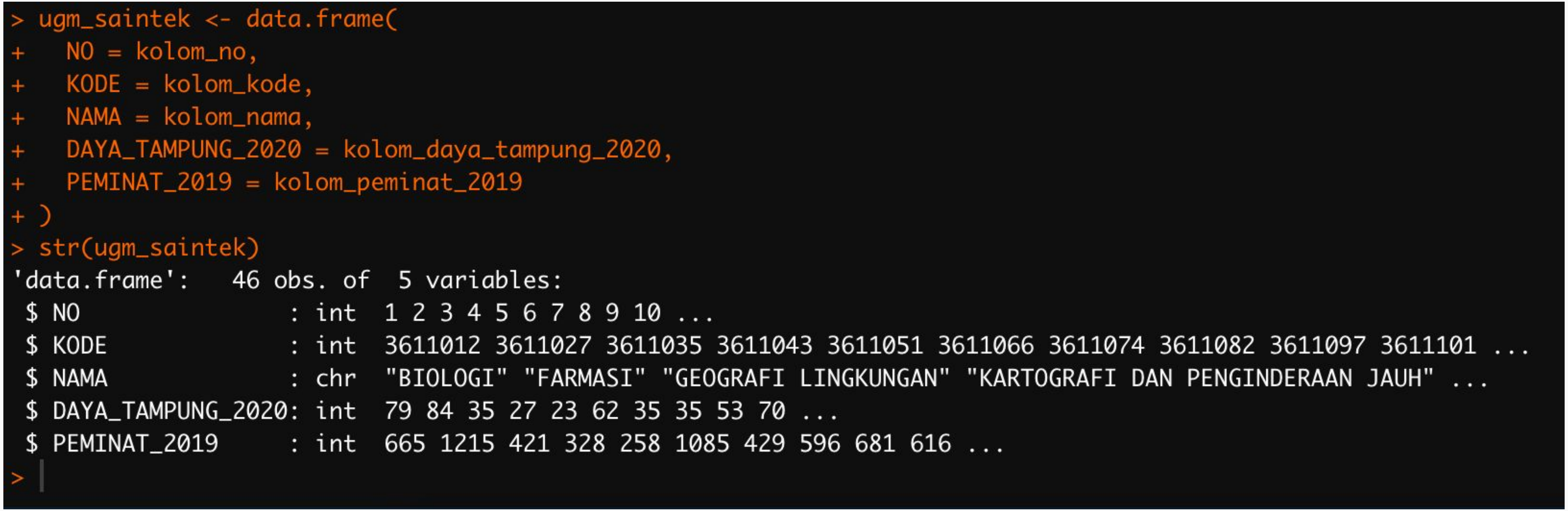


```
> kolom_kode <- webpage %>%
+     html_nodes('#jenis1 td:nth-child(2)') %>%
+     html_text() %>%
+     as.integer()
>
> kolom_kode
 [1] 3611012 3611027 3611035 3611043 3611051 3611066 3611074 3611082 3611097 3611101 3611155 3611163 3611171 3611186
[15] 3611194 3611205 3611213 3611221 3611244 3611252 3611267 3611275 3611283 3611291 3611302 3611317 3611325 3611333
[29] 3611341 3611356 3611364 3611372 3611387 3611395 3611406 3611414 3611422 3611437 3611445 3611453 3611461 3611476
[43] 3611484 3611492 3611503 3611511
> |
```

Continue by yourself for the rest!

Data Scraping Using R (10)

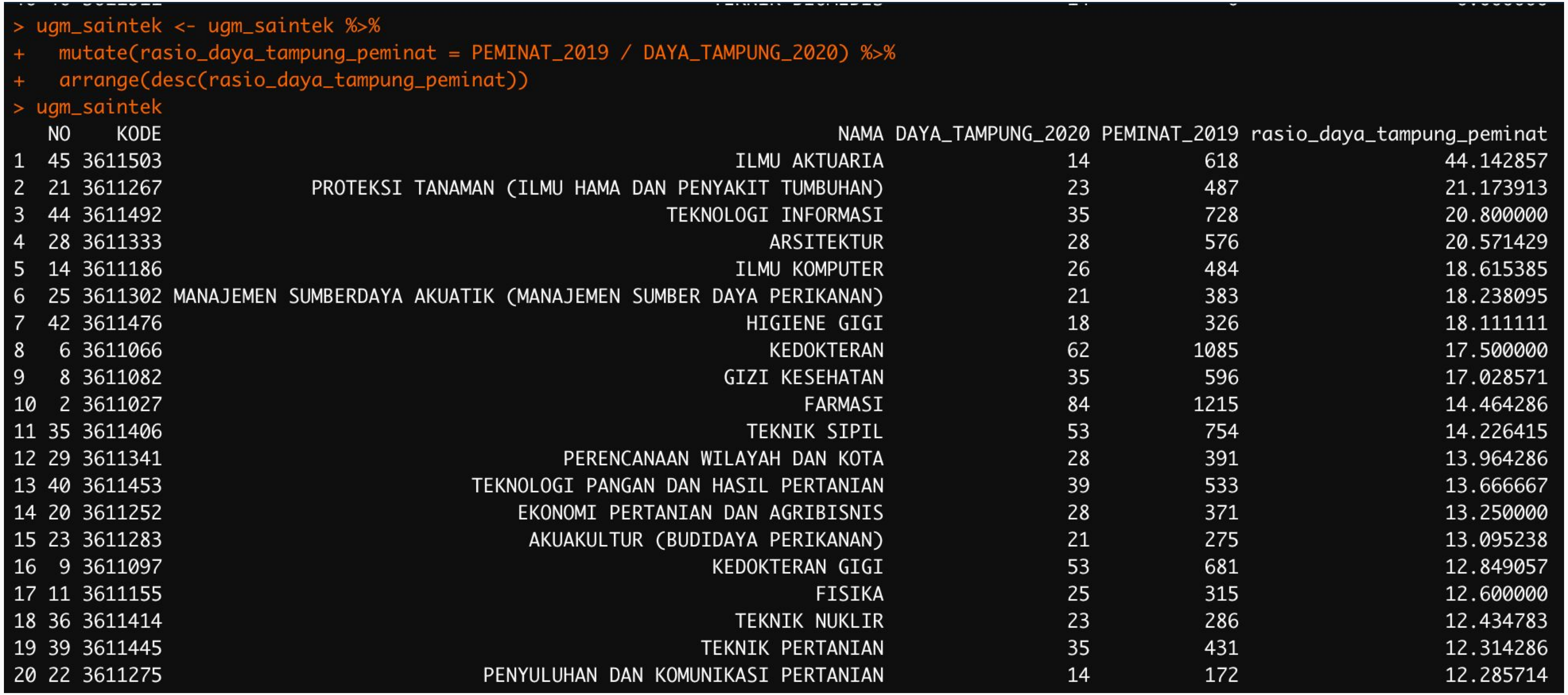
Merge all of it into one data frame



```
> ugm_saintek <- data.frame(
+   NO = kolom_no,
+   KODE = kolom_kode,
+   NAMA = kolom_nama,
+   DAYA_TAMPUNG_2020 = kolom_daya_tampung_2020,
+   PEMINAT_2019 = kolom_peminat_2019
+ )
> str(ugm_saintek)
'data.frame':	46 obs. of  5 variables:
 $ NO               : int  1 2 3 4 5 6 7 8 9 10 ...
 $ KODE             : int  3611012 3611027 3611035 3611043 3611051 3611066 3611074 3611082 3611097 3611101 ...
 $ NAMA             : chr  "BIOLOGI" "FARMASI" "GEOGRAFI LINGKUNGAN" "KARTOGRAFI DAN PENGINDERAAN JAUH" ...
 $ DAYA_TAMPUNG_2020: int  79 84 35 27 23 62 35 35 53 70 ...
 $ PEMINAT_2019     : int  665 1215 421 328 258 1085 429 596 681 616 ...
> 
```

Data Scraping Using R (11)

Simple aggregation using ugm_saintek



```
> ugm_saintek <- ugm_saintek %>%
+   mutate(rasio_daya_tampung_peminat = PEMINAT_2019 / DAYA_TAMPUNG_2020) %>%
+   arrange(desc(rasio_daya_tampung_peminat))
> ugm_saintek
   NO    KODE                                                    NAMA DAYA_TAMPUNG_2020 PEMINAT_2019 rasio_daya_tampung_peminat
1  45 3611503                                           ILMU AKTUARIA                14          618                  44.142857
2  21 3611267             PROTEKSI TANAMAN (ILMU HAMA DAN PENYAKIT TUMBUHAN)                23          487                  21.173913
3  44 3611492                                     TEKNOLOGI INFORMASI                35          728                  20.800000
4  28 3611333                                              ARSITEKTUR                28          576                  20.571429
5  14 3611186                                           ILMU KOMPUTER                26          484                  18.615385
6  25 3611302 MANAJEMEN SUMBERDAYA AKUATIK (MANAJEMEN SUMBER DAYA PERIKANAN)                21          383                  18.238095
7  42 3611476                                            HIGIENE GIGI                18          326                  18.111111
8   6 3611066                                             KEDOKTERAN                62         1085                  17.500000
9   8 3611082                                          GIZI KESEHATAN                35          596                  17.028571
10  2 3611027                                               FARMASI                84         1215                  14.464286
11 35 3611406                                           TEKNIK SIPIL                53          754                  14.226415
12 29 3611341                              PERENCANAAN WILAYAH DAN KOTA                28          391                  13.964286
13 40 3611453                          TEKNOLOGI PANGAN DAN HASIL PERTANIAN                39          533                  13.666667
14 20 3611252                             EKONOMI PERTANIAN DAN AGRIBISNIS                28          371                  13.250000
15 23 3611283                              AKUAKULTUR (BUDIDAYA PERIKANAN)                21          275                  13.095238
16  9 3611097                                         KEDOKTERAN GIGI                53          681                  12.849057
17 11 3611155                                                 FISIKA                25          315                  12.600000
18 36 3611414                                          TEKNIK NUKLIR                23          286                  12.434783
19 39 3611445                                       TEKNIK PERTANIAN                35          431                  12.314286
20 22 3611275                        PENYULUHAN DAN KOMUNIKASI PERTANIAN                14          172                  12.285714
```

Intro to Data Scraping Using R (12)

Check other packages!

RCurl

RCrawler