

# Wrangling Basics

Emily Malcolm-White

## Installing and Using Packages

Sometimes everything we need (data, functions, etc) are not available in base R. In R, expert users will package up useful things like data and functions into packages that be download and used.

First, you need to download the package from the right hand menu -> You only need to do this once.

In each new .qmd document, you need to call any packages you want to use but adding the code `library(packagename)` inside an R chunk.

### The tidyverse package

In this class we will use the `tidyverse` package a lot.

```
library(tidyverse) #<1>
```

① Loads the `tidyverse` package

There are actually many commonly used packages wrapped up inside one `tidyverse` package.

Today we are specifically going to be talking about the package `dplyr` which is useful to manipulating data sets.



Figure 1: Credit: <https://uopsych-r-bootcamp-2020.netlify.app/>



## can\_lang dataset

In this class, we are going to be working with a dataset relating to the languages spoken at home by Canadian residents. Many Indigenous peoples exist in Canada with their own languages and cultures. Sadly, colonization has led to the loss of many of these languages. This data is a subset of data collected during the 2016 census.

## Importing Data

What is a .csv file?

- It's plain text file that stores data
- Each value is separated by a comma (,) - hence the name "comma separated values"
- It's readable with tools like Excel, Google Sheets, R, and more.

**How do we import it into R?** Use `read.csv()`! Note that your data file (.csv) needs to be saved in the same folder as your notes template document (.qmd).

```
can_lang <- read.csv("data/can_lang.csv") #<1>
```

- ① Takes the `can_lang.csv` file (located in the same folder as your `.qmd` file), reads it into R, and saves it as the dataset `can_lang`

Alternatively, you can download it directly from the internet. Github user `ttimbers` hosts this file to share with the public at the link: [https://raw.githubusercontent.com/ttimbers/canlang/master/inst/extdata/can\\_lang.csv](https://raw.githubusercontent.com/ttimbers/canlang/master/inst/extdata/can_lang.csv)

```
can_lang <- read.csv("https://raw.githubusercontent.com/ttimbers/canlang/master/inst/extda
```

- ① Takes the dataset located at the given url, reads it into R, and saves it as the dataset `can_lang`

Let's take a look at this data for a minute to see what information has been recorded. In the environment in the top left, if you click on the word `can_lang` (not the blue play button, the word itself) it will open the object so you can see what is saved inside. Alternatively you can use the `head()` function to display just the first few rows of the dataset.

```
head(can_lang)
```

		category	language
1		Aboriginal languages	Aboriginal languages, n.o.s.
2	Non-Official & Non-Aboriginal languages		Afrikaans
3	Non-Official & Non-Aboriginal languages	Afro-Asiatic languages, n.i.e.	
4	Non-Official & Non-Aboriginal languages		Akan (Twi)
5	Non-Official & Non-Aboriginal languages		Albanian
6		Aboriginal languages	Algonquian languages, n.i.e.

  

	mother_tongue	most_at_home	most_at_work	lang_known
1	590	235	30	665
2	10260	4785	85	23415
3	1150	445	10	2775
4	13460	5985	25	22150
5	26895	13135	345	31930
6	45	10	0	120

## `filter()`

We can use the `filter` function to extract *rows* from the data that have a particular characteristic.

For example, we may be interested in only looking at only the languages in this dataset that are Aboriginal languages.



Figure 2: Artwork by @allisonhorst

Start with the `can_lang` dataset, the pipe `%>%` means apply the action on the following line to the previous line.

```
can_lang %>%                                     #<1>
  filter(category == "Aboriginal languages")      #<2>
```

- ① begin with the `can_lang` dataset
- ② only include the rows where the category variable is “Aboriginal languages”

	category	language	mother_tongue	most_at_home
1	Aboriginal languages	Aboriginal languages, n.o.s.	590	235
2	Aboriginal languages	Algonquian languages, n.i.e.	45	10
3	Aboriginal languages	Algonquin	1260	370
4	Aboriginal languages	Athabaskan languages, n.i.e.	50	10
5	Aboriginal languages	Atikamekw	6150	5465
6	Aboriginal languages	Babine (Wetsuwet'en)	110	20
7	Aboriginal languages	Beaver	190	50
8	Aboriginal languages	Blackfoot	2815	1110
9	Aboriginal languages	Carrier	1025	250
10	Aboriginal languages	Cayuga	45	10
11	Aboriginal languages	Chilcotin	655	255
12	Aboriginal languages	Comox	85	0
13	Aboriginal languages	Cree, n.o.s.	64050	37950
14	Aboriginal languages	Dakota	1210	255
15	Aboriginal languages	Dene	10700	7710
16	Aboriginal languages	Dogrib (Tlicho)	1650	1020
17	Aboriginal languages	Gitxsan (Gitksan)	880	315
18	Aboriginal languages	Gwich'in	255	50

19	Aboriginal languages	Haida	80	10
20	Aboriginal languages	Haisla	90	20
21	Aboriginal languages	Halkomelem	480	50
22	Aboriginal languages	Heiltsuk	100	5
23	Aboriginal languages	Inuinnaqtun (Inuvialuktun)	1020	165
24	Aboriginal languages	Inuit languages, n.i.e.	310	90
25	Aboriginal languages	Inuktitut	35210	29230
26	Aboriginal languages	Iroquoian languages, n.i.e.	35	5
27	Aboriginal languages	Kaska (Nahani)	180	20
28	Aboriginal languages	Kutenai	110	10
29	Aboriginal languages	Kwakiutl (Kwak'wala)	325	25
30	Aboriginal languages	Lillooet	315	25
31	Aboriginal languages	Malecite	300	55
32	Aboriginal languages	Mi'kmaq	6690	3565
33	Aboriginal languages	Michif	465	80
34	Aboriginal languages	Mohawk	985	255
35	Aboriginal languages	Montagnais (Innu)	10235	8585
36	Aboriginal languages	Moose Cree	105	10
37	Aboriginal languages	Naskapi	1205	1195
38	Aboriginal languages	Nisga'a	400	75
39	Aboriginal languages	North Slavey (Hare)	765	340
40	Aboriginal languages	Northern East Cree	315	110
41	Aboriginal languages	Northern Tutchone	220	30
42	Aboriginal languages	Nuu-chah-nulth (Nootka)	280	30
43	Aboriginal languages	Oji-Cree	12855	7905
44	Aboriginal languages	Ojibway	17885	6175
45	Aboriginal languages	Okanagan	275	80
46	Aboriginal languages	Oneida	60	15
47	Aboriginal languages	Ottawa (Odawa)	150	75
48	Aboriginal languages	Plains Cree	3065	1345
49	Aboriginal languages	Salish languages, n.i.e.	260	25
50	Aboriginal languages	Sarsi (Sarcee)	80	10
51	Aboriginal languages	Sekani	85	15
52	Aboriginal languages	Shuswap (Secwepemctsin)	445	50
53	Aboriginal languages	Siouan languages, n.i.e.	55	20
54	Aboriginal languages	Slavey, n.o.s.	280	105
55	Aboriginal languages	South Slavey	945	370
56	Aboriginal languages	Southern East Cree	45	15
57	Aboriginal languages	Southern Tutchone	70	5
58	Aboriginal languages	Squamish	40	5
59	Aboriginal languages	Stoney	3025	1950
60	Aboriginal languages	Straits	80	25
61	Aboriginal languages	Swampy Cree	1440	330

62	Aboriginal languages	Tahltan	95	5
63	Aboriginal languages	Thompson (Ntlakapamux)	335	20
64	Aboriginal languages	Tlingit	95	0
65	Aboriginal languages	Tsimshian	200	30
66	Aboriginal languages	Wakashan languages, n.i.e.	10	0
67	Aboriginal languages	Woods Cree	1840	800
	most_at_work	lang_known		
1	30	665		
2	0	120		
3	40	2480		
4	0	85		
5	1100	6645		
6	10	210		
7	0	340		
8	85	5645		
9	15	2100		
10	10	125		
11	15	1150		
12	0	185		
13	7800	86115		
14	20	1760		
15	770	13060		
16	165	2375		
17	10	1305		
18	10	360		
19	0	465		
20	0	175		
21	20	1060		
22	10	125		
23	30	1975		
24	15	470		
25	8795	40620		
26	0	115		
27	10	365		
28	0	170		
29	15	605		
30	15	790		
31	10	760		
32	915	9025		
33	10	1210		
34	30	2415		
35	2055	11445		
36	0	195		

37	370	1465
38	10	1055
39	95	1005
40	35	550
41	0	280
42	10	560
43	1080	15605
44	765	28580
45	20	820
46	0	185
47	0	205
48	95	5905
49	0	560
50	0	145
51	0	185
52	35	1305
53	0	140
54	10	675
55	35	1365
56	0	40
57	0	145
58	10	285
59	240	3675
60	15	365
61	10	2350
62	0	265
63	0	450
64	10	260
65	10	410
66	0	25
67	75	2665

Some notes:

- the aboriginal languages is text/categorical and so quotation marks are needed.
- R doesn't care about whether they are double quotation marks ("") or single (''). They work the same.
- If we don't assign it to an object, then it just prints out for us to see!

Oftentimes, we want to take our subset and give it a new name. This takes our subset and assigns it to a new dataset called `aboriginal_lang`.

```
aboriginal_lang <- can_lang %>% #<1>
  filter(category == "Aboriginal languages")
```

- ① The code `aboriginal_lang <-` takes the given data (the Aboriginal languages in the `can_lang` dataset) and saves it as a new object called `aboriginal_lang`.

Notes:

- Notice if you assign it to an object that it doesn't print out the contents.
- You'll see the new object in your environment on the top right —>

It can also be used with numeric criteria.

Suppose we want a list of all the languages in Canada that are spoken by less than 100 people as their mother tongue.

```
rare_lang <- can_lang %>% #<1>
  filter(mother_tongue < 100) #<2>
                                #<3>
```

- ① begin with the `can_lang` dataset
- ② only include the rows where the number of people who speak the language as their mother tongue is more than 100 people
- ③ data saved to the object `rare_lang`

The logical operators are given below:

Operator	Description
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to
!x	Not x
x   y	x OR y
x & y	x AND y

## select()

`select` is used to extract only certain *columns*. For example, perhaps we only want to print out a list names of the aboriginal languages (language column).



```

aboriginal_lang %>% #<1>
  select(language) #<2>

```

- ① Begin with the `aboriginal_lang` dataset
- ② only include the language column

```

      language
1  Aboriginal languages, n.o.s.
2  Algonquian languages, n.i.e.
3      Algonquin
4  Athabaskan languages, n.i.e.
5      Atikamekw
6      Babine (Wetsuwet'en)
7      Beaver
8      Blackfoot
9      Carrier
10     Cayuga
11     Chilcotin
12     Comox
13     Cree, n.o.s.
14     Dakota
15     Dene
16     Dogrib (Tlicho)
17     Gitxsan (Gitksan)
18     Gwich'in
19     Haida
20     Haisla
21     Halkomelem
22     Heiltsuk
23  Inuinnaqtun (Inuvialuktun)
24     Inuit languages, n.i.e.
25     Inuktitut
26  Iroquoian languages, n.i.e.
27     Kaska (Nahani)
28     Kutenai
29     Kwakiutl (Kwak'wala)
30     Lillooet
31     Malecite
32     Mi'kmaq
33     Michif
34     Mohawk
35     Montagnais (Innu)

```

```

36             Moose Cree
37             Naskapi
38             Nisga'a
39     North Slavey (Hare)
40     Northern East Cree
41     Northern Tutchone
42     Nuu-chah-nulth (Nootka)
43             Oji-Cree
44             Ojibway
45             Okanagan
46             Oneida
47             Ottawa (Odawa)
48             Plains Cree
49     Salish languages, n.i.e.
50             Sarsi (Sarcee)
51             Sekani
52     Shuswap (Secwepemctsin)
53     Siouan languages, n.i.e.
54             Slavey, n.o.s.
55             South Slavey
56     Southern East Cree
57     Southern Tutchone
58             Squamish
59             Stoney
60             Straits
61             Swampy Cree
62             Tahltan
63     Thompson (Ntlakapamux)
64             Tlingit
65             Tsimshian
66     Wakashan languages, n.i.e.
67             Woods Cree

```

We can combine criteria together as well in one command with multiple pipes:

```

can_lang %>%
  filter(category == "Aboriginal languages") %>%
  select(language)

```

```

              language
1 Aboriginal languages, n.o.s.
2 Algonquian languages, n.i.e.

```

3	Algonquin
4	Athabaskan languages, n.i.e.
5	Atikamekw
6	Babine (Wetsuwet'en)
7	Beaver
8	Blackfoot
9	Carrier
10	Cayuga
11	Chilcotin
12	Comox
13	Cree, n.o.s.
14	Dakota
15	Dene
16	Dogrib (Tlicho)
17	Gitxsan (Gitksan)
18	Gwich'in
19	Haida
20	Haisla
21	Halkomelem
22	Heiltsuk
23	Inuinnaqtun (Inuvialuktun)
24	Inuit languages, n.i.e.
25	Inuktitut
26	Iroquoian languages, n.i.e.
27	Kaska (Nahani)
28	Kutenai
29	Kwakiutl (Kwak'wala)
30	Lillooet
31	Malecite
32	Mi'kmaq
33	Michif
34	Mohawk
35	Montagnais (Innu)
36	Moose Cree
37	Naskapi
38	Nisga'a
39	North Slavey (Hare)
40	Northern East Cree
41	Northern Tutchone
42	Nuu-chah-nulth (Nootka)
43	Oji-Cree
44	Ojibway
45	Okanagan

```

46             Oneida
47         Ottawa (Odawa)
48         Plains Cree
49     Salish languages, n.i.e.
50         Sarsi (Sarcee)
51         Sekani
52     Shuswap (Secwepemctsin)
53     Siouan languages, n.i.e.
54         Slavey, n.o.s.
55         South Slavey
56         Southern East Cree
57         Southern Tutchone
58         Squamish
59         Stoney
60         Straits
61         Swampy Cree
62         Tahltan
63     Thompson (Ntlakapamux)
64         Tlingit
65         Tsimshian
66     Wakashan languages, n.i.e.
67         Woods Cree

```

## arrange()

The `arrange` function allows us to order the rows of the data frame by the values of a particular column.

For example, arrange all the aboriginal languages in canada by from most to least spoken as mother tongue.

```

aboriginal_lang %>%
  arrange(desc(mother_tongue))  #<1>

```

- ① arranges the languages from the language with the most to the least people who speak the language as their mother tongue

	category	language	mother_tongue	most_at_home
1	Aboriginal languages	Cree, n.o.s.	64050	37950
2	Aboriginal languages	Inuktitut	35210	29230
3	Aboriginal languages	Ojibway	17885	6175

4	Aboriginal languages	Oji-Cree	12855	7905
5	Aboriginal languages	Dene	10700	7710
6	Aboriginal languages	Montagnais (Innu)	10235	8585
7	Aboriginal languages	Mi'kmaq	6690	3565
8	Aboriginal languages	Atikamekw	6150	5465
9	Aboriginal languages	Plains Cree	3065	1345
10	Aboriginal languages	Stoney	3025	1950
11	Aboriginal languages	Blackfoot	2815	1110
12	Aboriginal languages	Woods Cree	1840	800
13	Aboriginal languages	Dogrib (Tlicho)	1650	1020
14	Aboriginal languages	Swampy Cree	1440	330
15	Aboriginal languages	Algonquin	1260	370
16	Aboriginal languages	Dakota	1210	255
17	Aboriginal languages	Naskapi	1205	1195
18	Aboriginal languages	Carrier	1025	250
19	Aboriginal languages	Inuinnaqtun (Inuvialuktun)	1020	165
20	Aboriginal languages	Mohawk	985	255
21	Aboriginal languages	South Slavey	945	370
22	Aboriginal languages	Gitksan (Gitksan)	880	315
23	Aboriginal languages	North Slavey (Hare)	765	340
24	Aboriginal languages	Chilcotin	655	255
25	Aboriginal languages	Aboriginal languages, n.o.s.	590	235
26	Aboriginal languages	Halkomelem	480	50
27	Aboriginal languages	Michif	465	80
28	Aboriginal languages	Shuswap (Secwepemctsin)	445	50
29	Aboriginal languages	Nisga'a	400	75
30	Aboriginal languages	Thompson (Ntlakapamux)	335	20
31	Aboriginal languages	Kwakiutl (Kwak'wala)	325	25
32	Aboriginal languages	Lillooet	315	25
33	Aboriginal languages	Northern East Cree	315	110
34	Aboriginal languages	Inuit languages, n.i.e.	310	90
35	Aboriginal languages	Malecite	300	55
36	Aboriginal languages	Nuu-chah-nulth (Nootka)	280	30
37	Aboriginal languages	Slavey, n.o.s.	280	105
38	Aboriginal languages	Okanagan	275	80
39	Aboriginal languages	Salish languages, n.i.e.	260	25
40	Aboriginal languages	Gwich'in	255	50
41	Aboriginal languages	Northern Tutchone	220	30
42	Aboriginal languages	Tsimshian	200	30
43	Aboriginal languages	Beaver	190	50
44	Aboriginal languages	Kaska (Nahani)	180	20
45	Aboriginal languages	Ottawa (Odawa)	150	75
46	Aboriginal languages	Babine (Wetsuwet'en)	110	20

47	Aboriginal languages	Kutenai	110	10
48	Aboriginal languages	Moose Cree	105	10
49	Aboriginal languages	Heiltsuk	100	5
50	Aboriginal languages	Tahltan	95	5
51	Aboriginal languages	Tlingit	95	0
52	Aboriginal languages	Haisla	90	20
53	Aboriginal languages	Comox	85	0
54	Aboriginal languages	Sekani	85	15
55	Aboriginal languages	Haida	80	10
56	Aboriginal languages	Sarsi (Sarcee)	80	10
57	Aboriginal languages	Straits	80	25
58	Aboriginal languages	Southern Tutchone	70	5
59	Aboriginal languages	Oneida	60	15
60	Aboriginal languages	Siouan languages, n.i.e.	55	20
61	Aboriginal languages	Athabaskan languages, n.i.e.	50	10
62	Aboriginal languages	Algonquian languages, n.i.e.	45	10
63	Aboriginal languages	Cayuga	45	10
64	Aboriginal languages	Southern East Cree	45	15
65	Aboriginal languages	Squamish	40	5
66	Aboriginal languages	Iroquoian languages, n.i.e.	35	5
67	Aboriginal languages	Wakashan languages, n.i.e.	10	0
most_at_work lang_known				
1	7800	86115		
2	8795	40620		
3	765	28580		
4	1080	15605		
5	770	13060		
6	2055	11445		
7	915	9025		
8	1100	6645		
9	95	5905		
10	240	3675		
11	85	5645		
12	75	2665		
13	165	2375		
14	10	2350		
15	40	2480		
16	20	1760		
17	370	1465		
18	15	2100		
19	30	1975		
20	30	2415		
21	35	1365		

22	10	1305
23	95	1005
24	15	1150
25	30	665
26	20	1060
27	10	1210
28	35	1305
29	10	1055
30	0	450
31	15	605
32	15	790
33	35	550
34	15	470
35	10	760
36	10	560
37	10	675
38	20	820
39	0	560
40	10	360
41	0	280
42	10	410
43	0	340
44	10	365
45	0	205
46	10	210
47	0	170
48	0	195
49	10	125
50	0	265
51	10	260
52	0	175
53	0	185
54	0	185
55	0	465
56	0	145
57	15	365
58	0	145
59	0	185
60	0	140
61	0	85
62	0	120
63	10	125
64	0	40

65	10	285
66	0	115
67	0	25

Note:

- use `arrange(variable)` to go from least to most
- use `arrange(desc(variable))` to go from most to least, `arrange(-variable)` also works

## slice()

The slice function will allow us to pick only a subset of the rows based on their numeric order (1st through last).

For example, if I want a list of the 10 most commonly spoken aboriginal languages.

```
aboriginal_lang %>%
  arrange(desc(mother_tongue)) %>%
  slice(1:10)    #<1>
```

- ① Only include the first 10 rows of the dataset

	category	language	mother_tongue	most_at_home
1	Aboriginal languages	Cree, n.o.s.	64050	37950
2	Aboriginal languages	Inuktitut	35210	29230
3	Aboriginal languages	Ojibway	17885	6175
4	Aboriginal languages	Oji-Cree	12855	7905
5	Aboriginal languages	Dene	10700	7710
6	Aboriginal languages	Montagnais (Innu)	10235	8585
7	Aboriginal languages	Mi'kmaq	6690	3565
8	Aboriginal languages	Atikamekw	6150	5465
9	Aboriginal languages	Plains Cree	3065	1345
10	Aboriginal languages	Stoney	3025	1950

  

	most_at_work	lang_known
1	7800	86115
2	8795	40620
3	765	28580
4	1080	15605
5	770	13060
6	2055	11445
7	915	9025



8	1100	6645
9	95	5905
10	240	3675

## mutate()

`mutate()` creates new columns that are functions of existing variables.



Figure 3: Artwork by @allisonhorst

For example, if I want to create a new column called `mother_tongue_K` which represents the number of people who speak the language their mother tongue in thousands. You may want to save this new dataset over top of the original dataset so you could use this new column in the future.

```
aboriginal_lang <- aboriginal_lang %>%
  mutate(mother_tongue_K = mother_tongue/1000) #<1>
```

- ① Creates a new column called `mother_tongue_K` calculated by taking the `mother_tongue` column and dividing it by 1000.

This can be useful for unit conversions. It also be useful for making new calculations based on existing data (for example, price and number of square feet could be used to calculate price per square foot).