

I am not saving  
it back to a df  
so I want to  
assign this <-

.---What all I learned to read the data---  
read\_excel - need to be careful in having sheet argument to capture sheets. I want  
to download before reading. download.file mode = "wb" as I am windows user  
read\_csv - what are the arguments I used n\_max, skip, etc..  
read\_csv2 - function what Europeans like  
read\_delim - I notice that it automatically detect the delimiter. Interesting !!

	country	1999	2000
1	Afghanistan	745	2666
2	Brazil	37737	80488
3	China	212258	213766

```
library(tidyverse)
table4a |>
# everything(), when to use ?
pivot_longer(-country, names_to = "year", values_to = "cases") |>

# Doing just for fun to capture the syntax
pivot_wider(names_from = year, values_from = cases) |>

# backticks ` be careful! Even for `Province/State`
pivot_longer(`1999`:`2000`, names_to = "year", values_to = "cases") |>
# syntax for case_when, what to do when there is na ?
# str_replace, case_when How imp is it to have TRUE ~ colname
mutate(country = str_replace(country, "Afghanistan", "Afg")) |>

# for fun
unite("test_combine", c(country, cases), sep = " ") |>

# for fun
separate(test_combine, c("country", "cases"), sep = " ") |>

# Should be careful after applying str operations
mutate(cases = as.numeric(cases)) |>
# I dont like country name to be country I want to rename it to my_country
rename(my_country = `country`) ## I am happy now!! .. There is janitor to clean for more columns at a time, link here
# str_detect, %in%, %within% for interval, ! to negate
filter(!country == "Brazil") |>
# -c(colname) to get everything else other than colname
select(country, cases) |>

# desc and without desc
arrange(desc(cases)) |>

# specify rows 1:3, pass as vector
slice(1) |>

# drop_na(colname), na.rm = TRUE as argument to function like sum, median etc..
drop_na() |>
# It is removing all columns that got NA in it. Maybe I don't want to do that way

# pull a column as vector
pull(country)
# Why don't I use select instead of pull. WAIT I remember I can't do that,
# as pull return vector back

# There are many more like this
# why not I try out everything with my own example and have a cheat sheet ready
# I can write more notes all over
```

as song.. as dot  
numeric..character..integer..double..integer.  
..logical..list..lalala Even there is a "is" song!

There is a base R way to filter mtcars[table4a\$country=="Brazil", ]

Operator	Example use	What it returns
[	mtcars[1:10, 2:4]	rows 1-10 for columns 2-4 of the data frame, as a data frame
[	mtcars[1:10, ]	rows 1-10 for all columns of the data frame, as a data frame
[	mtcars[1]	the first column of the data frame, as a data frame
[[	mtcars[[1]]	the first column of the data frame, as a vector
\$	mtcars\$ cyl	the column the corresponds to the name that follows the \$, as a vector

Note that \$ and [[ remove a level of structure from the data frame object (this happens with lists too).

adding as this is  
helpful!

'China'

more links and blog posts can be here!!