DR. ALVIN'S PUBLICATIONS

DATA WRANGLING WITH R

BY DR. ALVIN ANG

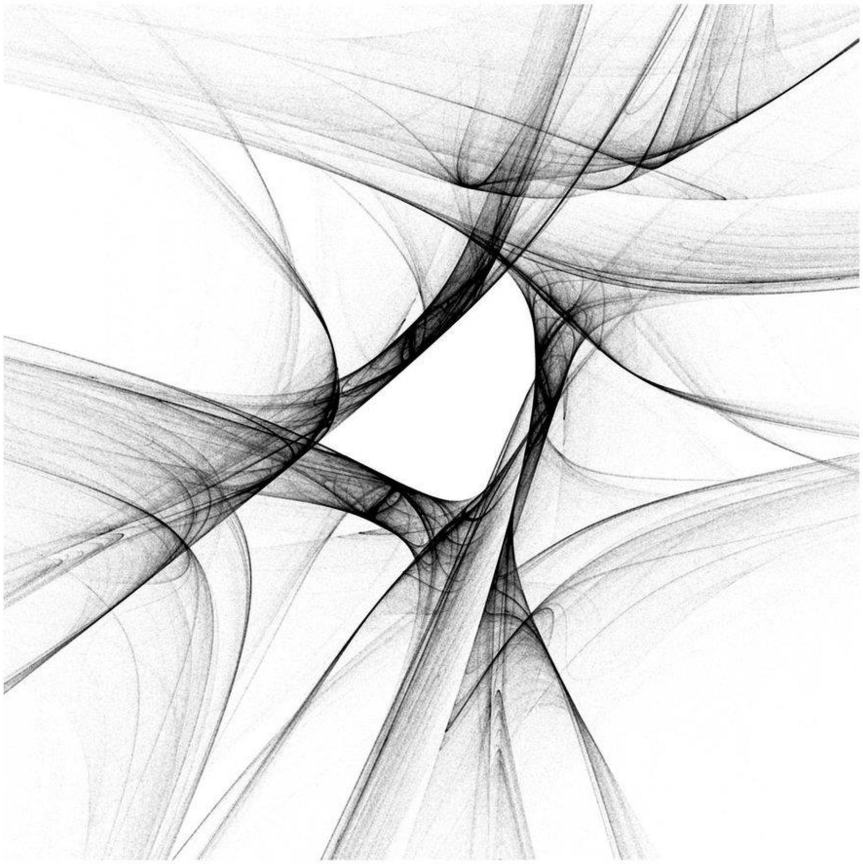
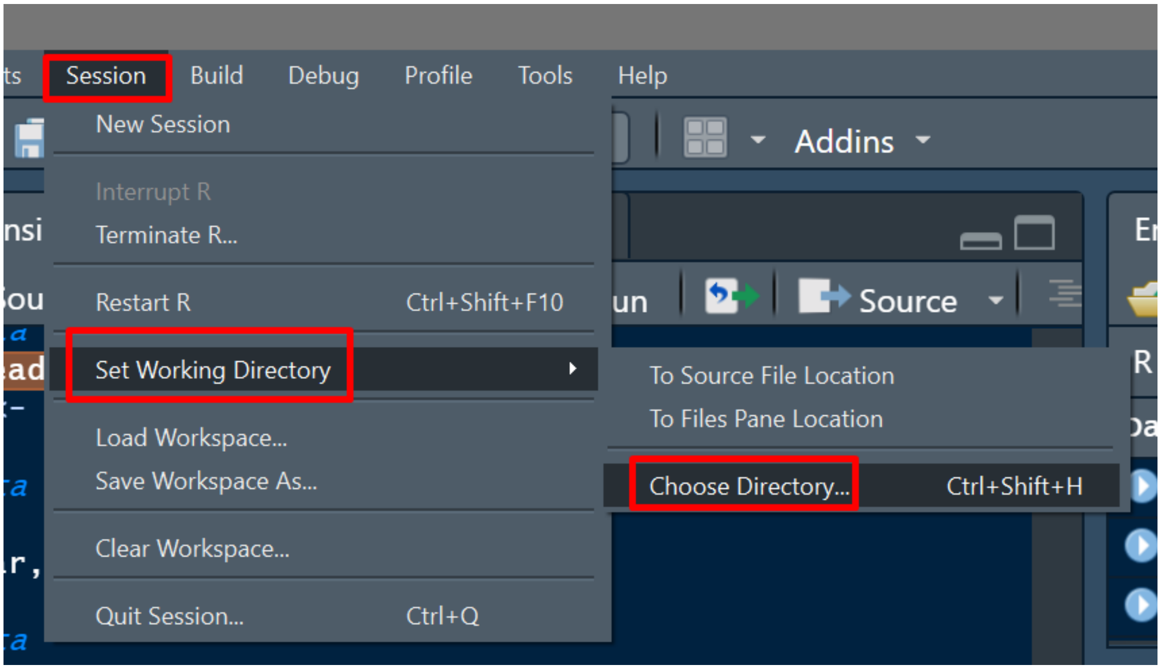


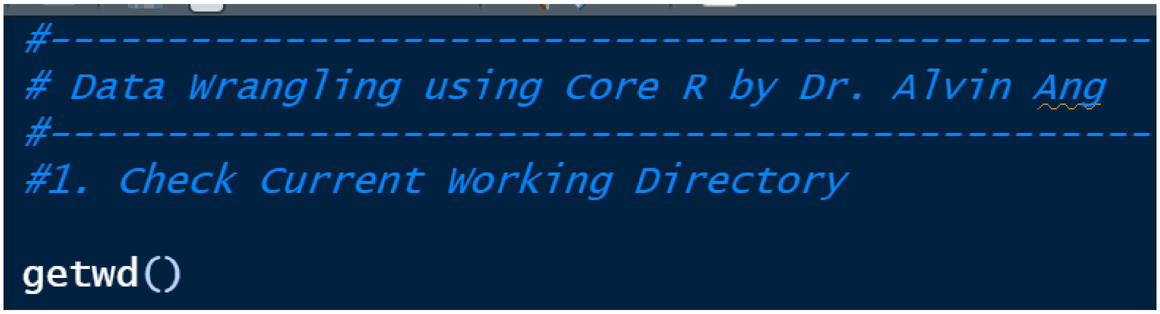
TABLE OF CONTENTS

I. DATA WRANGLING USING CORE R

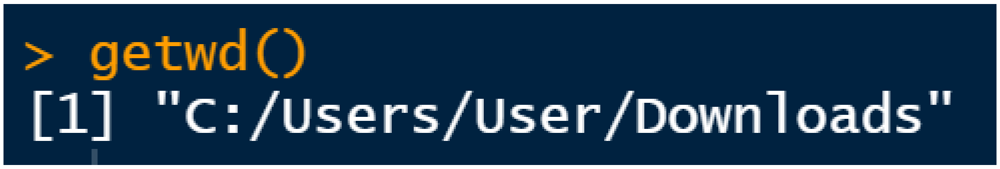
A. SET YOUR WORKING DIRECTORY TO YOUR DOWNLOADS FOLDER



1. CHECK YOUR CURRENT WORKING DIRECTORY



```
#-----------------------------------------------------
# Data Wrangling using Core R by Dr. Alvin Ang
#-----------------------------------------------------
#1. Check Current Working Directory

getwd()
```

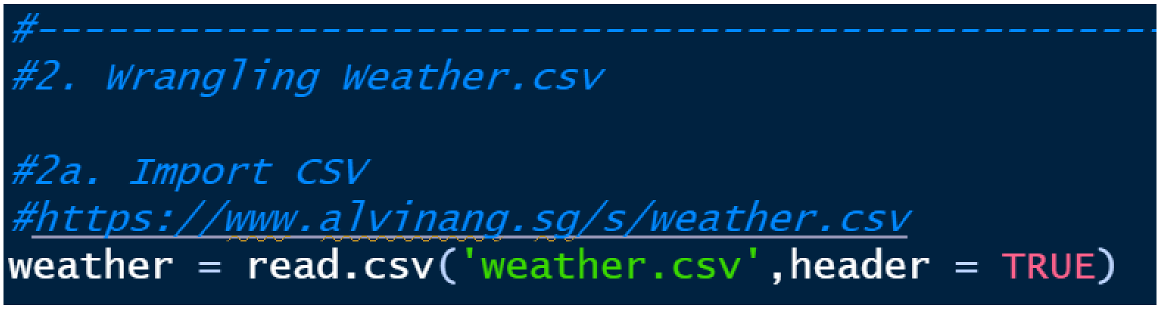


```
> getwd()
[1] "C:/Users/User/Downloads"
```

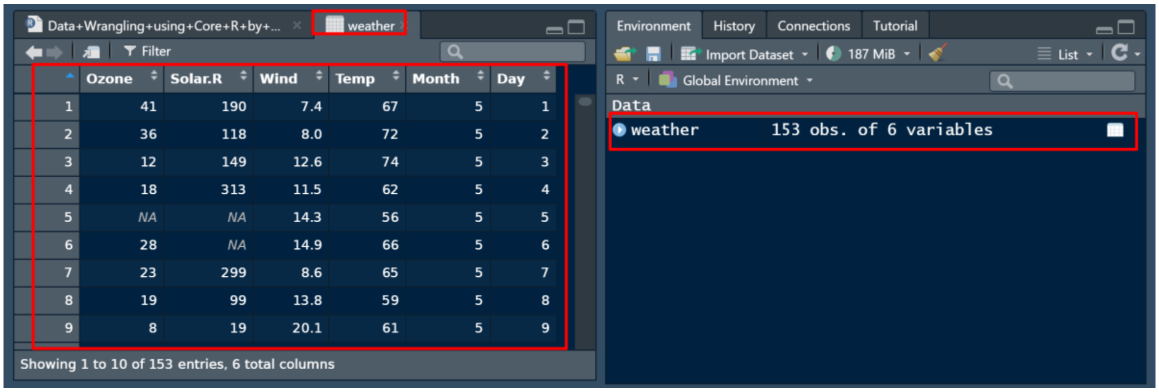
B. WRANGLING WEATHER.CSV

1. IMPORT CSV

https://www.alvinang.sg/s/weather.csv



```
#------------------------------------------------
#2. Wrangling Weather.csv

#2a. Import CSV
#https://www.alvinang.sg/s/weather.csv
weather = read.csv('weather.csv',header = TRUE)
```

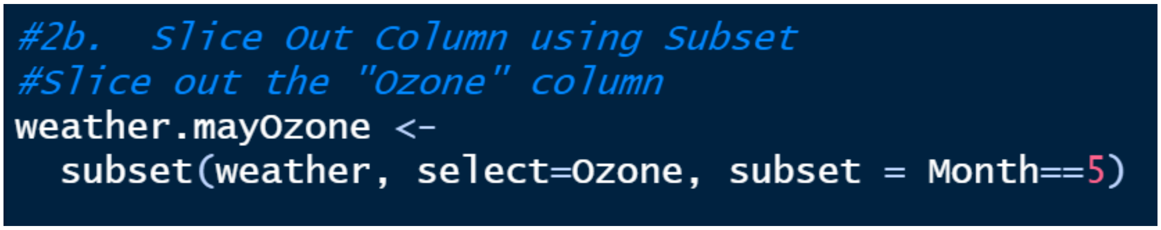


	Ozone	Solar.R	Wind	Temp	Month	Day
1	41	190	7.4	67	5	1
2	36	118	8.0	72	5	2
3	12	149	12.6	74	5	3
4	18	313	11.5	62	5	4
5	NA	NA	14.3	56	5	5
6	28	NA	14.9	66	5	6
7	23	299	8.6	65	5	7
8	19	99	13.8	59	5	8
9	8	19	20.1	61	5	9

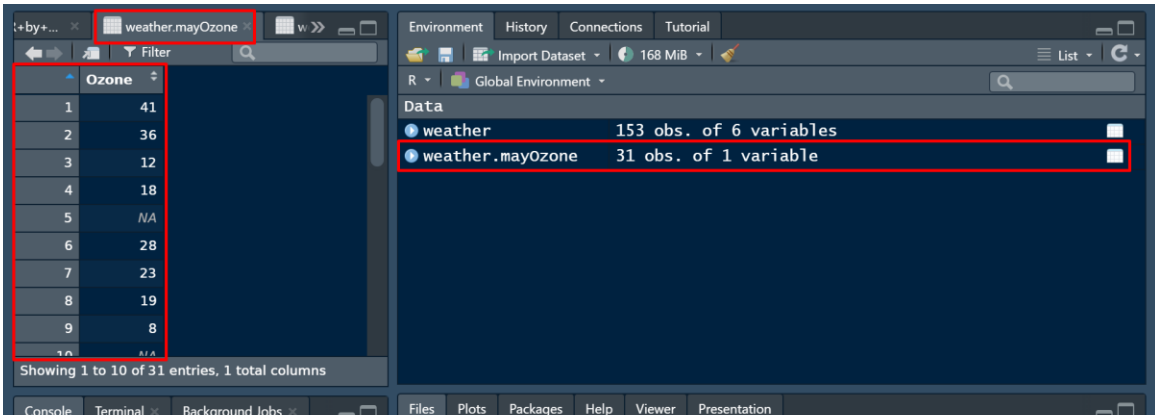
Showing 1 to 10 of 153 entries, 6 total columns

```
Data
weather          153 obs. of 6 variables
```

2. SLICE OUT COLUMN USING SUBSET



```
#2b.  Slice Out Column using Subset
#Slice out the "Ozone" column
weather.mayOzone <-
  subset(weather, select=Ozone, subset = Month==5)
```

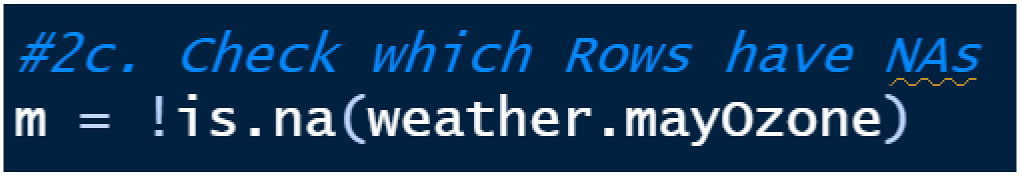


	Ozone
1	41
2	36
3	12
4	18
5	NA
6	28
7	23
8	19
9	8
10	NA

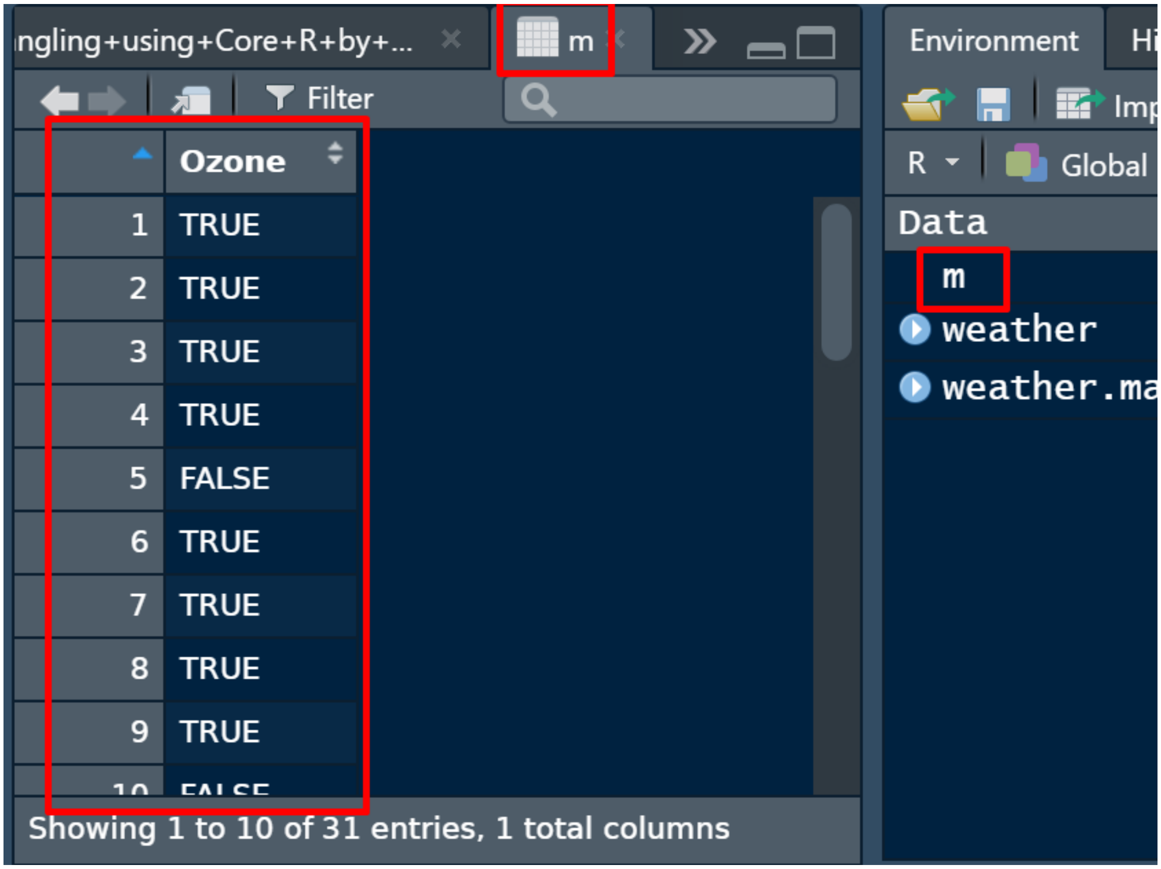
Showing 1 to 10 of 31 entries, 1 total columns

```
Data
weather              153 obs. of 6 variables
weather.mayOzone     31 obs. of 1 variable
```

3. CHECK WHICH ROWS HAVE NAS



```
#2c. Check which Rows have NAs
m = !is.na(weather.mayOzone)
```

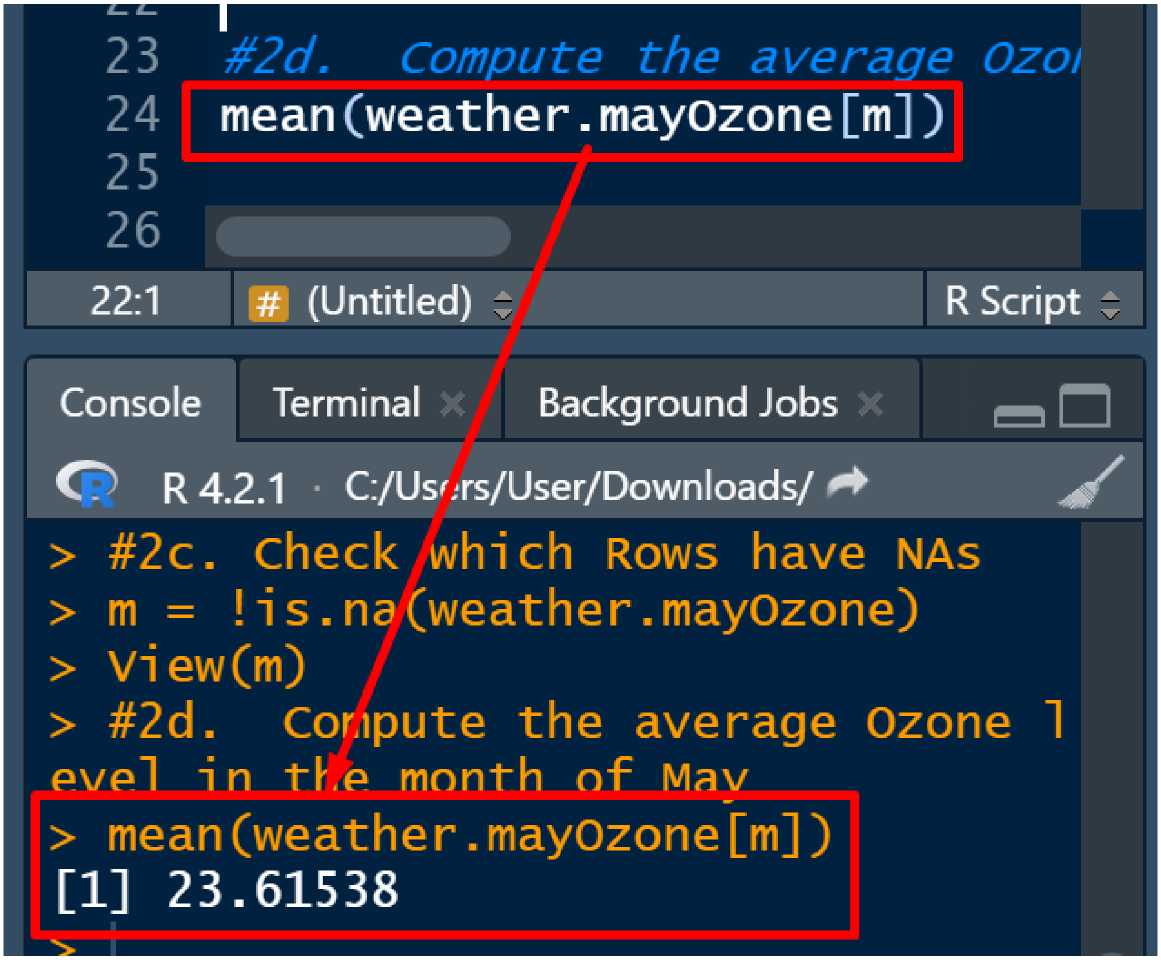


	Ozone
1	TRUE
2	TRUE
3	TRUE
4	TRUE
5	FALSE
6	TRUE
7	TRUE
8	TRUE
9	TRUE
10	FALSE

Showing 1 to 10 of 31 entries, 1 total columns

4. COMPUTE THE AVERAGE OZONE LEVEL IN THE MONTH OF MAY

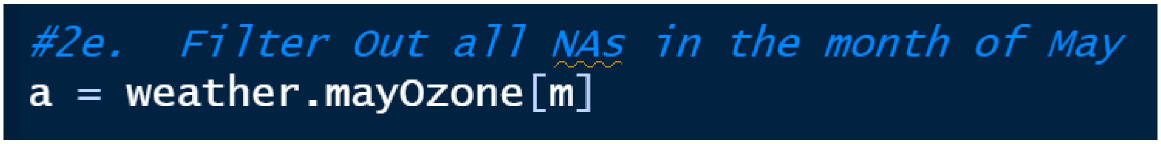
```
#2d.  Compute the average Ozone level in the month of May
mean(weather.mayOzone[m])
```



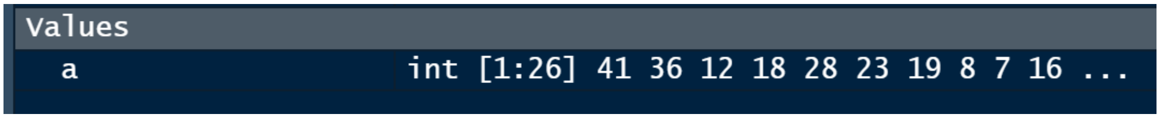
```
#2d.  Compute the average Ozo
mean(weather.mayOzone[m])
```

```
> #2c. Check which Rows have NAs
> m = !is.na(weather.mayOzone)
> View(m)
> #2d.  Compute the average Ozone l
evel in the month of May
> mean(weather.mayOzone[m])
[1] 23.61538
>
```

5. FILTER OUT ALL NAS IN THE MONTH OF MAY

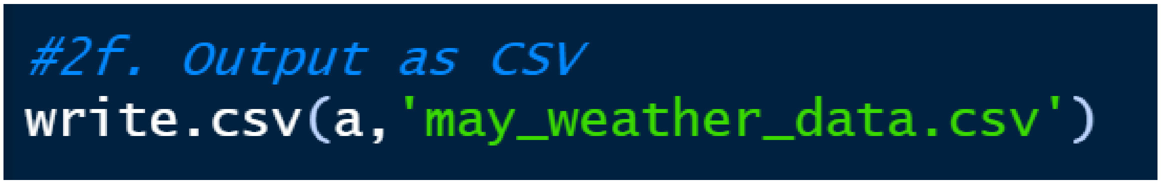


```
#2e.  Filter Out all NAs in the month of May
a = weather.mayOzone[m]
```

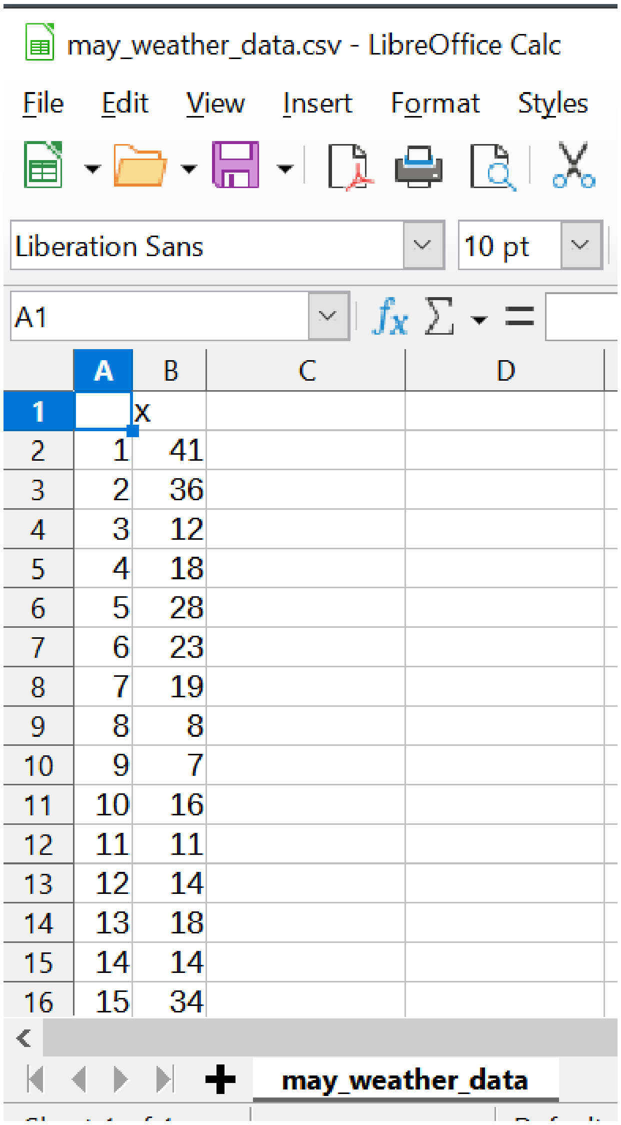


```
Values
  a                    int [1:26] 41 36 12 18 28 23 19 8 7 16 ...
```

6. OUTPUT AS CSV



```
#2f. Output as CSV
write.csv(a,'may_weather_data.csv')
```

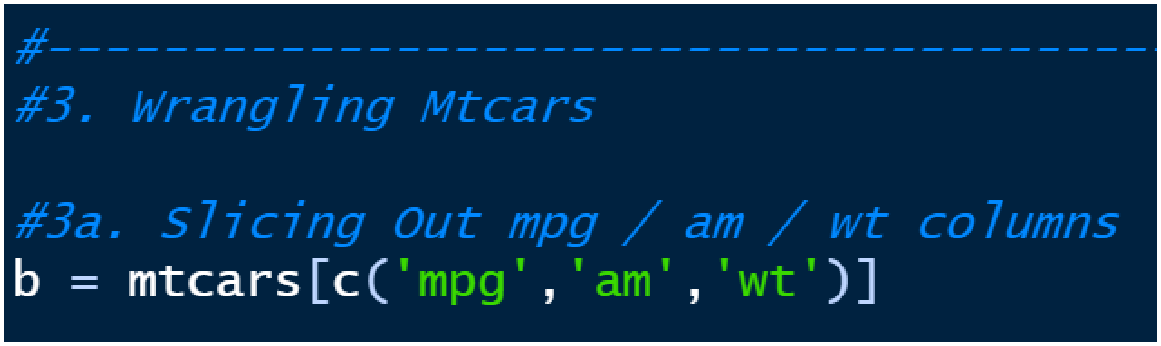


may_weather_data.csv - LibreOffice Calc
File Edit View Insert Format Styles
Liberation Sans
10 pt
A1
may_weather_data

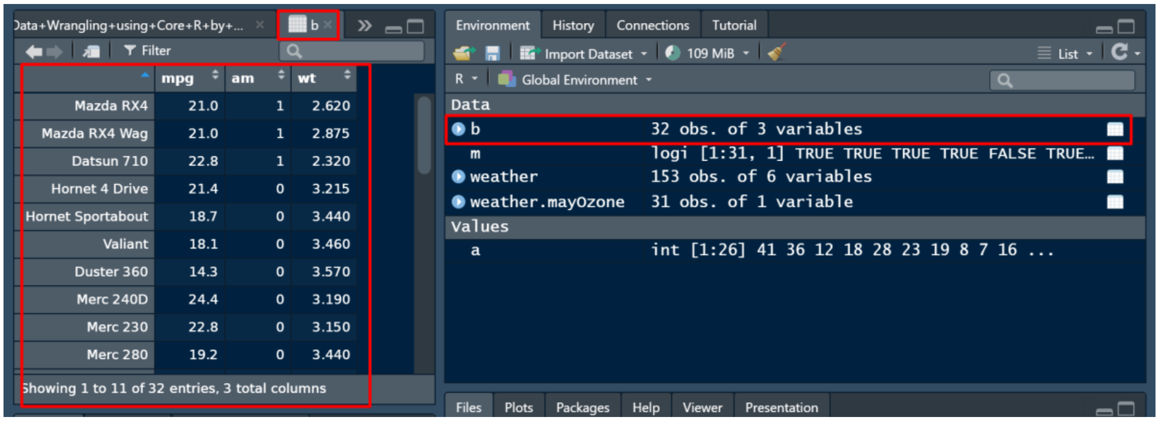
	x
1	41
2	36
3	12
4	18
5	28
6	23
7	19
8	8
9	7
10	16
11	11
12	14
13	18
14	14
15	34

C. WRANGLING MTCARS

1. SLICING OUT MPG / AM / WT COLUMNS



```
#------------------------------------------
#3. Wrangling Mtcars

#3a. Slicing Out mpg / am / wt columns
b = mtcars[c('mpg','am','wt')]
```

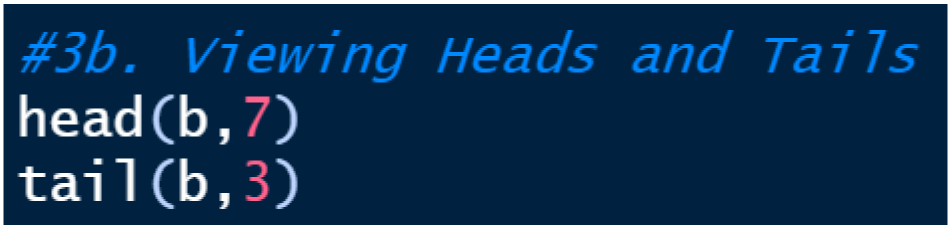


	mpg	am	wt
Mazda RX4	21.0	1	2.620
Mazda RX4 Wag	21.0	1	2.875
Datsun 710	22.8	1	2.320
Hornet 4 Drive	21.4	0	3.215
Hornet Sportabout	18.7	0	3.440
Valiant	18.1	0	3.460
Duster 360	14.3	0	3.570
Merc 240D	24.4	0	3.190
Merc 230	22.8	0	3.150
Merc 280	19.2	0	3.440

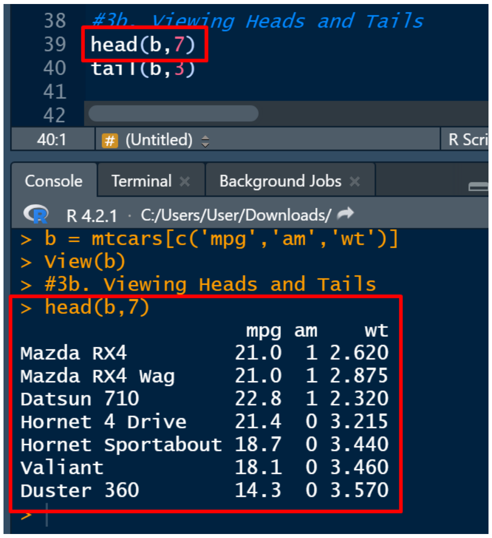
Showing 1 to 11 of 32 entries, 3 total columns

```
Data
 b                32 obs. of 3 variables
 m                logi [1:31, 1] TRUE TRUE TRUE TRUE FALSE TRUE…
 weather          153 obs. of 6 variables
 weather.mayOzone 31 obs. of 1 variable
Values
 a                int [1:26] 41 36 12 18 28 23 19 8 7 16 ...
```

2. PREVIEWING HEADS AND TAILS

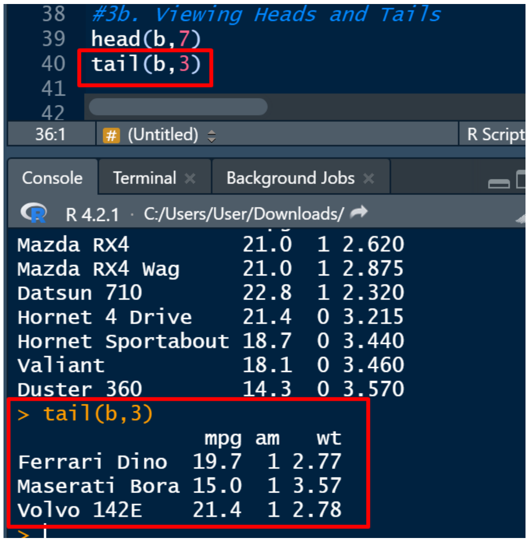


```
#3b. Viewing Heads and Tails
head(b,7)
tail(b,3)
```



```
#3b. Viewing Heads and Tails
head(b,7)
tail(b,3)
```

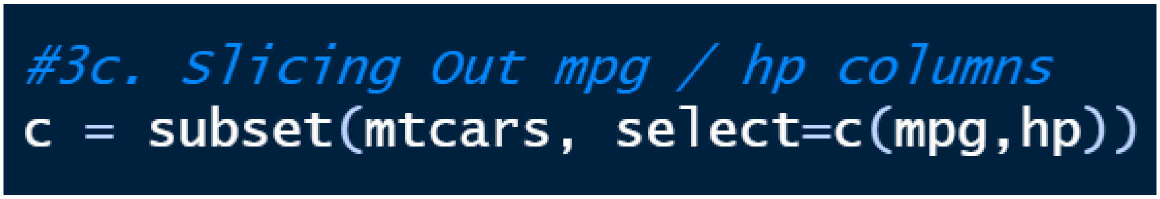
```
> b = mtcars[c('mpg','am','wt')]
> View(b)
> #3b. Viewing Heads and Tails
> head(b,7)
                   mpg am    wt
Mazda RX4         21.0  1 2.620
Mazda RX4 Wag     21.0  1 2.875
Datsun 710        22.8  1 2.320
Hornet 4 Drive    21.4  0 3.215
Hornet Sportabout 18.7  0 3.440
Valiant           18.1  0 3.460
Duster 360        14.3  0 3.570
>
```



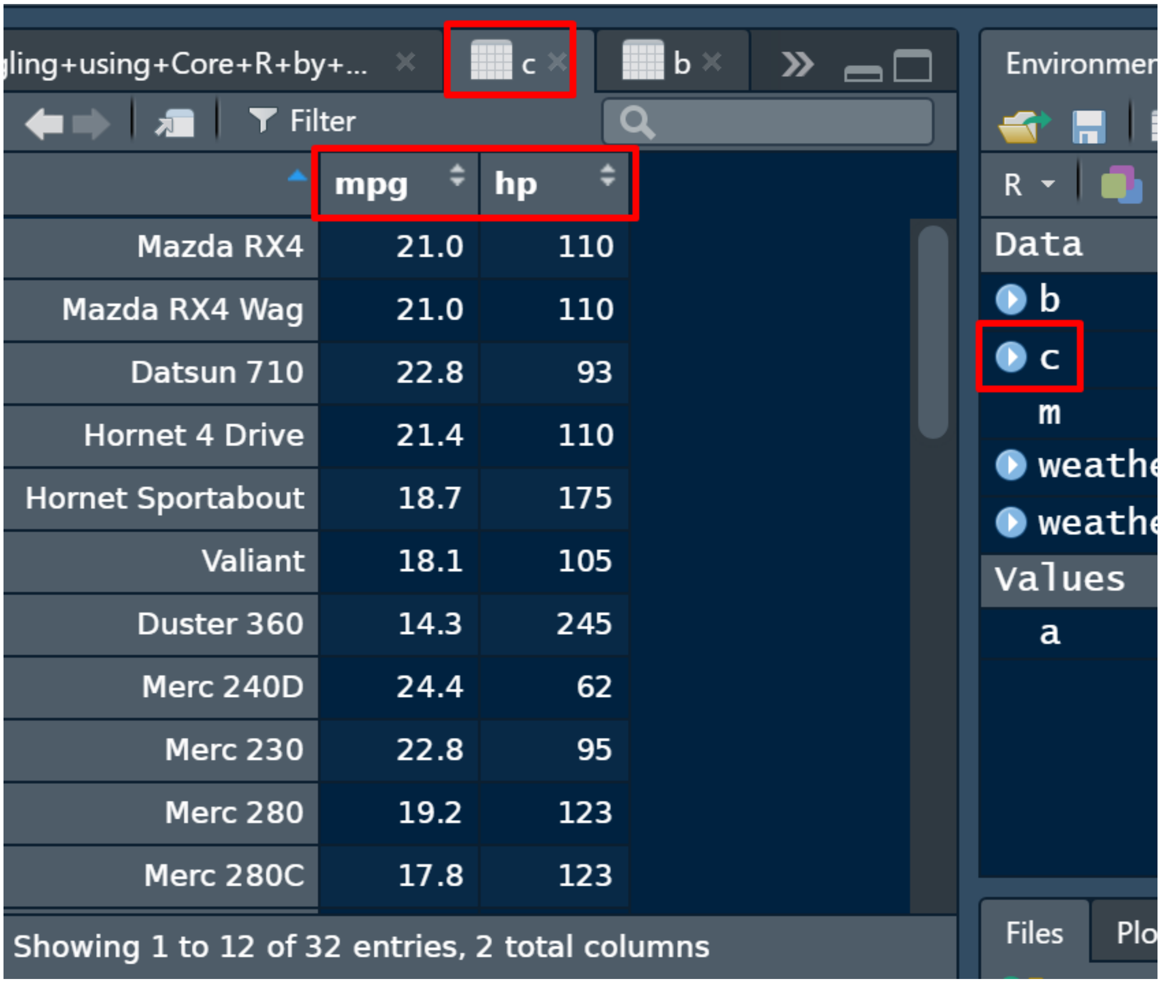
```
#3b. Viewing Heads and Tails
head(b,7)
tail(b,3)
```

```
Mazda RX4         21.0  1 2.620
Mazda RX4 Wag     21.0  1 2.875
Datsun 710        22.8  1 2.320
Hornet 4 Drive    21.4  0 3.215
Hornet Sportabout 18.7  0 3.440
Valiant           18.1  0 3.460
Duster 360        14.3  0 3.570
> tail(b,3)
              mpg am   wt
Ferrari Dino 19.7  1 2.77
Maserati Bora 15.0  1 3.57
Volvo 142E    21.4  1 2.78
>
```

3. SLICING OUT MPG / HP COLUMNS



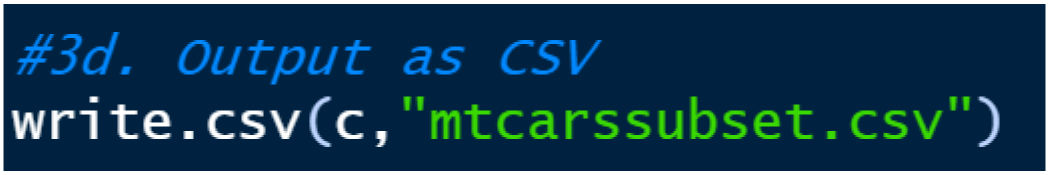
```
#3c. Slicing Out mpg / hp columns
c = subset(mtcars, select=c(mpg,hp))
```



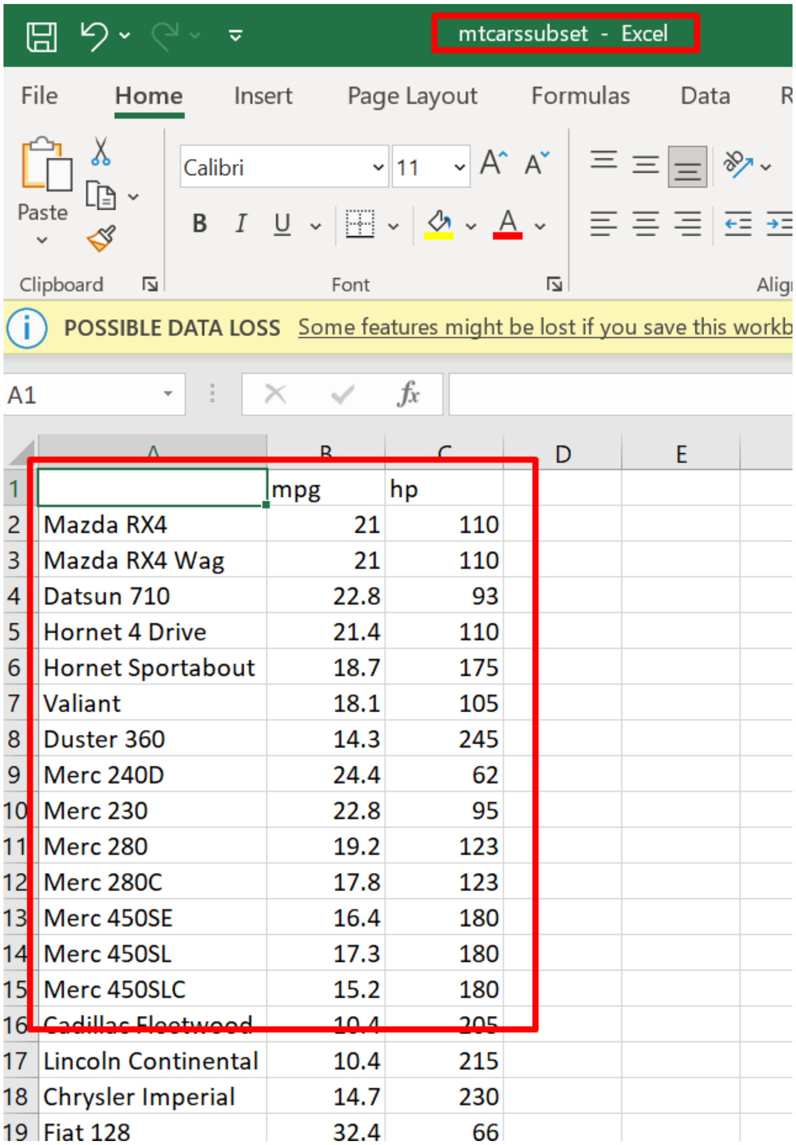
	mpg	hp
Mazda RX4	21.0	110
Mazda RX4 Wag	21.0	110
Datsun 710	22.8	93
Hornet 4 Drive	21.4	110
Hornet Sportabout	18.7	175
Valiant	18.1	105
Duster 360	14.3	245
Merc 240D	24.4	62
Merc 230	22.8	95
Merc 280	19.2	123
Merc 280C	17.8	123

Showing 1 to 12 of 32 entries, 2 total columns

4. OUTPUT AS CSV



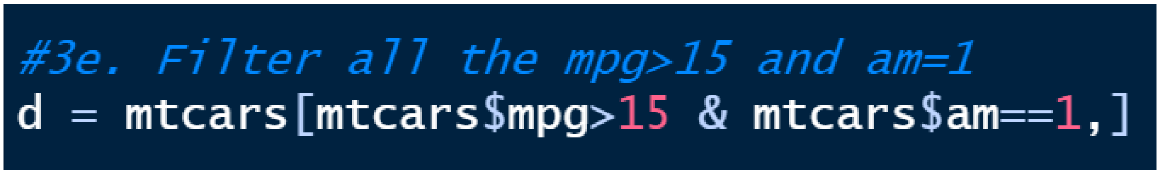
```
#3d. Output as CSV
write.csv(c,"mtcarssubset.csv")
```



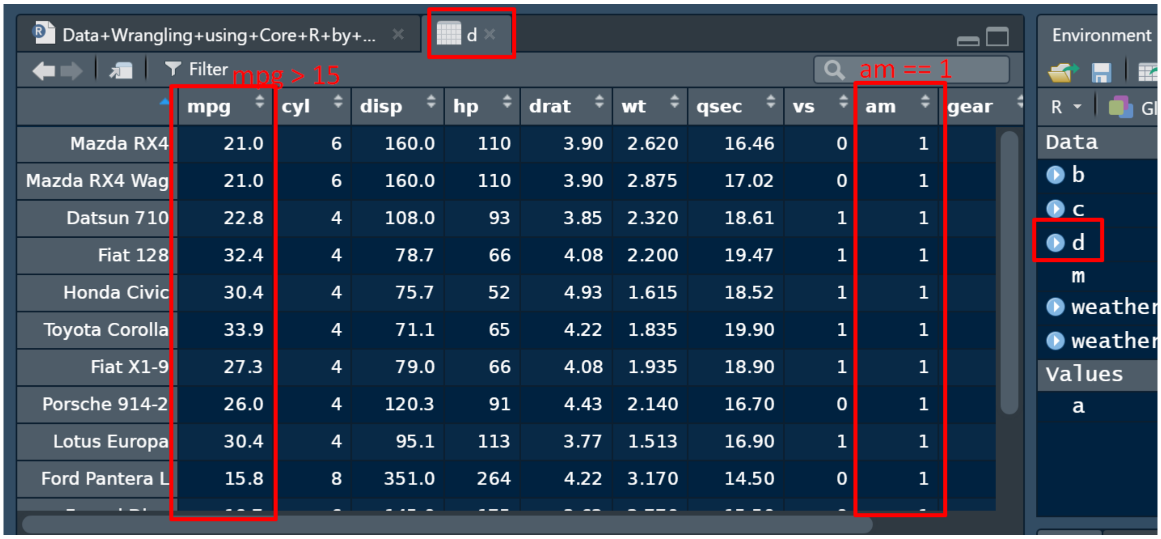
POSSIBLE DATA LOSS Some features might be lost if you save this workb

	A	B	C	D	E
1		mpg	hp		
2	Mazda RX4	21	110		
3	Mazda RX4 Wag	21	110		
4	Datsun 710	22.8	93		
5	Hornet 4 Drive	21.4	110		
6	Hornet Sportabout	18.7	175		
7	Valiant	18.1	105		
8	Duster 360	14.3	245		
9	Merc 240D	24.4	62		
10	Merc 230	22.8	95		
11	Merc 280	19.2	123		
12	Merc 280C	17.8	123		
13	Merc 450SE	16.4	180		
14	Merc 450SL	17.3	180		
15	Merc 450SLC	15.2	180		
16	Cadillac Fleetwood	10.4	205		
17	Lincoln Continental	10.4	215		
18	Chrysler Imperial	14.7	230		
19	Fiat 128	32.4	66		

5. FILTER ALL THE MPG > 15 AND AM = 1

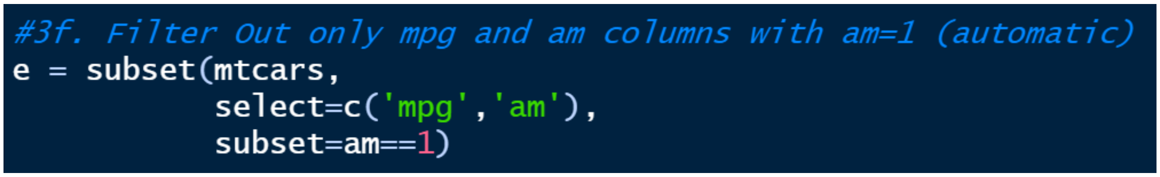


```
#3e. Filter all the mpg>15 and am=1
d = mtcars[mtcars$mpg>15 & mtcars$am==1,]
```

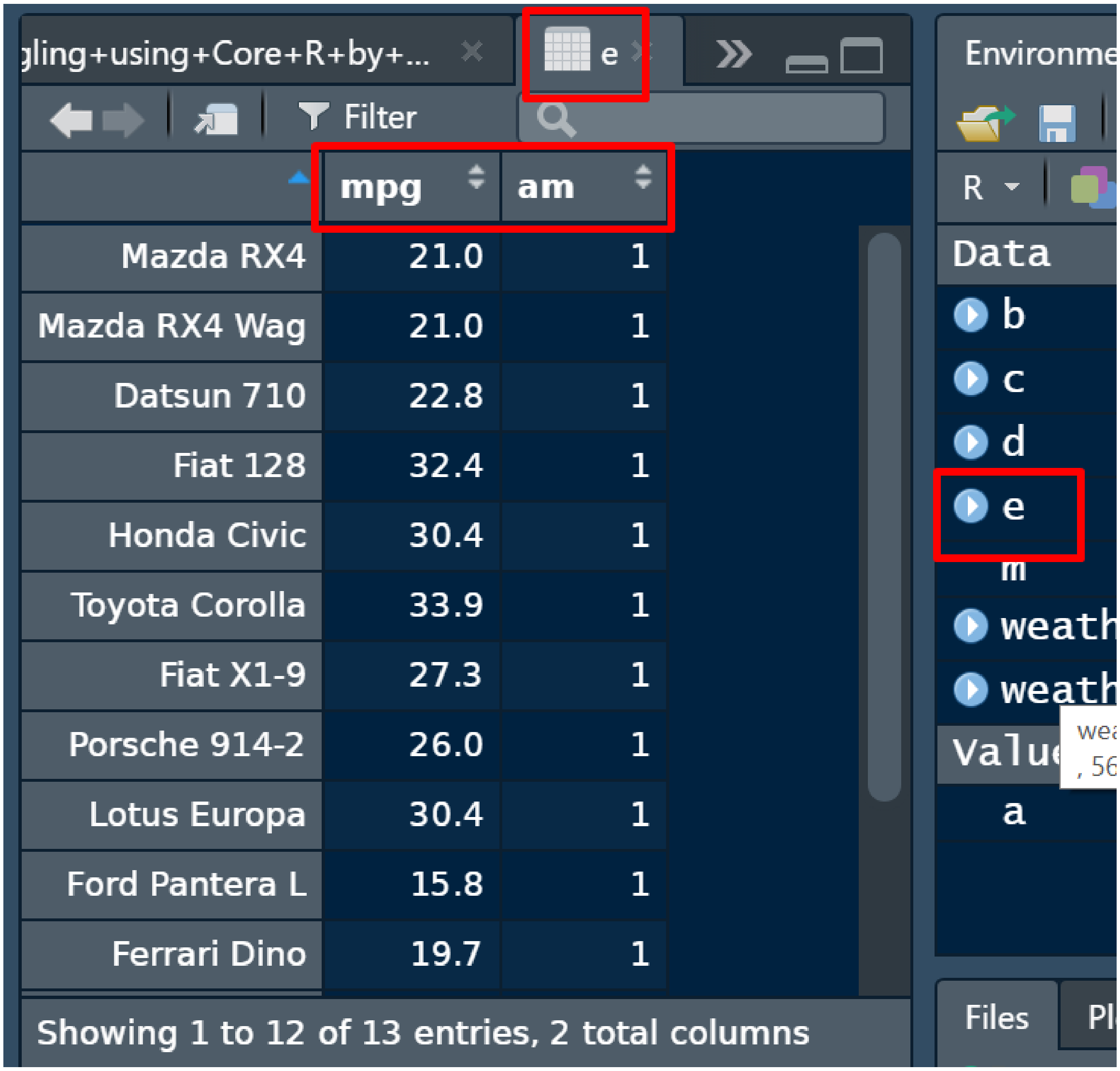


	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	

6. FILTER OUT ONLY MPG AND AM COLUMNS WITH AM = 1 (AUTOMATIC)



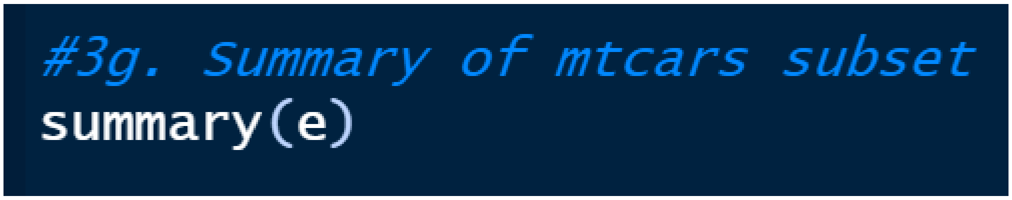
```
#3f. Filter Out only mpg and am columns with am=1 (automatic)
e = subset(mtcars,
           select=c('mpg','am'),
           subset=am==1)
```



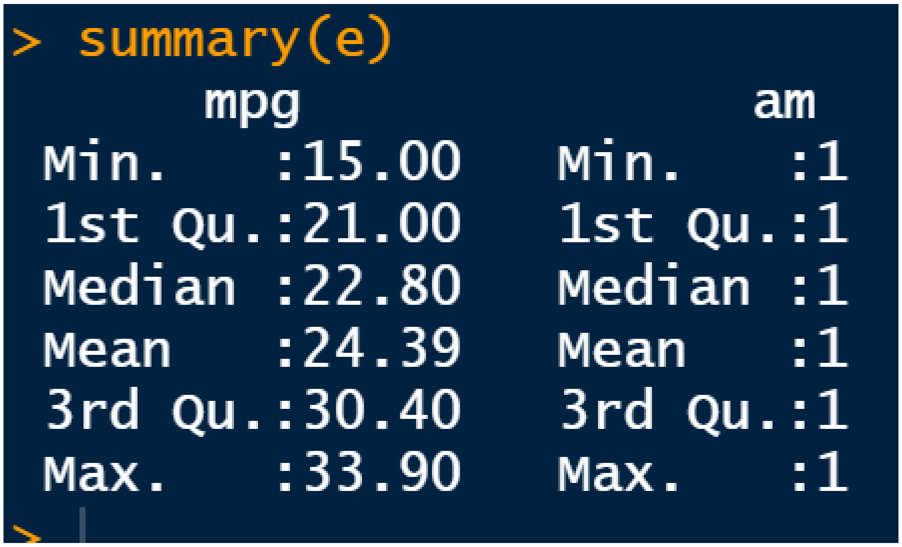
	mpg	am
Mazda RX4	21.0	1
Mazda RX4 Wag	21.0	1
Datsun 710	22.8	1
Fiat 128	32.4	1
Honda Civic	30.4	1
Toyota Corolla	33.9	1
Fiat X1-9	27.3	1
Porsche 914-2	26.0	1
Lotus Europa	30.4	1
Ford Pantera L	15.8	1
Ferrari Dino	19.7	1

Showing 1 to 12 of 13 entries, 2 total columns

7. SUMMARY OF MTCARS SUBSET

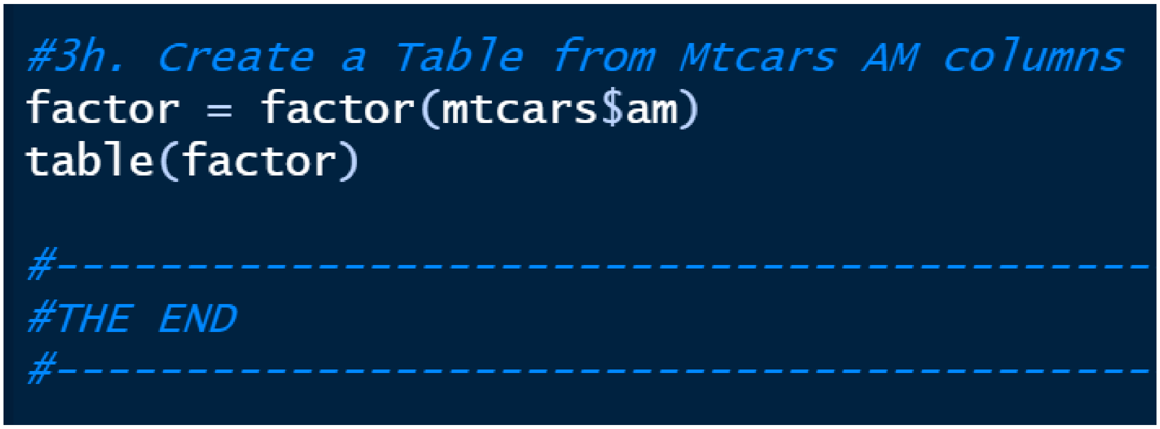


```
#3g. Summary of mtcars subset
summary(e)
```

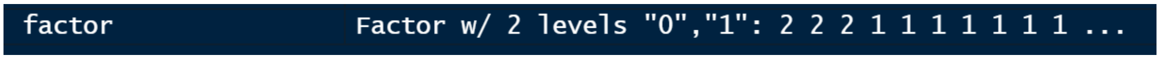


```
> summary(e)
      mpg                  am   
 Min.   :15.00   Min.   :1  
 1st Qu.:21.00   1st Qu.:1  
 Median :22.80   Median :1  
 Mean   :24.39   Mean   :1  
 3rd Qu.:30.40   3rd Qu.:1  
 Max.   :33.90   Max.   :1  
> 
```

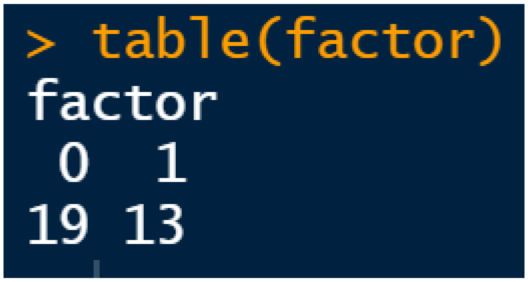
8. CREATE A TABLE FROM MTCARS AM COLUMNS



```
#3h. Create a Table from Mtcars AM columns
factor = factor(mtcars$am)
table(factor)


#------------------------------------------
#THE END
#------------------------------------------
```



```
factor                 Factor w/ 2 levels "0","1": 2 2 2 1 1 1 1 1 1 1 ...
```



```
> table(factor)
factor
 0  1 
19 13 
```

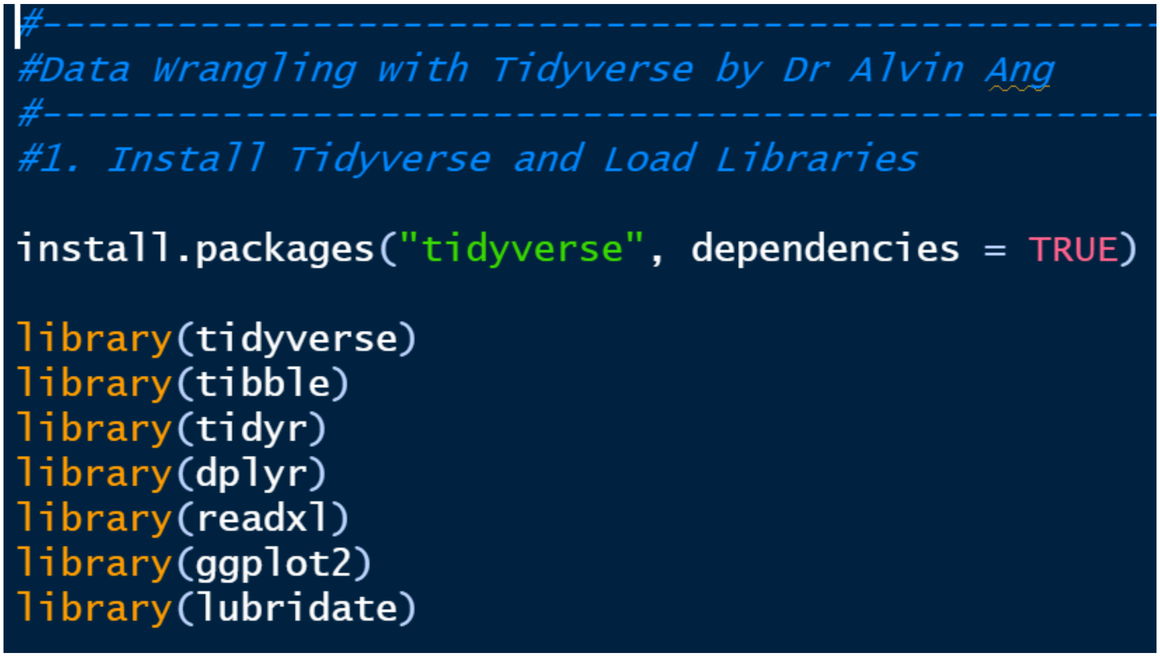
II. DATA WRANGLING USING TIDYVERSE

https://www.alvinang.sg/s/Data-Wrangling-with-Tidyverse-by-Dr-Alvin-Ang.R

Tons of great Data Wrangling with R here:

https://www.marsja.se/how-to-rename-column-or-columns-in-r-with-dplyr/

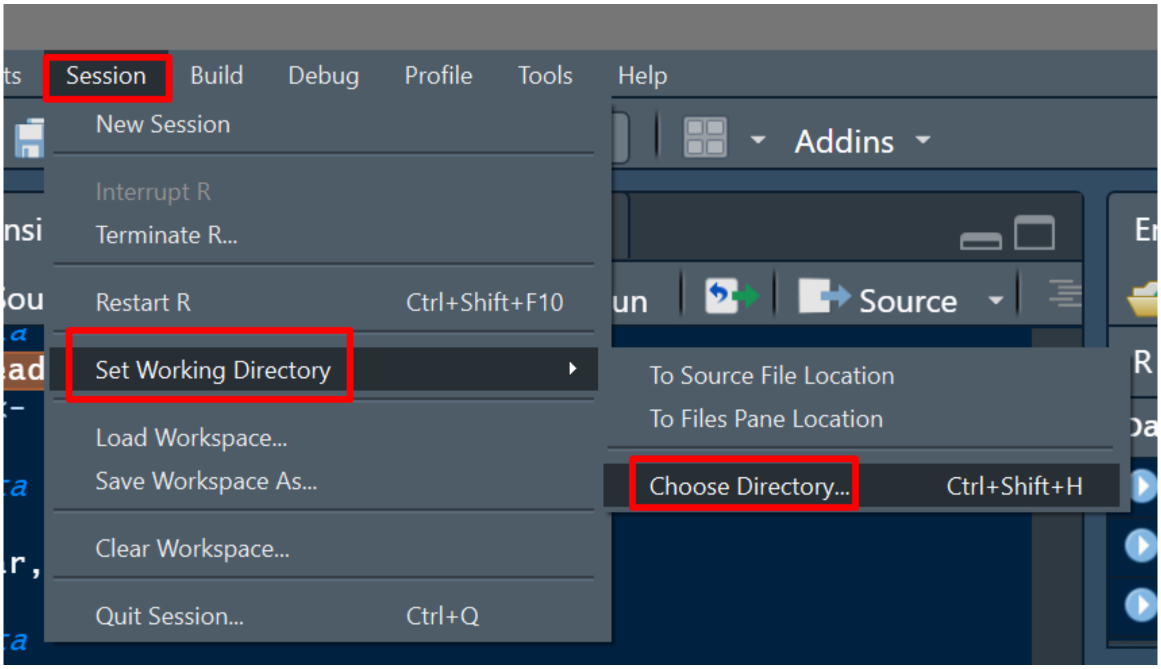
A. INSTALLING TIDYVERSE PACKAGE



```r
#---------------------------------------------------
#Data Wrangling with Tidyverse by Dr Alvin Ang
#---------------------------------------------------
#1. Install Tidyverse and Load Libraries

install.packages("tidyverse", dependencies = TRUE)

library(tidyverse)
library(tibble)
library(tidyr)
library(dplyr)
library(readxl)
library(ggplot2)
library(lubridate)
```

B. READING IN CSV

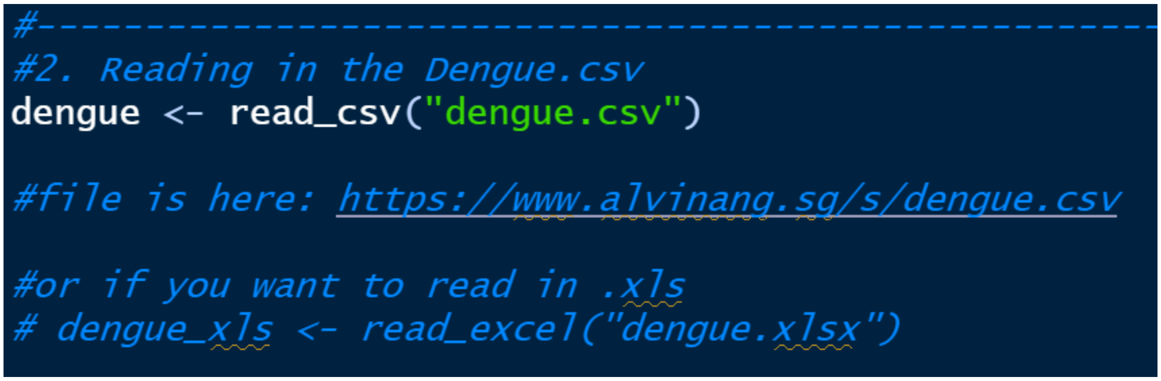
File can be found here: https://www.alvinang.sg/s/dengue.csv



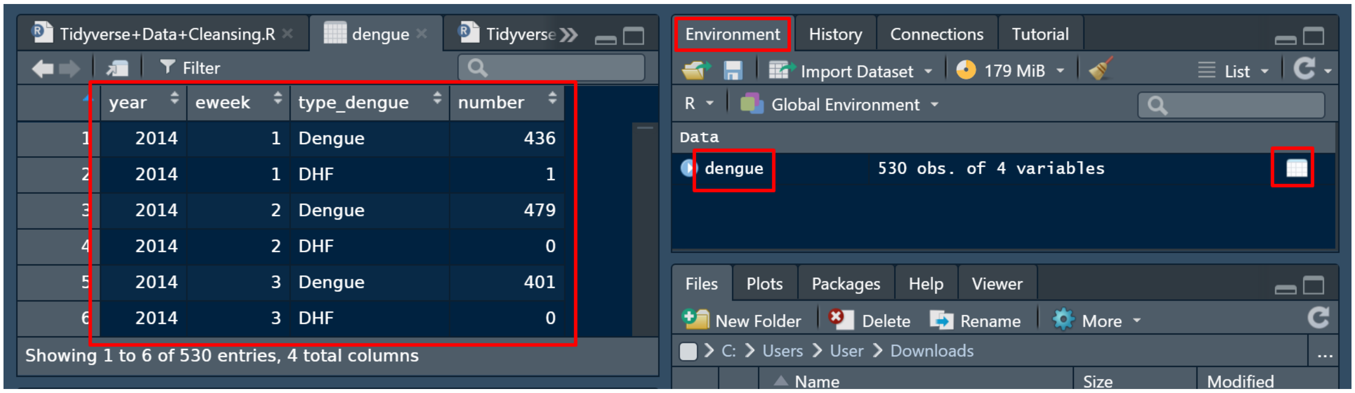
- Do you know where you stored the dengue.csv downloaded file?

- Most probably is in your download folder.

- Make sure that you set the working directory to that folder (download folder)… so that it can import in the CSV.



```r
#-----------------------------------------------------------------
#2. Reading in the Dengue.csv
dengue <- read_csv("dengue.csv")

#file is here: https://www.alvinang.sg/s/dengue.csv

#or if you want to read in .xls
# dengue_xls <- read_excel("dengue.xlsx")
```

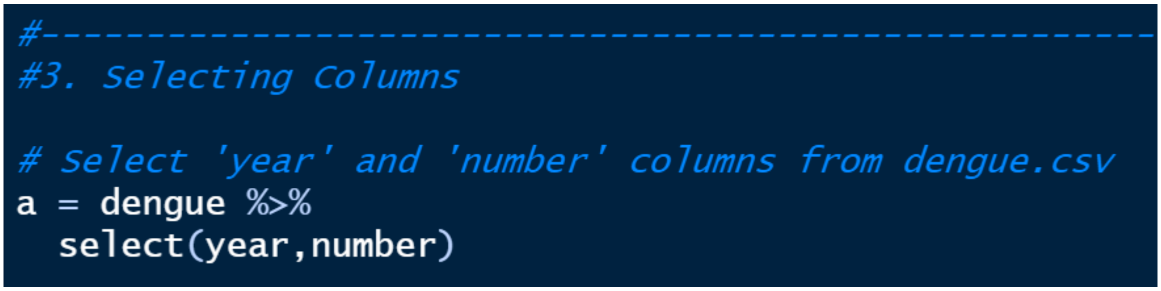


	year	eweek	type_dengue	number
1	2014	1	Dengue	436
2	2014	1	DHF	1
3	2014	2	Dengue	479
4	2014	2	DHF	0
5	2014	3	Dengue	401
6	2014	3	DHF	0

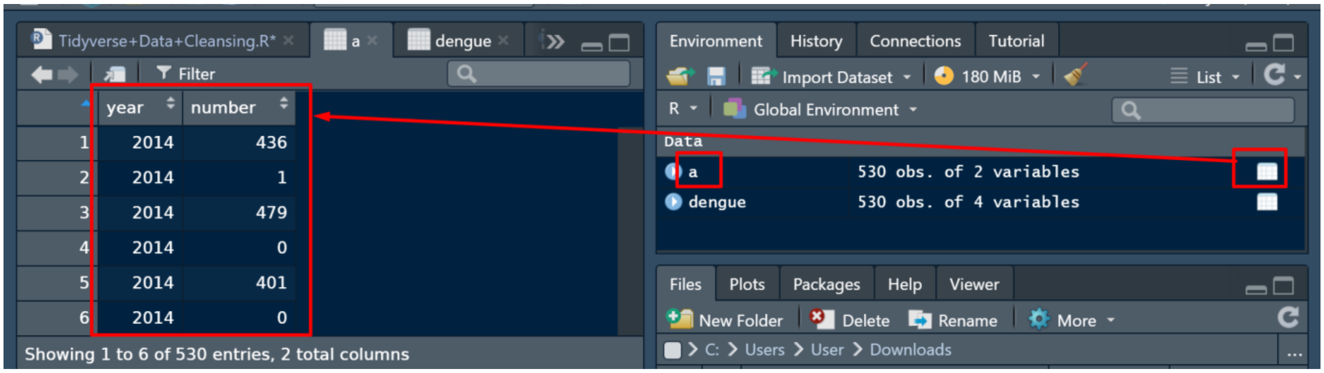
Showing 1 to 6 of 530 entries, 4 total columns

Data
dengue 530 obs. of 4 variables

C. SELECTING COLUMNS



```
#------------------------------------------------------
#3. Selecting Columns

# Select 'year' and 'number' columns from dengue.csv
a = dengue %>%
  select(year,number)
```



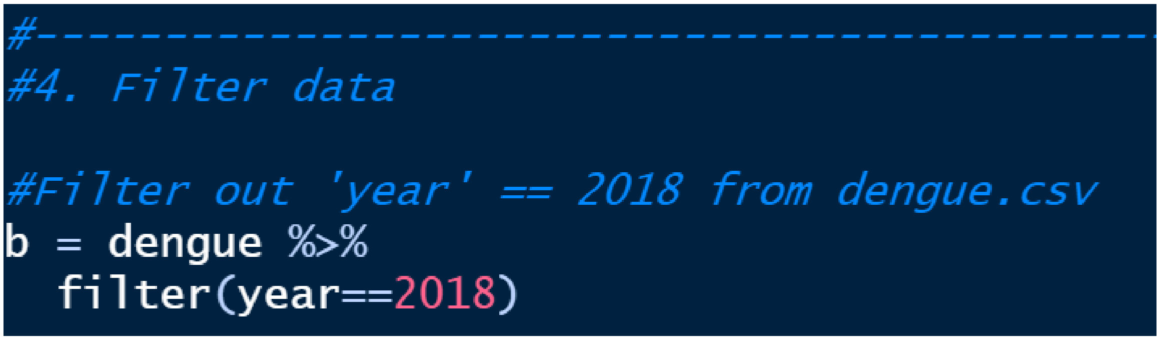
	year	number
1	2014	436
2	2014	1
3	2014	479
4	2014	0
5	2014	401
6	2014	0

Showing 1 to 6 of 530 entries, 2 total columns

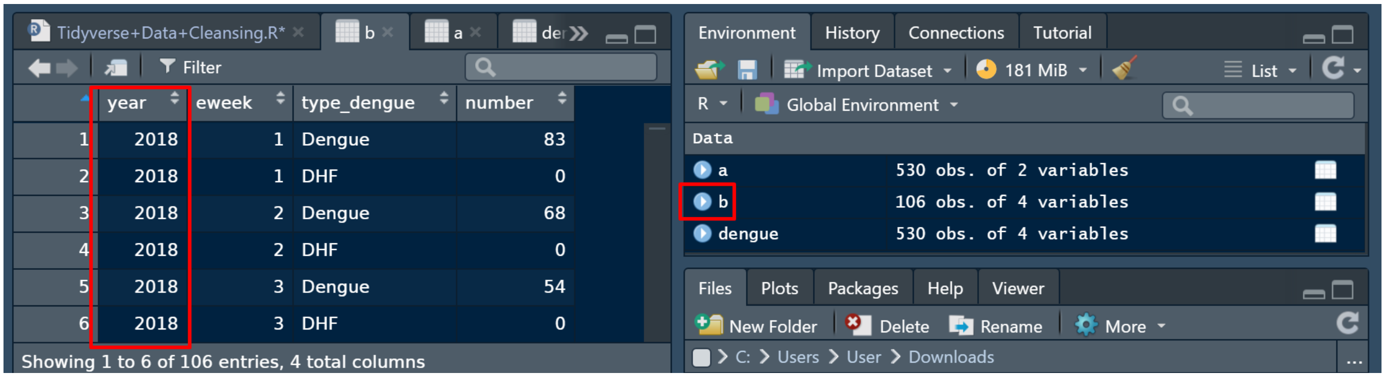
```
Data
a             530 obs. of 2 variables
dengue        530 obs. of 4 variables
```

D. FILTERING DATA

1. FILTER OUT 'YEAR' == 2018 FROM DENGUE.CSV



```
#---------------------------------------------------
#4. Filter data

#Filter out 'year' == 2018 from dengue.csv
b = dengue %>%
  filter(year==2018)
```



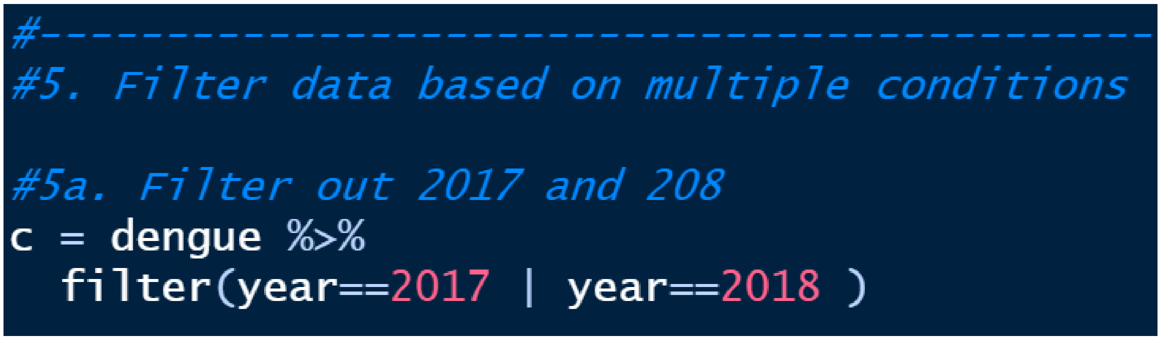
	year	eweek	type_dengue	number
1	2018	1	Dengue	83
2	2018	1	DHF	0
3	2018	2	Dengue	68
4	2018	2	DHF	0
5	2018	3	Dengue	54
6	2018	3	DHF	0

Showing 1 to 6 of 106 entries, 4 total columns

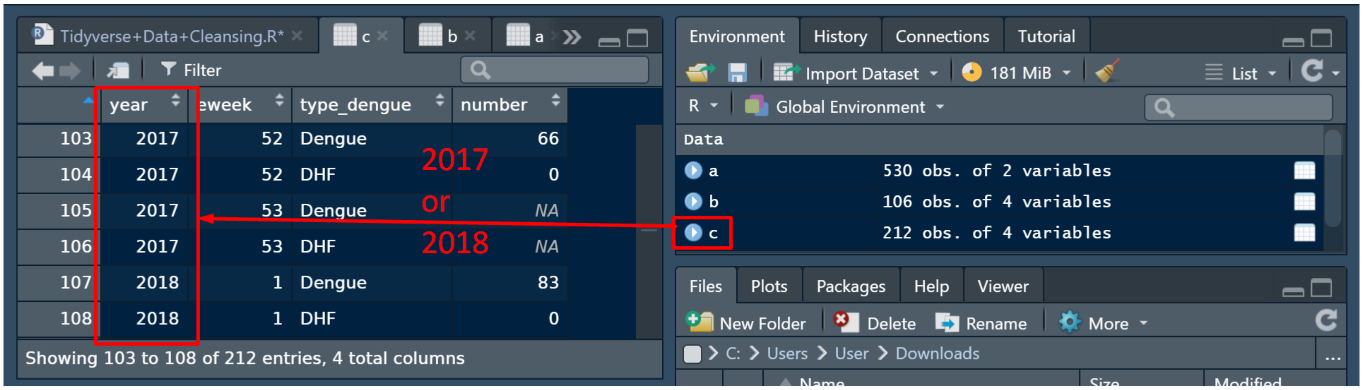
```
Data
a         530 obs. of 2 variables
b         106 obs. of 4 variables
dengue    530 obs. of 4 variables
```

E. FILTER DATA BASED ON MULTIPLE CONDITIONS

1. FILTER OUT 2017 AND 2018



```
#------------------------------------------------
#5. Filter data based on multiple conditions

#5a. Filter out 2017 and 208
c = dengue %>%
  filter(year==2017 | year==2018 )
```

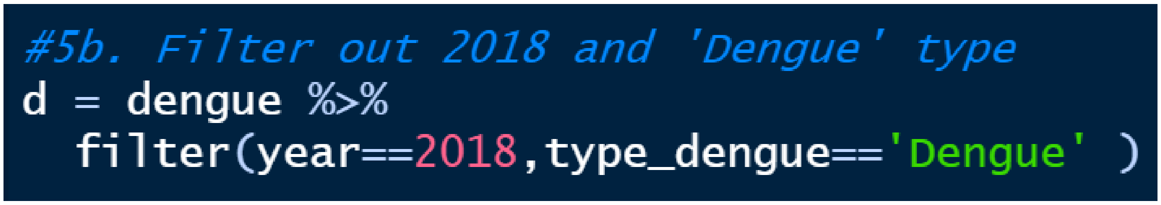


	year	eweek	type_dengue	number
103	2017	52	Dengue	66
104	2017	52	DHF	0
105	2017	53	Dengue	NA
106	2017	53	DHF	NA
107	2018	1	Dengue	83
108	2018	1	DHF	0

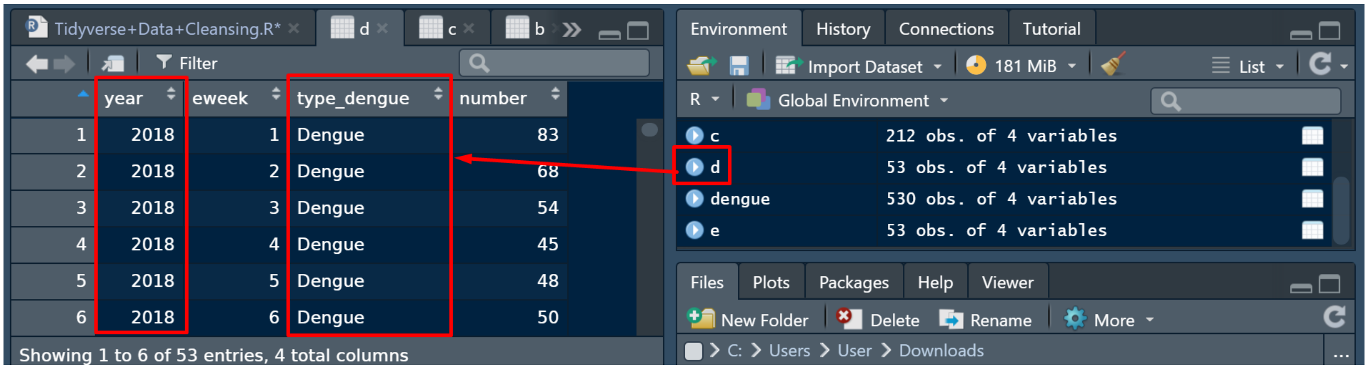
Showing 103 to 108 of 212 entries, 4 total columns

```
Data
a          530 obs. of 2 variables
b          106 obs. of 4 variables
c          212 obs. of 4 variables
```

2. FILTER OUT 2018 AND DENGUE TYPE



```
#5b. Filter out 2018 and 'Dengue' type
d = dengue %>%
  filter(year==2018,type_dengue=='Dengue' )
```

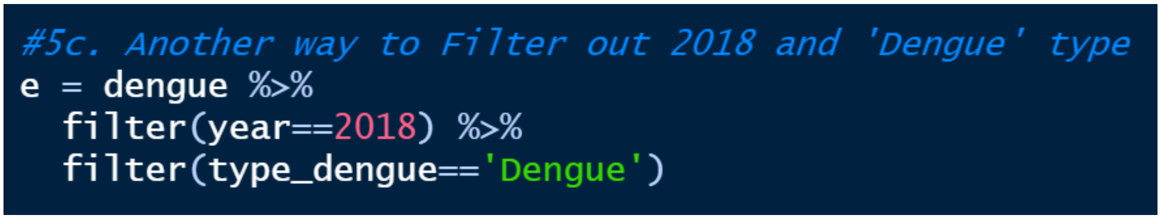


	year	eweek	type_dengue	number
1	2018	1	Dengue	83
2	2018	2	Dengue	68
3	2018	3	Dengue	54
4	2018	4	Dengue	45
5	2018	5	Dengue	48
6	2018	6	Dengue	50

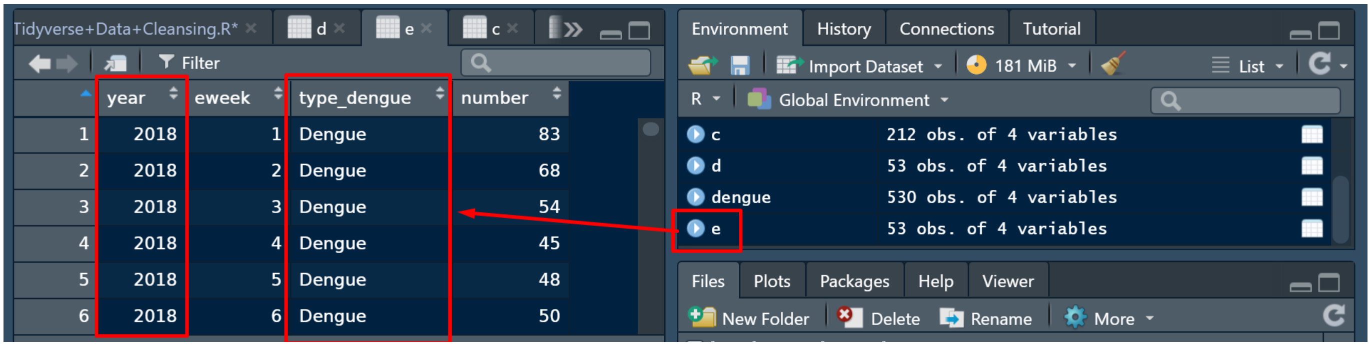
Showing 1 to 6 of 53 entries, 4 total columns

```
c          212 obs. of 4 variables
d          53 obs. of 4 variables
dengue     530 obs. of 4 variables
e          53 obs. of 4 variables
```

3. ANOTHER WAY TO FILTER OUT 2018 AND DENGUE TYPE



```
#5c. Another way to Filter out 2018 and 'Dengue' type
e = dengue %>%
  filter(year==2018) %>%
  filter(type_dengue=='Dengue')
```

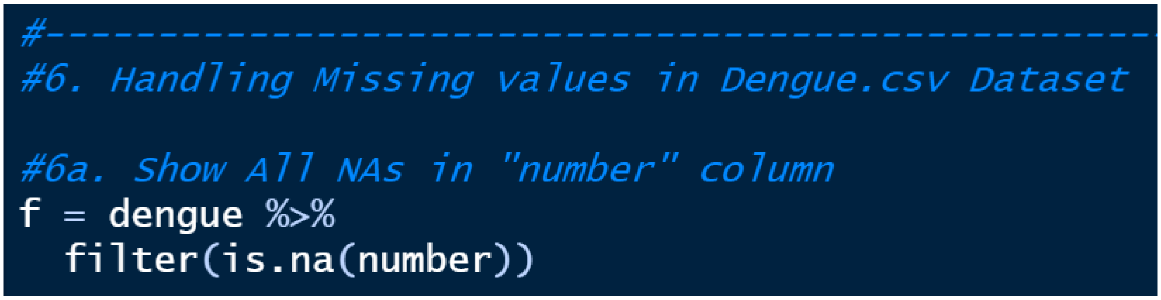


	year	eweek	type_dengue	number
1	2018	1	Dengue	83
2	2018	2	Dengue	68
3	2018	3	Dengue	54
4	2018	4	Dengue	45
5	2018	5	Dengue	48
6	2018	6	Dengue	50

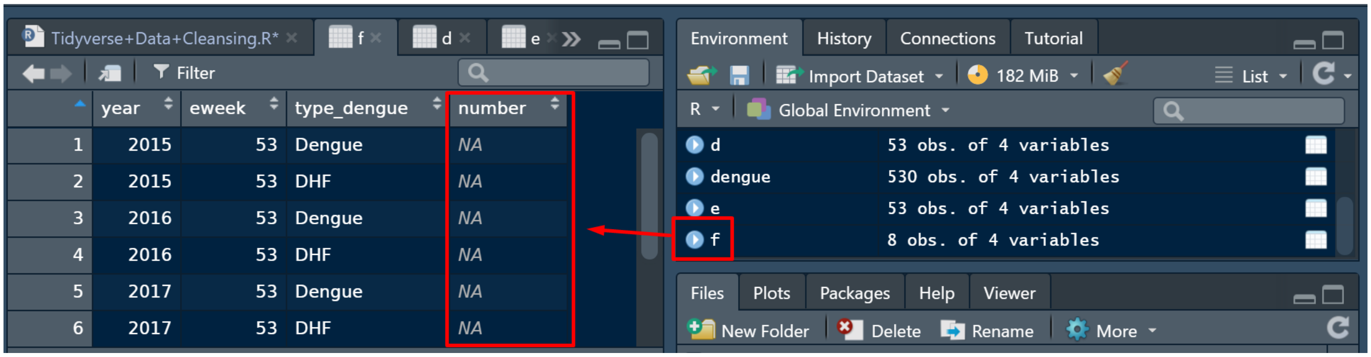
```
c         212 obs. of 4 variables
d         53 obs. of 4 variables
dengue    530 obs. of 4 variables
e         53 obs. of 4 variables
```

F. HANDLING MISSING VALUES IN DENGUE.CSV DATASET

1. SHOW ALL NAS IN "NUMBER" COLUMN



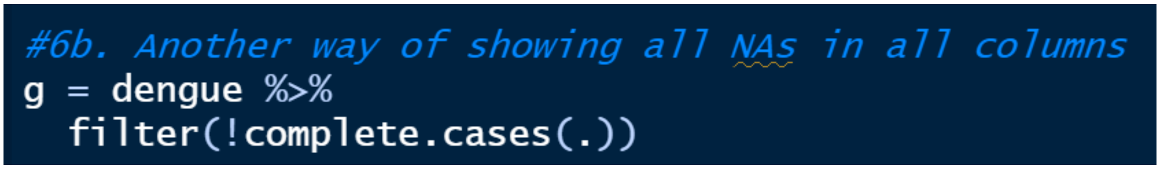
```
#-------------------------------------------------
#6. Handling Missing values in Dengue.csv Dataset

#6a. Show All NAs in "number" column
f = dengue %>%
  filter(is.na(number))
```



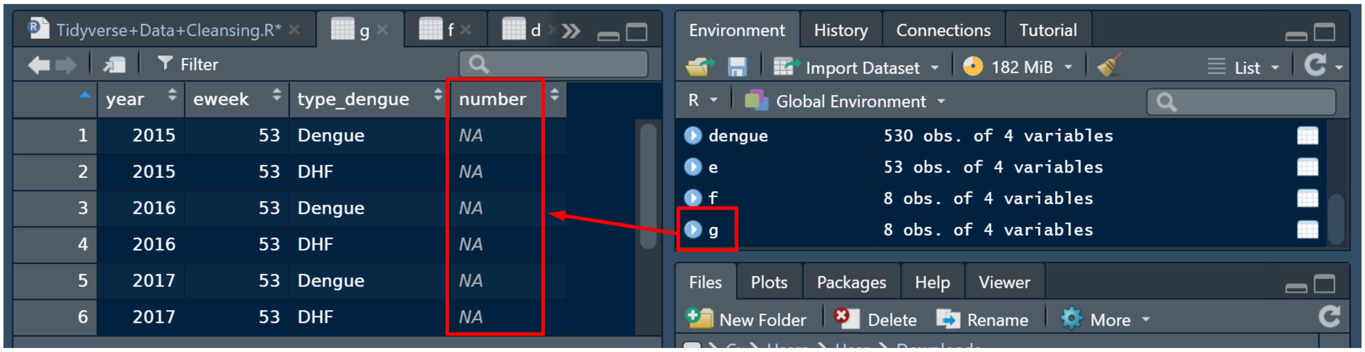
	year	eweek	type_dengue	number
1	2015	53	Dengue	NA
2	2015	53	DHF	NA
3	2016	53	Dengue	NA
4	2016	53	DHF	NA
5	2017	53	Dengue	NA
6	2017	53	DHF	NA

```
d        53 obs. of 4 variables
dengue   530 obs. of 4 variables
e        53 obs. of 4 variables
f        8 obs. of 4 variables
```

2. ANOTHER WAY OF SHOWING ALL NAS IN ALL COLUMNS



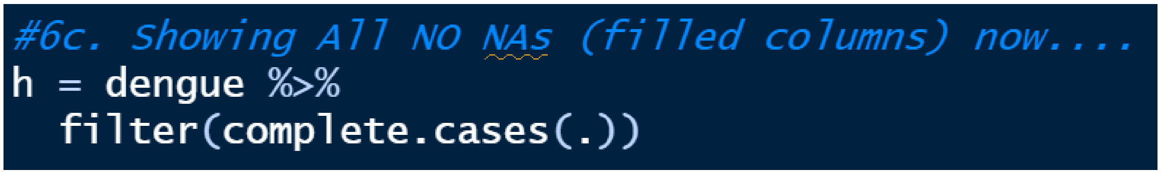
```
#6b. Another way of showing all NAs in all columns
g = dengue %>%
  filter(!complete.cases(.))
```



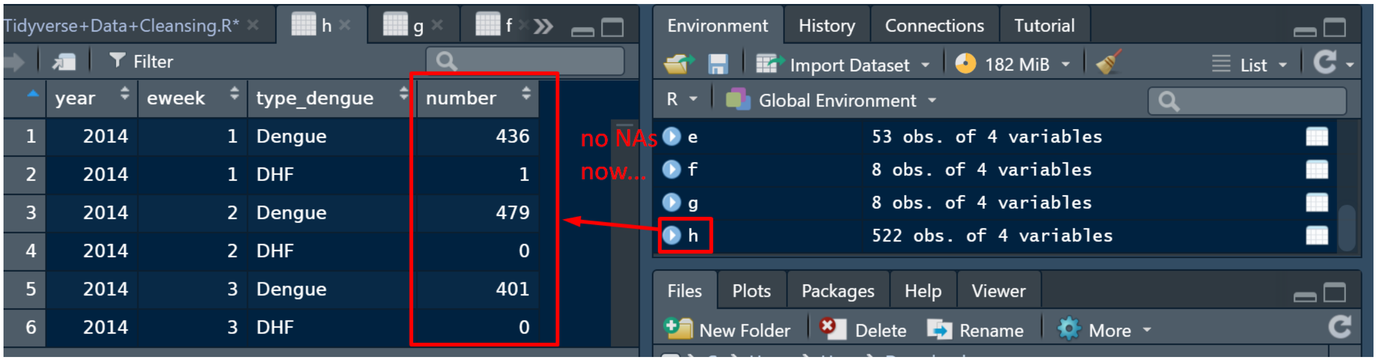
	year	eweek	type_dengue	number
1	2015	53	Dengue	NA
2	2015	53	DHF	NA
3	2016	53	Dengue	NA
4	2016	53	DHF	NA
5	2017	53	Dengue	NA
6	2017	53	DHF	NA

```
dengue   530 obs. of 4 variables
e        53 obs. of 4 variables
f        8 obs. of 4 variables
g        8 obs. of 4 variables
```

3. SHOWING ALL NO NAS (FILLED COLUMNS) NOW



```
#6c. Showing All NO NAs (filled columns) now....
h = dengue %>%
  filter(complete.cases(.))
```

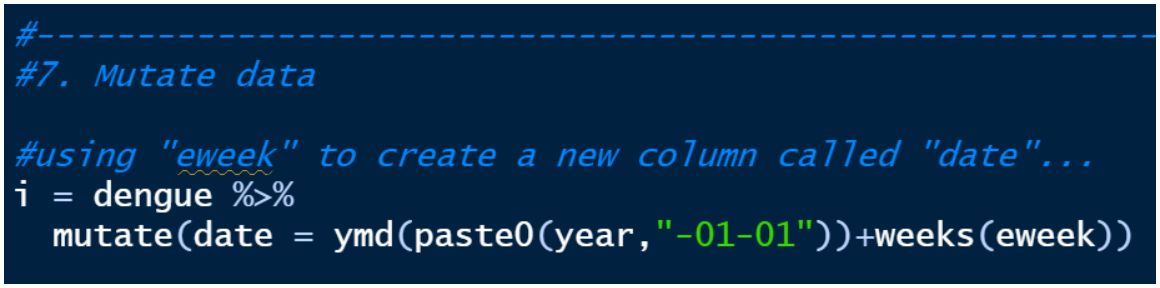


	year	eweek	type_dengue	number
1	2014	1	Dengue	436
2	2014	1	DHF	1
3	2014	2	Dengue	479
4	2014	2	DHF	0
5	2014	3	Dengue	401
6	2014	3	DHF	0

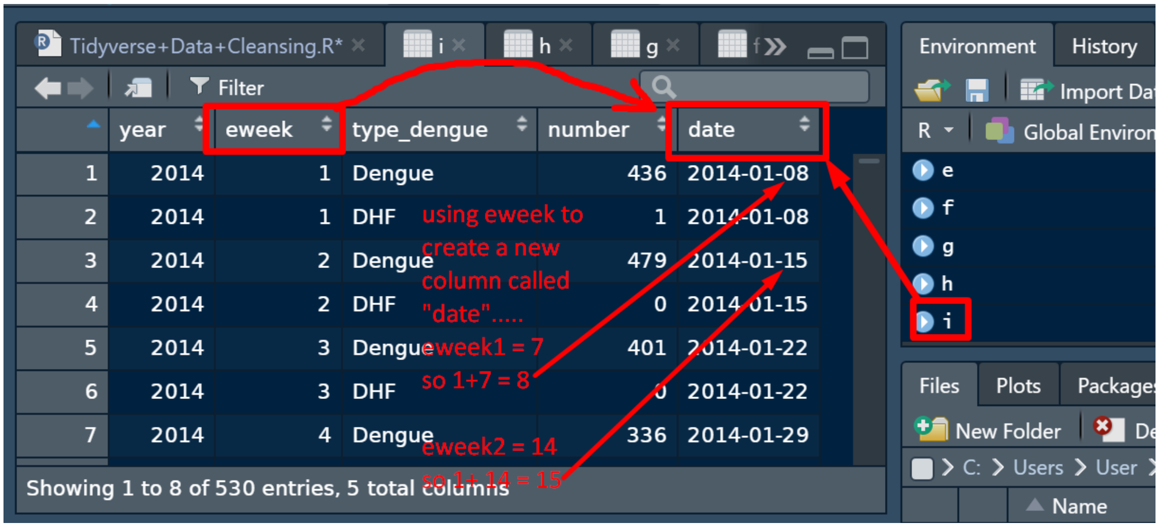
no NAs now...

```
e    53 obs. of 4 variables
f    8 obs. of 4 variables
g    8 obs. of 4 variables
h    522 obs. of 4 variables
```

G. MUTATE DATA



```
#------------------------------------------------------------
#7. Mutate data

#using "eweek" to create a new column called "date"...
i = dengue %>%
  mutate(date = ymd(paste0(year,"-01-01"))+weeks(eweek))
```



	year	eweek	type_dengue	number	date
1	2014	1	Dengue	436	2014-01-08
2	2014	1	DHF	1	2014-01-08
3	2014	2	Dengue	479	2014-01-15
4	2014	2	DHF	0	2014-01-15
5	2014	3	Dengue	401	2014-01-22
6	2014	3	DHF	0	2014-01-22
7	2014	4	Dengue	336	2014-01-29

Showing 1 to 8 of 530 entries, 5 total columns

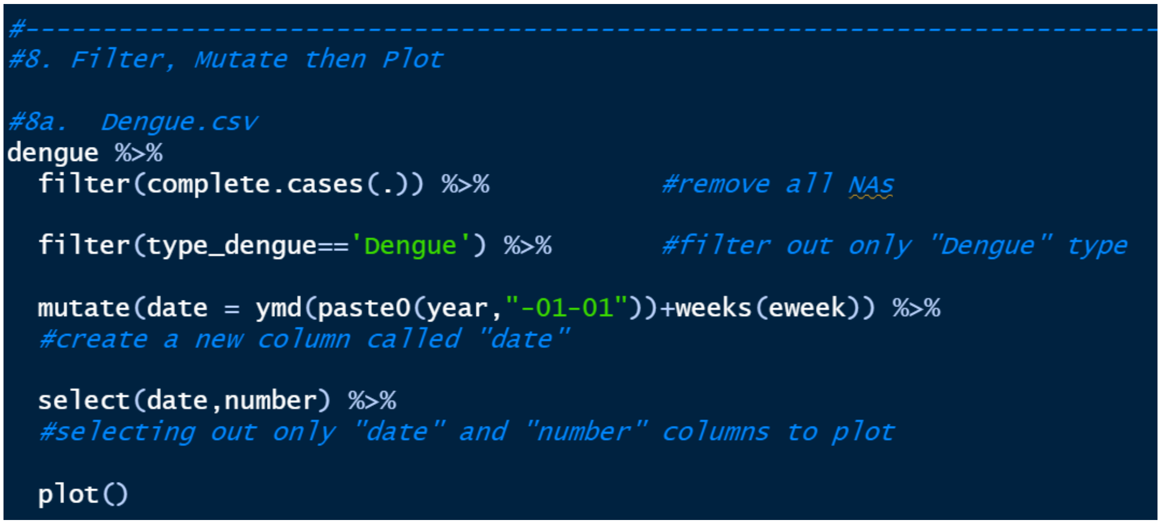
using eweek to create a new column called "date".....

eweek1 = 7 so 1+7 = 8

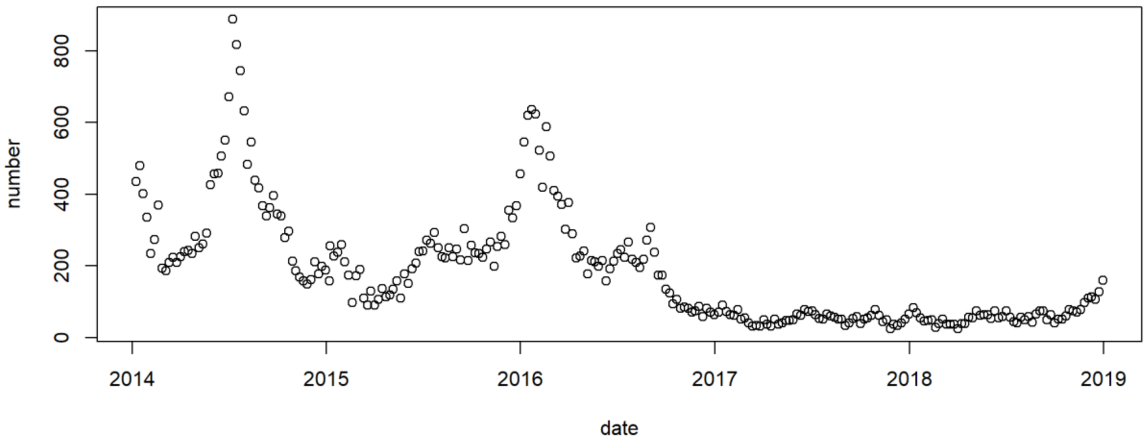
eweek2 = 14 so 1+ 14 = 15

H. FILTER, MUTATE THEN PLOT

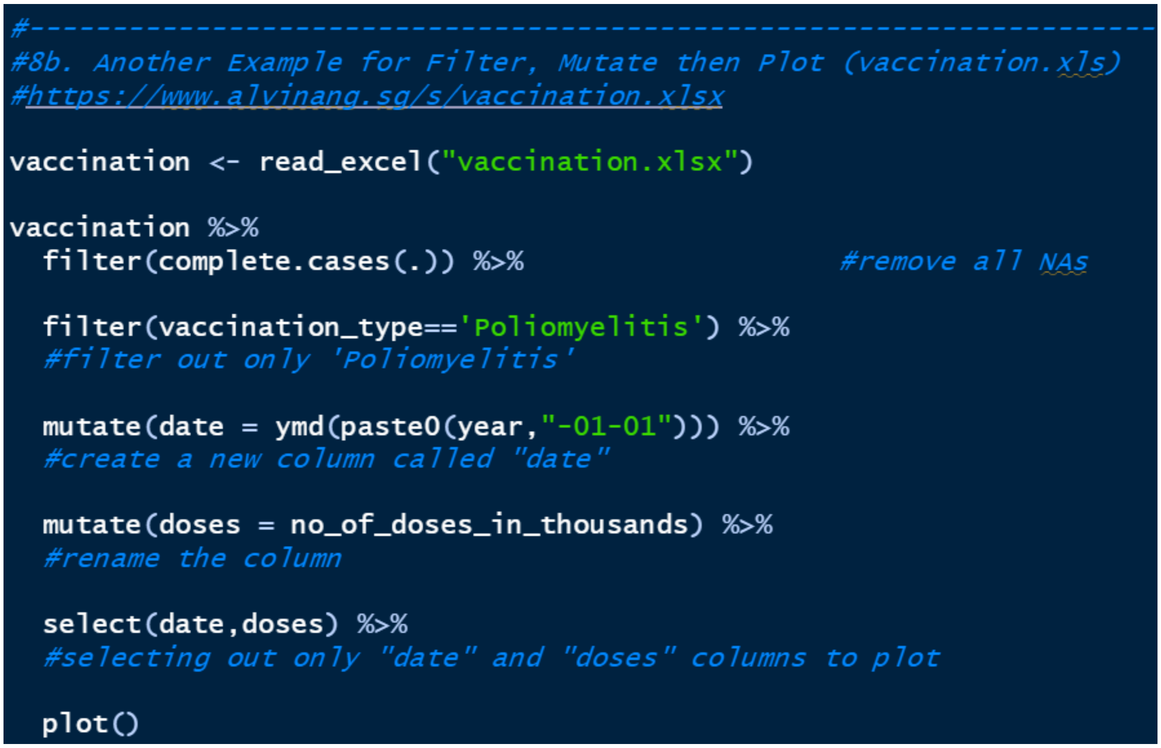
1. DENGUE.CSV



```
#---------------------------------------------------------------------
#8. Filter, Mutate then Plot

#8a.  Dengue.csv
dengue %>%
  filter(complete.cases(.)) %>%           #remove all NAs

  filter(type_dengue=='Dengue') %>%       #filter out only "Dengue" type

  mutate(date = ymd(paste0(year,"-01-01"))+weeks(eweek)) %>%
  #create a new column called "date"

  select(date,number) %>%
  #selecting out only "date" and "number" columns to plot

  plot()
```

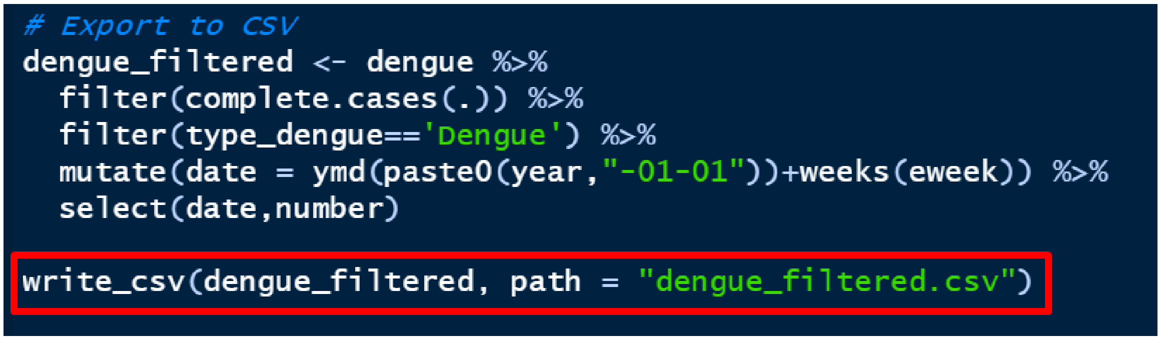


2. ANOTHER EXAMPLE FOR FILTER, MUTATE THEN PLOT (VACCINATION.XLS)

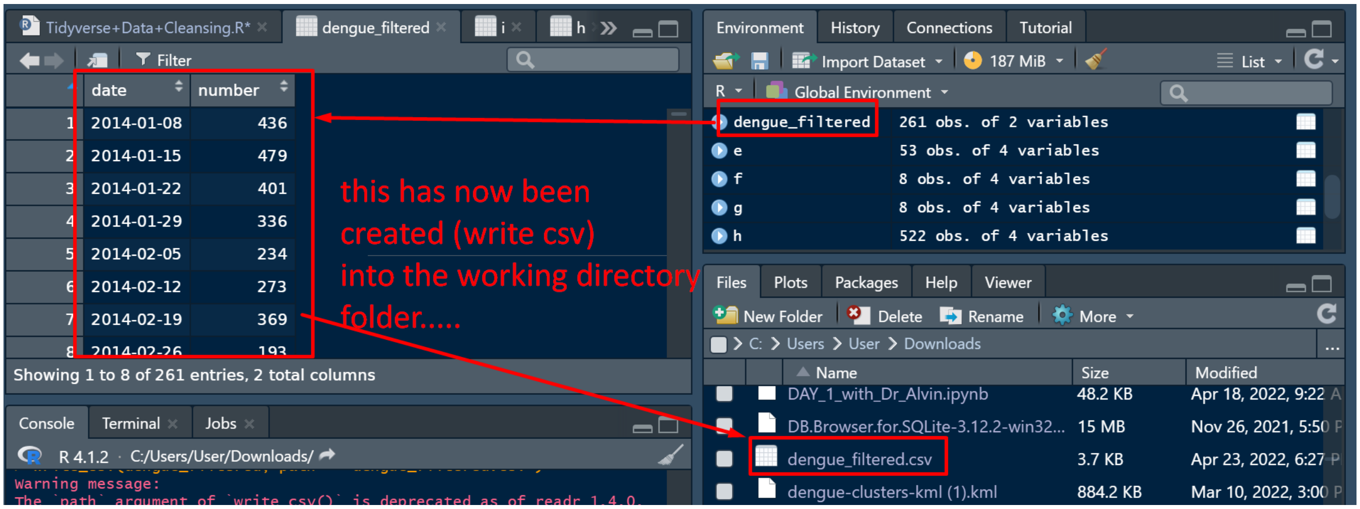


```
#--------------------------------------------------------------------
#8b. Another Example for Filter, Mutate then Plot (vaccination.xls)
#https://www.alvinang.sg/s/vaccination.xlsx

vaccination <- read_excel("vaccination.xlsx")

vaccination %>%
  filter(complete.cases(.)) %>%                     #remove all NAs

  filter(vaccination_type=='Poliomyelitis') %>%
  #filter out only 'Poliomyelitis'

  mutate(date = ymd(paste0(year,"-01-01"))) %>%
  #create a new column called "date"

  mutate(doses = no_of_doses_in_thousands) %>%
  #rename the column

  select(date,doses) %>%
  #selecting out only "date" and "doses" columns to plot

  plot()
```

3. FILTER, MUTATE THEN EXPORT TO CSV



```r
# Export to CSV
dengue_filtered <- dengue %>%
  filter(complete.cases(.)) %>%
  filter(type_dengue=='Dengue') %>%
  mutate(date = ymd(paste0(year,"-01-01"))+weeks(eweek)) %>%
  select(date,number)

write_csv(dengue_filtered, path = "dengue_filtered.csv")
```



	date	number
1	2014-01-08	436
2	2014-01-15	479
3	2014-01-22	401
4	2014-01-29	336
5	2014-02-05	234
6	2014-02-12	273
7	2014-02-19	369
8	2014-02-26	193

Showing 1 to 8 of 261 entries, 2 total columns

```
dengue_filtered    261 obs. of 2 variables
e                  53 obs. of 4 variables
f                  8 obs. of 4 variables
g                  8 obs. of 4 variables
h                  522 obs. of 4 variables
```

this has now been created (write csv) into the working directory folder.....

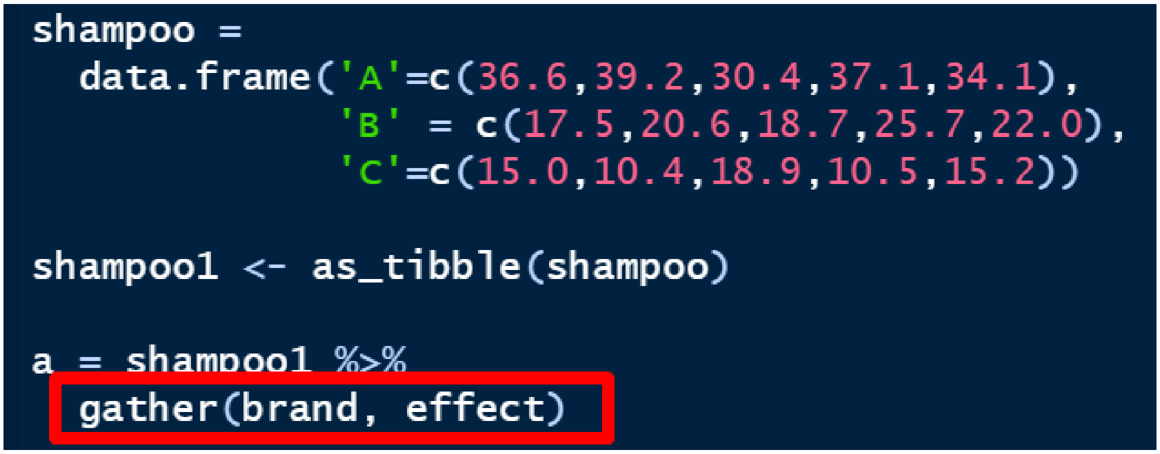
C: > Users > User > Downloads

Name	Size	Modified
DAY_1_with_Dr_Alvin.ipynb	48.2 KB	Apr 18, 2022, 9:22 A
DB.Browser.for.SQLite-3.12.2-win32...	15 MB	Nov 26, 2021, 5:50 P
dengue_filtered.csv	3.7 KB	Apr 23, 2022, 6:27 P
dengue-clusters-kml (1).kml	884.2 KB	Mar 10, 2022, 3:00 P

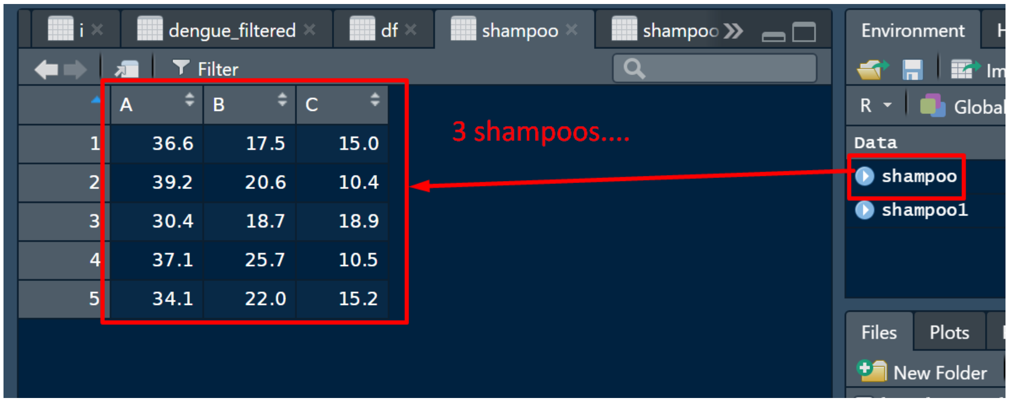
R 4.1.2 · C:/Users/User/Downloads/

```
Warning message:
The `path` argument of `write_csv()` is deprecated as of readr 1.4.0.
```

4. USING GATHER TO PIVOT DATA

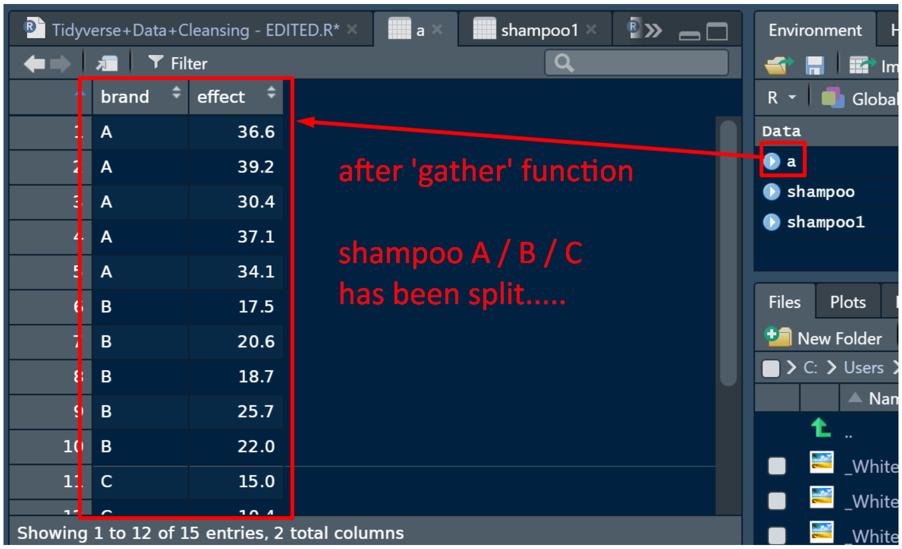


```
shampoo =
  data.frame('A'=c(36.6,39.2,30.4,37.1,34.1),
                      'B' = c(17.5,20.6,18.7,25.7,22.0),
                      'C'=c(15.0,10.4,18.9,10.5,15.2))

shampoo1 <- as_tibble(shampoo)

a = shampoo1 %>%
  gather(brand, effect)
```



	A	B	C
1	36.6	17.5	15.0
2	39.2	20.6	10.4
3	30.4	18.7	18.9
4	37.1	25.7	10.5
5	34.1	22.0	15.2

3 shampoos....



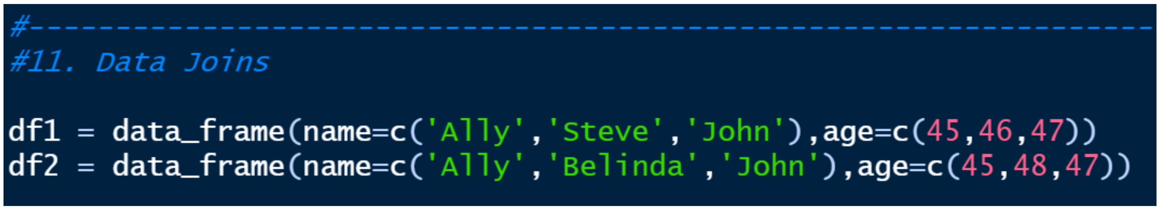
	brand	effect
1	A	36.6
2	A	39.2
3	A	30.4
4	A	37.1
5	A	34.1
6	B	17.5
7	B	20.6
8	B	18.7
9	B	25.7
10	B	22.0
11	C	15.0
12	C	10.4

after 'gather' function

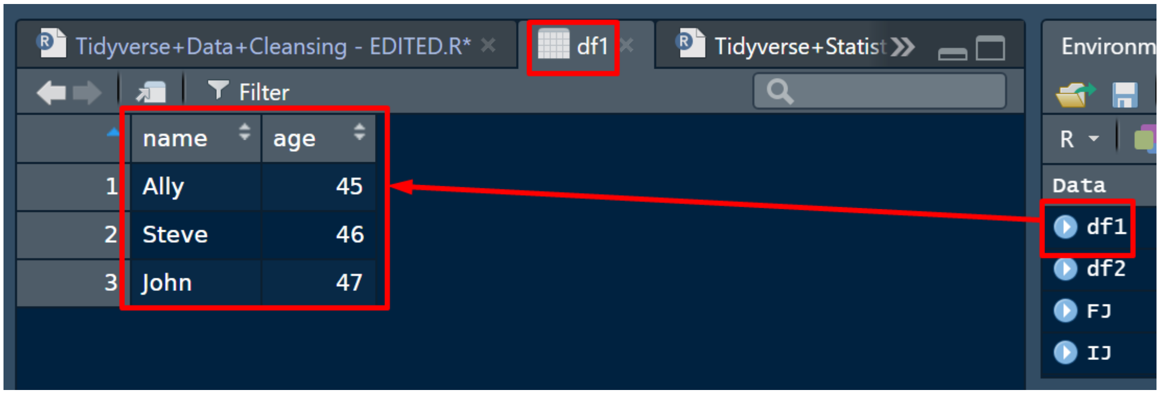
shampoo A / B / C has been split.....

Showing 1 to 12 of 15 entries, 2 total columns

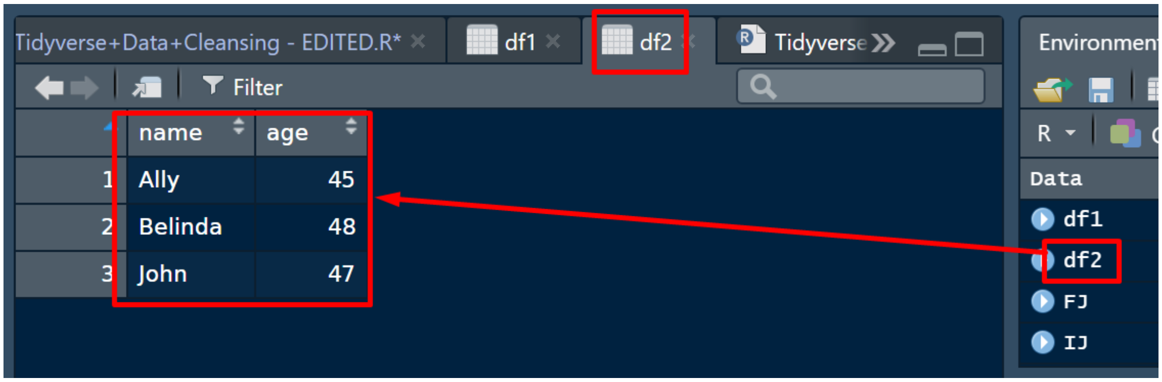
I. JOINS



```
#-------------------------------------------------------------------
#11. Data Joins

df1 = data_frame(name=c('Ally','Steve','John'),age=c(45,46,47))
df2 = data_frame(name=c('Ally','Belinda','John'),age=c(45,48,47))
```

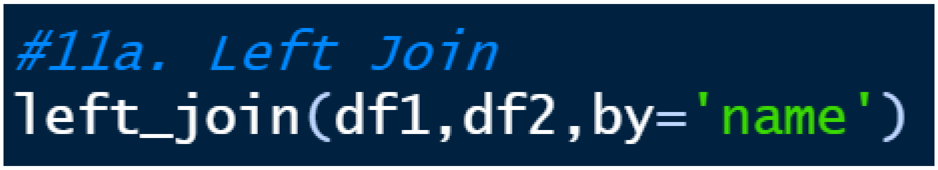


	name	age
1	Ally	45
2	Steve	46
3	John	47

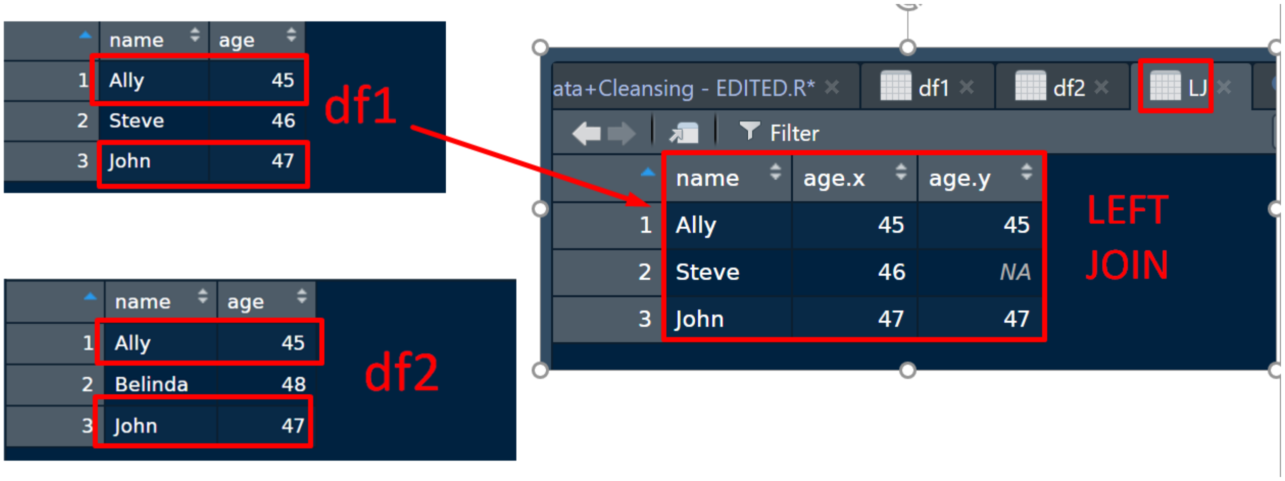


	name	age
1	Ally	45
2	Belinda	48
3	John	47

1. LEFT JOIN



```
#11a. Left Join
left_join(df1,df2,by='name')
```

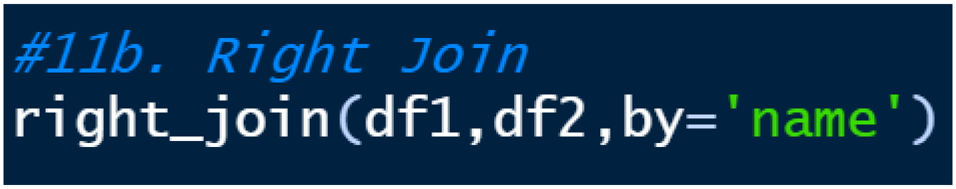


	name	age
1	Ally	45
2	Steve	46
3	John	47

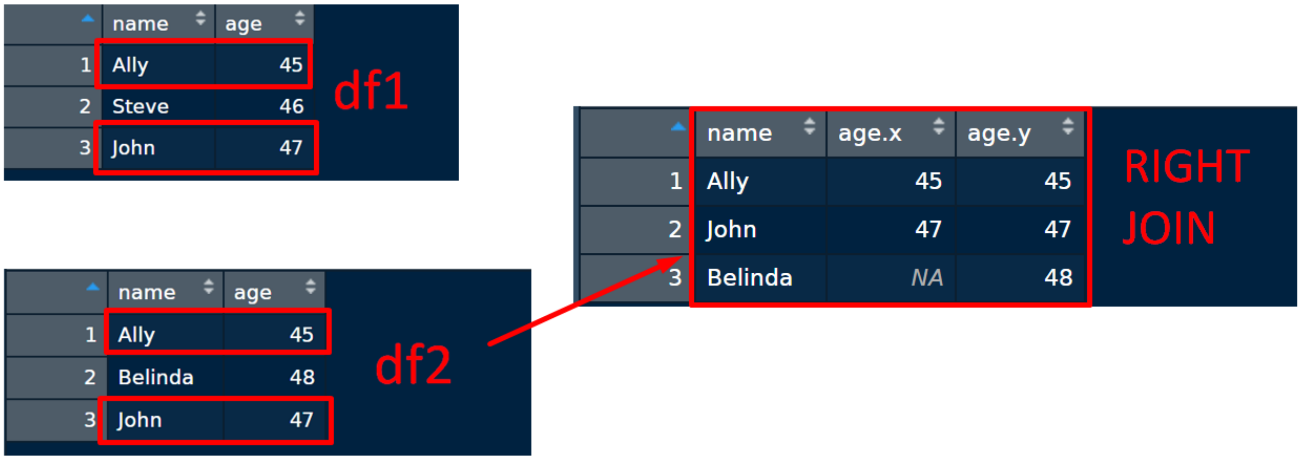
	name	age
1	Ally	45
2	Belinda	48
3	John	47

	name	age.x	age.y
1	Ally	45	45
2	Steve	46	*NA*
3	John	47	47

2. RIGHT JOIN



```
#11b. Right Join
right_join(df1,df2,by='name')
```

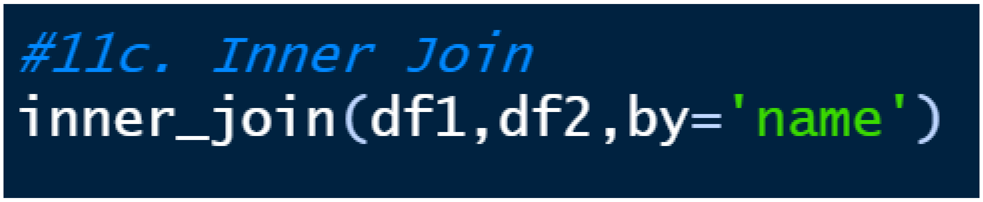


	name	age
1	Ally	45
2	Steve	46
3	John	47

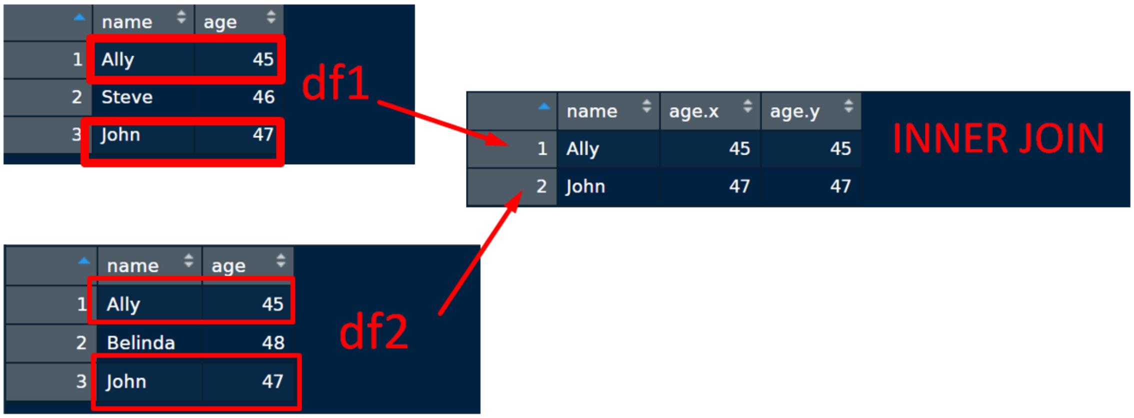
	name	age
1	Ally	45
2	Belinda	48
3	John	47

	name	age.x	age.y
1	Ally	45	45
2	John	47	47
3	Belinda	*NA*	48

3. INNER JOIN



```
#11c. Inner Join
inner_join(df1,df2,by='name')
```

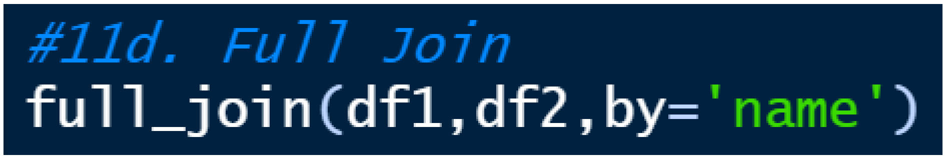


	name	age
1	Ally	45
2	Steve	46
3	John	47

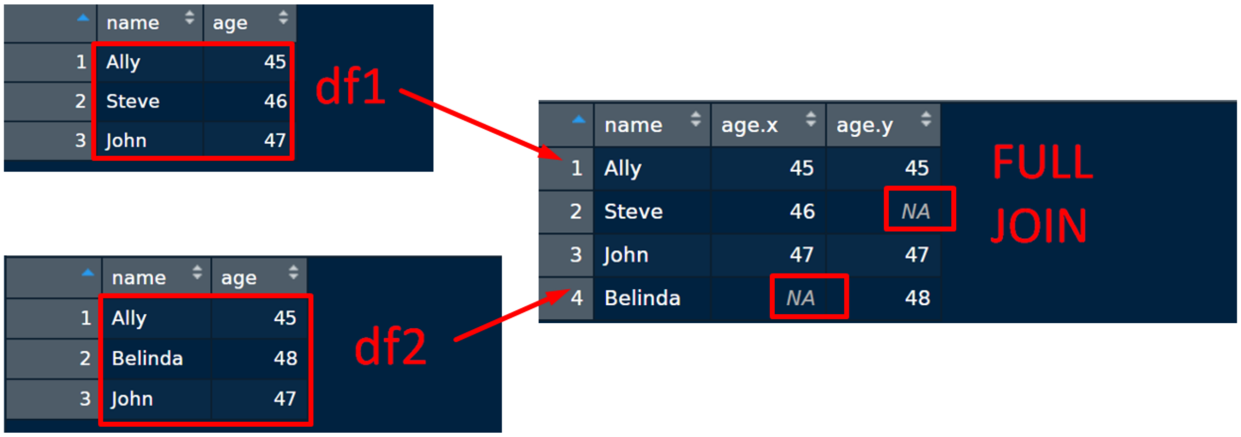
	name	age.x	age.y
1	Ally	45	45
2	John	47	47

	name	age
1	Ally	45
2	Belinda	48
3	John	47

4. FULL JOIN



```
#11d. Full Join
full_join(df1,df2,by='name')
```

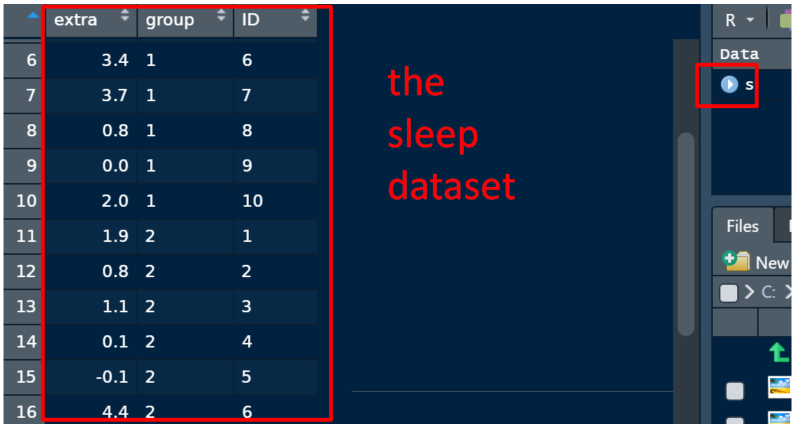


	name	age
1	Ally	45
2	Steve	46
3	John	47

	name	age
1	Ally	45
2	Belinda	48
3	John	47

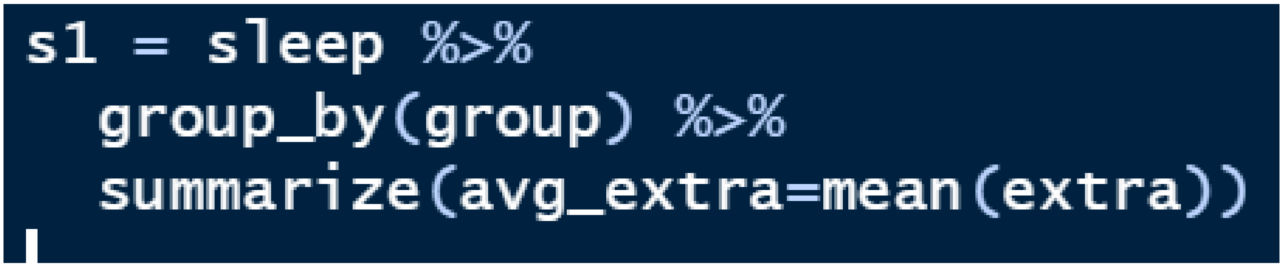
	name	age.x	age.y
1	Ally	45	45
2	Steve	46	NA
3	John	47	47
4	Belinda	NA	48

J. GROUPBY

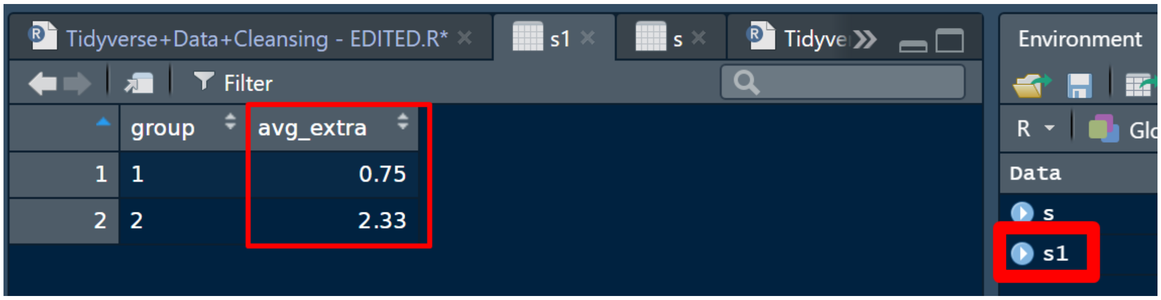


	extra	group	ID
6	3.4	1	6
7	3.7	1	7
8	0.8	1	8
9	0.0	1	9
10	2.0	1	10
11	1.9	2	1
12	0.8	2	2
13	1.1	2	3
14	0.1	2	4
15	-0.1	2	5
16	4.4	2	6

the sleep dataset

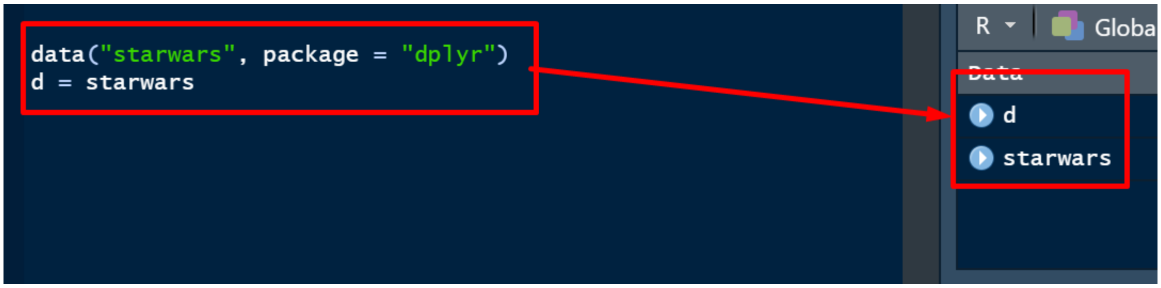


```
s1 = sleep %>%
  group_by(group) %>%
  summarize(avg_extra=mean(extra))
```

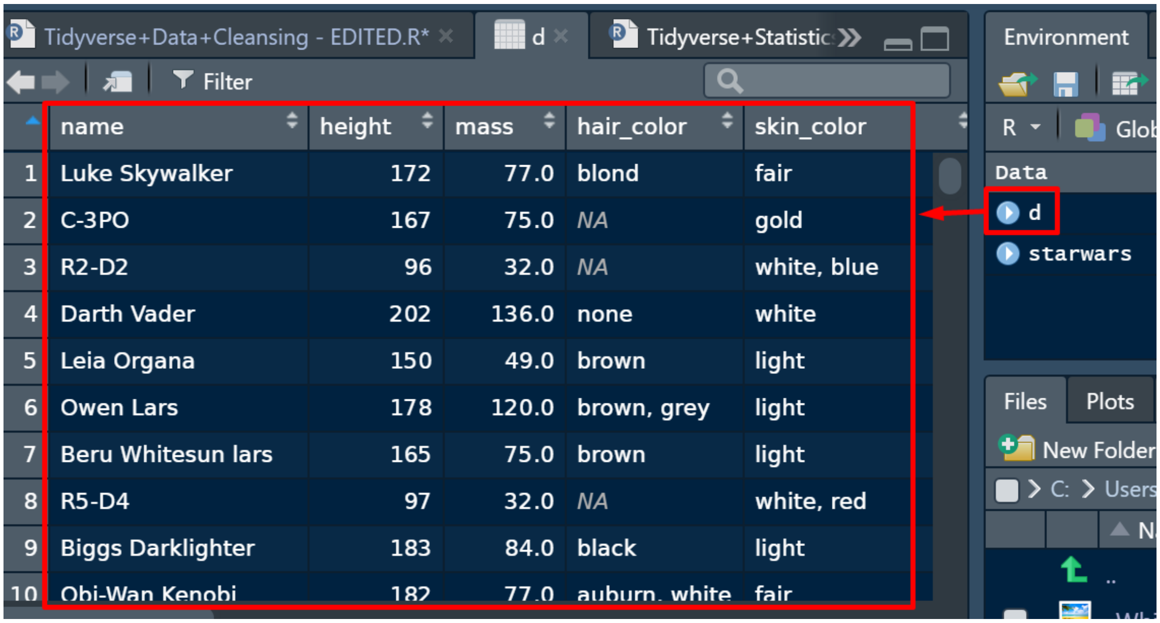


	group	avg_extra
1	1	0.75
2	2	2.33

K. REMOVING COLUMN

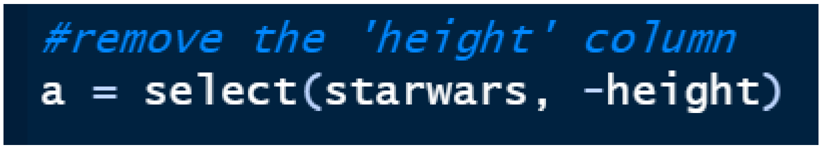


```
data("starwars", package = "dplyr")
d = starwars
```

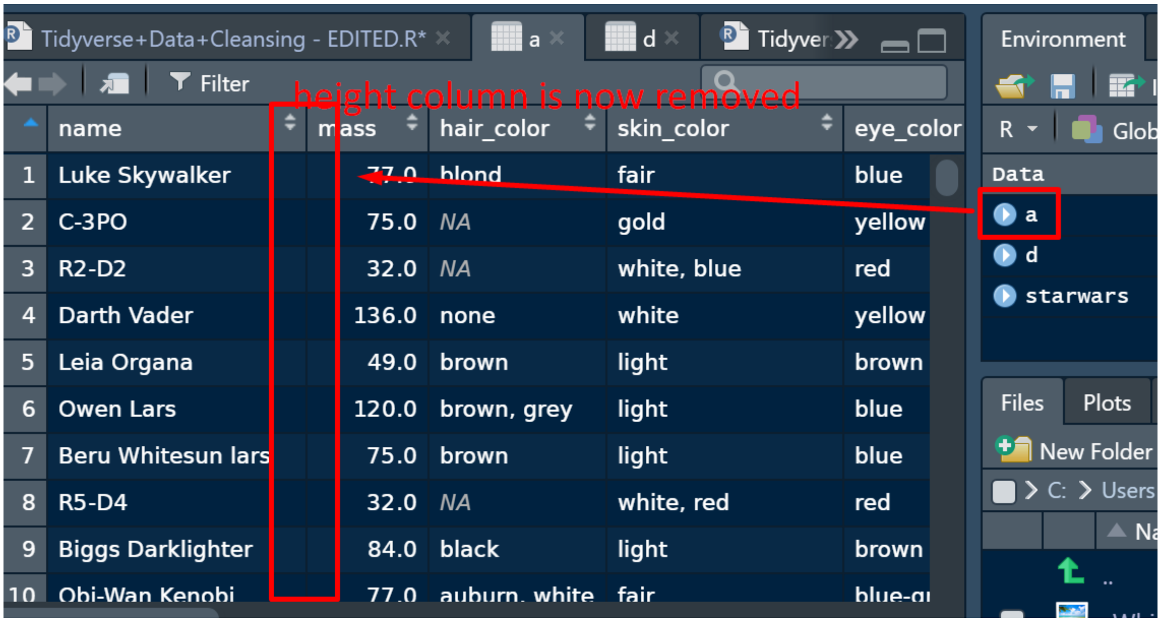


	name	height	mass	hair_color	skin_color
1	Luke Skywalker	172	77.0	blond	fair
2	C-3PO	167	75.0	NA	gold
3	R2-D2	96	32.0	NA	white, blue
4	Darth Vader	202	136.0	none	white
5	Leia Organa	150	49.0	brown	light
6	Owen Lars	178	120.0	brown, grey	light
7	Beru Whitesun lars	165	75.0	brown	light
8	R5-D4	97	32.0	NA	white, red
9	Biggs Darklighter	183	84.0	black	light
10	Obi-Wan Kenobi	182	77.0	auburn, white	fair

1. REMOVING THE ‘HEIGHT’ COLUMN



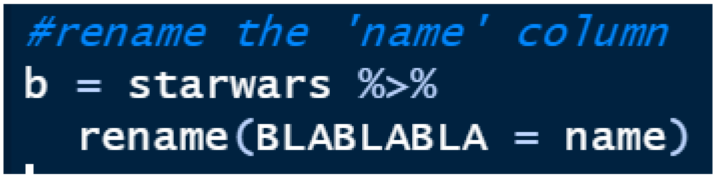
```r
#remove the 'height' column
a = select(starwars, -height)
```



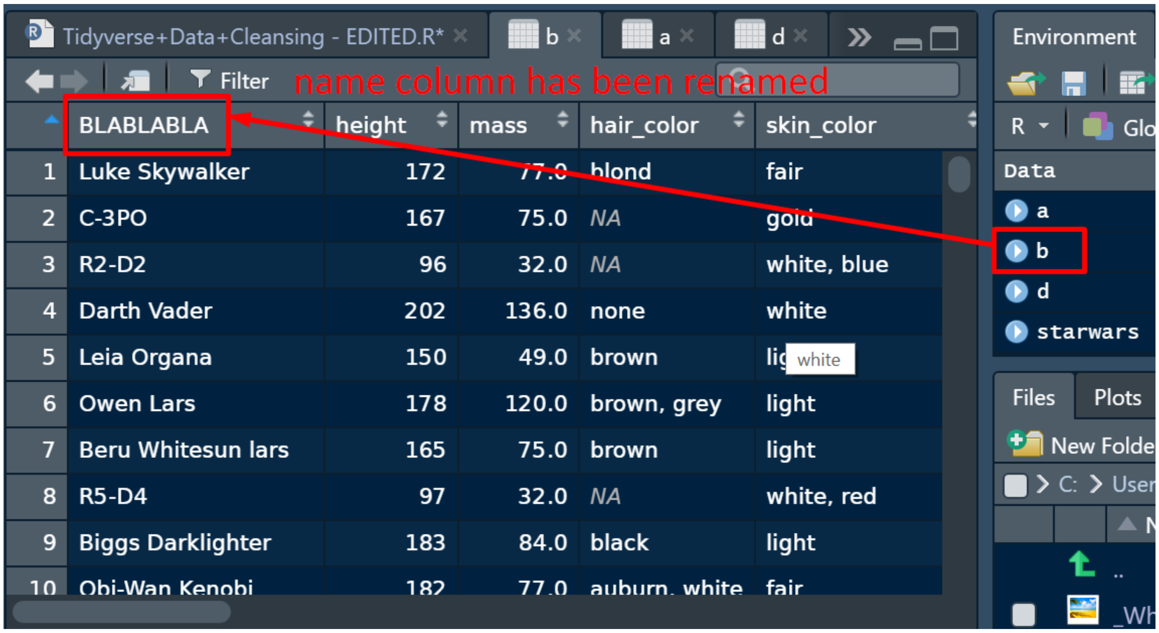
height column is now removed

	name	mass	hair_color	skin_color	eye_color
1	Luke Skywalker	77.0	blond	fair	blue
2	C-3PO	75.0	NA	gold	yellow
3	R2-D2	32.0	NA	white, blue	red
4	Darth Vader	136.0	none	white	yellow
5	Leia Organa	49.0	brown	light	brown
6	Owen Lars	120.0	brown, grey	light	blue
7	Beru Whitesun lars	75.0	brown	light	blue
8	R5-D4	32.0	NA	white, red	red
9	Biggs Darklighter	84.0	black	light	brown
10	Obi-Wan Kenobi	77.0	auburn, white	fair	blue-gı

2. RENAME THE 'NAME' COLUMN



```
#rename the 'name' column
b = starwars %>%
  rename(BLABLABLA = name)
```

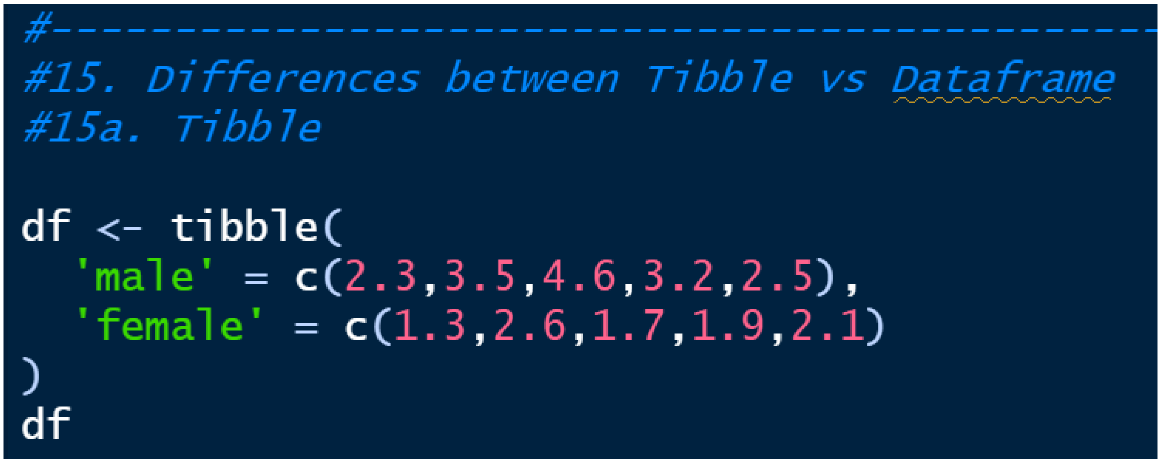


name column has been renamed

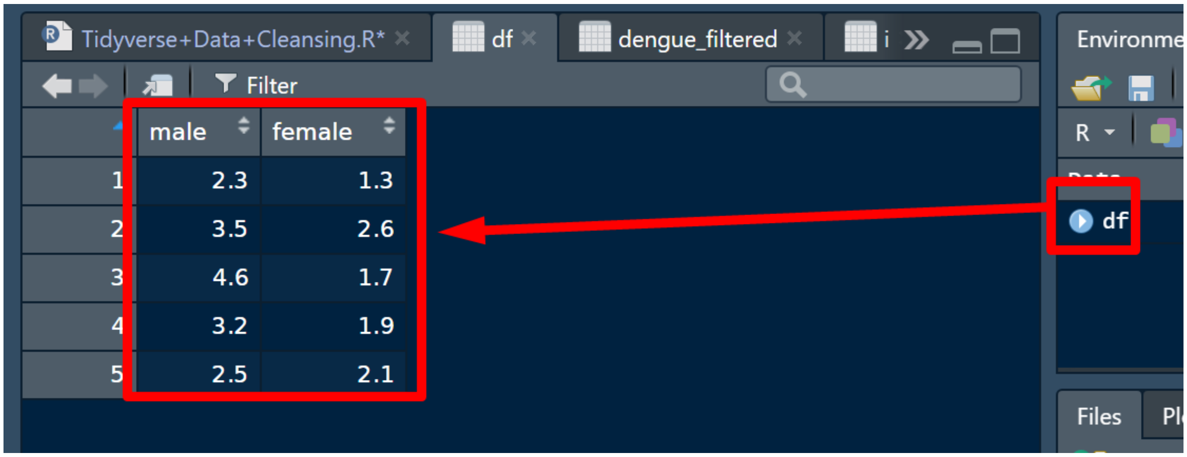
	BLABLABLA	height	mass	hair_color	skin_color
1	Luke Skywalker	172	77.0	blond	fair
2	C-3PO	167	75.0	NA	gold
3	R2-D2	96	32.0	NA	white, blue
4	Darth Vader	202	136.0	none	white
5	Leia Organa	150	49.0	brown	lig white
6	Owen Lars	178	120.0	brown, grey	light
7	Beru Whitesun lars	165	75.0	brown	light
8	R5-D4	97	32.0	NA	white, red
9	Biggs Darklighter	183	84.0	black	light
10	Obi-Wan Kenobi	182	77.0	auburn, white	fair

L. DIFFERENCES BETWEEN TIBBLE VS DATAFRAME

1. TIBBLE

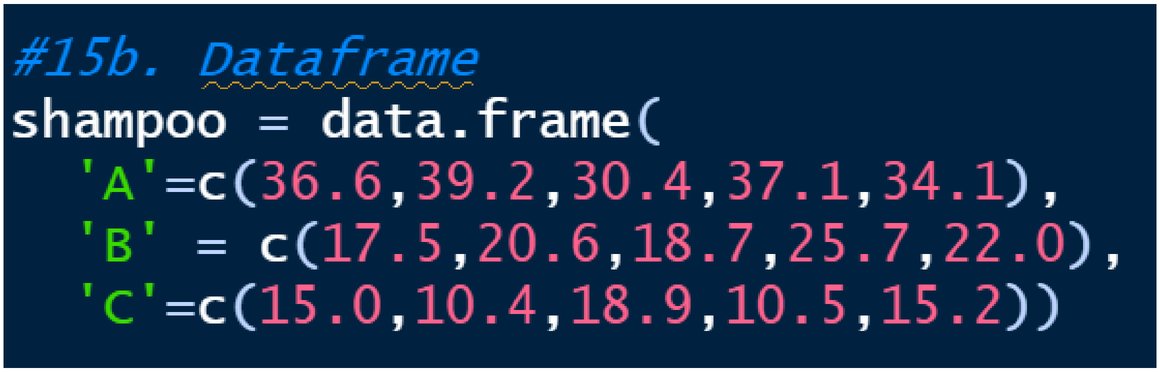


```
#-----------------------------------------------
#15. Differences between Tibble vs Dataframe
#15a. Tibble

df <- tibble(
  'male' = c(2.3,3.5,4.6,3.2,2.5),
  'female' = c(1.3,2.6,1.7,1.9,2.1)
)
df
```

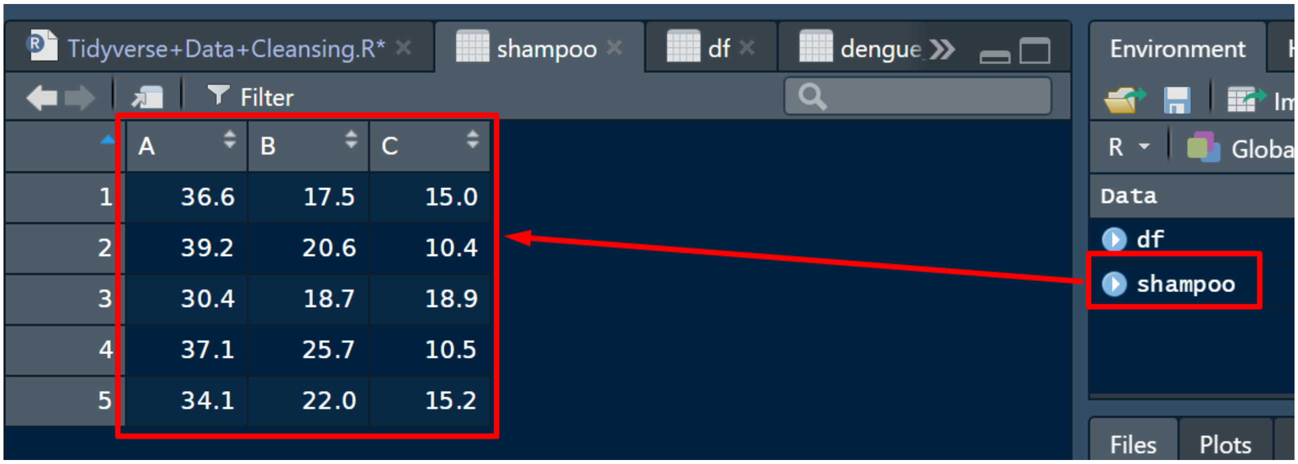


	male	female
1	2.3	1.3
2	3.5	2.6
3	4.6	1.7
4	3.2	1.9
5	2.5	2.1

2. DATAFRAME

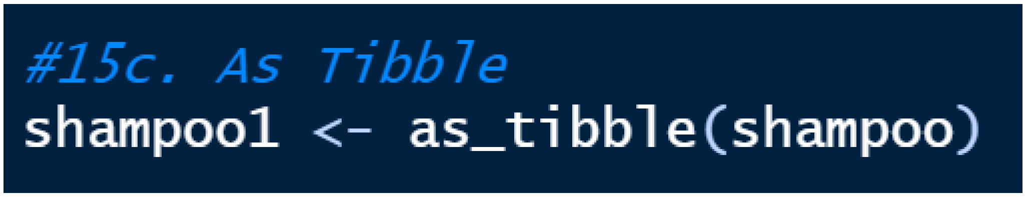


```
#15b. Dataframe
shampoo = data.frame(
  'A'=c(36.6,39.2,30.4,37.1,34.1),
  'B' = c(17.5,20.6,18.7,25.7,22.0),
  'C'=c(15.0,10.4,18.9,10.5,15.2))
```

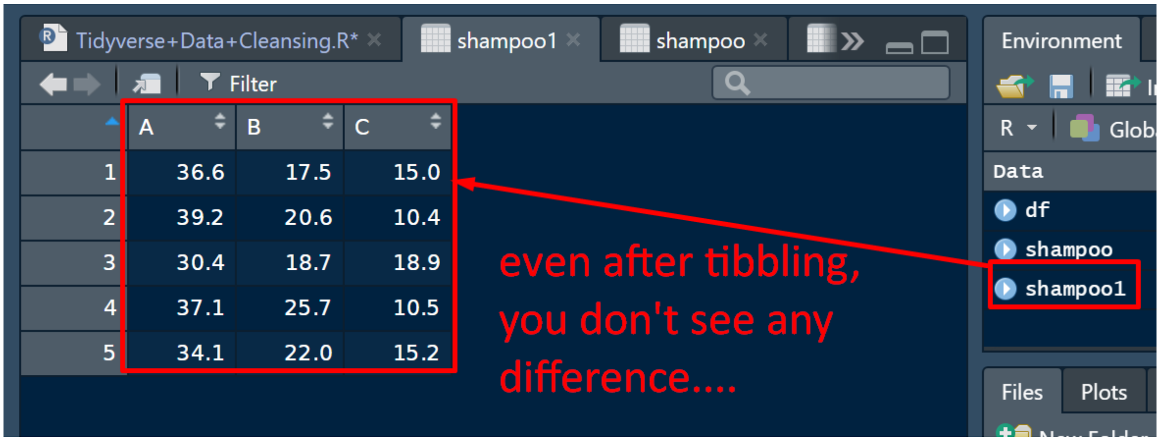


	A	B	C
1	36.6	17.5	15.0
2	39.2	20.6	10.4
3	30.4	18.7	18.9
4	37.1	25.7	10.5
5	34.1	22.0	15.2

3. AS TIBBLE



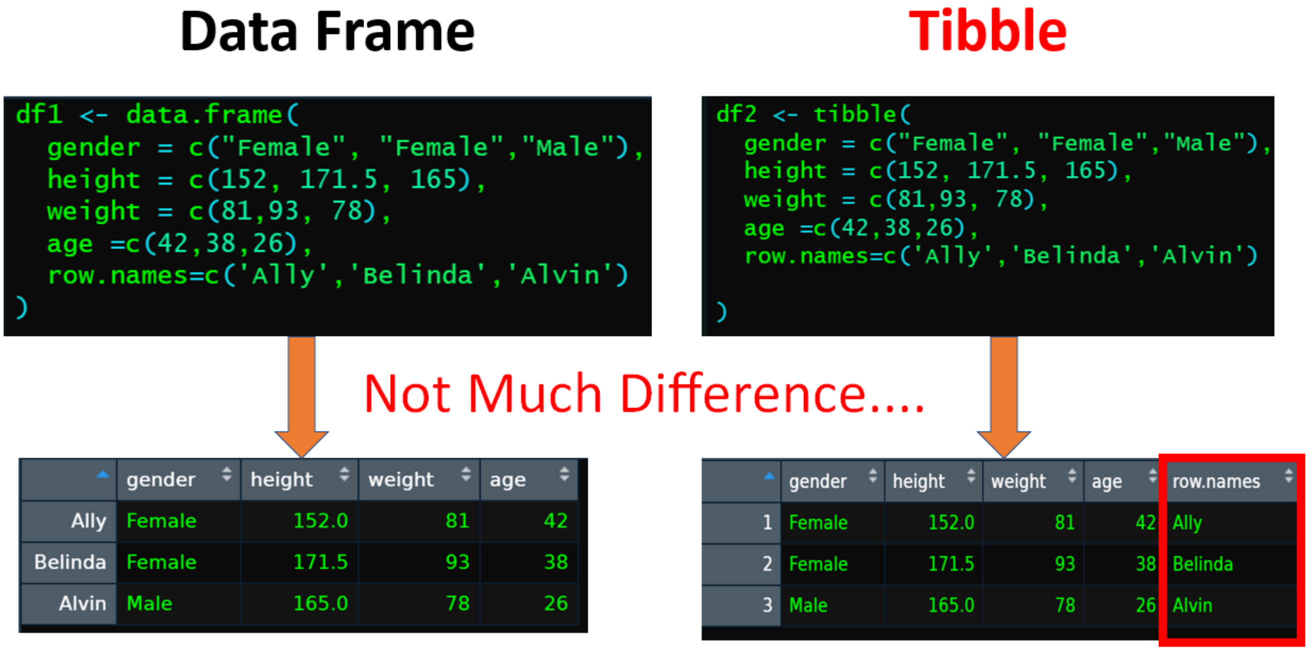
```
#15c. As Tibble
shampoo1 <- as_tibble(shampoo)
```



	A	B	C
1	36.6	17.5	15.0
2	39.2	20.6	10.4
3	30.4	18.7	18.9
4	37.1	25.7	10.5
5	34.1	22.0	15.2

even after tibbling, you don't see any difference....

4. COMPARING CONVERSION



Data Frame

```
df1 <- data.frame(
  gender = c("Female", "Female","Male"),
  height = c(152, 171.5, 165),
  weight = c(81,93, 78),
  age =c(42,38,26),
  row.names=c('Ally','Belinda','Alvin')
)
```

Tibble

```
df2 <- tibble(
  gender = c("Female", "Female","Male"),
  height = c(152, 171.5, 165),
  weight = c(81,93, 78),
  age =c(42,38,26),
  row.names=c('Ally','Belinda','Alvin')

)
```

Not Much Difference....

	gender	height	weight	age
Ally	Female	152.0	81	42
Belinda	Female	171.5	93	38
Alvin	Male	165.0	78	26

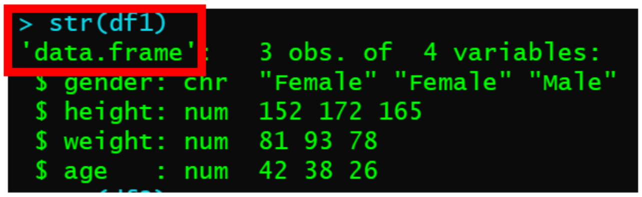
	gender	height	weight	age	row.names
1	Female	152.0	81	42	Ally
2	Female	171.5	93	38	Belinda
3	Male	165.0	78	26	Alvin

- There’s not much visible difference between a Data Frame vs Tibble…..

- Except that Tibble adds an extra column….

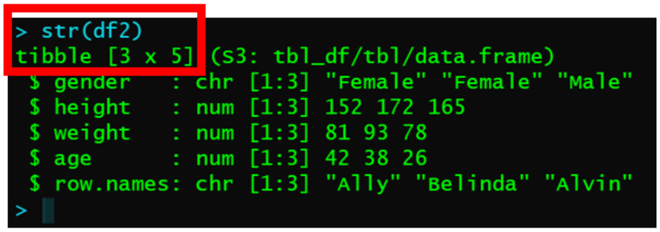
5. COMPARING STRUCTURE (STR)

Data Frame



```
> str(df1)
'data.frame':	3 obs. of  4 variables:
 $ gender: chr  "Female" "Female" "Male"
 $ height: num  152 172 165
 $ weight: num  81 93 78
 $ age   : num  42 38 26
```

Tibble



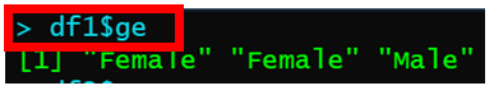
```
> str(df2)
tibble [3 x 5] (S3: tbl_df/tbl/data.frame)
 $ gender   : chr [1:3] "Female" "Female" "Male"
 $ height   : num [1:3] 152 172 165
 $ weight   : num [1:3] 81 93 78
 $ age      : num [1:3] 42 38 26
 $ row.names: chr [1:3] "Ally" "Belinda" "Alvin"
>
```

The Structure of a Data Frame vs Tibble also don't show much difference....

- Even if you look at the strucure…they display the same things….

6. COMPARING RETRIEVING COLUMNS

Data Frame

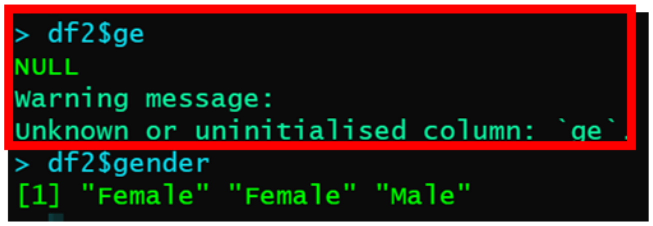


```
> df1$ge
[1] "Female" "Female" "Male"
```

Even though the proper column name
Is called "gender"….. If you use a DataFrame,
You can misspell it as $ge and it will still
Show the column items….

This might cause future errors if you accidentally
Call out the wrong column with similar column
"ge" headings…..

Tibble



```
> df2$ge
NULL
Warning message:
Unknown or uninitialised column: `ge`.
> df2$gender
[1] "Female" "Female" "Male"
```

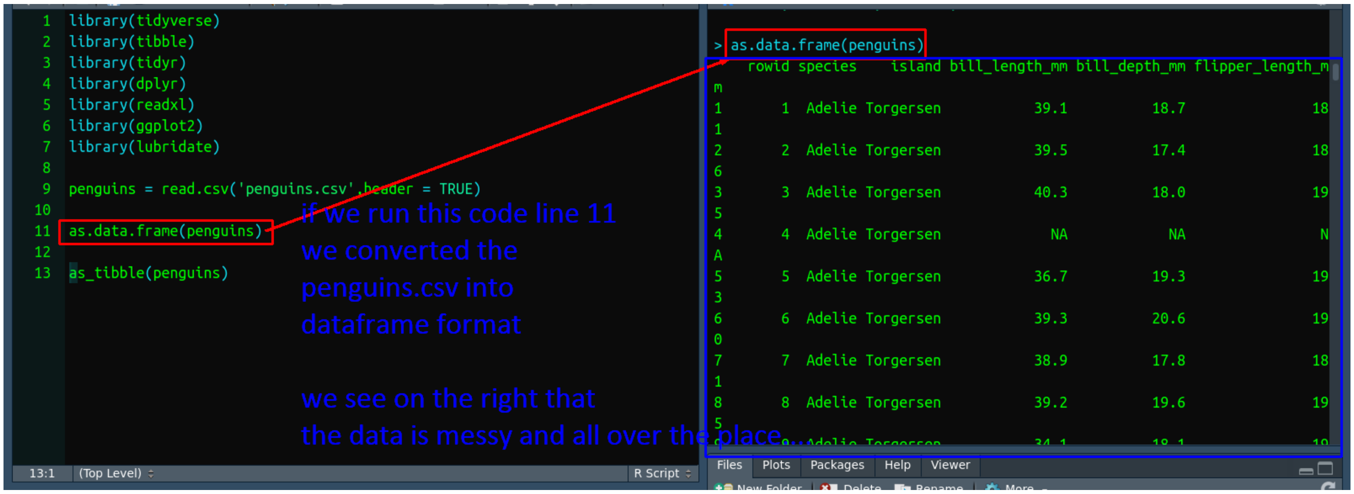
However, for Tibble, you are not able to
Display the column items if you misspell
The column name…..it will show an error…

You have to type out the whole "$gender"
To get the items…..

This prevents future errors…..

- But you are not able to use short forms for the column names to call out the items…

7. COMPARING DISPLAY



```
1  library(tidyverse)
2  library(tibble)
3  library(tidyr)
4  library(dplyr)
5  library(readxl)
6  library(ggplot2)
7  library(lubridate)
8
9  penguins = read.csv('penguins.csv',header = TRUE)
10
11 as.data.frame(penguins)
12
13 as_tibble(penguins)
```

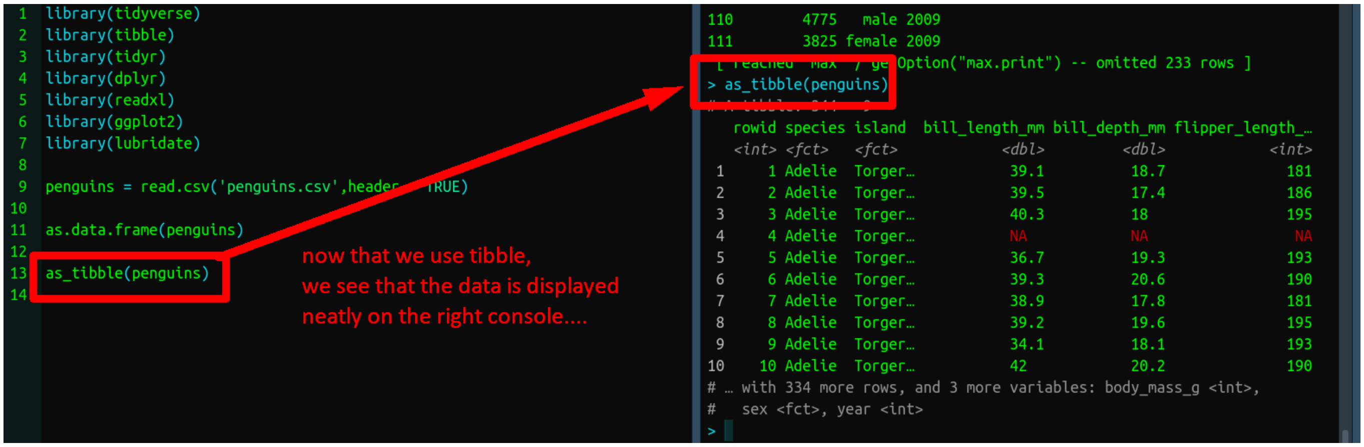
if we run this code line 11 we converted the penguins.csv into dataframe format

we see on the right that the data is messy and all over the place...

```
> as.data.frame(penguins)
   rowid species    island bill_length_mm bill_depth_mm flipper_length_m
m
1      1  Adelie Torgersen           39.1          18.7               18
1
2      2  Adelie Torgersen           39.5          17.4               18
6
3      3  Adelie Torgersen           40.3          18.0               19
5
4      4  Adelie Torgersen             NA            NA                N
A
5      5  Adelie Torgersen           36.7          19.3               19
3
6      6  Adelie Torgersen           39.3          20.6               19
0
7      7  Adelie Torgersen           38.9          17.8               18
1
8      8  Adelie Torgersen           39.2          19.6               19
5
9      9  Adelie Torgersen           34.1          18.1               19
```

- The Penguins dataset can be found here: https://www.alvinang.sg/s/penguins.csv

- For dataframe as shown above, we see that it displays very messily in the console….



```
library(tidyverse)
library(tibble)
library(tidyr)
library(dplyr)
library(readxl)
library(ggplot2)
library(lubridate)

penguins = read.csv('penguins.csv',header = TRUE)

as.data.frame(penguins)

as_tibble(penguins)
```

now that we use tibble, we see that the data is displayed neatly on the right console....

```
110         4775   male 2009
111         3825 female 2009
 [ reached 'max' / getOption("max.print") -- omitted 233 rows ]
> as_tibble(penguins)
# A tibble: 344 x 9
   rowid species island  bill_length_mm bill_depth_mm flipper_length_…
   <int> <fct>   <fct>            <dbl>         <dbl>            <int>
 1     1 Adelie  Torger…           39.1          18.7              181
 2     2 Adelie  Torger…           39.5          17.4              186
 3     3 Adelie  Torger…           40.3          18                195
 4     4 Adelie  Torger…           NA            NA                 NA
 5     5 Adelie  Torger…           36.7          19.3              193
 6     6 Adelie  Torger…           39.3          20.6              190
 7     7 Adelie  Torger…           38.9          17.8              181
 8     8 Adelie  Torger…           39.2          19.6              195
 9     9 Adelie  Torger…           34.1          18.1              193
10    10 Adelie  Torger…           42            20.2              190
# … with 334 more rows, and 3 more variables: body_mass_g <int>,
#   sex <fct>, year <int>
>
```

- For tibble, we now see that its displayed neatly….in the console….

```
#---------
#THE END
#---------
```

ABOUT THE AUTHOR

ABOUT DR. ALVIN ANG

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He was a previously a Professor, Scientist and Financial Consultant. Currently, he owns multiple self-started businesses and is a Personal/Business Advisor.

More about him at www.AlvinAng.sg