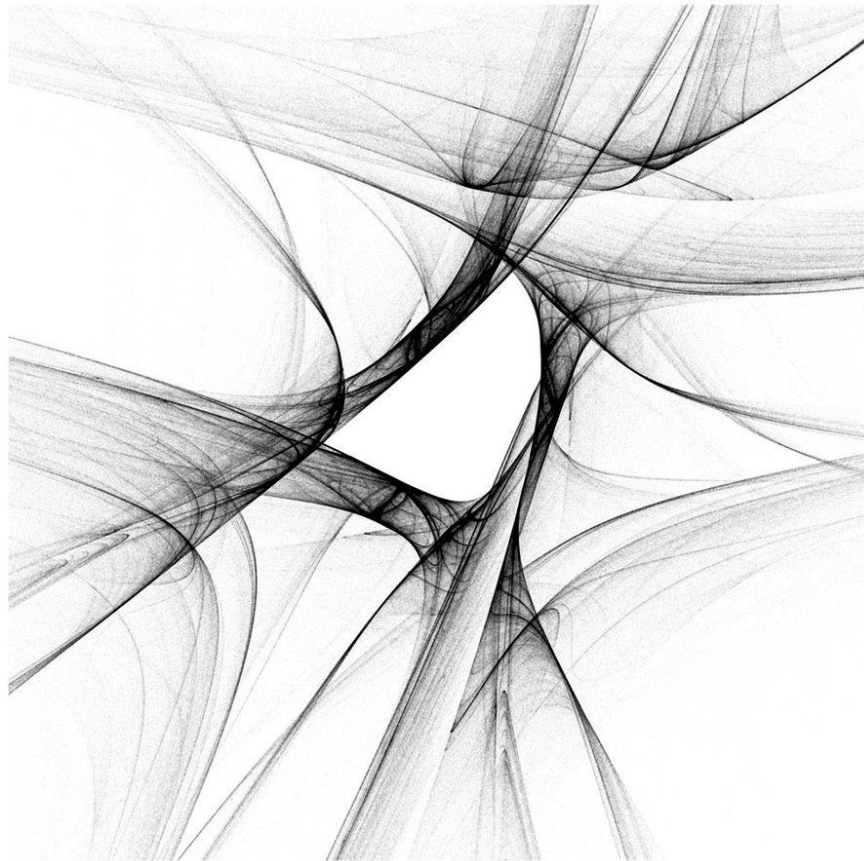


DR. ALVIN'S PUBLICATIONS

DATA WRANGLING WITH R

BY DR. ALVIN ANG



COPYRIGHTED BY DR ALVIN ANG
WWW.ALVINANG.SG

TABLE OF CONTENTS

Table of Contents.....2

I. Data Wrangling using Core R.....4

A. Set your Working Directory to your Downloads Folder4

1. Check Your Current Working Directory.....4

B. Wrangling Weather.csv5

1. Import CSV5

2. Slice Out Column using Subset.....6

3. Check Which Rows Have NAs.....7

4. Compute the Average Ozone Level in the Month of May8

5. Filter Out All NAs in the Month of May.....9

6. Output as CSV10

C. Wrangling Mtcars11

1. Slicing Out mpg / am / wt Columns11

2. Previewing Heads and Tails.....12

3. Slicing out mpg / hp columns.....13

4. Output as CSV14

5. Filter All the mpg > 15 and am = 115

6. Filter Out only mpg and am columns with am = 1 (automatic)16

7. Summary of Mtcars Subset.....17

8. Create a Table from Mtcars AM Columns.....18

II. Data Wrangling Using Tidyverse19

A. Installing Tidyverse Package19

B. Reading in CSV20

C. Selecting Columns.....22

D. Filtering Data23

1. Filter out 'Year' == 2018 from Dengue.csv.....23

E. Filter Data Based on Multiple Conditions24

1. Filter Out 2017 and 201824

2. Filter out 2018 and Dengue Type.....25

3. Another Way to Filter Out 2018 and Dengue Type.....26

F. Handling Missing Values in Dengue.csv Dataset27

1. Show All NAs in "number" column27

2. Another Way of Showing All NAs in All Columns27

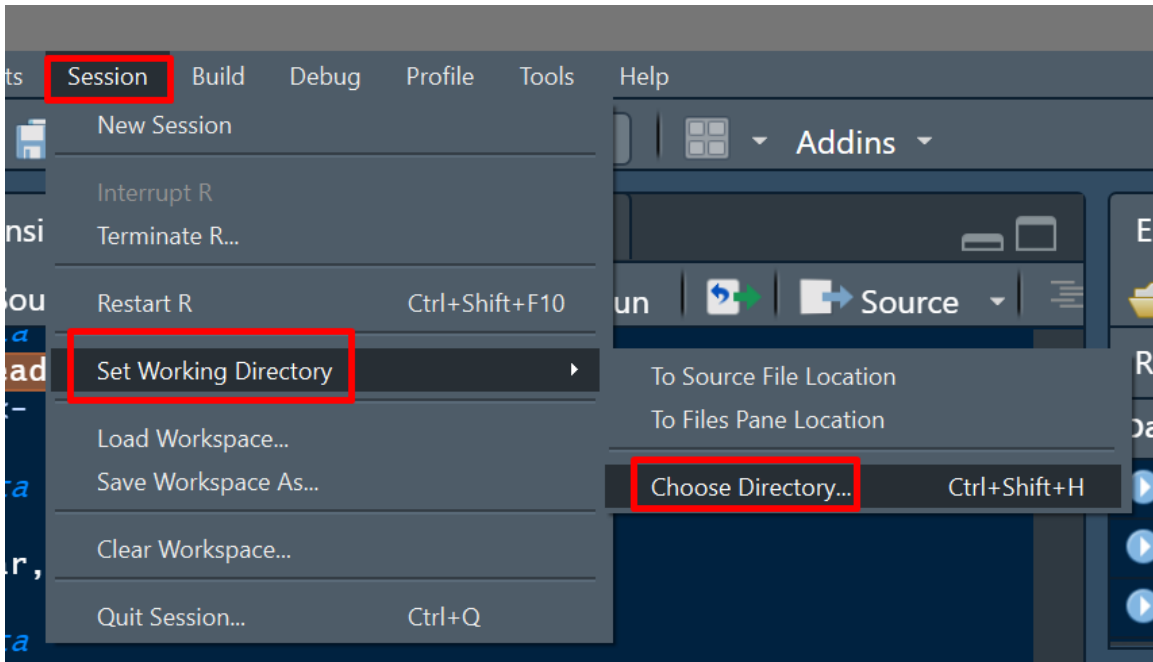
3. Showing All NO NAs (filled columns) now.....28

G. Mutate Data29

H. Filter, Mutate then Plot.....	30
1. Dengue.csv	30
2. Another Example for Filter, Mutate then Plot (Vaccination.xls)	31
3. Filter, Mutate then Export to CSV	32
4. Using Gather to Pivot DAta	33
I. Joins	34
1. Left Join	35
2. Right Join	36
3. Inner Join.....	37
4. Full Join	38
J. Groupby	39
K. Removing Column.....	40
1. Removing the 'height' Column.....	41
2. Rename the 'name' Column	42
L. Differences between Tibble vs Dataframe.....	43
1. Tibble	43
2. Dataframe	44
3. As Tibble.....	45
4. Comparing Conversion.....	46
5. Comparing Structure (STR).....	47
6. Comparing Retrieving Columns.....	48
7. Comparing Display	49
<i>About The Author</i>	50
About Dr. Alvin Ang.....	50

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()
```

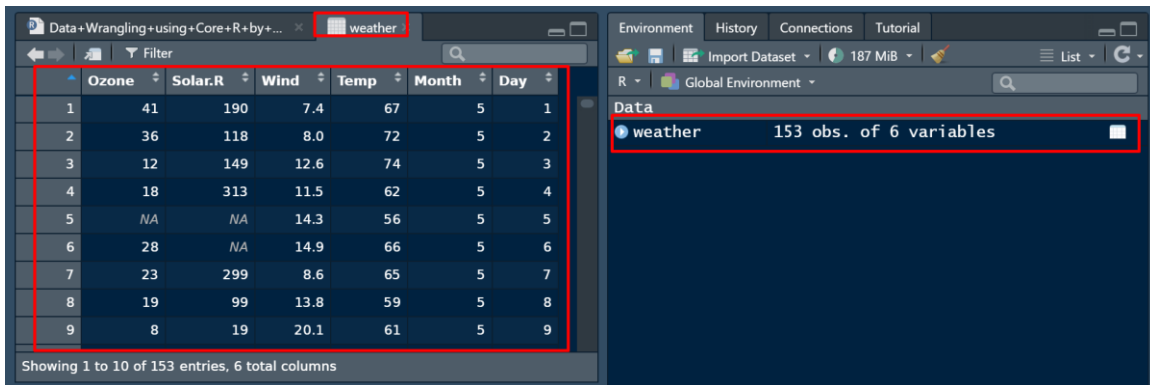
```
> getwd()  
[1] "C:/Users/User/Downloads"
```

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)
```

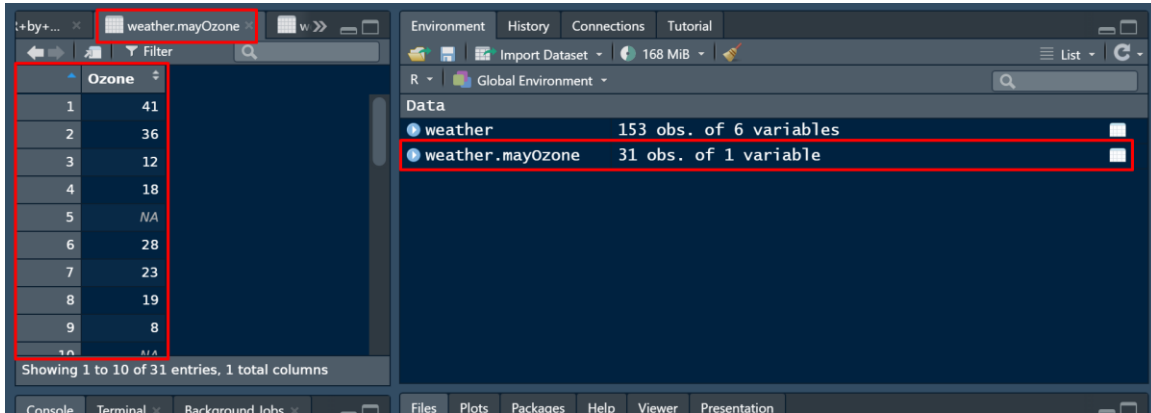


The screenshot shows the RStudio interface with the 'weather' dataset loaded. The table view displays the first 10 rows of data, and the Environment pane shows the dataset with 153 observations and 6 variables.

	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

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)
```



The screenshot displays the RStudio interface. The Environment pane on the right shows the variable 'weather.mayOzone' with 31 observations of 1 variable. The Data viewer on the left shows the first 10 rows of the 'Ozone' column.

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

3. CHECK WHICH ROWS HAVE NAs

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

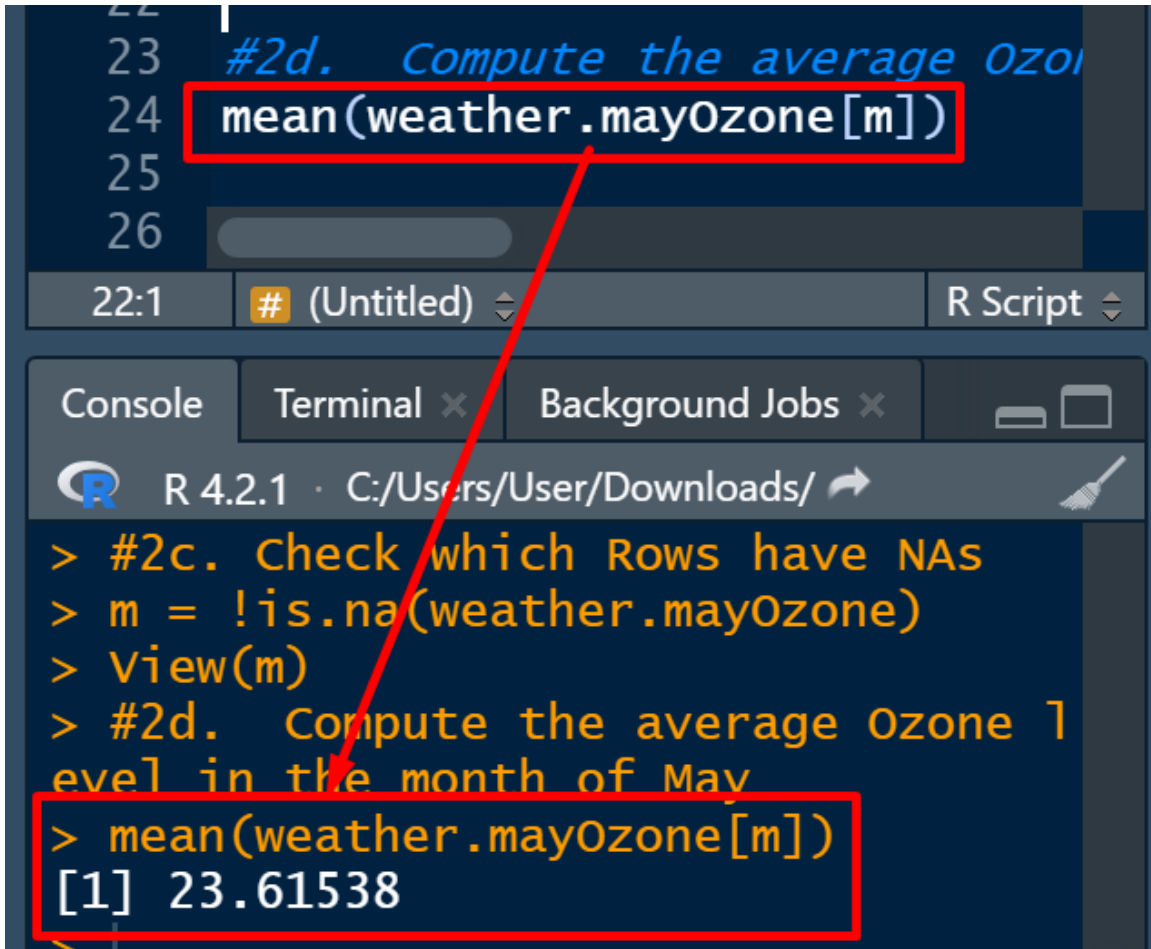
The screenshot shows the RStudio interface. The main window displays a data table with 10 rows and 1 column named 'Ozone'. The values are TRUE for rows 1-4, 6-9, and FALSE for rows 5 and 10. A red box highlights the first 10 rows. The Environment pane on the right shows the variable 'm' is defined. The Data pane shows the variable 'm' is defined.

	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])
```



```
23 #2d. Compute the average Ozone level in the month of May  
24 mean(weather.mayOzone[m])  
25  
26
```

22:1 # (Untitled) R Script

Console Terminal Background Jobs

R 4.2.1 · C:/Users/User/Downloads/

```
> #2c. Check which Rows have NAs  
> m = !is.na(weather.mayOzone)  
> View(m)  
> #2d. Compute the average Ozone level 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

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

may_weather_data

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')]
```

The screenshot shows the RStudio interface. The Environment pane on the right displays the variable 'b' with 32 observations of 3 variables. The Data pane on the left shows the first 11 rows of the sliced data, with columns 'mpg', 'am', and 'wt' highlighted. The status bar at the bottom indicates 'Showing 1 to 11 of 32 entries, 3 total columns'.

	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

2. PREVIEWING HEADS AND TAILS

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

```
38 #3b. Viewing Heads and Tails  
39 head(b,7)  
40 tail(b,3)  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
```

R 4.2.1 · C:/Users/User/Downloads/

```
> 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

```
38 #3b. Viewing Heads and Tails  
39 head(b,7)  
40 tail(b,3)  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
```

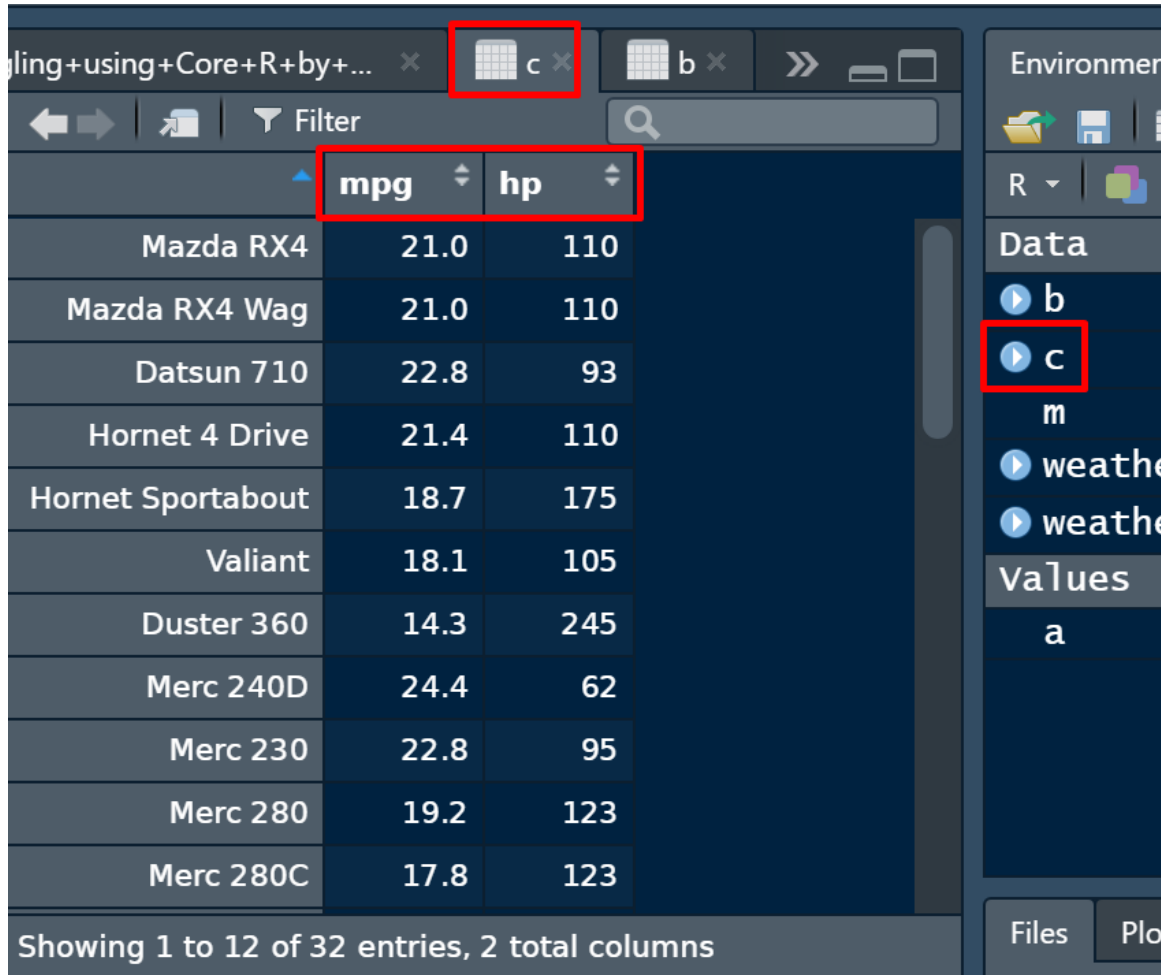
R 4.2.1 · C:/Users/User/Downloads/

```
> b = mtcars[c('mpg', 'am', 'wt')]  
> View(b)  
> #3b. Viewing Heads and Tails  
> head(b,7)  
> 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))
```



The screenshot shows the RStudio interface. The main window displays a data table with two columns, 'mpg' and 'hp', highlighted with red boxes. The 'Data' pane on the right shows a list of objects, with 'c' highlighted in red. The status bar at the bottom indicates 'Showing 1 to 12 of 32 entries, 2 total columns'.

	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

4. OUTPUT AS CSV

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

The screenshot shows the Microsoft Excel interface with the 'mtcarssubset - Excel' window title. The ribbon is set to 'Home'. A yellow warning banner at the top states 'POSSIBLE DATA LOSS Some features might be lost if you save this workb'. The active cell is A1. The following table is displayed in the spreadsheet:

	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,]
```

The screenshot shows the RStudio interface with a data frame 'd' displayed in the viewer. The data frame contains 10 rows of car data, filtered based on the conditions mpg > 15 and am == 1. The 'mpg' column is highlighted with a red box, and the 'am' column is also highlighted with a red box. The 'd' object is highlighted in the Environment pane on the right.

	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)
```

The screenshot shows the RStudio interface. The main window displays a data table with two columns: 'mpg' and 'am'. The rows list various car models and their corresponding mpg values and automatic transmission status (am). The 'mpg' and 'am' column headers are highlighted with a red box. In the Environment pane on the right, the object 'e' is highlighted with a red box. The status bar at the bottom indicates 'Showing 1 to 12 of 13 entries, 2 total columns'.

	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

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:

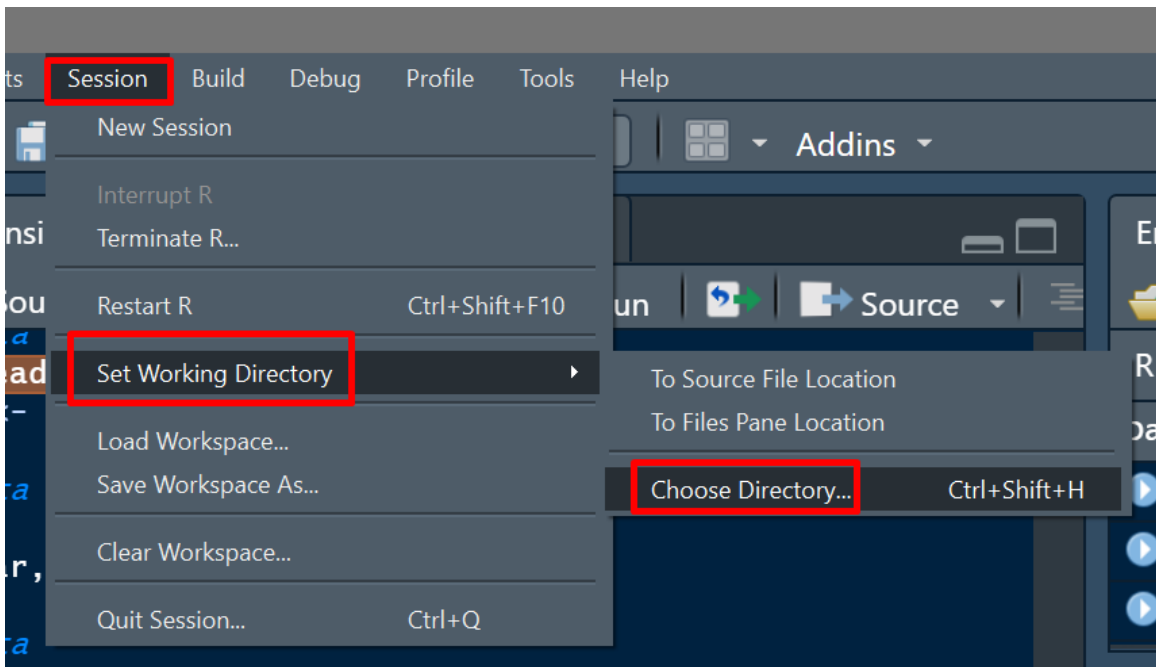
<https://www.marsja.se/how-to-rename-column-or-columns-in-r-with-dplyr/>

A. INSTALLING TIDYVERSE PACKAGE

```
#-----  
#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.

```
#-----  
#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")
```

The screenshot shows the RStudio interface. The Environment pane on the right displays the 'dengue' dataset with 530 observations and 4 variables. The Data pane on the left shows a preview of the first 6 rows of the dataset, which are highlighted with a red box. The columns are 'year', 'week', 'type_dengue', and 'number'.

	year	week	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

C. SELECTING COLUMNS

```
#-----  
#3. Selecting Columns  
  
# select 'year' and 'number' columns from dengue.csv  
a = dengue %>%  
  select(year,number)
```

The screenshot shows the RStudio interface. The main window displays a data frame with 6 rows and 2 columns: 'year' and 'number'. The data is as follows:

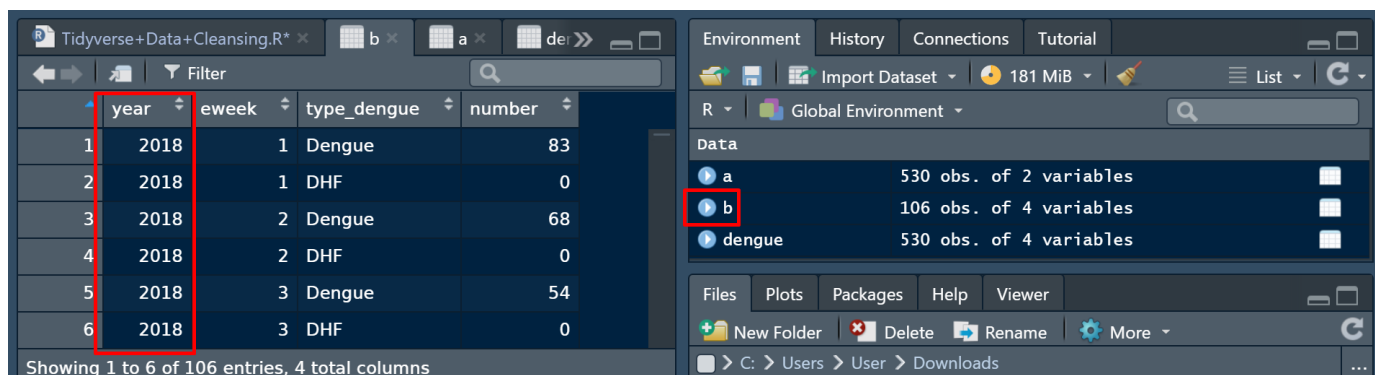
	year	number
1	2014	436
2	2014	1
3	2014	479
4	2014	0
5	2014	401
6	2014	0

The Environment pane on the right shows the 'Data' section with two objects: 'a' (530 obs. of 2 variables) and 'dengue' (530 obs. of 4 variables). A red box highlights the 'a' object, and a red arrow points from it to the data frame in the main window. The Files pane at the bottom shows the current directory as 'C:\Users\User\Downloads'.

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)
```



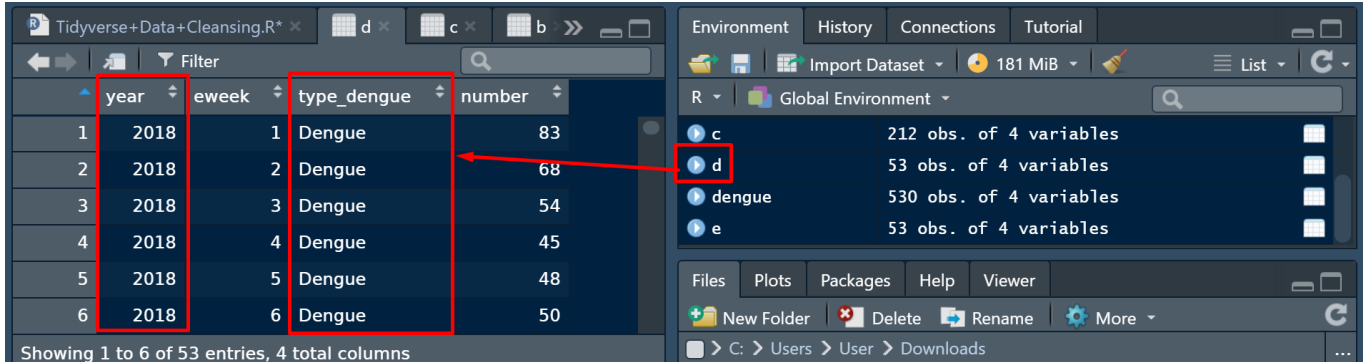
The screenshot displays the RStudio interface. The console on the left shows the R code used to filter the data. The Environment pane on the right shows the resulting data frame 'b' with 106 observations. The Data pane shows the structure of the data frame 'b'.

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

Showing 1 to 6 of 106 entries, 4 total columns

2. FILTER OUT 2018 AND DENGUE TYPE

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



The screenshot displays the RStudio interface. On the left, a data table is shown with columns: year, eweek, type_dengue, and number. The first six rows are highlighted with a red box, showing all '2018' and 'Dengue' entries. On the right, the Environment pane shows a list of objects: 'c' (212 obs.), 'd' (53 obs.), 'dengue' (530 obs.), and 'e' (53 obs.). The object 'd' is highlighted with a red box, and a red arrow points from it to the 'type_dengue' column in the table. The status bar at the bottom indicates 'Showing 1 to 6 of 53 entries, 4 total columns'.

	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

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')
```

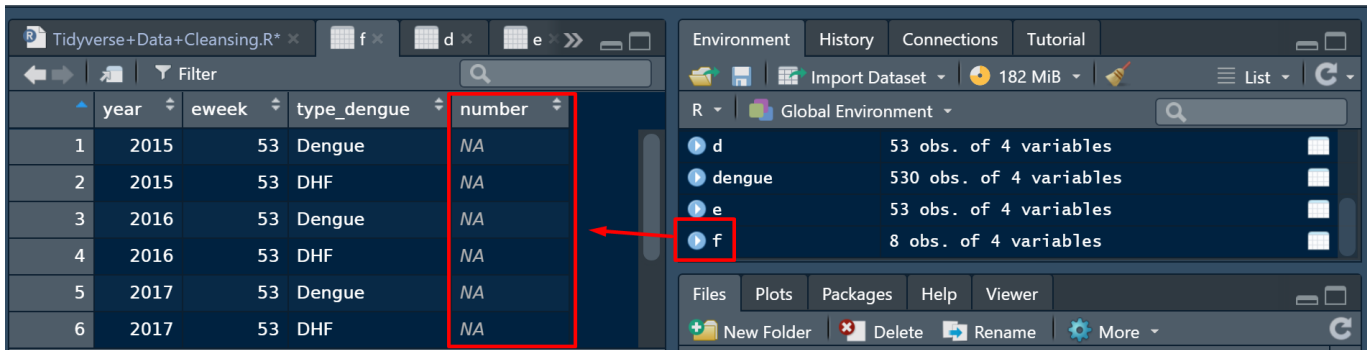
The screenshot displays the RStudio interface. On the left, a data table is shown with columns: year, eweek, type_dengue, and number. The table contains 6 rows of data for the year 2018, all of which are of the 'Dengue' type. A red box highlights the first three columns (year, eweek, type_dengue) for all rows. On the right, the Environment pane shows the Global Environment with four objects: 'c' (212 obs. of 4 variables), 'd' (53 obs. of 4 variables), 'dengue' (530 obs. of 4 variables), and 'e' (53 obs. of 4 variables). A red box highlights the 'e' object, and a red arrow points from this box to the 'type_dengue' column of the data table.

	year	ewek	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

F. HANDLING MISSING VALUES IN DENGUE.CSV DATASET

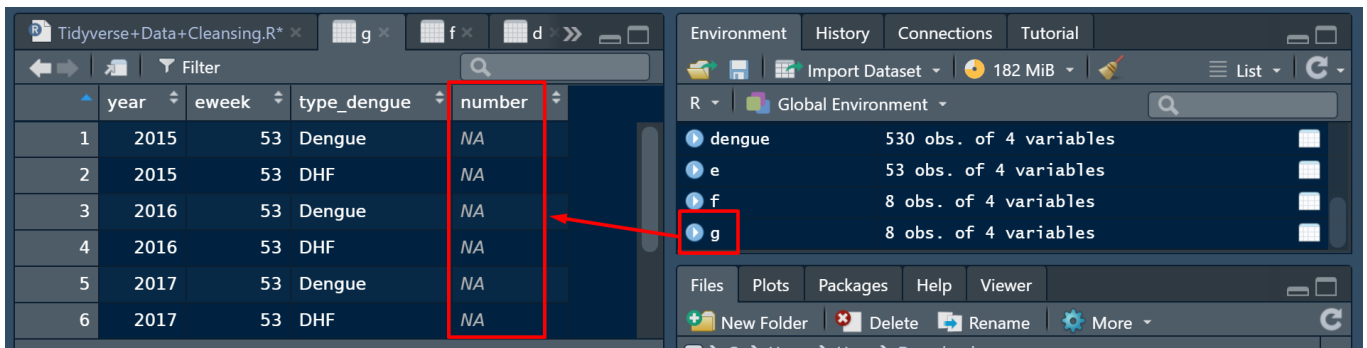
1. SHOW ALL NAs IN "NUMBER" COLUMN

```
#-----  
#6. Handling Missing values in Dengue.csv Dataset  
  
#6a. Show All NAs in "number" column  
f = dengue %>%  
  filter(is.na(number))
```



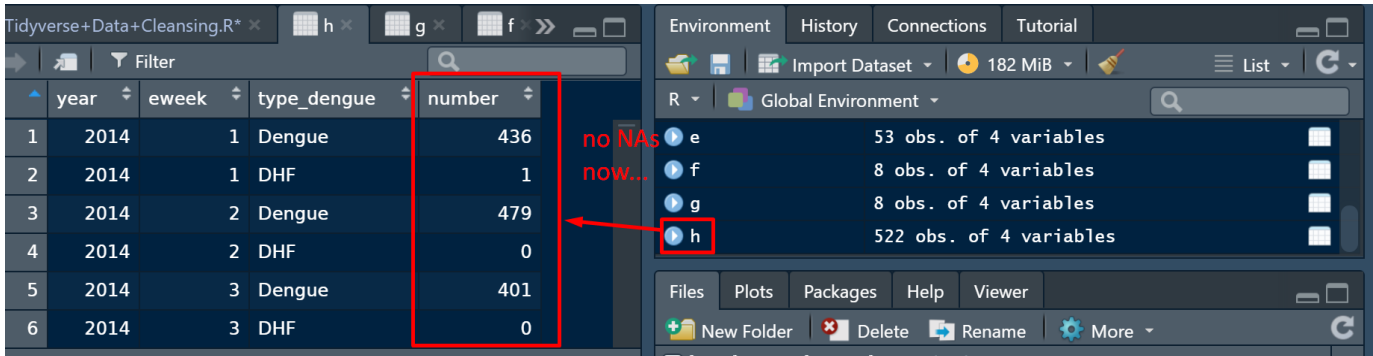
2. ANOTHER WAY OF SHOWING ALL NAs IN ALL COLUMNS

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



3. SHOWING ALL NO NAs (FILLED COLUMNS) NOW

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



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(eweeek))
```

using eweeek to create a new column called "date"

so 1+7 = 8

eweeek2 = 14

so 1+ 14 = 15

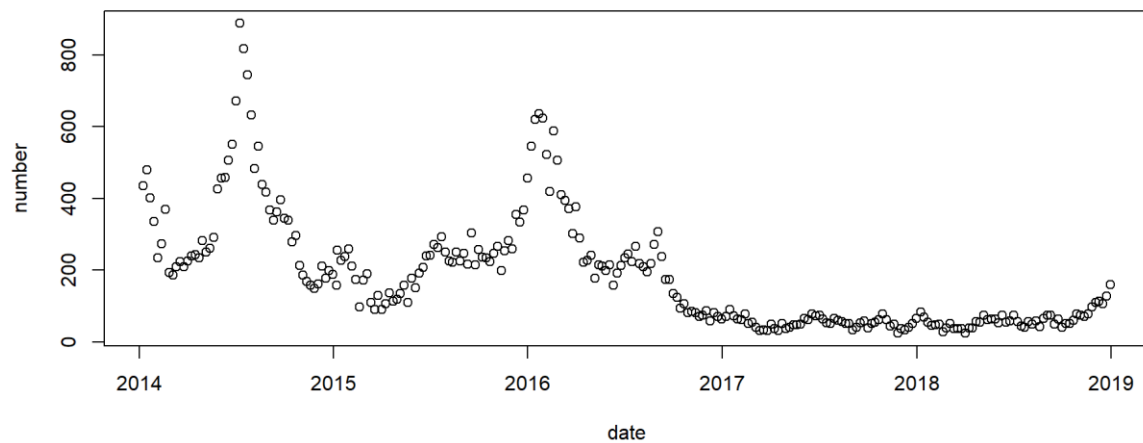
	year	eweeek	type_dengue	number	date
1	2014	1	Dengue	436	2014-01-08
2	2014	1	DHF	1	2014-01-08
3	2014	2	Dengue	479	2014-01-15
4	2014	2	DHF	0	2014-01-15
5	2014	3	Dengue	401	2014-01-22
6	2014	3	DHF	0	2014-01-22
7	2014	4	Dengue	336	2014-01-29

Showing 1 to 8 of 530 entries, 5 total columns

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(eweeek)) %>%  
  #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

```
# 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")
```

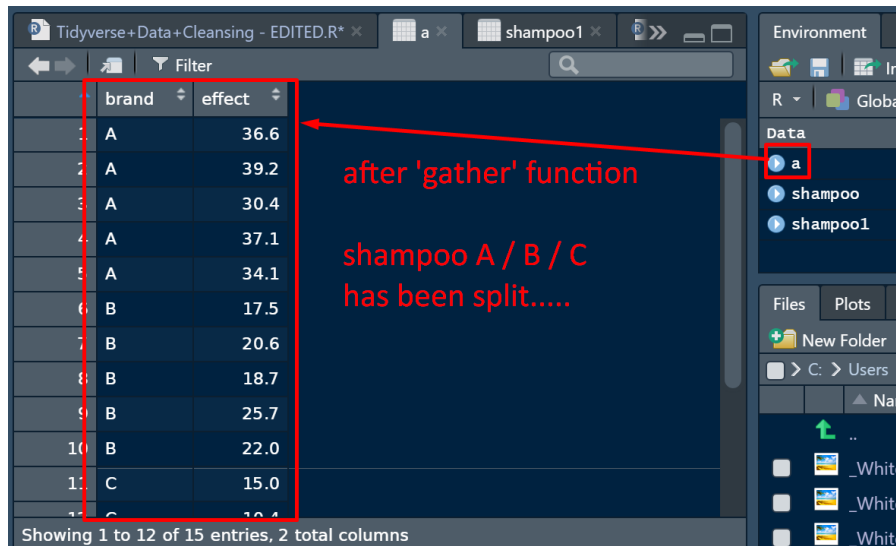
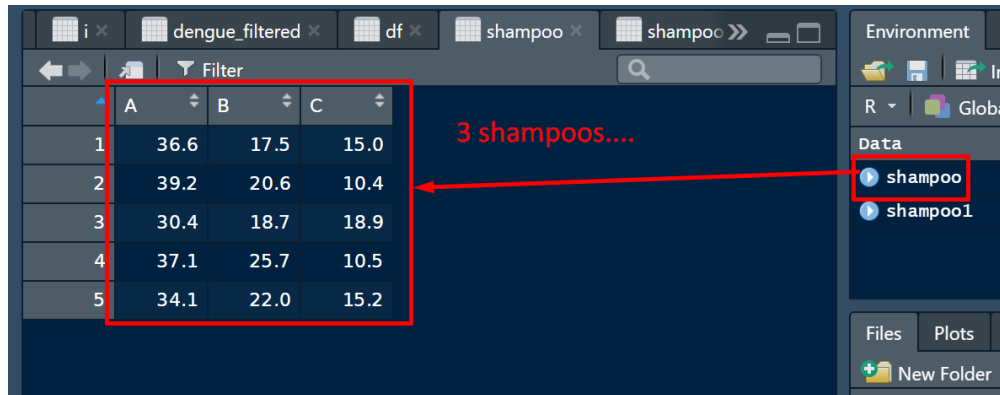
The screenshot shows the RStudio interface. On the left, the 'Environment' pane displays the 'dengue_filtered' data frame with 261 observations and 2 variables. Below it, the 'Files' pane shows the 'dengue_filtered.csv' file in the 'Downloads' folder. A red box highlights the 'dengue_filtered' data frame in the Environment pane and the 'dengue_filtered.csv' file in the Files pane. A red arrow points from the text 'this has now been created (write csv) into the working directory folder.....' to the 'dengue_filtered.csv' file. The console shows a warning message: 'Warning message: The path argument of write_csv() is deprecated as of readr 1.4.0.'

Environment	Size	Modified
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	

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

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)
```



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

df1

	name	age
1	Ally	45
2	Belinda	48
3	John	47

df2

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

LEFT JOIN

2. RIGHT JOIN

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

	name	age
1	Ally	45
2	Steve	46
3	John	47

df1

	name	age
1	Ally	45
2	Belinda	48
3	John	47

df2

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

RIGHT JOIN

3. INNER JOIN

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



4. FULL JOIN

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

	name	age
1	Ally	45
2	Steve	46
3	John	47

df1

	name	age
1	Ally	45
2	Belinda	48
3	John	47

df2

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

FULL
JOIN

J. GROUPBY

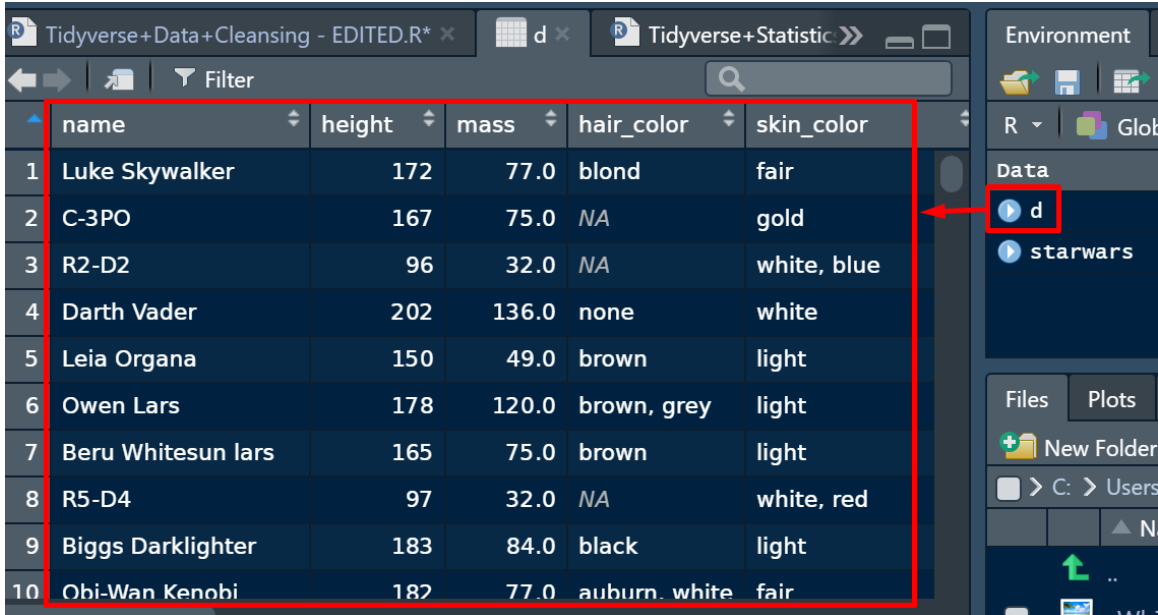
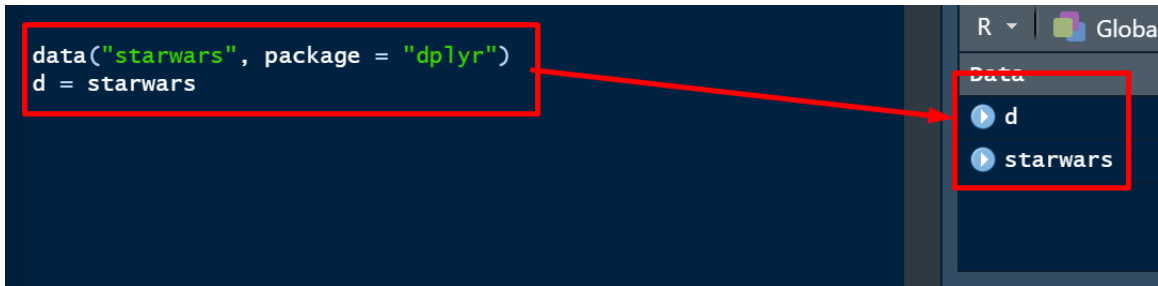
	extra	group	ID
6	3.4	1	6
7	3.7	1	7
8	0.8	1	8
9	0.0	1	9
10	2.0	1	10
11	1.9	2	1
12	0.8	2	2
13	1.1	2	3
14	0.1	2	4
15	-0.1	2	5
16	4.4	2	6

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

group	avg_extra
1	0.75
2	2.33

K. REMOVING COLUMN

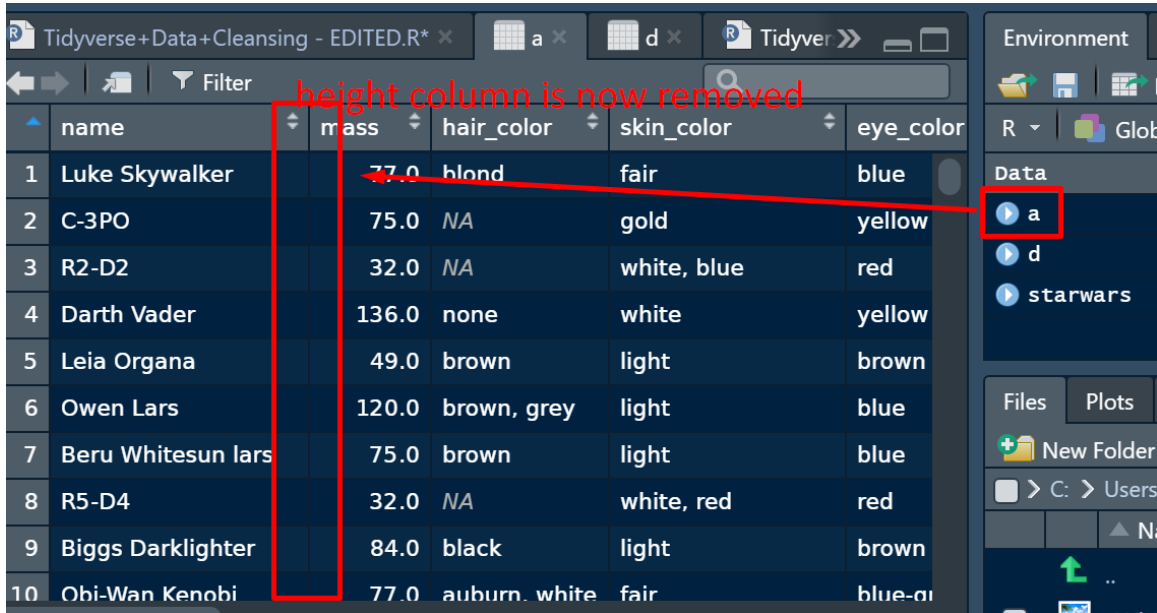
```
data("starwars", package = "dplyr")
d = starwars
```



	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

```
#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-gr

2. RENAME THE 'NAME' COLUMN

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

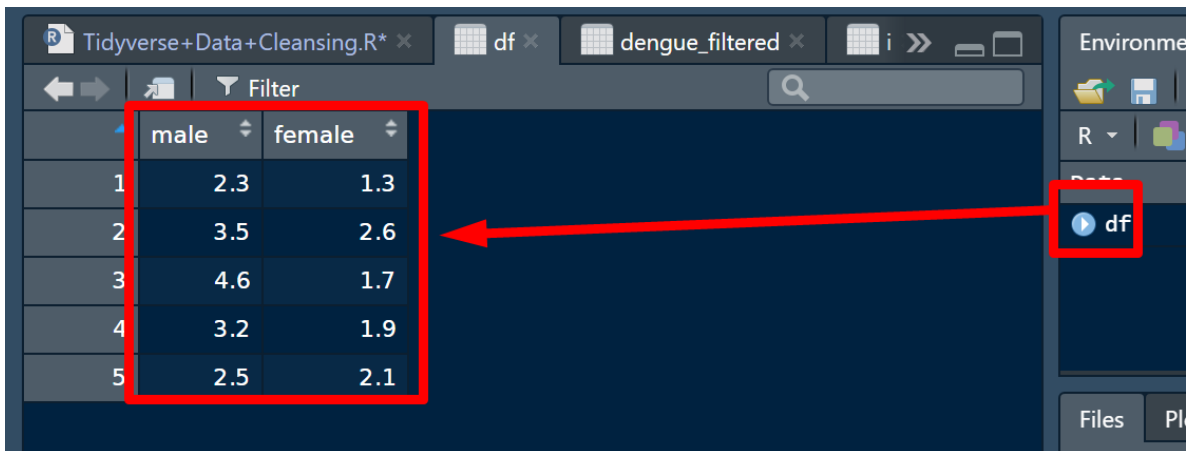
The screenshot shows the RStudio interface. The main window displays a data table with the following columns: BLABLABLA, height, mass, hair_color, and skin_color. The first row of data is: 1, Luke Skywalker, 172, 77.0, blond, fair. The Environment pane on the right shows a list of objects: a, b, d, and starwars. The object 'b' is highlighted with a red box. A red arrow points from the 'b' box to the 'BLABLABLA' column header. A red text overlay at the top of the table area reads "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	light skin 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
```



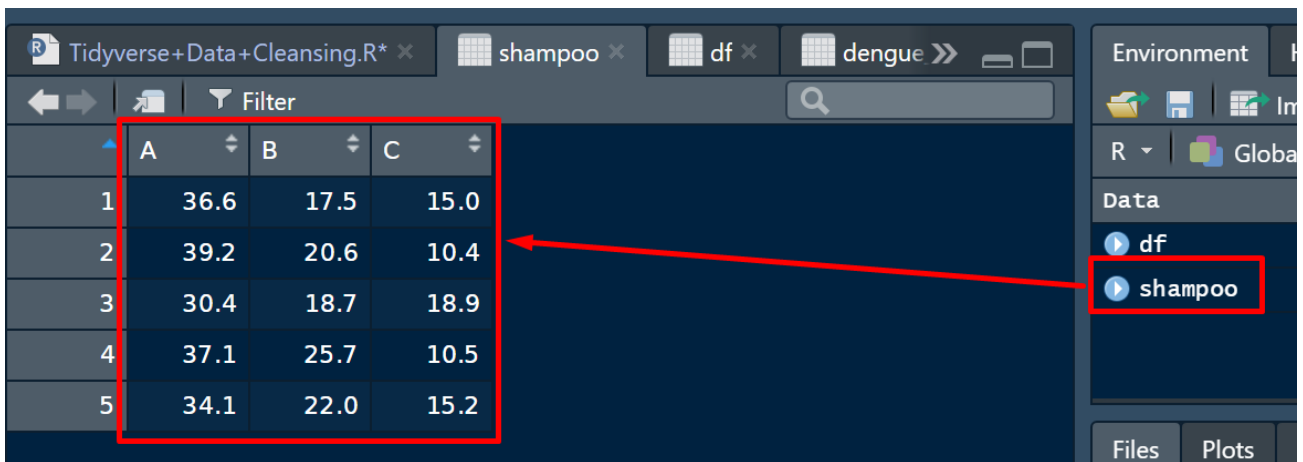
The screenshot shows the RStudio interface. The environment pane on the right contains a variable named 'df'. A red box highlights this variable. A red arrow points from this box to a data preview table in the center of the screen. The table has two columns, 'male' and 'female', and five rows of data.

	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))
```



The screenshot shows the RStudio interface with a data frame named 'shampoo' loaded. The data is displayed in a table view with columns A, B, and C. The values are as follows:

	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

The 'shampoo' object is highlighted in the Environment pane on the right side of the interface. A red arrow points from the 'shampoo' object in the Environment pane to the data table.

3. AS TIBBLE

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

The screenshot shows the RStudio interface. The main window displays a data frame with columns A, B, and C. The data is as follows:

	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

The Environment pane on the right shows the following objects:

- df
- shampoo
- shampoo1

A red box highlights the 'shampoo1' object in the Environment pane. A red arrow points from this box to the data frame in the main window. The text 'even after tibbling, you don't see any difference....' is written in red next to the arrow.

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')  
)
```

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

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')  
)
```

	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

Not Much Difference....

- 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 structure...they display the same things....

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
1         1  Adelle Torgersen      39.1          18.7             181
2         2  Adelle Torgersen      39.5          17.4             186
3         3  Adelle Torgersen      40.3           18              195
4         4  Adelle Torgersen      NA           NA                NA
5         5  Adelle Torgersen      36.7          19.3             193
6         6  Adelle Torgersen      39.3          20.6             190
7         7  Adelle Torgersen      38.9          17.8             181
8         8  Adelle Torgersen      39.2          19.6             195
9         9  Adelle Torgersen      34.1          18.1             193
10        10  Adelle Torgersen       42           20.2             190

```

- 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....

```

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)
14

```

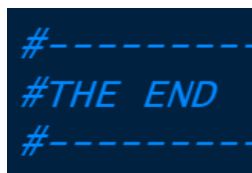
now that we use tibble, we see that the data is displayed neatly on the right console....

```

110      4775  male 2009
111      3825  female 2009
[ Reached max. # of options ("max.print") -- omitted 233 rows ]
> as_tibble(penguins)
# A tibble: 344 x 7
  rowid species island bill_length_mm bill_depth_mm flipper_length_mm
  <int> <fct>   <fct>         <dbl>         <dbl>         <int>
1     1  Adelle Torgersen 39.1           18.7           181
2     2  Adelle Torgersen 39.5           17.4           186
3     3  Adelle Torgersen 40.3            18            195
4     4  Adelle Torgersen NA              NA              NA
5     5  Adelle Torgersen 36.7           19.3           193
6     6  Adelle Torgersen 39.3           20.6           190
7     7  Adelle Torgersen 38.9           17.8           181
8     8  Adelle Torgersen 39.2           19.6           195
9     9  Adelle Torgersen 34.1           18.1           193
10    10  Adelle Torgersen 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....



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