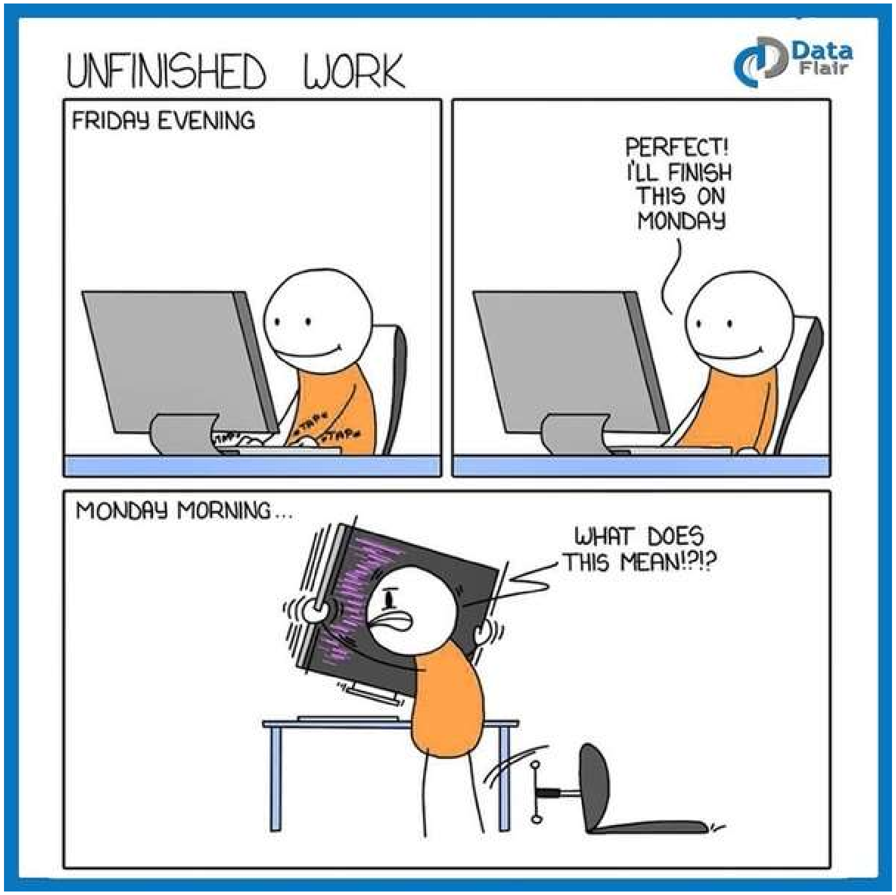
THE TIDYVERSE

Class 2: Data Transformation

Questions and recap

- Did you survive the exercises?
- Any comments on the teaching so far?
- What is a variable?
- What is a vector?
- What is a data.frame?
- Experimental Setup.
- What is a lurking variable?

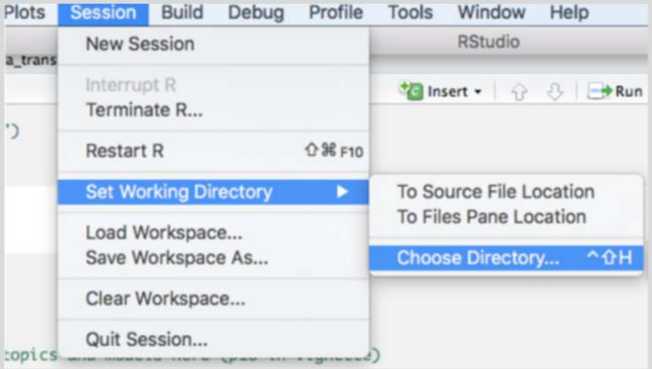


Let’s get some data going

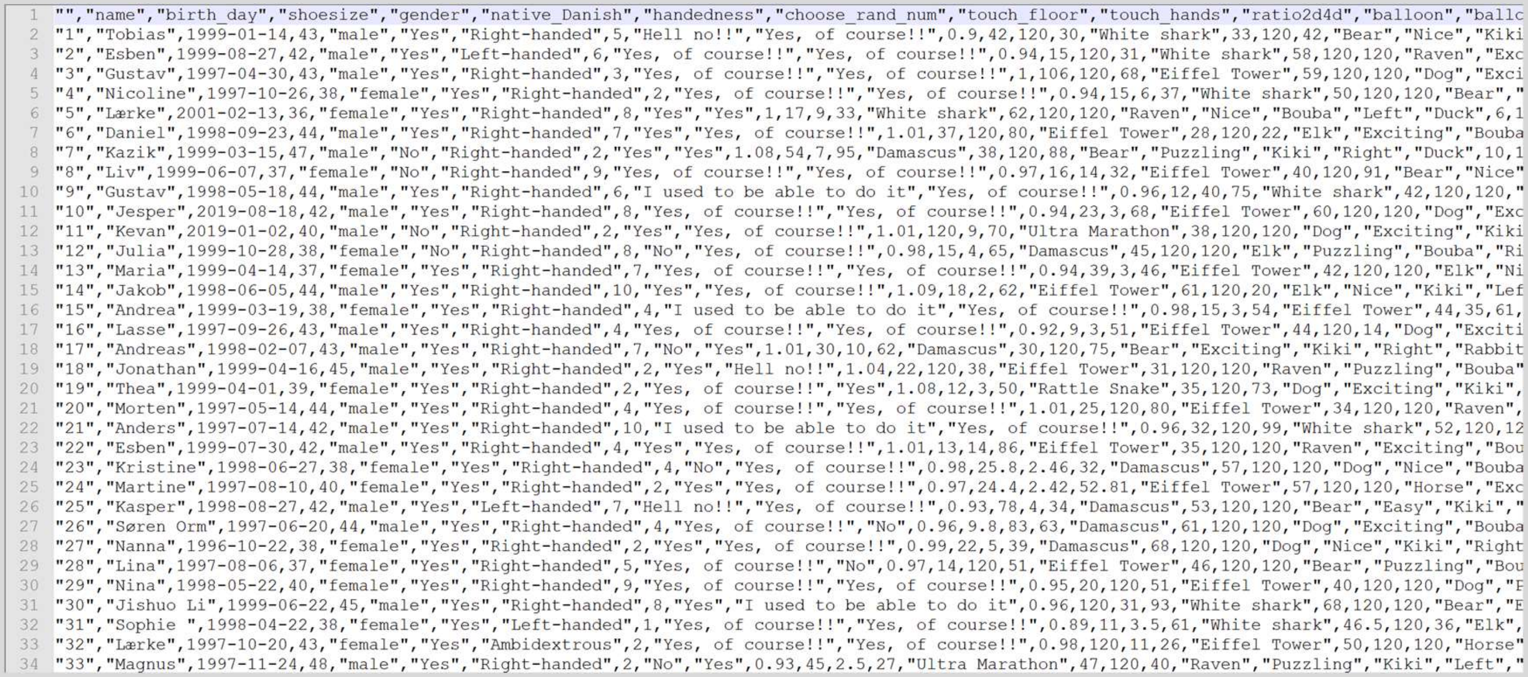
- Fake data is nice, fine… and boring
- We want real data! And we want your data
- What is a data set? – The file version of a data.frame
  - Often ends with .csv
- Throw your data in a folder and let’s rock!

Working directory: Where R searches for files

- Can only have one active working directory (WD) at a time
- Find the current WD with the function: `getwd()`
- Can be changed with the function: `setwd("path/to/dir")`
- … Or using the dropdown if you don't feel like a H4CK3R yet
- Best practise: Save your script where you want your WD and set path when looking in different folders.



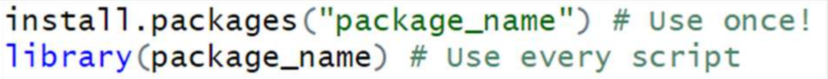
How does *raw* data look?



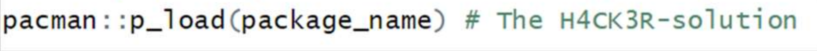
```
"","name","birth_day","shoesize","gender","native_Danish","handedness","choose_rand_num","touch_floor","touch_hands","ratio2d4d","balloon","ballc
"1","Tobias",1999-01-14,43,"male","Yes","Right-handed",5,"Hell no!!","Yes, of course!!",0.9,42,120,30,"White shark",33,120,42,"Bear","Nice","Kiki
"2","Esben",1999-08-27,42,"male","Yes","Left-handed",6,"Yes, of course!!","Yes, of course!!",0.94,15,120,31,"White shark",58,120,120,"Raven","Exc
"3","Gustav",1997-04-30,43,"male","Yes","Right-handed",3,"Yes, of course!!","Yes, of course!!",1,106,120,68,"Eiffel Tower",59,120,120,"Dog","Exci
"4","Nicoline",1997-10-26,38,"female","Yes","Right-handed",2,"Yes, of course!!","Yes, of course!!",0.94,15,6,37,"White shark",50,120,120,"Bear","
"5","Lærke",2001-02-13,36,"female","Yes","Right-handed",8,"Yes","Yes",1,17,9,33,"White shark",62,120,120,"Raven","Nice","Bouba","Left","Duck",6,1
"6","Daniel",1998-09-23,44,"male","Yes","Right-handed",7,"Yes","Yes, of course!!",1.01,37,120,80,"Eiffel Tower",28,120,22,"Elk","Exciting","Bouba
"7","Kazik",1999-03-15,47,"male","No","Right-handed",2,"Yes","Yes",1.08,54,7,95,"Damascus",38,120,88,"Bear","Puzzling","Kiki","Right","Duck",10,1
"8","Liv",1999-06-07,37,"female","No","Right-handed",9,"Yes, of course!!","Yes, of course!!",0.97,16,14,32,"Eiffel Tower",40,120,91,"Bear","Nice"
"9","Gustav",1998-05-18,44,"male","Yes","Right-handed",6,"I used to be able to do it","Yes, of course!!",0.96,12,40,75,"White shark",42,120,120,"
"10","Jesper",2019-08-18,42,"male","Yes","Right-handed",8,"Yes, of course!!","Yes, of course!!",0.94,23,3,68,"Eiffel Tower",60,120,120,"Dog","Exc
"11","Kevan",2019-01-02,40,"male","No","Right-handed",2,"Yes","Yes, of course!!",1.01,120,9,70,"Ultra Marathon",38,120,120,"Dog","Exciting","Kiki
"12","Julia",1999-10-28,38,"female","No","Right-handed",8,"No","Yes, of course!!",0.98,15,4,65,"Damascus",45,120,120,"Elk","Puzzling","Bouba","Ri
"13","Maria",1999-04-14,37,"female","Yes","Right-handed",7,"Yes, of course!!","Yes, of course!!",0.94,39,3,46,"Eiffel Tower",42,120,120,"Elk","Ni
"14","Jakob",1998-06-05,44,"male","Yes","Right-handed",10,"Yes","Yes, of course!!",1.09,18,2,62,"Eiffel Tower",61,120,20,"Elk","Nice","Kiki","Lef
"15","Andrea",1999-03-19,38,"female","Yes","Right-handed",4,"I used to be able to do it","Yes, of course!!",0.98,15,3,54,"Eiffel Tower",44,35,61,
"16","Lasse",1997-09-26,43,"male","Yes","Right-handed",4,"Yes, of course!!","Yes, of course!!",0.92,9,3,51,"Eiffel Tower",44,120,14,"Dog","Exciti
"17","Andreas",1998-02-07,43,"male","Yes","Right-handed",7,"No","Yes",1.01,30,10,62,"Damascus",30,120,75,"Bear","Exciting","Kiki","Right","Rabbit
"18","Jonathan",1999-04-16,45,"male","Yes","Right-handed",2,"Yes","Hell no!!",1.04,22,120,38,"Eiffel Tower",31,120,120,"Raven","Puzzling","Bouba"
"19","Thea",1999-04-01,39,"female","Yes","Right-handed",2,"Yes, of course!!","Yes",1.08,12,3,50,"Rattle Snake",35,120,73,"Dog","Exciting","Kiki",
"20","Morten",1997-05-14,44,"male","Yes","Right-handed",4,"Yes, of course!!","Yes, of course!!",1.01,25,120,80,"Eiffel Tower",34,120,120,"Raven",
"21","Anders",1997-07-14,42,"male","Yes","Right-handed",10,"I used to be able to do it","Yes, of course!!",0.96,32,120,99,"White shark",52,120,12
"22","Esben",1999-07-30,42,"male","Yes","Right-handed",4,"Yes","Yes, of course!!",1.01,13,14,86,"Eiffel Tower",35,120,120,"Raven","Exciting","Bou
"23","Kristine",1998-06-27,38,"female","Yes","Right-handed",4,"No","Yes, of course!!",0.98,25.8,2.46,32,"Damascus",57,120,120,"Dog","Nice","Bouba
"24","Martine",1997-08-10,40,"female","Yes","Right-handed",2,"Yes","Yes, of course!!",0.97,24.4,2.42,52.81,"Eiffel Tower",57,120,120,"Horse","Exc
"25","Kasper",1998-08-27,42,"male","Yes","Left-handed",7,"Hell no!!","Yes, of course!!",0.93,78,4,34,"Damascus",53,120,120,"Bear","Easy","Kiki","
"26","Søren Orm",1997-06-20,44,"male","Yes","Right-handed",4,"Yes, of course!!","No",0.96,9.8,83,63,"Damascus",61,120,120,"Dog","Exciting","Bouba
"27","Nanna",1996-10-22,38,"female","Yes","Right-handed",2,"Yes","Yes, of course!!",0.99,22,5,39,"Damascus",68,120,120,"Dog","Nice","Kiki","Right
"28","Lina",1997-08-06,37,"female","Yes","Right-handed",5,"Yes, of course!!","No",0.97,14,120,51,"Eiffel Tower",46,120,120,"Bear","Puzzling","Bou
"29","Nina",1998-05-22,40,"female","Yes","Right-handed",9,"Yes, of course!!","Yes, of course!!",0.95,20,120,51,"Eiffel Tower",40,120,120,"Dog","P
"30","Jishuo Li",1999-06-22,45,"male","Yes","Right-handed",8,"Yes","I used to be able to do it",0.96,120,31,93,"White shark",68,120,120,"Bear","E
"31","Sophie ",1998-04-22,38,"female","Yes","Left-handed",1,"Yes, of course!!","Yes, of course!!",0.89,11,3.5,61,"White shark",46.5,120,36,"Elk",
"32","Lærke",1997-10-20,43,"female","Yes","Ambidextrous",2,"Yes, of course!!","Yes, of course!!",0.98,120,11,26,"Eiffel Tower",50,120,120,"Horse"
"33","Magnus",1997-11-24,48,"male","Yes","Right-handed",2,"No","Yes",0.93,45,2.5,27,"Ultra Marathon",47,120,40,"Raven","Puzzling","Kiki","Left","
```

Packages: Stealing code, but… better

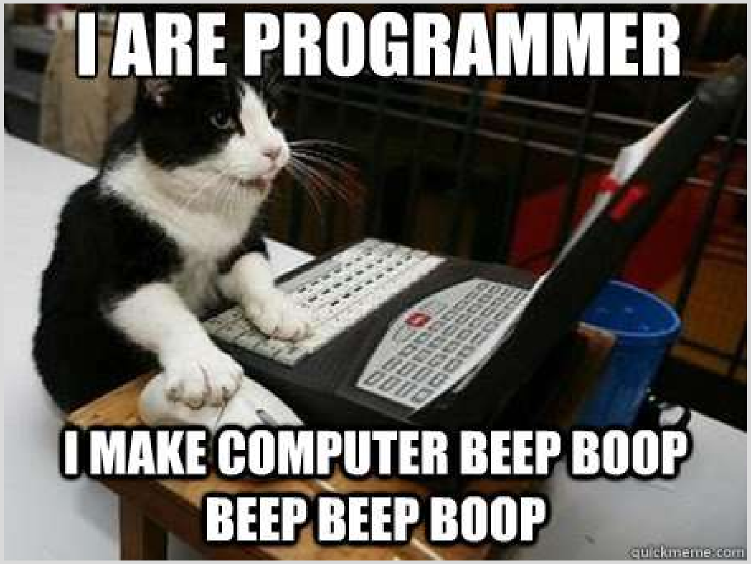
- The fundamental unit of shareable (stealable) code
- Bundles together code, data, documentation et al.
- Useful functions:



```
install.packages("package_name") # Use once!
library(package_name) # Use every script
```



```
pacman::p_load(package_name) # The H4CK3R-solution
```



ONE LIBRARY TO RULE THEM ALL...

Philosophy of Tidyverse

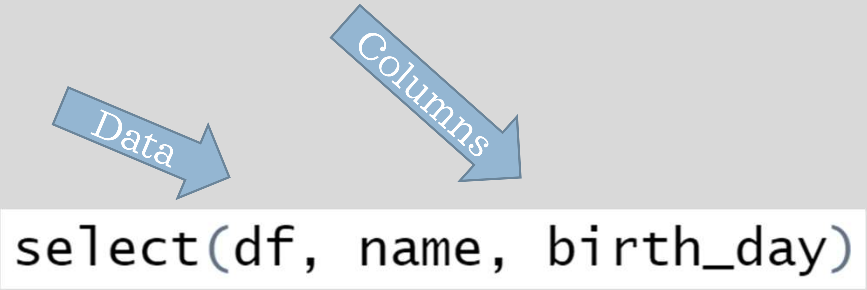
- Data should be TIDY
  - One observation per row, one variable per column, one type per table
- Functions operate like verbs
  - One action per function
  - Can be stringed together
  - Create beautiful poetry (we'll get there!)
- Design for humans not computers
  - All the magic is behind the scenes

Main functions of tidyverse/all online stores

- Filter observations by values: `filter()`
- Reorder the rows: `arrange()`
- Select columns by name: `select()`
- Create new column or change existing one: `mutate()`
- Summarise variable into single value: `summmarise()` / `summarize()`

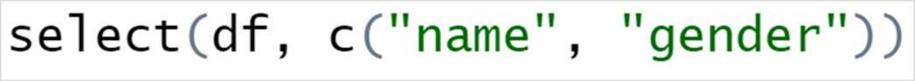
SELECT()

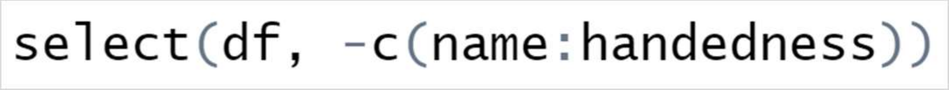
- Choosing relevant columns
- All sorts of nifty tricks

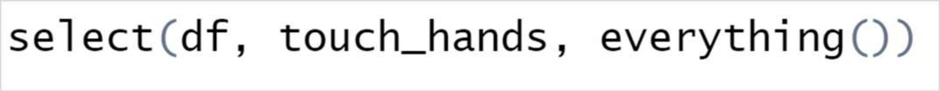

Data

Columns

```
select(df, name, birth_day)
```

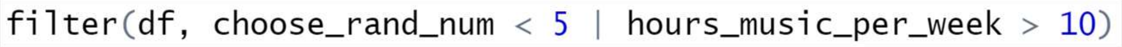

```
select(df, c("name", "gender"))
```


```
select(df, -c(name:handedness))
```

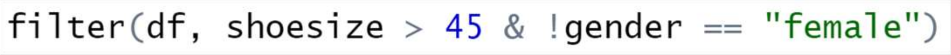

```
select(df, touch_hands, everything())
```

FILTER()

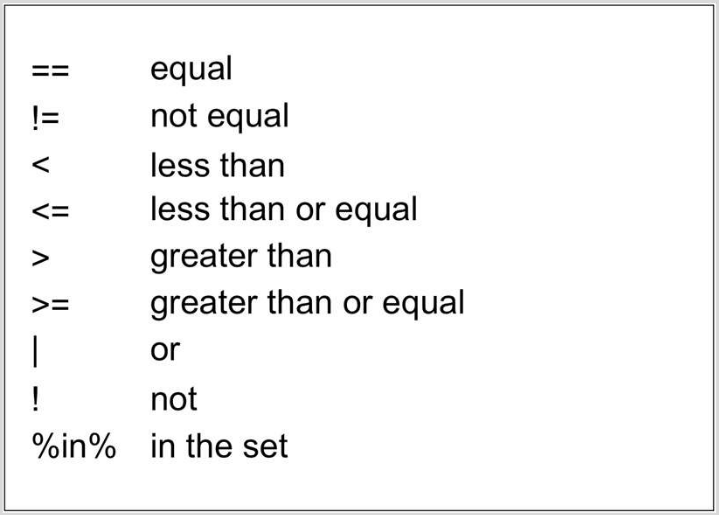
- Choosing relevant rows
- Based on logic operators
- Examples:



```
filter(df, choose_rand_num < 5 | hours_music_per_week > 10)
```



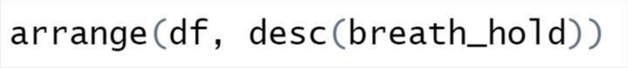
```
filter(df, shoesize > 45 & !gender == "female")
```



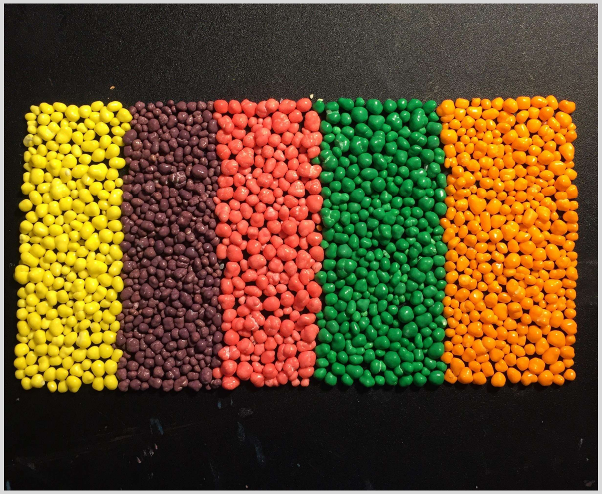
==	equal
!=	not equal
<	less than
<=	less than or equal
>	greater than
>=	greater than or equal
|	or
!	not
%in%	in the set

ARRANGE()

- Sorts stuff
- If you want to sort by descending use desc i.e

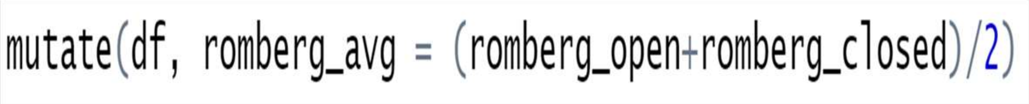


```
arrange(df, desc(breath_hold))
```

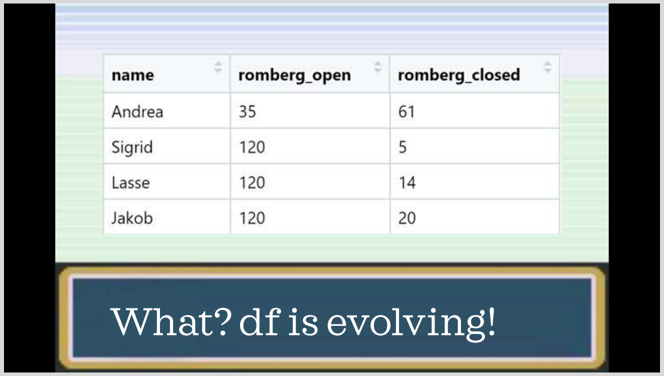


Mutate()

- For changing columns or creating new ones
- Will be added as new last column
- Example

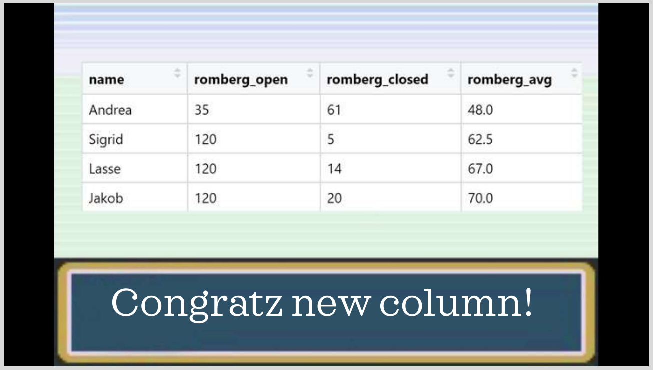


```
mutate(df, romberg_avg = (romberg_open+romberg_closed)/2)
```



name	romberg_open	romberg_closed
Andrea	35	61
Sigrid	120	5
Lasse	120	14
Jakob	120	20

What? df is evolving!

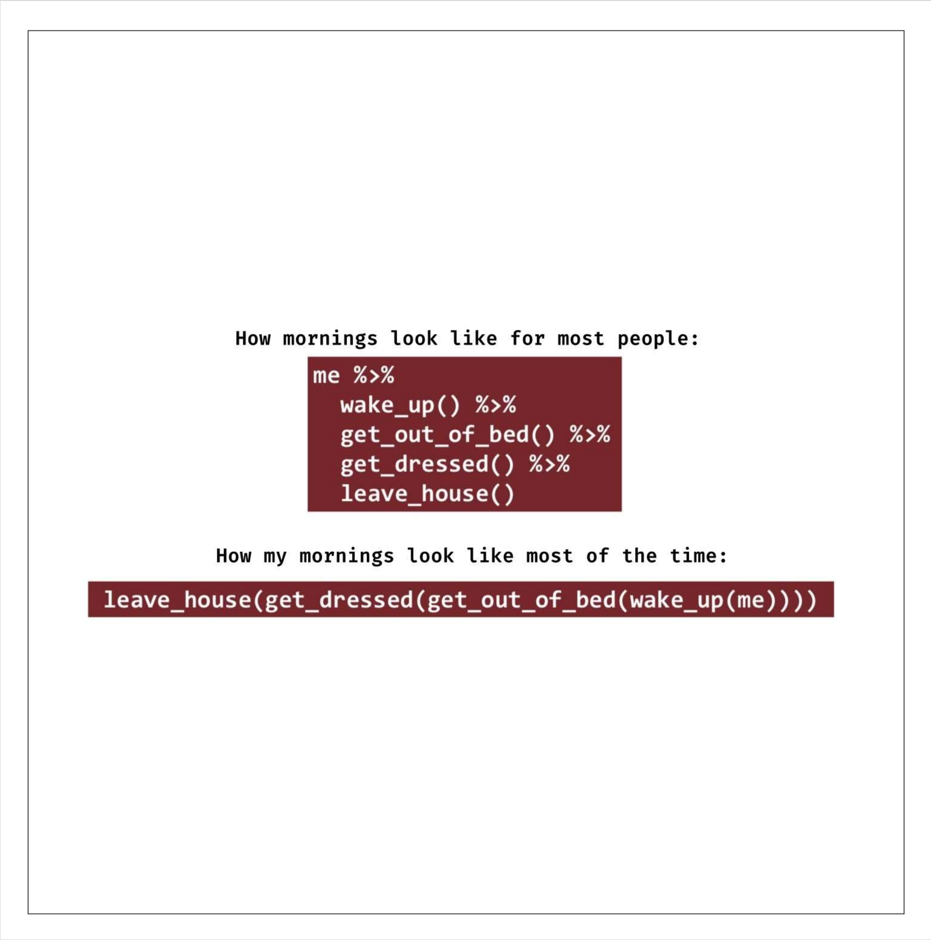


name	romberg_open	romberg_closed	romberg_avg
Andrea	35	61	48.0
Sigrid	120	5	62.5
Lasse	120	14	67.0
Jakob	120	20	70.0

Congratz new column!

Where's the beauty?

%>%
Ceci n'est pas un pipe.
www.tidyverse.org



How mornings look like for most people:

```
me %>%
  wake_up() %>%
  get_out_of_bed() %>%
  get_dressed() %>%
  leave_house()
```

How my mornings look like most of the time:

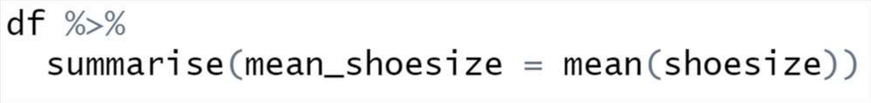
```
leave_house(get_dressed(get_out_of_bed(wake_up(me))))
```

Pipes: The glue of the tidyverse

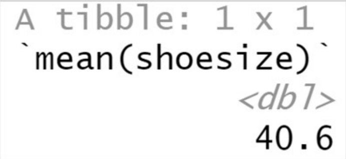
- Elegant way to string together functions
- Technically takes previous output as first argument…
- But can just be read as ”and then”
- Shortcut: ctrl/cmd + shift + m

Summarise() and group_by()

- Collapses data into a single row

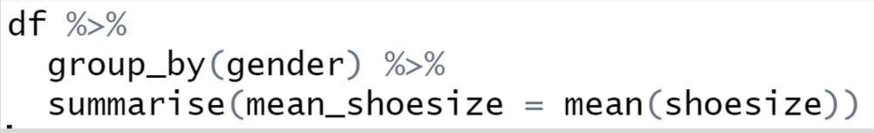


```
df %>%
  summarise(mean_shoesize = mean(shoesize))
```

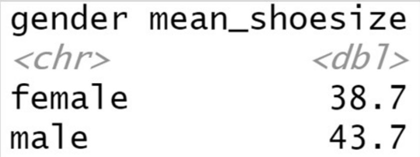


```
A tibble: 1 x 1
`mean(shoesize)`
           <dbl>
            40.6
```

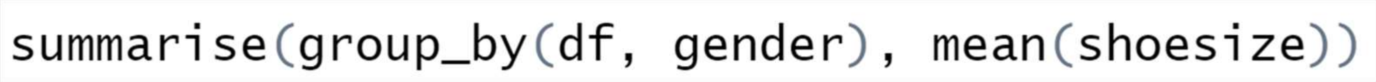
- Most useful with a group_by()



```
df %>%
  group_by(gender) %>%
  summarise(mean_shoesize = mean(shoesize))
```



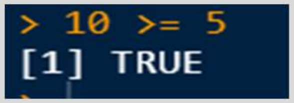
```
gender mean_shoesize
<chr>          <dbl>
female          38.7
male            43.7
```



```
summarise(group_by(df, gender), mean(shoesize))
```

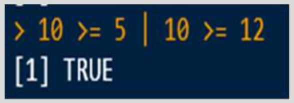
Insider tip: Logical Operators:

- Last time we learned stuff like this:

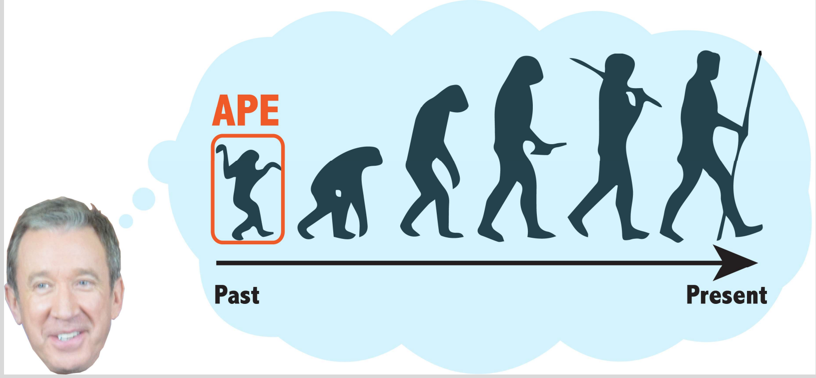


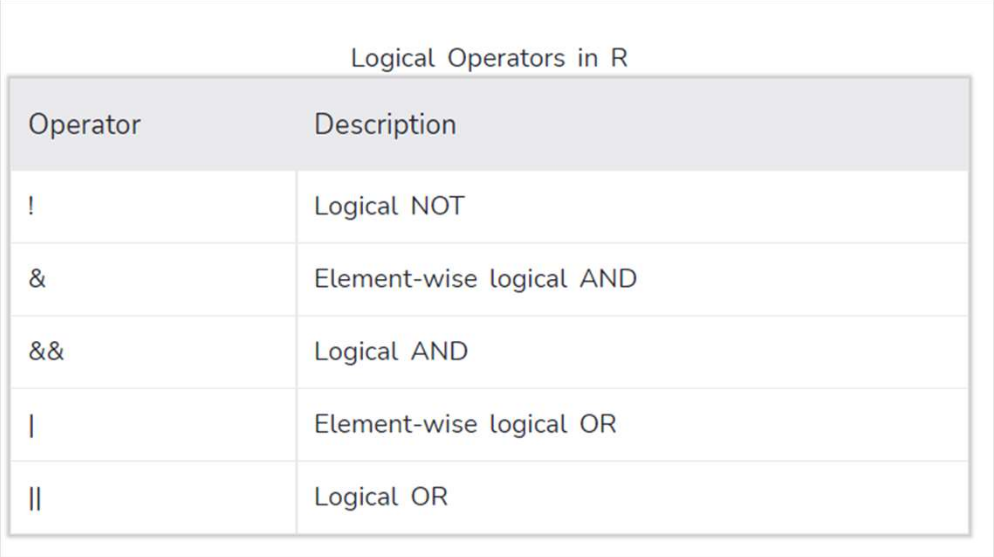
```
> 10 >= 5
[1] TRUE
```

- Now we evolve:



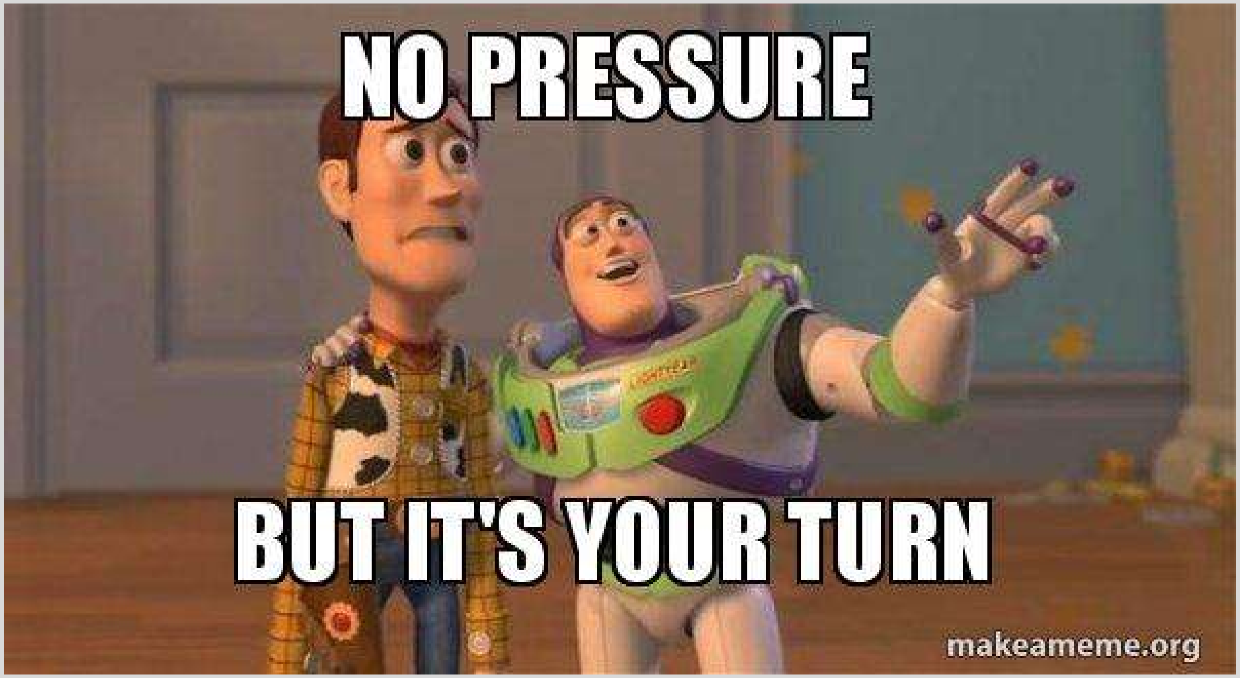
```
> 10 >= 5 | 10 >= 12
[1] TRUE
```



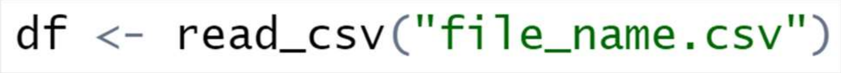


Logical Operators in R

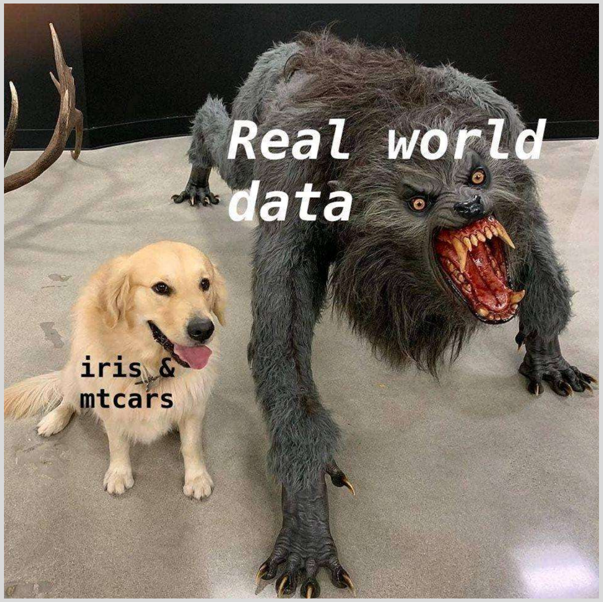
Operator	Description
!	Logical NOT
&	Element-wise logical AND
&&	Logical AND
|	Element-wise logical OR
||	Logical OR


NO PRESSURE
BUT IT'S YOUR TURN
makeameme.org

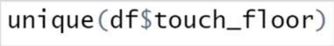
Loading data

- Make sure you have the right working directory (getwd())
- 

```
df <- read_csv("file_name.csv")
```



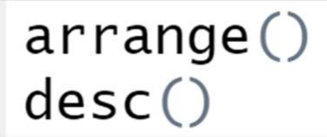
Exercise 1: filter()

- Find student(s) that:
  1. Have shoesize 39 or bigger
  2. Were able to touch the floor (hint: check  `unique(df$touch_floor)`)
  3. Were able to hold their breath for longer than average
  4. Could balance a balloon between 13 and 60 seconds
  5. Extra: all of the above (extra extra: using pipes!)

Exercise 2: arrange()

1. Sort data so it starts with the slowest tongue twister person
2. Sort data by one argument to find the student who performed best at the Romberg task (you define best)
3. How could you improve your Romberg ranking table?

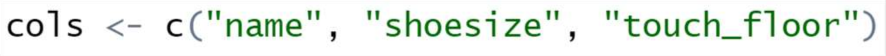
Commands you'll need:



```
arrange()
desc()
```

Exercise 3: select()

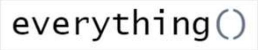
1. What happens if you select the same column multiple times?
2. Make the following vector



```r
cols <- c("name", "shoesize", "touch_floor")
```

What happens when you select with this vector?

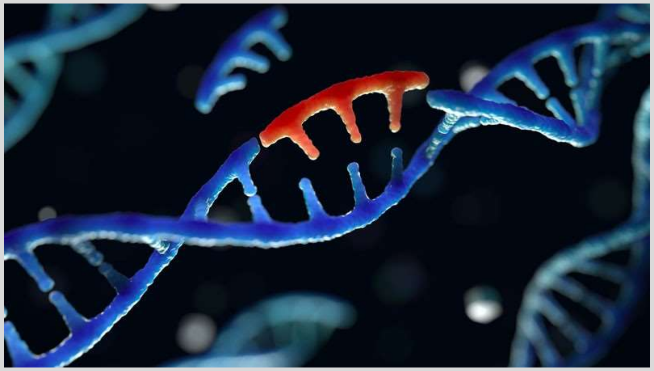
3. Rearrange the columns of the dataframe with gender and shoesize first. Hint: use



```r
everything()
```

Exercise 4: mutate()

- The tongue twister had 99 words. Make a new column called ”words_per_sec” where tyou calculate how many words each student said per second.
- Currently breath_hold is in seconds. Convert it into two new columns called ”breath_min” and ”breath_sec” containing the number of whole minutes (achieved by dividing using `%/%`) and remaining seconds respectively
- BONUS: create a new column calculating how far is from the average breath_hold



Exercise 5: summarise()

1. Is there a gender difference when it comes to balloon balancing? (hint: group_by and %>% are your friends!)
2. Is there a relation between sound level preference and which cola was chosen?
3. Does handedness influence tongue twisting speed?
    1. Add a column to the summary which contains number of people in each group (e.g. number of right handed people). Hint: look at the `n()` -function
    2. Does this tell us something about the reliability?

Nice resources (shout out to Anita + Kenneth)

- ”R for data science” – a bible written by Jesus (aka. Wickham) himself!)
  - https://r4ds.had.co.nz/
- R programming tutorials
  - https://www.datamentor.io/r-programming/#tutorial
- Cheat sheets O.o
  - https://rstudio.com/resources/cheatsheets/

THANK YOU