

Intro to R for Biologists

IBiS Special Topics, Fall 2021

Class 4: Pou5F1, 2021

Erik Andersen and Shelby Blythe

**The most important part of data analysis
is to *think* about your data**

What do you want to test?

How will you show that conclusion?

Put the goal at the top of your markdown report

Thinking like a computer

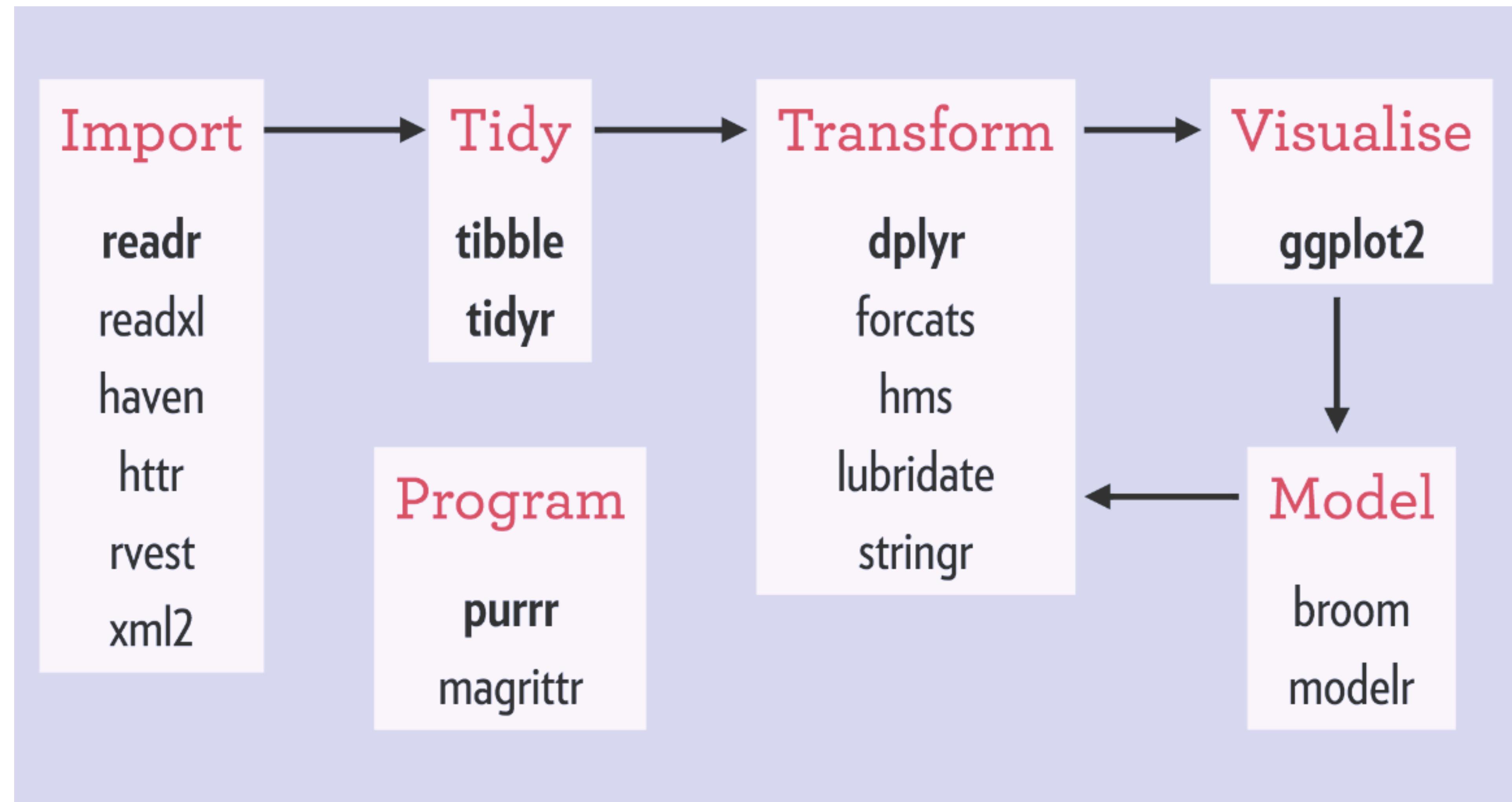
1. Figure out what you want to do
2. Describe those tasks in words
3. Describe those tasks in code

What is the Tidyverse?

- Collection of packages for data manipulation, exploration, and visualization that share a common syntax
- Intended to make data scientists more productive by guiding them through workflows
- Allows for connections between tools



Packages in Tidyverse



Starwars dataset



```
library(tidyverse)  
data(starwars)  
View(starwars)
```

variables

	name	height	mass	hair_color	skin_color	eye_color	birth_year	gender	homeworld	species	films	vehicles	starships
1	Luke Skywalker	172	77.0	blond	fair	blue	19.0	male	Tatooine	Human	c("Revenge of the Sith", "Return of the Jedi", "The Emp...	c("Snowspeeder", "Imperial Speeder Bike")	c("X-wing", "Imperial shuttle")
2	C-3PO	167	75.0	NA	gold	yellow	112.0	NA	Tatooine	Droid	c("Attack of the Clones", "The Phantom Menace", "Rev...	character(0)	character(0)
3	R2-D2	96	32.0	NA	white, blue	red	33.0	NA	Naboo	Droid	c("Attack of the Clones", "The Phantom Menace", "Rev...	character(0)	character(0)
4	Darth Vader	202	136.0	none	white	yellow	41.9	male	Tatooine	Human	c("Revenge of the Sith", "Return of the Jedi", "The Emp...	character(0)	TIE Advanced x1
5	Leia Organa	150	49.0	brown	light	brown	19.0	female	Alderaan	Human	c("Revenge of the Sith", "Return of the Jedi", "The Emp...	Imperial Speeder Bike	character(0)
6	Owen Lars	178	120.0	brown, grey	light	blue	52.0	male	Tatooine	Human	c("Attack of the Clones", "Revenge of the Sith", "A New...	character(0)	character(0)
7	Beru Whitesun lars	165	75.0	brown	light	blue	47.0	female	Tatooine	Human	c("Attack of the Clones", "Revenge of the Sith", "A New...	character(0)	character(0)
8	R5-D4	97	32.0	NA	white, red	red	NA	NA	Tatooine	Droid	A New Hope	character(0)	character(0)
9	Biggs Darklighter	183	84.0	black	light	brown	24.0	male	Tatooine	Human	A New Hope	character(0)	X-wing
10	Obi-Wan Kenobi	182	77.0	auburn, white	fair	blue-gray	57.0	male	Stewjon	Human	c("Attack of the Clones", "The Phantom Menace", "Rev...	Tribubble bongo	c("Jedi starfighter", "Trade Federation cruiser", "Naboo...
11	Anakin Skywalker	188	84.0	blond	fair	blue	41.9	male	Tatooine	Human	c("Attack of the Clones", "The Phantom Menace", "Rev...	c("Zephyr-G swoop bike", "XJ-6 airspeeder")	c("Trade Federation cruiser", "Jedi Interceptor", "Nabo...
12	Wilhuff Tarkin	180	NA	auburn, grey	fair	blue	64.0	male	Eriadu	Human	c("Revenge of the Sith", "A New Hope")	character(0)	character(0)
13	Chewbacca	228	112.0	brown	unknown	blue	200.0	male	Kashyyyk	Wookiee	c("Revenge of the Sith", "Return of the Jedi", "The Emp...	AT-ST	c("Millennium Falcon", "Imperial shuttle")
14	Han Solo	180	80.0	brown	fair	brown	29.0	male	Corellia	Human	c("Return of the Jedi", "The Empire Strikes Back", "A N...	character(0)	c("Millennium Falcon", "Imperial shuttle")
15	Greedo	173	74.0	NA	green	black	44.0	male	Rodia	Rodian	A New Hope	character(0)	character(0)
16	Jabba Desilijic Tiure	175	1358.0	NA	green-tan, brown	orange	600.0	hermaphrodite	Nal Hutta	Hutt	c("The Phantom Menace", "Return of the Jedi", "A New ...	character(0)	character(0)
17	Wedge Antilles	170	77.0	brown	fair	hazel	21.0	male	Corellia	Human	c("Return of the Jedi", "The Empire Strikes Back", "A N...	Snowspeeder	X-wing
18	Jek Tono Porkins	180	110.0	brown	fair	blue	NA	male	Bestine IV	Human	A New Hope	character(0)	X-wing
19	Yoda	66	17.0	white	green	brown	896.0	male	NA	Yoda's species	c("Attack of the Clones", "The Phantom Menace", "Rev...	character(0)	character(0)
20	Palpatine	170	75.0	grey	pale	yellow	82.0	male	Naboo	Human	c("Attack of the Clones", "The Phantom Menace", "Rev...	character(0)	character(0)
21	Boba Fett	183	78.2	black	fair	brown	31.5	male	Kamino	Human	c("Attack of the Clones", "Return of the Jedi", "The Em...	character(0)	Slave 1
22	IG-88	200	140.0	none	metal	red	15.0	none	NA	Droid	The Empire Strikes Back	character(0)	character(0)
23	Bossk	190	113.0	none	green	red	53.0	male	Trandosha	Trandoshan	The Empire Strikes Back	character(0)	character(0)
24	Lando Calrissian	177	79.0	black	dark	brown	31.0	male	Socorro	Human	c("Return of the Jedi", "The Empire Strikes Back")	character(0)	Millennium Falcon

observations

values

Starwars dataset



library(tidyverse)
data(starwars)
View(starwars)

MESSY DATA

variables

	name	height	mass	hair_color	skin_color	eye_color	birth_year	gender	homeworld	species	films	vehicles	starships
1	Luke Skywalker	172	77.0	blond	fair	blue	19.0	male	Tatooine	Human	c("Revenge of the Sith", "Return of the Jedi", "The Empire...")	c("Snowspeeder", "Imperial Speeder Bike")	c("X-wing", "Imperial shuttle")
2	C-3PO	167	75.0	NA	gold	yellow	112.0	NA	Tatooine	Droid	c("Attack of the Clones", "The Phantom Menace", "Rev...")	character(0)	character(0)
3	R2-D2	96	32.0	NA	white, blue	red	33.0	NA	Naboo	Droid	c("Attack of the Clones", "The Phantom Menace", "Rev...")	character(0)	character(0)
4	Darth Vader	202	136.0	none	white	yellow	41.9	male	Tatooine	Human	c("Revenge of the Sith", "Return of the Jedi", "The Empire...")	character(0)	TIE Advanced x1
5	Leia Organa	150	49.0	brown	light	brown	19.0	female	Alderaan	Human	c("Revenge of the Sith", "Return of the Jedi", "The Empire...")	Imperial Speeder Bike	character(0)
6	Owen Lars	178	120.0	brown, grey	light	blue	52.0	male	Tatooine	Human	c("Attack of the Clones", "Revenge of the Sith", "A New...")	character(0)	character(0)
7	Beru Whitesun lars	165	75.0	brown	light	blue	47.0	female	Tatooine	Human	c("Attack of the Clones", "Revenge of the Sith", "A New...")	character(0)	character(0)
8	R5-D4	97	32.0	NA	white, red	red	NA	NA	Tatooine	Droid	A New Hope	character(0)	character(0)
9	Biggs Darklighter	183	84.0	black	light	brown	24.0	male	Tatooine	Human	A New Hope	character(0)	X-wing
10	Obi-Wan Kenobi	182	77.0	auburn, white	fair	blue-gray	57.0	male	Stewjon	Human	c("Attack of the Clones", "The Phantom Menace", "Rev...")	Tribubble bongo	c("Jedi starfighter", "Trade Federation cruiser", "Naboo...")
11	Anakin Skywalker	188	84.0	blond	fair	blue	41.9	male	Tatooine	Human	c("Attack of the Clones", "The Phantom Menace", "Rev...")	c("Zephyr-G swoop bike", "XJ-6 airspeeder")	c("Trade Federation cruiser", "Jedi Interceptor", "Nabo...")
12	Wilhuff Tarkin	180	NA	auburn, grey	fair	blue	64.0	male	Eriadu	Human	c("Revenge of the Sith", "A New Hope")	character(0)	character(0)
13	Chewbacca	228	112.0	brown	unknown	blue	200.0	male	Kashyyyk	Wookiee	c("Revenge of the Sith", "Return of the Jedi", "The Empire...")	AT-ST	c("Millennium Falcon", "Imperial shuttle")
14	Han Solo	180	80.0	brown	fair	brown	29.0	male	Corellia	Human	c("Return of the Jedi", "The Empire Strikes Back", "A N...")	character(0)	c("Millennium Falcon", "Imperial shuttle")
15	Greedo	173	74.0	NA	green	black	44.0	male	Rodia	Rodian	A New Hope	character(0)	character(0)
16	Jabba Desilijic Tiure	175	1358.0	NA	green-tan, brown	orange	600.0	hermaphrodite	Nal Hutta	Hutt	c("The Phantom Menace", "Return of the Jedi", "A New ...")	character(0)	character(0)
17	Wedge Antilles	170	77.0	brown	fair	hazel	21.0	male	Corellia	Human	c("Return of the Jedi", "The Empire Strikes Back", "A N...")	Snowspeeder	X-wing
18	Jek Tono Porkins	180	110.0	brown	fair	blue	NA	male	Bestine IV	Human	A New Hope	character(0)	X-wing
19	Yoda	66	17.0	white	green	brown	896.0	male	NA	Yoda's species	c("Attack of the Clones", "The Phantom Menace", "Rev...")	character(0)	character(0)
20	Palpatine	170	75.0	grey	pale	yellow	82.0	male	Naboo	Human	c("Attack of the Clones", "The Phantom Menace", "Rev...")	character(0)	character(0)
21	Boba Fett	183	78.2	black	fair	brown	31.5	male	Kamino	Human	c("Attack of the Clones", "Return of the Jedi", "The Em...")	character(0)	Slave 1
22	IG-88	200	140.0	none	metal	red	15.0	none	NA	Droid	The Empire Strikes Back	character(0)	character(0)
23	Bossk	190	113.0	none	green	red	53.0	male	Trandosha	Trandoshan	The Empire Strikes Back	character(0)	character(0)
24	Lando Calrissian	177	79.0	black	dark	brown	31.0	male	Socorro	Human	c("Return of the Jedi", "The Empire Strikes Back")	character(0)	Millennium Falcon

observations

values

Introduction: dplyr

- Collection of functions as **verbs** to easily describe what you want to do with your data

Functions:

- `filter()` to keep rows based on values
- `select()` to keep columns based on names
- `mutate()` to add new (or change existing) columns
- `group_by()` to group rows by columns
- `summarize()` to condense multiple columns
- `arrange()` to reorder the rows
- `rename()` to give columns new names
- `distinct()` to keep unique values





`dplyr::filter()`

`dplyr::filter()` to keep rows based on values

dplyr::filter()

dplyr::filter(dataframe, condition(s))

starwars

only humans

1. Figure out what you want to do

2. Describe your goal in words

Filter the starwars dataframe to only include humans

3. Describe your goal in code

dplyr::filter(starwars, ?)

dplyr::filter() to keep rows based on values



dplyr::filter()

	name	height	mass	hair_color	skin_color	eye_color	birth_year	gender	homeworld	species	films	vehicles	starships
1	Luke Skywalker	172	77.0	blond	fair	blue	19.0	male	Tatooine	Human	c("Revenge of the Sith", "Return of the Jedi", "The Empire Strikes Back", "The Phantom Menace", "Attack of the Clones", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	c("Snowspeeder", "Imperial Speeder Bike")	c("X-wing", "Imperial shuttle")
2	C-3PO	167	75.0	NA	gold	yellow	112.0	NA	Tatooine	Droid	c("Attack of the Clones", "The Phantom Menace", "Revenge of the Sith", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	character(0)	character(0)
3	R2-D2	96	32.0	NA	white, blue	red	33.0	NA	Naboo	Droid	c("Attack of the Clones", "The Phantom Menace", "Revenge of the Sith", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	character(0)	character(0)
4	Darth Vader	202	136.0	none	white	yellow	41.9	male	Tatooine	Human	c("Revenge of the Sith", "Return of the Jedi", "The Empire Strikes Back", "The Phantom Menace", "Attack of the Clones", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	character(0)	TIE Advanced x1
5	Leia Organa	150	49.0	brown	light	brown	19.0	female	Alderaan	Human	c("Revenge of the Sith", "Return of the Jedi", "The Empire Strikes Back", "The Phantom Menace", "Attack of the Clones", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	Imperial Speeder Bike	character(0)
6	Owen Lars	178	120.0	brown, grey	light	blue	52.0	male	Tatooine	Human	c("Attack of the Clones", "Revenge of the Sith", "A New Hope", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	character(0)	character(0)
7	Beru Whitesun lars	165	75.0	brown	light	blue	47.0	female	Tatooine	Human	c("Attack of the Clones", "Revenge of the Sith", "A New Hope", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	character(0)	character(0)
8	R5-D4	97	32.0	NA	white, red	red	NA	NA	Tatooine	Droid	A New Hope	character(0)	character(0)
9	Biggs Darklighter	183	84.0	black	light	brown	24.0	male	Tatooine	Human	A New Hope	character(0)	X-wing
10	Obi-Wan Kenobi	182	77.0	auburn, white	fair	blue-gray	57.0	male	Stewjon	Human	c("Attack of the Clones", "The Phantom Menace", "Revenge of the Sith", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	Tribubble bongo	c("Jedi starfighter", "Trade Federation cruiser", "Naboo Starship", "Millennium Falcon", "Imperial shuttle")
11	Anakin Skywalker	188	84.0	blond	fair	blue	41.9	male	Tatooine	Human	c("Attack of the Clones", "The Phantom Menace", "Revenge of the Sith", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	c("Zephyr-G swoop bike", "XJ-6 airspeeder")	c("Trade Federation cruiser", "Jedi Interceptor", "Naboo Starship", "Millennium Falcon", "Imperial shuttle")
12	Wilhuff Tarkin	180	NA	auburn, grey	fair	blue	64.0	male	Eriadu	Human	c("Revenge of the Sith", "A New Hope")	character(0)	character(0)
13	Chewbacca	228	112.0	brown	unknown	blue	200.0	male	Kashyyyk	Wookiee	c("Revenge of the Sith", "Return of the Jedi", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	AT-ST	c("Millennium Falcon", "Imperial shuttle")
14	Han Solo	180	80.0	brown	fair	brown	29.0	male	Corellia	Human	c("Return of the Jedi", "The Empire Strikes Back", "A New Hope", "The Force Awakens")	character(0)	c("Millennium Falcon", "Imperial shuttle")
15	Greedo	173	74.0	NA	green	black	44.0	male	Rodia	Rodian	A New Hope	character(0)	character(0)
16	Jabba Desilijic Tiure	175	1358.0	NA	green-tan, brown	orange	600.0	hermaphrodite	Nal Hutta	Hutt	c("The Phantom Menace", "Return of the Jedi", "A New Hope", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	character(0)	character(0)
17	Wedge Antilles	170	77.0	brown	fair	hazel	21.0	male	Corellia	Human	c("Return of the Jedi", "The Empire Strikes Back", "A New Hope", "The Force Awakens")	Snowspeeder	X-wing
18	Jek Tono Porkins	180	110.0	brown	fair	blue	NA	male	Bestine IV	Human	A New Hope	character(0)	X-wing
19	Yoda	66	17.0	white	green	brown	896.0	male	NA	Yoda's species	c("Attack of the Clones", "The Phantom Menace", "Revenge of the Sith", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	character(0)	character(0)
20	Palpatine	170	75.0	grey	pale	yellow	82.0	male	Naboo	Human	c("Attack of the Clones", "The Phantom Menace", "Revenge of the Sith", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	character(0)	character(0)
21	Boba Fett	183	78.2	black	fair	brown	31.5	male	Kamino	Human	c("Attack of the Clones", "Return of the Jedi", "The Empire Strikes Back", "The Return of the Jedi", "The Force Awakens")	character(0)	Slave 1
22	IG-88	200	140.0	none	metal	red	15.0	none	NA	Droid	The Empire Strikes Back	character(0)	character(0)
23	Bossk	190	113.0	none	green	red	53.0	male	Trandosha	Trandoshan	The Empire Strikes Back	character(0)	character(0)
24	Lando Calrissian	177	79.0	black	dark	brown	31.0	male	Socorro	Human	c("Return of the Jedi", "The Empire Strikes Back")	character(0)	Millennium Falcon

species == "Human"

dplyr::filter() to keep rows based on values



dplyr::filter()

dplyr::filter(dataframe, condition(s))

starwars

only humans

1. Figure out what you want to do

2. Describe your goal in words

Filter the starwars dataframe to only include humans

3. Describe your goal in code

```
dplyr::filter(starwars, species == "Human")
```

dplyr::filter() to keep rows based on values



dplyr::filter()

	name	height	mass	hair_color	skin_color	eye_color	birth_year	gender	homeworld	species	films	vehicles	starships
1	Luke Skywalker	172	77.0	blond	fair	blue	19.0	male	Tatooine	Human	c("Revenge of the Sith", "Return of the Jedi", "The Emp...	c("Snowspeeder", "Imperial Speeder Bike")	c("X-wing", "Imperial shuttle")
2	Darth Vader	202	136.0	none	white	yellow	41.9	male	Tatooine	Human	c("Revenge of the Sith", "Return of the Jedi", "The Emp...	character(0)	TIE Advanced x1
3	Leia Organa	150	49.0	brown	light	brown	19.0	female	Alderaan	Human	c("Revenge of the Sith", "Return of the Jedi", "The Emp...	Imperial Speeder Bike	character(0)
4	Owen Lars	178	120.0	brown, grey	light	blue	52.0	male	Tatooine	Human	c("Attack of the Clones", "Revenge of the Sith", "A New...	character(0)	character(0)
5	Beru Whitesun lars	165	75.0	brown	light	blue	47.0	female	Tatooine	Human	c("Attack of the Clones", "Revenge of the Sith", "A New...	character(0)	character(0)
6	Biggs Darklighter	183	84.0	black	light	brown	24.0	male	Tatooine	Human	A New Hope	character(0)	X-wing
7	Obi-Wan Kenobi	182	77.0	auburn, white	fair	blue-gray	57.0	male	Stewjon	Human	c("Attack of the Clones", "The Phantom Menace", "Rev...	Tribubble bongo	c("Jedi starfighter", "Trade Federation cruiser", "Naboo...
8	Anakin Skywalker	188	84.0	blond	fair	blue	41.9	male	Tatooine	Human	c("Attack of the Clones", "The Phantom Menace", "Rev...	c("Zephyr-G swoop bike", "XJ-6 airspeeder")	c("Trade Federation cruiser", "Jedi Interceptor", "Nabo...
9	Wilhuff Tarkin	180	NA	auburn, grey	fair	blue	64.0	male	Eriadu	Human	c("Revenge of the Sith", "A New Hope")	character(0)	character(0)
10	Han Solo	180	80.0	brown	fair	brown	29.0	male	Corellia	Human	c("Return of the Jedi", "The Empire Strikes Back", "A N...	character(0)	c("Millennium Falcon", "Imperial shuttle")
11	Wedge Antilles	170	77.0	brown	fair	hazel	21.0	male	Corellia	Human	c("Return of the Jedi", "The Empire Strikes Back", "A N...	Snowspeeder	X-wing
12	Jek Tono Porkins	180	110.0	brown	fair	blue	NA	male	Bestine IV	Human	A New Hope	character(0)	X-wing
13	Palpatine	170	75.0	grey	pale	yellow	82.0	male	Naboo	Human	c("Attack of the Clones", "The Phantom Menace", "Rev...	character(0)	character(0)
14	Boba Fett	183	78.2	black	fair	brown	31.5	male	Kamino	Human	c("Attack of the Clones", "Return of the Jedi", "The Em...	character(0)	Slave 1
15	Lando Calrissian	177	79.0	black	dark	brown	31.0	male	Socorro	Human	c("Return of the Jedi", "The Empire Strikes Back")	character(0)	Millennium Falcon
16	Lobot	175	79.0	none	light	blue	37.0	male	Bespin	Human	The Empire Strikes Back	character(0)	character(0)
17	Mon Mothma	150	NA	auburn	fair	blue	48.0	female	Chandrila	Human	Return of the Jedi	character(0)	character(0)
18	Arvel Crynyd	NA	NA	brown	fair	brown	NA	male	NA	Human	Return of the Jedi	character(0)	A-wing
19	Qui-Gon Jinn	193	89.0	brown	fair	blue	92.0	male	NA	Human	The Phantom Menace	Tribubble bongo	character(0)
20	Finis Valorum	170	NA	blond	fair	blue	91.0	male	Coruscant	Human	The Phantom Menace	character(0)	character(0)
21	Shmi Skywalker	163	NA	black	fair	brown	72.0	female	Tatooine	Human	c("Attack of the Clones", "The Phantom Menace")	character(0)	character(0)

```
dplyr::filter(starwars, species == "Human")
```

dplyr::filter() to keep rows based on values



dplyr::filter()

```
dplyr::filter(dataframe, condition(s))
```



You can also filter based on multiple conditions

Filter the starwars dataframe to include only male humans.

```
dplyr::filter(starwars, species == "Human", gender == "male")
```

Filter the starwars dataframe to include humans and droids.

```
dplyr::filter(starwars, species %in% c("Human", "Droid"))
```

dplyr::filter() to keep rows based on values



dplyr::filter()

```
dplyr::filter(dataframe, condition(s))
```

Filter starwars to include only people with height less than 100.

```
dplyr::filter(starwars, height < 100)
```

Keep all non-humans

```
dplyr::filter(starwars, species != "Human")
```

Remove characters with no known species

```
dplyr::filter(starwars, !is.na(species))
```

dplyr::filter() to keep rows based on values





pipe ($\%>\%$)

takes output of left side and makes it input of right side

Piping in Tidyverse

Filter the starwars dataframe to include only male humans.

Without pipes

```
dplyr::filter(starwars, species == "Human", gender == "male")
```

With pipes

```
starwars %>%  
  dplyr::filter(species == "Human") %>%  
  dplyr::filter(gender == "male")
```

1. start with the starwars dataframe

2. filter to keep only humans

3. filter to keep only males

**Can be useful to combine several tidyverse
“verbs” to one block of code**



takes output of left side and makes it input of right side



`dplyr::select()`

`dplyr::select()` to keep columns based on names

dplyr::select()

```
dplyr::select(dataframe, columns_to_keep)
```

starwars

name, species, films

Select only name, species, and films variables from the starwars dataframe

```
dplyr::select(starwars, name, species, films)
```



Deselect a column with -column_name

Select all columns except the gender

```
dplyr::select(starwars, -gender)
```

dplyr::select() to keep columns based on names



dplyr::select()

```
dplyr::select(dataframe, columns_to_keep)
```



Select a range of columns with column1:column2

Select columns name, height, mass, hair color, and skin color

```
dplyr::select(starwars, name:skin_color)
```

Select all columns except hair color, skin color, and eye color

```
dplyr::select(starwars, -hair_color, -skin_color, -eye_color)
```

```
dplyr::select(starwars, -(hair_color:eye_color))
```

dplyr::select() to keep columns based on names



dplyr::select()

```
dplyr::select(dataframe, columns_to_keep)
```



Reorder columns with select too!

Select columns name, height, mass, hair color, and skin color

```
dplyr::select(starwars, name:skin_color)
```

Select columns name, mass, skin color, hair color, and height (in order)

```
dplyr::select(starwars, name, mass, skin_color, hair_color, height)
```

dplyr::select() to keep columns based on names



dplyr::select()

```
dplyr::select(dataframe, new_name = old_name)
```



You can also use select() to rename columns

Rename the “name” column to “character” and the “mass” column to “weight” using select()

```
renamed <- starwars %>%  
  dplyr::select(character = name, weight = mass)
```

character	weight
Luke Skywalker	77.0
C-3PO	75.0
R2-D2	32.0
Darth Vader	136.0
Leia Organa	49.0
Owen Lars	120.0

dplyr::select() to keep columns based on names



Combine filter and select

```
dplyr::filter(dataframe, condition(s))
```

```
dplyr::select(dataframe, columns_to_keep)
```

Challenge:

[Keep: height greater than 100 &

Keep: humans &

Remove: brown hair color &

Remove: vehicles &

Keep: name, homeworld, height, species, hair color]



dplyr::filter() to keep rows based on values

dplyr::select() to keep columns based on names



Combine filter and select

	name	homeworld	height	species	hair_color
1	Luke Skywalker	Tatooine	172	Human	blond
2	Darth Vader	Tatooine	202	Human	none
3	Owen Lars	Tatooine	178	Human	brown, grey
4	Biggs Darklighter	Tatooine	183	Human	black
5	Obi-Wan Kenobi	Stewjon	182	Human	auburn, white
6	Anakin Skywalker	Tatooine	188	Human	blond
7	Wilhuff Tarkin	Eriadu	180	Human	auburn, grey
8	Palpatine	Naboo	170	Human	grey
9	Boba Fett	Kamino	183	Human	black
10	Lando Calrissian	Socorro	177	Human	black
11	Lobot	Bespin	175	Human	none
12	Mon Mothma	Chandrila	150	Human	auburn
13	Finis Valorum	Coruscant	170	Human	blond
14	Shmi Skywalker	Tatooine	163	Human	black
15	Mace Windu	Haruun Kal	188	Human	none
16	Gregar Typho	Naboo	185	Human	black
17	Dooku	Serenno	193	Human	white

```
starwars %>%
  dplyr::filter(height > 100) %>%
  dplyr::filter(species == "Human") %>%
  dplyr::filter(hair_color != "brown") %>%
  dplyr::select(-vehicles) %>%
  dplyr::select(name, homeworld, height, species, hair_color)
```



dplyr::rename()

`dplyr::rename()` to give columns new names

dplyr::rename()

```
dplyr::rename(dataframe, new_name = old_name)
```

Rename the “name” column to “character”

```
renamed <- starwars %>%  
  dplyr::rename(character = name)
```

Rename the “name” column to “character” and the “mass” column to “weight”

```
renamed <- starwars %>%  
  dplyr::rename(character = name, weight = mass)
```

dplyr::rename() to give columns new names



dplyr::rename()

```
dplyr::rename(dataframe, new_name = old_name)
```



compare `select()` and `rename()` for renaming columns

**Rename the “name” column to “character” and the “mass” column to “weight”
using `select()` and `rename()`**

`dplyr::select()`

character	weight
Luke Skywalker	77.0
C-3PO	75.0
R2-D2	32.0
Darth Vader	136.0
Leia Organa	49.0
Owen Lars	120.0

`dplyr::rename()`

character	height	weight	hair_color	skin_color	eye_color	birt
Luke Skywalker	172	77.0	blond	fair	blue	
C-3PO	167	75.0	NA	gold	yellow	
R2-D2	96	32.0	NA	white, blue	red	
Darth Vader	202	136.0	none	white	yellow	
Leia Organa	150	49.0	brown	light	brown	
Owen Lars	178	120.0	brown, grey	light	blue	

`dplyr::rename()` to give columns new names





dplyr::mutate()

`dplyr::mutate()` to add new (or change existing) columns

dplyr::mutate()

```
dplyr::mutate(dataframe, new_column = expression)
```

1. Figure out what you want to do

calculate BMI of all starwars characters

2. Describe your goal in words

cm

$$\text{BMI} = \text{weight} \text{ (kg)} / [\text{height} \text{ (m)}]^2$$

	name	height	mass	hair_color	skin_color	eye_color
1	Luke Skywalker	172	77.0	blond	fair	blue
2	C-3PO	167	75.0	NA	gold	yellow
3	R2-D2	96	32.0	NA	white, blue	red
4	Darth Vader	202	136.0	none	white	yellow

?starwars

dplyr::mutate() to add new (or change existing) columns



dplyr::mutate()

```
dplyr::mutate(dataframe, new_column = expression)
```

1. Figure out what you want to do

calculate BMI of all starwars characters

2. Describe your goal in words

$$\text{BMI} = \text{weight} \text{ (kg)} / [\text{height} \text{ (m)}]^2$$

First: convert height in cm to height in meters

Second: calculate BMI as new column

dplyr::mutate() to add new (or change existing) columns



dplyr::mutate()

```
dplyr::mutate(dataframe, new_column = expression)
```

starwars

height_m

height / 100

2. Describe your goal in words

$$\text{BMI} = \text{weight} \text{ (kg)} / [\text{height} \text{ (m)}]^2$$

First: convert height in cm to height in meters

Second: calculate BMI as new column

3. Describe your goal in code

```
new_starwars <- dplyr::mutate(starwars, height_m = height / 100)
```

dplyr::mutate() to add new (or change existing) columns



dplyr::mutate()

```
dplyr::mutate(dataframe, new_column = expression)
```

new_starwars

bmi

mass / height_m^2

2. Describe your goal in words

$$\text{BMI} = \text{weight} \text{ (kg)} / [\text{height} \text{ (m)}]^2$$

First: convert height in cm to height in meters

Second: calculate BMI as new column

3. Describe your goal in code

```
new_starwars <- starwars %>%  
  dplyr::mutate(height_m = height / 100) %>%  
  dplyr::mutate(bmi = mass / height_m^2)
```

dplyr::mutate() to add new (or change existing) columns



dplyr::mutate()

```
dplyr::mutate(dataframe, new_column = expression)
```

Select only name, height, mass, height_m, and bmi and view dataframe

```
new_starwars <- new_starwars %>%  
  dplyr::select(name, height, mass, height_m, bmi)
```

```
View(new_starwars)
```

name	height	mass	height_m	bmi
Luke Skywalker	172	77.0	1.72	26.02758
C-3PO	167	75.0	1.67	26.89232
R2-D2	96	32.0	0.96	34.72222
Darth Vader	202	136.0	2.02	33.33007
Leia Organa	150	49.0	1.50	21.77778
Owen Lars	178	120.0	1.78	37.87401
Beru Whitesun Lars	165	75.0	1.65	27.54821
R5-D4	97	32.0	0.97	34.00999
Biggs Darklighter	183	84.0	1.83	25.08286

dplyr::mutate() to add new (or change existing) columns



dplyr::mutate()

```
dplyr::mutate(dataframe, new_column = expression)
```



We can change existing columns if we use the same name

```
new_starwars <- starwars %>%  
  dplyr::mutate(height = height / 100) %>%  
  dplyr::mutate(bmi = mass / height^2)
```

View(new_starwars)

name	height	mass	bmi
Luke Skywalker	1.72	77.0	26.02758
C-3PO	1.67	75.0	26.89232
R2-D2	0.96	32.0	34.72222
Darth Vader	2.02	136.0	33.33007
Leia Organa	1.50	49.0	21.77778
Owen Lars	1.78	120.0	37.87401

dplyr::mutate() to add new (or change existing) columns



dplyr::mutate()

```
dplyr::mutate(dataframe, new_column = expression)
```

Make a new column to find the average height (in meters)

```
starwars %>%
```

```
  dplyr::mutate(height = hei
```

```
  dplyr::mutate(avg_height =
```

```
?mean
```

name	height	mass	avg_height
Palpatine	1.70	75.0	NA
Boba Fett	1.83	78.2	NA
IG-88	2.00	140.0	NA
Bossk	1.90	113.0	NA
Lando Calrissian	1.77	79.0	NA
Lobot	1.75	79.0	NA
Ackbar	1.80	83.0	NA
Mon Mothma	1.50	NA	NA
Arvel Crynyd	NA	NA	NA
Wicket Systri Warrick	0.88	20.0	NA
Nien Nunb	1.60	68.0	NA
Qui-Gon Jinn	1.93	89.0	NA
Nute Gunray	1.91	90.0	NA
Finis Valorum	1.70	NA	NA

dplyr::mutate() to add new (or change existing) columns



dplyr::mutate()

```
dplyr::mutate(dataframe, new_column = expression)
```

Make a new column to find the average height (in meters)

```
starwars %>%
  dplyr::mutate(
    dplyr::mutate(
```

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

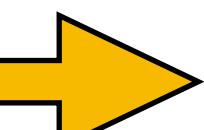
Usage

```
mean(x, ...)

## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
```

Arguments

- x** An R object. Currently there are methods for numeric/logical vectors and [date](#), [date-time](#) and [time interval](#) objects. Complex vectors are allowed for `trim = 0`, only.
- trim** the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the mean is computed. Values of `trim` outside that range are taken as the nearest endpoint.
- na.rm** a logical value indicating whether `NA` values should be stripped before the computation proceeds.
- ...** further arguments passed to or from other methods.



`dplyr::mutate()` to add new (or change existing) columns



dplyr::mutate()

```
dplyr::mutate(dataframe, new_column = expression)
```

Make a new column to find the average height (in meters)

```
starwars %>%
  dplyr::mutate(height = height / 100) %>%
dplyr::mutate(avg_height = mean(height, na.rm = TRUE))
```

name	height	mass	avg_height
Luke Skywalker	172	77.0	1.74358
C-3PO	167	75.0	1.74358
R2-D2	96	32.0	1.74358
Darth Vader	202	136.0	1.74358
Leia Organa	150	49.0	1.74358
Owen Lars	178	120.0	1.74358
Beru Whitesun Lars	165	75.0	1.74358
R5-D4	97	32.0	1.74358

dplyr::mutate() to add new (or change existing) columns



dplyr::mutate()

```
dplyr::mutate(dataframe, new_column = expression)
```

Standardize the heights of the starwars characters (height / avg_height)

```
starwars %>%  
  dplyr::mutate(height = height / 100) %>%  
  dplyr::mutate(avg_height = mean(height, na.rm = TRUE)) %>%  
  dplyr::mutate(std_height = height / avg_height)
```

name	height	mass	avg_height	std_height
Luke Skywalker	1.72	77.0	1.74358	0.9864760
C-3PO	1.67	75.0	1.74358	0.9577993
R2-D2	0.96	32.0	1.74358	0.5505912
Darth Vader	2.02	136.0	1.74358	1.1585357
Leia Organa	1.50	49.0	1.74358	0.8602988
Owen Lars	1.78	120.0	1.74358	1.0208879
Beru Whitesun Lars	1.65	75.0	1.74358	0.9463287
R5-D4	0.97	32.0	1.74358	0.5563266
Biggs Darklighter	1.83	84.0	1.74358	1.0495645
Obi-Wan Kenobi	1.82	77.0	1.74358	1.0438292

dplyr::mutate() to add new (or change existing) columns



dplyr::mutate()

```
dplyr::mutate(dataframe, new_column = expression)
```

Make a new column to see if each character is above or below the average height

hint: try using ifelse(condition, if_true_do_this, if_false_do_this)

```
test <- starwars %>%
  dplyr::mutate(height = height / 100) %>%
  dplyr::mutate(avg_height = mean(height, na.rm = TRUE)) %>%
  dplyr::mutate(std_height = height / avg_height) %>%
  dplyr::select(name, height, mass, avg_height, std_height) %>%
dplyr::mutate(relative_height =
  ifelse(std_height > 1, "above", "below")) %>%
dplyr::filter(relative_height == "below")
```

Bonus: make a new dataframe that only keeps characters with heights BELOW average

dplyr::mutate() to add new (or change existing) columns



dplyr::mutate()

```
dplyr::mutate(dataframe, new_column = expression)
```

Make a new column to see if each character is above or below the average height

hint: try using ifelse(condition, if_true_do_this, if_false_do_this)

```
test <- starwars %>%
  dplyr::mutate(height = height / 100) %>%
  dplyr::mutate(avg_height = mean(height, na.rm = TRUE)) %>%
  dplyr::mutate(std_height = height / avg_height) %>%
  dplyr::select(name, height, mass, avg_height, std_height) %>%
dplyr::mutate(relative_height =
    ifelse(std_height > 1, "above", "below")) %>%
dplyr::filter(std_height < 1)
```

Bonus: make a new dataframe that only keeps characters with heights BELOW average

dplyr::mutate() to add new (or change existing) columns



dplyr::mutate()

```
dplyr::mutate(dataframe, new_column = expression)
```

Make a new column to see if each character is above or below the average height

hint: try using ifelse(condition, if_true_do_this, if_false_do_this)

```
test <- starwars %>%
  dplyr::mutate(height = height / 100) %>%
  dplyr::mutate(avg_height = mean(height, na.rm = TRUE)) %>%
  dplyr::mutate(std_height = height / avg_height) %>%
  dplyr::select(name, height, mass, avg_height, std_height) %>%
dplyr::mutate(relative_height =
    ifelse(std_height > 1, "above", "below")) %>%
dplyr::filter(height < avg_height)
```

Bonus: make a new dataframe that only keeps characters with heights BELOW average

dplyr::mutate() to add new (or change existing) columns





dplyr::group_by()

`dplyr::group_by()` to group rows by columns

dplyr::group_by()

```
dplyr::group_by(dataframe, column_to_group_by)
```



Doesn't change how the data look, changes how the data interact with other dplyr verbs

Group starwars dataframe by gender

```
grouped_starwars <- starwars %>%  
  dplyr::group_by(gender)
```

```
View(starwars)  
View(grouped_starwars)
```

dplyr::group_by() to group rows by columns



dplyr::group_by()

dplyr::group_by(dataframe, column_to_group_by)

ungrouped

	name	height	mass	hair_color	skin_color	eye_color	birth_year	gender
1	Luke Skywalker	172	77.0	blond	fair	blue	19.0	male
2	C-3PO	167	75.0	NA	gold	yellow	112.0	NA
3	R2-D2	96	32.0	NA	white, blue	red	33.0	NA
4	Darth Vader	202	136.0	none	white	yellow	41.9	male
5	Leia Organa	150	49.0	brown	light	brown	19.0	female
6	Owen Lars	178	120.0	brown, grey	light	blue	52.0	male
7	Beru Whitesun Lars	165	75.0	brown	light	blue	47.0	female
8	R5-D4	97	32.0	NA	white, red	red	NA	NA
9	Biggs Darklighter	183	84.0	black	light	brown	24.0	male
10	Obi-Wan Kenobi	182	77.0	auburn, white	fair	blue-gray	57.0	male
11	Anakin Skywalker	188	84.0	blond	fair	blue	41.9	male
12	Wilhuff Tarkin	180	NA	auburn, grey	fair	blue	64.0	male
13	Chewbacca	228	112.0	brown	unknown	blue	200.0	male
14	Han Solo	180	80.0	brown	fair	brown	29.0	male
15	Greedo	173	74.0	NA	green	black	44.0	male
16	Jabba Desilijic Tiure	175	1358.0	NA	green-tan, brown	orange	600.0	hermaphrodite
17	Wedge Antilles	170	77.0	brown	fair	hazel	21.0	male
18	Jek Tono Porkins	180	110.0	brown	fair	blue	NA	male
19	Yoda	66	17.0	white	green	brown	896.0	male
20	Palpatine	170	75.0	grey	pale	yellow	82.0	male
21	Boba Fett	183	78.2	black	fair	brown	31.5	male
22	IG-88	200	140.0	none	metal	red	15.0	none
23	Bossk	190	113.0	none	green	red	53.0	male
24	Lando Calrissian	177	79.0	black	dark	brown	31.0	male

grouped by gender

	name	height	mass	hair_color	skin_color	eye_color	birth_year	gender
1	Luke Skywalker	172	77.0	blond	fair	blue	19.0	male
2	C-3PO	167	75.0	NA	gold	yellow	112.0	NA
3	R2-D2	96	32.0	NA	white, blue	red	33.0	NA
4	Darth Vader	202	136.0	none	white	yellow	41.9	male
5	Leia Organa	150	49.0	brown	light	brown	19.0	female
6	Owen Lars	178	120.0	brown, grey	light	blue	52.0	male
7	Beru Whitesun Lars	165	75.0	brown	light	blue	47.0	female
8	R5-D4	97	32.0	NA	white, red	red	NA	NA
9	Biggs Darklighter	183	84.0	black	light	brown	24.0	male
10	Obi-Wan Kenobi	182	77.0	auburn, white	fair	blue-gray	57.0	male
11	Anakin Skywalker	188	84.0	blond	fair	blue	41.9	male
12	Wilhuff Tarkin	180	NA	auburn, grey	fair	blue	64.0	male
13	Chewbacca	228	112.0	brown	unknown	blue	200.0	male
14	Han Solo	180	80.0	brown	fair	brown	29.0	male
15	Greedo	173	74.0	NA	green	black	44.0	male
16	Jabba Desilijic Tiure	175	1358.0	NA	green-tan, brown	orange	600.0	hermaphrodite
17	Wedge Antilles	170	77.0	brown	fair	hazel	21.0	male
18	Jek Tono Porkins	180	110.0	brown	fair	blue	NA	male
19	Yoda	66	17.0	white	green	brown	896.0	male
20	Palpatine	170	75.0	grey	pale	yellow	82.0	male
21	Boba Fett	183	78.2	black	fair	brown	31.5	male
22	IG-88	200	140.0	none	metal	red	15.0	none
23	Bossk	190	113.0	none	green	red	53.0	male
24	Lando Calrissian	177	79.0	black	dark	brown	31.0	male

dplyr::group_by() to group rows by columns



dplyr::group_by()

```
dplyr::group_by(dataframe, column_to_group_by)
```



Doesn't change how the data look, changes how the data interact with other dplyr verbs

Group starwars dataframe by gender

```
grouped_starwars <- starwars %>%  
  dplyr::group_by(gender)
```

Calculate average height *PER GENDER*

hint: group by gender FIRST

```
grouped_starwars <- starwars %>%  
  dplyr::group_by(gender) %>%  
  dplyr::mutate(avg_height = mean(height, na.rm = TRUE))
```

dplyr::group_by() to group rows by columns



dplyr::group_by()

```
dplyr::group_by(dataframe, column_to_group_by)
```

Calculate average height *PER GENDER*

name	gender	height	avg_height
Luke Skywalker	male	1.72	1.792373
C-3PO	NA	1.67	1.200000
R2-D2	NA	0.96	1.200000
Darth Vader	male	2.02	1.792373
Leia Organa	female	1.50	1.654706
Owen Lars	male	1.78	1.792373
Beru Whitesun lars	female	1.65	1.654706
R5-D4	NA	0.97	1.200000
Biggs Darklighter	male	1.83	1.792373
Obi-Wan Kenobi	male	1.82	1.792373
Anakin Skywalker	male	1.88	1.792373
Wilhuff Tarkin	male	1.80	1.792373
Chewbacca	male	2.28	1.792373
Han Solo	male	1.80	1.792373

Calculate average height

name	gender	height	avg_height
Luke Skywalker	male	1.72	1.74358
C-3PO	NA	1.67	1.74358
R2-D2	NA	0.96	1.74358
Darth Vader	male	2.02	1.74358
Leia Organa	female	1.50	1.74358
Owen Lars	male	1.78	1.74358
Beru Whitesun lars	female	1.65	1.74358
R5-D4	NA	0.97	1.74358
Biggs Darklighter	male	1.83	1.74358
Obi-Wan Kenobi	male	1.82	1.74358
Anakin Skywalker	male	1.88	1.74358
Wilhuff Tarkin	male	1.80	1.74358
Chewbacca	male	2.28	1.74358
Han Solo	male	1.80	1.74358

dplyr::group_by() to group rows by columns



dplyr::group_by()

```
dplyr::group_by(dataframe, column_to_group_by)
```

- ★ dplyr::ungroup() removes all groups

Ungroup your grouped dataframe and re-calculate average height

```
ungrouped_starwars <- grouped_starwars %>%  
  dplyr::ungroup() %>%  
  dplyr::mutate(avg_height = mean(height, na.rm = TRUE))
```

dplyr::group_by() to group rows by columns



dplyr::group_by()

```
dplyr::group_by(dataframe, column_to_group_by)
```

- ★ You can also group by multiple columns

Calculate the average height per gender AND eye color

```
grouped_starwars <- starwars %>%  
  dplyr::group_by(gender, eye_color) %>%  
  dplyr::mutate(avg_height = mean(height, na.rm = TRUE))
```

dplyr::group_by() to group rows by columns





dplyr::summarize()
dplyr::summarise()

dplyr::summarize() to condense multiple columns

dplyr::summarize()

```
dplyr::summarize(dataframe, new_column = expression)
```



summarize() is similar to mutate() but only keeps grouped columns

Calculate average height *PER GENDER*

hint: group by gender FIRST

```
grouped_starwars <- starwars %>%  
  dplyr::group_by(gender) %>%  
  dplyr::summarize(avg_height = mean(height, na.rm = TRUE))
```

gender	avg_height
NA	120.0000
female	165.4706
hermaphrodite	175.0000
male	179.2373
none	200.0000

dplyr::summarize() to condense multiple columns



dplyr::summarize()



★ summarize() is similar to mutate() but only keeps grouped columns

Compare **mutate()** and **summarize()** to calculate average height by gender

```
dplyr::mutate(avg_height = mean(height, na.rm = TRUE))
```

name	height	mass	hair_color	skin_color	eye_color	birth_year	gender	homeworld	species	films	vehicles	starships	avg_height
Luke Skywalker	172	77.0	blond	fair	blue	19.0	male	Tatooine	Human	c("Revenge of the Sith", "Return of the Jedi", "The Emp...	c("Snowspeeder", "Imperial Speeder Bike")	c("X-wing", "Imperial shuttle")	179.2373
C-3PO	167	75.0	NA	gold	yellow	112.0	NA	Tatooine	Droid	c("Attack of the Clones", "The Phantom Menace", "Rev...	character(0)	character(0)	120.0000
R2-D2	96	32.0	NA	white, blue	red	33.0	NA	Naboo	Droid	c("Attack of the Clones", "The Phantom Menace", "Rev...	character(0)	character(0)	120.0000
Darth Vader	202	136.0	none	white	yellow	41.9	male	Tatooine	Human	c("Revenge of the Sith", "Return of the Jedi", "The Emp...	character(0)	TIE Advanced x1	179.2373
Leia Organa	150	49.0	brown	light	brown	19.0	female	Alderaan	Human	c("Revenge of the Sith", "Return of the Jedi", "The Emp...	Imperial Speeder Bike	character(0)	165.4706
Owen Lars	178	120.0	brown, grey	light	blue	52.0	male	Tatooine	Human	c("Attack of the Clones", "Revenge of the Sith", "A New...	character(0)	character(0)	179.2373
Beru Whitesun Lars	165	75.0	brown	light	blue	47.0	female	Tatooine	Human	c("Attack of the Clones", "Revenge of the Sith", "A New...	character(0)	character(0)	165.4706
R5-D4	97	32.0	NA	white, red	red	NA	NA	Tatooine	Droid	A New Hope	character(0)	character(0)	120.0000

```
dplyr::summarize(avg_height = mean(height, na.rm = TRUE))
```

gender	avg_height
NA	120.0000
female	165.4706
hermaphrodite	175.0000
male	179.2373
none	200.0000

dplyr::summarize() to condense multiple columns





`dplyr::arrange()`

`dplyr::arrange()` to reorder the rows

dplyr::arrange()

```
dplyr::arrange(dataframe, variable)
```

Arrange the starwars dataframe by homeworld

```
arranged_df <- dplyr::arrange(starwars, homeworld)
```

homeworld arranged A > Z



use desc() to arrange in descending order

```
arranged_df <- dplyr::arrange(starwars, desc(homeworld))
```

homeworld arranged Z > A

dplyr::arrange() to reorder the rows



dplyr::arrange()

dplyr::arrange(dataframe, variable)

Arrange starwars by species first then height

```
arranged_df <- dplyr::arrange(starwars, species, height)
```

name	height	mass	hair_color	skin_color	eye_color	birth_year	gender	homeworld	species
Ratts Tyerell	79	15.0	none	grey, blue	unknown	NA	male	Aleen Minor	Aleena
Dexter Jettster	198	102.0	none	brown	yellow	NA	male	Ojom	Besalisk
Ki-Adi-Mundi	198	82.0	white	pale	yellow	92.0	male	Cerea	Cerean
Mas Amedda	196	NA	none	blue	blue	NA	male	Champala	Chagrian
Zam Wesell	168	55.0	blonde	fair, green, yellow	yellow	NA	female	Zolan	Clawdite
R2-D2	96	32.0	NA	white, blue	red	33.0	NA	Naboo	Droid
R5-D4	97	32.0	NA	white, red	red	NA	NA	Tatooine	Droid
C-3PO	167	75.0	NA	gold	yellow	112.0	NA	Tatooine	Droid
IG-88	200	140.0	none	metal	red	15.0	none	NA	Droid
BB8	NA	NA	none	none	black	NA	none	NA	Droid
Sebulba	112	40.0	none	grey, red	orange	NA	male	Malastare	Dug
Wicket Systri Warrick	88	20.0	brown	brown	brown	8.0	male	Endor	Ewok
Poggle the Lesser	183	80.0	none	green	yellow	NA	male	Geonosis	Geonosian
Jar Jar Binks	196	66.0	none	orange	orange	52.0	male	Naboo	Gungan

dplyr::arrange() to reorder the rows





dplyr::distinct()

`dplyr::distinct()` to keep unique rows

dplyr::distinct()

```
dplyr::distinct(dataframe, variable)
```

Keep unique species values

```
species_df <- dplyr::distinct(starwars, species)
```

The first value of each unique species is kept



use `.keep_all=TRUE` to keep all data in the data frame

```
species_df <- dplyr::distinct(starwars, species, .keep_all=TRUE)
```

dplyr::distinct() to keep unique rows



dplyr::distinct()

```
dplyr::distinct(dataframe, variable)
```

Keep unique species values

```
species_df <- dplyr::distinct(starwars, species)
```

▲	species	▼
1	Human	
2	Droid	
3	Wookiee	
4	Rodian	
5	Hutt	
6	Yoda's species	
7	Trandoshan	
8	Mon Calamari	
9	Ewok	
10	Sullustan	
11	Neimodian	
12	Gungan	
13	NA	
14	Toydarian	
15	Dug	
16	Zabrak	
17	Twi'lek	
18	Vulptereen	
19	Xexto	
20	Toong	
21	Cerean	
22	Nautolan	
23	Tholothian	
24	Iktotchi	

dplyr::distinct() to keep unique rows



dplyr::distinct()

dplyr::distinct(dataframe, variable)

Keep unique species values

```
species_df <- dplyr::distinct(starwars, species, .keep_all=TRUE)
```

#	name	height	mass	hair_color	skin_color	eye_color	birth_year	sex	gender	homeworld	species
1	Luke Skywalker	172	77.0	blond	fair	blue	19	male	masculine	Tatooine	Human
2	C-3PO	167	75.0	NA	gold	yellow	112	none	masculine	Tatooine	Droid
3	Chewbacca	228	112.0	brown	unknown	blue	200	male	masculine	Kashyyyk	Wookiee
4	Greedo	173	74.0	NA	green	black	44	male	masculine	Rodia	Rodian
5	Jabba Desilijic Tiure	175	1358.0	NA	green-tan, brown	orange	600	hermaphroditic	masculine	Nal Hutta	Hutt
6	Yoda	66	17.0	white	green	brown	896	male	masculine	NA	Yoda's species
7	Bossk	190	113.0	none	green	red	53	male	masculine	Trandosha	Trandoshan
8	Ackbar	180	83.0	none	brown mottle	orange	41	male	masculine	Mon Cala	Mon Calamari
9	Wicket Systri Warrick	88	20.0	brown	brown	brown	8	male	masculine	Endor	Ewok
10	Nien Nunb	160	68.0	none	grey	black	NA	male	masculine	Sullust	Sullustan
11	Nute Gunray	191	90.0	none	mottled green	red	NA	male	masculine	Cato Neimoidia	Neimodian
12	Jar Jar Binks	196	66.0	none	orange	orange	52	male	masculine	Naboo	Gungan
13	Ric Olié	183	NA	brown	fair	blue	NA	NA	NA	Naboo	NA
14	Watto	137	NA	black	blue, grey	yellow	NA	male	masculine	Toydaria	Toydarian
15	Sebulba	112	40.0	none	grey, red	orange	NA	male	masculine	Malastare	Dug
16	Darth Maul	175	80.0	none	red	yellow	54	male	masculine	Dathomir	Zabrak
17	Bib Fortuna	180	NA	none	pale	pink	NA	male	masculine	Ryloth	Twi'lek
18	Dud Bolt	94	45.0	none	blue, grey	yellow	NA	male	masculine	Vulpter	Vulptereen
19	Gasgano	122	NA	none	white, blue	black	NA	male	masculine	Troiken	Xexto
20	Ben Quadinaros	163	65.0	none	grey, green, yellow	orange	NA	male	masculine	Tund	Toong
21	Ki-Adi-Mundi	198	82.0	white	pale	yellow	92	male	masculine	Cerea	Cerean
22	Kit Fisto	196	87.0	none	green	black	NA	male	masculine	Glee Anselm	Nautolan
23	Adi Gallia	184	50.0	none	dark	blue	NA	female	feminine	Coruscant	Tholothian
24	Saesee Tiin	188	NA	none	pale	orange	NA	male	masculine	Iktotchi	Iktotchi

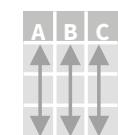
dplyr::distinct() to keep unique rows



Data transformation with dplyr :: CHEAT SHEET



dplyr functions work with pipes and expect **tidy data**. In tidy data:



&



Each **variable** is in its own **column**

Each **observation**, or **case**, is in its own **row**



$x \%>% f(y)$ becomes $f(x, y)$

Summarise Cases

Apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).

summary function →

- **summarise(.data, ...)**
Compute table of summaries.
`summarise(mtcars, avg = mean(mpg))`
- **count(.data, ..., wt = NULL, sort = FALSE, name = NULL)**
Count number of rows in each group defined by the variables in ... Also **tally()**.
`count(mtcars, cyl)`

Group Cases

Use **group_by(.data, ..., .add = FALSE, .drop = TRUE)** to create a "grouped" copy of a table grouped by columns in ... dplyr functions will manipulate each "group" separately and combine the results.



Use **rowwise(.data, ...)** to group data into individual rows. dplyr functions will compute results for each row. Also apply functions to list-columns. See tidyverse cheat sheet for list-column workflow.



ungroup(x, ...) Returns ungrouped copy of table.
`ungroup(g_mtcars)`

Manipulate Cases

EXTRACT CASES

Row functions return a subset of rows as a new table.

- **filter(.data, ..., .preserve = FALSE)** Extract rows that meet logical criteria.
`filter(mtcars, mpg > 20)`
- **distinct(.data, ..., .keep_all = FALSE)** Remove rows with duplicate values.
`distinct(mtcars, gear)`
- **slice(.data, ..., .preserve = FALSE)** Select rows by position.
`slice(mtcars, 10:15)`
- **slice_sample(.data, ..., n, prop, weight_by = NULL, replace = FALSE)** Randomly select rows. Use n to select a number of rows and prop to select a fraction of rows.
`slice_sample(mtcars, n = 5, replace = TRUE)`
- **slice_min(.data, order_by, ..., n, prop, with_ties = TRUE)** and **slice_max()** Select rows with the lowest and highest values.
`slice_min(mtcars, mpg, prop = 0.25)`
- **slice_head(.data, ..., n, prop)** and **slice_tail()**
Select the first or last rows.
`slice_head(mtcars, n = 5)`

Logical and boolean operators to use with filter()

<code>==</code>	<code><</code>	<code><=</code>	<code>is.na()</code>	<code>%in%</code>	<code> </code>	<code>xor()</code>
<code>!=</code>	<code>></code>	<code>>=</code>	<code>!is.na()</code>	<code>!</code>	<code>&</code>	

See [?base::Logic](#) and [?Comparison](#) for help.

ARRANGE CASES

- **arrange(.data, ..., .by_group = FALSE)** Order rows by values of a column or columns (low to high), use with **desc()** to order from high to low.
`arrange(mtcars, mpg)`
`arrange(mtcars, desc(mpg))`

ADD CASES

- **add_row(.data, ..., .before = NULL, .after = NULL)**
Add one or more rows to a table.
`add_row(cars, speed = 1, dist = 1)`

Manipulate Variables

EXTRACT VARIABLES

Column functions return a set of columns as a new vector or table.

- **pull(.data, var = -1, name = NULL, ...)** Extract column values as a vector, by name or index.
`pull(mtcars, wt)`
- **select(.data, ...)** Extract columns as a table.
`select(mtcars, mpg, wt)`
- **relocate(.data, ..., .before = NULL, .after = NULL)**
Move columns to new position.
`relocate(mtcars, mpg, cyl, .after = last_col())`

Use these helpers with select() and across()

e.g. `select(mtcars, mpg:cyl)`

<code>contains(match)</code>	<code>num_range(prefix, range)</code>	:	e.g. <code>mpg:cyl</code>
<code>ends_with(match)</code>	<code>all_of(x)/any_of(x, ..., vars)</code>	-	e.g. <code>gear</code>
<code>starts_with(match)</code>	<code>matches(match)</code>		<code>everything()</code>

MANIPULATE MULTIPLE VARIABLES AT ONCE

- **across(.cols, .funs, ..., .names = NULL)** Summarise or mutate multiple columns in the same way.
`summarise(mtcars, across(everything(), mean))`
- **c_across(.cols)** Compute across columns in row-wise data.
`transmute(rowwise(UKgas), total = sum(c_across(1:2)))`

MAKE NEW VARIABLES

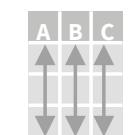
Apply **vectorized functions** to columns. Vectorized functions take vectors as input and return vectors of the same length as output (see back).

- **vectorized function**
- **mutate(.data, ..., .keep = "all", .before = NULL, .after = NULL)** Compute new column(s). Also **add_column()**, **add_count()**, and **add_tally()**.
`mutate(mtcars, gpm = 1 / mpg)`
- **transmute(.data, ...)** Compute new column(s), drop others.
`transmute(mtcars, gpm = 1 / mpg)`
- **rename(.data, ...)** Rename columns. Use **rename_with()** to rename with a function.
`rename(cars, distance = dist)`

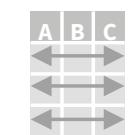
Data transformation with dplyr :: CHEAT SHEET



dplyr functions work with pipes and expect **tidy data**. In tidy data:



&



Each **variable** is in its own **column**

Each **observation**, or **case**, is in its own **row**

pipes
 $x \%>% f(y)$ becomes $f(x, y)$

Summarise Cases

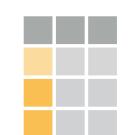
Apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).

summary function



summarise(.data, ...)

Compute table of summaries.
summarise(mtcars, avg = mean(mpg))

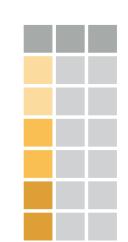


count(.data, ..., wt = NULL, sort = FALSE, name = NULL) Count number of rows in each group defined by the variables in ... Also **tally()**.
count(mtcars, cyl)

Group Cases

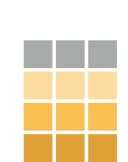


Use **group_by(.data, ..., .add = FALSE, .drop = TRUE)** to create a "grouped" copy of a table grouped by columns in ... dplyr functions will manipulate each "group" separately and combine the results.



mtcars %>% group_by(cyl) %>% summarise(avg = mean(mpg))

Use **rowwise(.data, ...)** to group data into individual rows. dplyr functions will compute results for each row. Also apply functions to list-columns. See tidyverse cheat sheet for list-column workflow.



starwars %>% rowwise() %>% mutate(film_count = length(films))



ungroup(x, ...) Returns ungrouped copy of table.
ungroup(g_mtcars)

Manipulate Cases

EXTRACT CASES

Row functions return a subset of rows as a new table.

filter(.data, ..., .preserve = FALSE) Extract rows that meet logical criteria.
filter(mtcars, mpg > 20)

distinct(.data, ..., .keep_all = FALSE) Remove rows with duplicate values.
distinct(mtcars, gear)

slice(.data, ..., .preserve = FALSE) Select rows by position.
slice(mtcars, 10:15)

slice_sample(.data, ..., n, prop, weight_by = NULL, replace = FALSE) Randomly select rows. Use n to select a number of rows and prop to select a fraction of rows.
slice_sample(mtcars, n = 5, replace = TRUE)

slice_min(.data, order_by, ..., n, prop, with_ties = TRUE) and **slice_max()** Select rows with the lowest and highest values.
slice_min(mtcars, mpg, prop = 0.25)

slice_head(.data, ..., n, prop) and **slice_tail()** Select the first or last rows.
slice_head(mtcars, n = 5)

Logical and boolean operators to use with filter()

<code>==</code>	<code><</code>	<code><=</code>	<code>is.na()</code>	<code>%in%</code>	<code> </code>	<code>xor()</code>
<code>!=</code>	<code>></code>	<code>>=</code>	<code>!is.na()</code>	<code>!</code>	<code>&</code>	

See [?base::Logic](#) and [?Comparison](#) for help.

ARRANGE CASES

arrange(.data, ..., .by_group = FALSE) Order rows by values of a column or columns (low to high), use with **desc()** to order from high to low.
arrange(mtcars, mpg)
arrange(mtcars, desc(mpg))

ADD CASES

add_row(.data, ..., .before = NULL, .after = NULL) Add one or more rows to a table.
add_row(cars, speed = 1, dist = 1)

Manipulate Variables

EXTRACT VARIABLES

Column functions return a set of columns as a new vector or table.

pull(.data, var = -1, name = NULL, ...) Extract column values as a vector, by name or index.
pull(mtcars, wt)

select(.data, ...) Extract columns as a table.
select(mtcars, mpg, wt)

relocate(.data, ..., .before = NULL, .after = NULL) Move columns to new position.
relocate(mtcars, mpg, cyl, .after = last_col())

Use these helpers with select() and across()

e.g. `select(mtcars, mpg:cyl)`

`contains(match)`

`ends_with(match)`

`starts_with(match)`

`num_range(prefix, range)` ; e.g. `mpg:cyl`

`all_of(x)/any_of(x, ..., vars)` ; e.g., `gear`

`matches(match)`

`everything()`

MANIPULATE MULTIPLE VARIABLES AT ONCE

across(.cols, .funs, ..., .names = NULL) Summarise or mutate multiple columns in the same way.
summarise(mtcars, across(everything(), mean))

c_across(.cols) Compute across columns in row-wise data.
transmute(rowwise(UKgas), total = sum(c_across(1:2)))

MAKE NEW VARIABLES

Apply **vectorized functions** to columns. Vectorized functions take vectors as input and return vectors of the same length as output (see back).

vectorized function

mutate(.data, ..., .keep = "all", .before = NULL, .after = NULL) Compute new column(s). Also **add_column()**, **add_count()**, and **add_tally()**.
mutate(mtcars, gpm = 1 / mpg)

transmute(.data, ...) Compute new column(s), drop others.
transmute(mtcars, gpm = 1 / mpg)

rename(.data, ...) Rename columns. Use **rename_with()** to rename with a function.
rename(cars, distance = dist)

Tidyverse resources

- <https://www.tidyverse.org>
- <https://www.rstudio.com/resources/cheatsheets>
- <https://cran.r-project.org/web/packages/dplyr/vignettes/dplyr.html>
- <https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html>

