Dplyr

Jennifer Brosnahan

7/23/2020

The purpose of this project is to explore basic data manipulation verbs of dplyr.

Dataset = starwars

Summary of dplyr - 5 useful commands

- arrange() = reorder rows
- filter() = pick observations of interest
- select() = pick variables of interest
- mutate() = add new variables that are functions of existing variables
- summarise() = collapse many values to a summary

Loading packages

```
library(tidyverse)
library(dplyr)
```

Importing data

```
starwars <- read.csv(file.path('C:/Users/jlbro/OneDrive/R Studio projects/Tidy Data', 'starwars.csv'))
dim(starwars)
```

```
## [1] 87 10
```

```
head(starwars)
```

	name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>	eye_color <chr>	birth_year <chr>
1	Luke Skywalker	172	77	blond	fair	blue	19BBY
2	C-3PO	167	75	*NA*	gold	yellow	112BBY
3	R2-D2	96	32	*NA*	white, blue	red	33BBY
4	Darth Vader	202	136	none	white	yellow	41.9BBY
5	Leia Organa	150	49	brown	light	brown	19BBY
6	Owen Lars	178	120	brown, grey	light	blue	52BBY

6 rows | 1-8 of 11 columns

Filter rows with filter()

Allows you to subset rows in a data frame

```
starwars %>% filter(skin_color == 'light', eye_color == 'brown')
```

name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>	eye_color <chr>	birth_year <chr>

name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>	eye_color <chr>	birth_year <chr>
Leia Organa	150	49	brown	light	brown	19BBY
Biggs Darklighter	183	84	black	light	brown	24BBY
CordÃ©	157	*NA*	brown	light	brown	*NA*
DormÃ©	165	*NA*	brown	light	brown	*NA*
Raymus Antilles	188	79	brown	light	brown	*NA*
Poe Dameron	*NA*	*NA*	brown	light	brown	*NA*
PadmÃ© Amidala	165	45	brown	light	brown	46BBY

7 rows | 1-7 of 10 columns

Arrange rows with arrange(). Similar to filter() except that instead of filtering or selecting rows, it reorders them. It takes a dataframe, and a set of column names to order by.

```
starwars %>% arrange(height, mass)
```

name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>
Yoda	66	17	white	green
Ratts Tyerell	79	15	none	grey, blue
Wicket Systri Warrick	88	20	brown	brown
Dud Bolt	94	45	none	blue, grey
R2-D2	96	32	*NA*	white, blue
R4-P17	96	*NA*	none	silver, red
R5-D4	97	32	*NA*	white, red
Sebulba	112	40	none	grey, red
Gasgano	122	*NA*	none	white, blue
Watto	137	*NA*	black	blue, grey

1-10 of 87 rows | 1-5 of 10 columns

Previous **1** 2 3 4 5 6 … 9 Next

Use desc() to order a column in descending order

```
starwars %>% arrange(desc(height, mass))
```

name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>
Yarael Poof	264	*NA*	none	white
Tarfful	234	136	brown	brown
Lama Su	229	88	none	grey
Chewbacca	228	112	brown	*NA*
Roos Tarpals	224	82	none	grey
Grievous	216	159	none	brown, white

name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>
Taun We	213	*NA*	none	grey
Rugor Nass	206	*NA*	none	green
Tion Medon	206	80	none	grey
Darth Vader	202	136	none	white

1-10 of 87 rows | 1-5 of 10 columns

Previous **1** 2 3 4 5 6 … 9 Next

Slice: Choose rows using their position with slice(). Selecting row numbers 5 through 10.

```
starwars %>% slice(5:10)
```

name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>	eye_color <chr>	birth_year <chr>
Leia Organa	150	49	brown	light	brown	19BBY
Owen Lars	178	120	brown, grey	light	blue	52BBY
Beru Whitesun lars	165	75	brown	light	blue	47BBY
R5-D4	97	32	*NA*	white, red	red	*NA*
Biggs Darklighter	183	84	black	light	brown	24BBY
Obi-Wan Kenobi	182	77	auburn, white	fair	blue-gray	57BBY

6 rows | 1-7 of 10 columns

Slice: You can slice the head or tail of a df

```
starwars %>% slice_head(n = 3)
```

name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>	eye_color <chr>	birth_year <chr>
Luke Skywalker	172	77	blond	fair	blue	19BBY
C-3PO	167	75	*NA*	gold	yellow	112BBY
R2-D2	96	32	*NA*	white, blue	red	33BBY

3 rows | 1-7 of 10 columns

Slice: You can slice a SAMPLE of a df

```
starwars %>% slice_sample(n = 5)
```

name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>	eye_color <chr>	birth_year <chr>
Mas Amedda	196	*NA*	none	blue	blue	*NA*
Jar Jar Binks	196	66	none	orange	orange	52BBY
Nien Nunb	160	68	none	grey	black	*NA*
Gregar Typho	185	85	black	dark	brown	*NA*
Obi-Wan Kenobi	182	77	auburn, white	fair	blue-gray	57BBY

5 rows | 1-7 of 10 columns

Slice: You can slice a proportion of a sample using prop=

```
starwars %>% slice_sample(prop = .1)
```

name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>	eye_color <chr>	birth_year <chr>
Bail Prestor Organa	191	*NA*	black	tan	brown	67BBY
Dooku	193	80	white	fair	brown	102BBY
Saesee Tiin	188	*NA*	none	pale	orange	*NA*
Finn	*NA*	*NA*	black	dark	dark	*NA*
Boba Fett	183	78.2	black	fair	brown	31.5BBY
Rugor Nass	206	*NA*	none	green	orange	*NA*
Ayla Secura	178	55	none	blue	hazel	48BBY
Kit Fisto	196	87	none	green	black	*NA*

8 rows | 1-7 of 10 columns

Slice: Use slice_min or slice_max to select rows with highest or lowest values of a variable. Note that we first must choose only values which are not NA.

```
starwars %>%
  filter(!is.na(height)) %>%
  slice_max(height, n = 3)
```

name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>	eye_color <chr>	birth_year <chr>	gender <chr>
Yarael Poof	264	*NA*	none	white	yellow	*NA*	male
Tarfful	234	136	brown	brown	blue	*NA*	male
Lama Su	229	88	none	grey	black	*NA*	male

3 rows | 1-8 of 10 columns

Select: Select columns with select(). Often, only a few columns are actually needed in large df's. Select() allows you to rapidly zoom in on useful subset

```
# Select columns by name
starwars %>% select(hair_color, skin_color, eye_color)
```

hair_color <chr>	skin_color <chr>	eye_color <chr>
blond	fair	blue
NA	gold	yellow
NA	white, blue	red
none	white	yellow
brown	light	brown
brown, grey	light	blue

hair_color <chr>	skin_color <chr>	eye_color <chr>
brown	light	blue
NA	white, red	red
black	light	brown
auburn, white	fair	blue-gray

1-10 of 87 rows Previous **1** 2 3 4 5 6 … 9 Next

```
# Select all columns between hair_color and eye_color (inclusive)
starwars %>% select(hair_color:eye_color)
```

hair_color <chr>	skin_color <chr>	eye_color <chr>
blond	fair	blue
NA	gold	yellow
NA	white, blue	red
none	white	yellow
brown	light	brown
brown, grey	light	blue
brown	light	blue
NA	white, red	red
black	light	brown
auburn, white	fair	blue-gray

1-10 of 87 rows Previous **1** 2 3 4 5 6 … 9 Next

```
# Select all columns except those from hair_color to eye_color
starwars %>% select(-(hair_color:eye_color))
```

name <chr>	height <int>	mass <chr>	birth_year <chr>	gender <chr>	homeworld <chr>
Luke Skywalker	172	77	19BBY	male	Tatooine
C-3PO	167	75	112BBY	*NA*	Tatooine
R2-D2	96	32	33BBY	*NA*	Naboo
Darth Vader	202	136	41.9BBY	male	Tatooine
Leia Organa	150	49	19BBY	female	Alderaan
Owen Lars	178	120	52BBY	male	Tatooine
Beru Whitesun lars	165	75	47BBY	female	Tatooine
R5-D4	97	32	*NA*	*NA*	Tatooine
Biggs Darklighter	183	84	24BBY	male	Tatooine
Obi-Wan Kenobi	182	77	57BBY	male	Stewjon

1-10 of 87 rows | 1-6 of 7 columns Previous **1** 2 3 4 5 6 … 9 Next

```
# Select all columns ending with color
starwars %>% select(ends_with('color'))
```

hair_color <chr>	skin_color <chr>	eye_color <chr>
blond	fair	blue
NA	gold	yellow
NA	white, blue	red
none	white	yellow
brown	light	brown
brown, grey	light	blue
brown	light	blue
NA	white, red	red
black	light	brown
auburn, white	fair	blue-gray

1-10 of 87 rows Previous **1** 2 3 4 5 6 … 9 Next

Rename variables using rename()

```
starwars %>% rename(home_world = homeworld)
```

name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>
Luke Skywalker	172	77	blond	fair
C-3PO	167	75	*NA*	gold
R2-D2	96	32	*NA*	white, blue
Darth Vader	202	136	none	white
Leia Organa	150	49	brown	light
Owen Lars	178	120	brown, grey	light
Beru Whitesun lars	165	75	brown	light
R5-D4	97	32	*NA*	white, red
Biggs Darklighter	183	84	black	light
Obi-Wan Kenobi	182	77	auburn, white	fair

1-10 of 87 rows | 1-5 of 10 columns Previous **1** 2 3 4 5 6 … 9 Next

Mutate: Adding new columns!

```
starwars %>% mutate(height_m = height/100)
```

name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>
Luke Skywalker	172	77	blond	fair
C-3PO	167	75	*NA*	gold

name <chr>	height <int>	mass <chr>	hair_color <chr>	skin_color <chr>
R2-D2	96	32	*NA*	white, blue
Darth Vader	202	136	none	white
Leia Organa	150	49	brown	light
Owen Lars	178	120	brown, grey	light
Beru Whitesun lars	165	75	brown	light
R5-D4	97	32	*NA*	white, red
Biggs Darklighter	183	84	black	light
Obi-Wan Kenobi	182	77	auburn, white	fair

1-10 of 87 rows | 1-5 of 11 columns

Previous **1** 2 3 4 5 6 … 9 Next

```
# We can't see the height in meters we just calculated, but we can by using select command
starwars %>%
  mutate(height_m = height/100) %>%
  select(height_m, height, everything())
```

height_m <dbl>	height <int>	name <chr>	mass <chr>	hair_color <chr>
1.72	172	Luke Skywalker	77	blond
1.67	167	C-3PO	75	*NA*
0.96	96	R2-D2	32	*NA*
2.02	202	Darth Vader	136	none
1.50	150	Leia Organa	49	brown
1.78	178	Owen Lars	120	brown, grey
1.65	165	Beru Whitesun lars	75	brown
0.97	97	R5-D4	32	*NA*
1.83	183	Biggs Darklighter	84	black
1.82	182	Obi-Wan Kenobi	77	auburn, white

1-10 of 87 rows | 1-5 of 11 columns

Previous **1** 2 3 4 5 6 … 9 Next

Relocate: Change column order

```
starwars %>% relocate(gender:homeworld, .before = height)
```

name <chr>	gender <chr>	homeworld <chr>	height <int>	mass <chr>
Luke Skywalker	male	Tatooine	172	77
C-3PO	*NA*	Tatooine	167	75
R2-D2	*NA*	Naboo	96	32
Darth Vader	male	Tatooine	202	136
Leia Organa	female	Alderaan	150	49
Owen Lars	male	Tatooine	178	120

name <chr>	gender <chr>	homeworld <chr>	height <int>	mass <chr>
Beru Whitesun lars	female	Tatooine	165	75
R5-D4	*NA*	Tatooine	97	32
Biggs Darklighter	male	Tatooine	183	84
Obi-Wan Kenobi	male	Stewjon	182	77

1-10 of 87 rows | 1-5 of 10 columns

Previous **1** 2 3 4 5 6 … 9 Next

Summarise values with summarise()

```
starwars %>% summarise(height = mean(height, na.rm = TRUE))
```

height <dbl>
174.358

1 row

Combining functions with %>%

This is difficult code

```
summarise(
  select(
    group_by(starwars, species, gender),
    height, mass),
  height = mean(height, na.rm = TRUE),
  mass = mean(mass, na.rm = TRUE))
```

```
## Adding missing grouping variables: `species`, `gender`
```

```
## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA
```

```
## `summarise()` regrouping output by 'species' (override with `.groups` argument)
```

species <chr>	gender <chr>	height <dbl>	mass <dbl>
Aleena	male	79.0000	*NA*
Besalisk	male	198.0000	*NA*
Cerean	male	198.0000	*NA*
Chagrian	male	196.0000	*NA*
Clawdite	female	168.0000	*NA*
Droid	none	200.0000	*NA*
Droid	*NA*	120.0000	*NA*
Dug	male	112.0000	*NA*
Ewok	male	88.0000	*NA*
Geonosian	male	183.0000	*NA*

Easier way to code using %>%!!!

```
starwars %>%
  group_by(species, gender) %>%
  select(height, mass) %>%
  summarise(
    height = mean(height, na.rm = TRUE),
    mass = mean(mass, na.rm = TRUE)
  )
```

```
## Adding missing grouping variables: `species`, `gender`
```

```
## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(mass, na.rm = TRUE): argument is not numeric or logical:
## returning NA
```

```
## `summarise()` regrouping output by 'species' (override with `.groups` argument)
```

species <chr>	gender <chr>	height <dbl>	mass <dbl>
Aleena	male	79.0000	*NA*
Besalisk	male	198.0000	*NA*
Cerean	male	198.0000	*NA*
Chagrian	male	196.0000	*NA*
Clawdite	female	168.0000	*NA*
Droid	none	200.0000	*NA*
Droid	*NA*	120.0000	*NA*
Dug	male	112.0000	*NA*
Ewok	male	88.0000	*NA*
Geonosian	male	183.0000	*NA*