# Basics of "dplyr"

Tidyverse

Gaston Sanchez

CC BY-NC-SA 4.0

STAT 33B, Fall 2025

### About

In this slides we provide a quick introduction to the R package "dplyr", which is part of the so-called "tidyverse".

### The Tidyverse

"tidyverse" is a set of packages for doing data science in  ${\sf R}$ 

https://www.tidyverse.org

Hadley Wickham is the leading author of the initial packages (they used to be referred as the "hadleyverse").

Nowadays, Tidyverse packages are made by many of the same people that make RStudio.

## Tidyverse Packages

They provide alternatives to R's built-in tools for:

- Reading files (package "readr")
- Manipulating data frames (packages "dplyr", "tidyr", "tibble")
- Making visualizations (package "ggplot2")
- Manipulating strings (package "stringr")
- Manipulating factors (package "forcats")
- ► Functional programming (package "purrr")

### The Tidyverse

The Tidyverse packages are popular but controversial, because some of them use a syntax different from base R.

library(tidyverse)

RStudio cheat sheets (mostly for Tidyverse packages):

https://rstudio.com/resources/cheatsheets/

# About tibbles

### Motivation

To illustrate some of the ideas presented in this part of the course, I'll use variations of a toy data example

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

### **Tibbles**

A **tibble** (deformation of the word "table") is Tidyverse's improved version of an R data frame.

Compared to an ordinary data frame, a tibble:

- Prints differently
- Default to drop = FALSE for the subset operator [
- Don't allow partial matching for the dollar operator \$

For all intents and purposes, treat tibbles as data frames.

### Toy Example

```
# data.frame
dat <- data.frame(</pre>
 name = c('Anakin', 'Padme', 'Luke', 'Leia'),
 gender = c('male', 'female', 'male', 'female'),
 height = c(1.88, 1.65, 1.72, 1.50)
# tibble
tbl <- tibble(
 name = c('Anakin', 'Padme', 'Luke', 'Leia'),
 gender = c('male', 'female', 'male', 'female'),
 height = c(1.88, 1.65, 1.72, 1.50)
```

## Toy Example

## 3 Luke male 1.72 ## 4 Leia female 1.5

```
dat
##
      name gender height
## 1
    Anakin male 1.88
    Padme female 1.65
## 2
## 3 Luke male 1.72
## 4 Leia female 1.50
tbl
## # A tibble: 4 x 3
##
    name gender height
    <chr> <chr> <dbl>
##
## 1 Anakin male 1.88
## 2 Padme female 1.65
```

### data.frame versus tibble

For the tibble, using [ to subset a single value **does not** drop the data frame:

```
class(dat[, 1])
## [1] "character"
class(dat[, 1, drop = FALSE])
## [1] "data.frame"
class(tbl[, 1])
## [1] "tbl_df" "tbl" "data.frame"
```

### data.frame versus tibble

For the tibble, the dollar operator \$ does not allow partial matches:

```
# partial match (column account)
dat$acc

## NULL

tbl$acc

## Warning: Unknown or uninitialised column: `acc`.
## NULL
```

### data.frame and tibble

There are as functions to convert from/to tibbles:

```
# Convert tibble to data frame
class(as.data.frame(tbl))

## [1] "data.frame"

# Convert data frame to tibble
class(as_tibble(dat))

## [1] "tbl_df" "tbl" "data.frame"
```

# "dplyr" main verbs

- ▶ filter()
- ▶ select()
- ▶ slice()
- arrange()
- mutate()
- group\_by()
- summarise()

# Structure of "dplyr" verbs

- First argument is a data frame (or tibble)
- Subsequent arguments say what to do with data frame
- ► Always return a data frame (or tibble)
- Never modify in place

# slice

Select rows based on index positions

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

name	gender	height
Anakin	male	1.88

slice(dat, 1)

dat

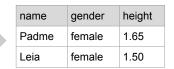
name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

name	gender	height
Anakin	male	1.88
Padme	female	1.65

slice(dat, 1:2)

### dat

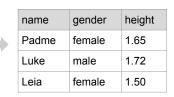
name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



slice(dat, c(2, 4))

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



slice(dat, -1)

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

name	gender	height
Padme	female	1.65
Leia	female	1.50

$$slice(dat, -c(1,3))$$

# select

Select one or more columns

#### dat

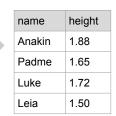
name	gender	height	name
Anakin	male	1.88	Anakin
Padme	female	1.65	Padme
Luke	male	1.72	Luke
Leia	female	1.50	Leia

```
# equivalent
select(dat, "name")
```

select(dat, name)

#### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



```
select(dat, name, height)
# equivalent
select(dat, "name", "height")
```

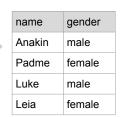
### dat

name	gender	height		height	name
Anakin	male	1.88	$\Rightarrow$	1.88	Anakin
Padme	female	1.65		1.65	Padme
Luke	male	1.72		1.72	Luke
Leia	female	1.50		1.50	Leia

select(dat, height, name)

### dat

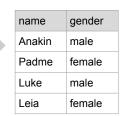
name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



select(dat, -height)

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



select(dat, name:gender)

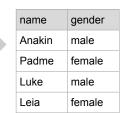
### dat

name	gender	height	name
Anakin	male	1.88	Anakin
Padme	female	1.65	Padme
Luke	male	1.72	Luke
Leia	female	1.50	Leia

select(dat, 1)

### dat

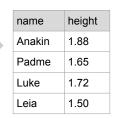
name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



select(dat, 1:2)

### dat

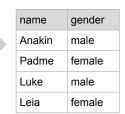
name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



select(dat, c(1, 3))

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



select(dat, -3)

# filter

Select (subset) rows based on a condition

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

name	gender	height
Padme	female	1.65
Leia	female	1.50

filter(dat, gender == "female")

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

name	gender	height
Luke	male	1.72

```
filter(dat, name == "Luke")
```

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

name	gender	height
Luke	male	1.72
Leia	female	1.50

filter(dat, name %in% c("Luke", "Leia"))

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

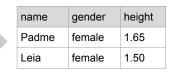
name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72

```
filter(dat, name != "Leia")
```

## filter: example 5

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

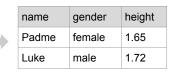


filter(dat, height < 1.70)</pre>

## filter: example 6

dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



filter(dat, height > 1.6 & height < 1.8)</pre>

### filter: example 7

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

name	gender	height
Padme	female	1.65

## arrange

Arrange rows based on values of one or more columns

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



arrange(dat, name)

### dat

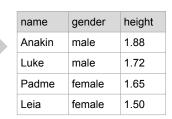
name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



arrange(dat, height)

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



arrange(dat, desc(height))

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



arrange(dat, gender)

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

name	gender	height
Padme	female	1.65
Leia	female	1.50
Luke	male	1.72
Anakin	male	1.88

arrange(dat, gender, desc(name))

## mutate

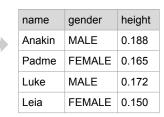
Add new columns or transform existing columns

### dat

name	gender	height	name	gender	height
Anakin	male	1.88	Anakin	male	0.188
Padme	female	1.65	Padme	female	0.165
Luke	male	1.72	Luke	male	0.172
Leia	female	1.50	Leia	female	0.150

mutate(dat, height = height / 10)

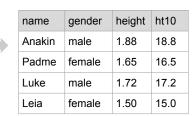
name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



```
mutate(dat,
    height = height / 10,
    gender = toupper(gender))
```

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



mutate(dat, ht10 = height \* 10)

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



```
mutate(dat, num = row_number())
# equivalent
mutate(dat, num = 1:n())
```

# **Grouped Summaries**

Summarize data, and grouped-by operations

## summarize: example 1

name	gender	height		total
Anakin	male	1.88	$\Rightarrow$	6.75
Padme	female	1.65		
Luke	male	1.72		
Leia	female	1.50		

```
summarize(dat, total = sum(height))
# equivalent
summarise(dat, total = sum(height))
```

## summarize: example 2

### dat

name	gender	height	avg
Anakin	male	1.88	1.6875
Padme	female	1.65	
Luke	male	1.72	
Leia	female	1.50	

summarize(dat, avg = mean(height))

## summarize: example 3

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

avg	med
1.6875	1.685

```
summarize(dat,
  avg = mean(height),
  med = median(height))
```

name	gender	height		gender	min
Anakin	male	1.88	$\Rightarrow$	female	1.58
Padme	female	1.65		male	1.8
Luke	male	1.72			
Leia	female	1.50			

```
by_gender <- group_by(dat, gender)
summarize(by_gender, avg = mean(height))</pre>
```

### dat

name	gender	height	gender
Anakin	male	1.88	female
Padme	female	1.65	male
Luke	male	1.72	
Leia	female	1.50	

```
by_gender <- group_by(dat, gender)
summarize(by gender, min = min(height))</pre>
```

avg

1.5

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

gender	min	max
female	1.5	1.65
male	1.72	1.88

```
by_gender <- group_by(dat, gender)
summarize(by_gender,
  min = min(height),
  max = max(height))</pre>
```

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

gender	avg	sd
female	1.58	0.106
male	1.8	0.113

```
summarize(
  group_by(dat, gender),
  avg = mean(height),
  sd = sd(height))
```

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50

gender	avg	sd
male	1.8	0.113
female	1.58	0.106

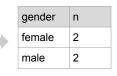
```
arrange(
   summarize(group_by(dat, gender),
     avg = mean(height),
   sd = sd(height)),
   desc(avg))
```

# Other Functions

## other examples

### dat

name	gender	height
Anakin	male	1.88
Padme	female	1.65
Luke	male	1.72
Leia	female	1.50



count(dat, gender)

## other examples

### dat

name	gender	height	gender
Anakin	male	1.88	male
Padme	female	1.65	female
Luke	male	1.72	
Leia	female	1.50	

distinct(dat, gender)

n\_distinct(select(dat, gender)) ---- 2