

Lecture 16: Oct 10, 2018

Grammar of Data

- *Interrogating Data*
- *dplyr*
- *Split-Apply-Combine*
- *Resources*

James Balamuta
STAT 385 @ UIUC



Announcements

- **hw06** is due **Friday, Oct 12th, 2018** at **6:00 PM**
- **Office Hour Changes**
 - **John Lee's** are now from **4 - 5 PM** on **WF**
 - **Hassan Kamil's** are now from **2:30 - 3:30 PM** on **TR**
- **Quiz 07** covers Week 6 contents @ [CBTF](#).
 - Window: Oct 9th - 11th
 - Sign up: <https://cbtf.engr.illinois.edu/sched>
- Want to review your homework or quiz grades?
Schedule an appointment.

Last Time

- **Designing a Graphic**

- Emphasis the data's narrative.
- Be ware of *Simpson's paradox*, *Apophenia*, and lying with graphics.

- **CRAP**

- **C**ontrast, **R**epetition, **A**lignment, **P**roximity
- Tenets of Gestalt Design

- **Chart Junk**

- Useless embellishment on the plot that impacts clarity of plot

- **Modern Graphics**

- Ability to modify display or view data over time.

Lecture Objectives

- Deriving appropriate domain questions for data.
- Describe the three stages of Split-Apply-Combine.
- Explain and apply the grammar of data to manipulate data.

Interrogating Data

Example Data

... tidied version of enrollment figures ...

Year	Gender	Enrolled
Undergrad	Men	18,345
Undergrad	Women	15,267
Undergrad	Unknown	12
Professional	Men	352
Professional	Women	640
Professional	Unknown	0
Graduate	Men	7,173
Graduate	Women	6,028
Graduate	Unknown	9

9 x 3

enrolled_fa2017

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

Questions

... digging into the data ...

1. What **variable holds** admission figures / gender / class?

```
enrolled_fa2017$Enrollment  
enrolled_fa2017[["Gender"]]  
enrolled_fa2017[, "Year"]
```

2. Are undergraduates **found** in the data?

```
enrolled_fa2017$Year == "Undergraduate"
```

3. Who has **highest enrollment** amount?

```
enrolled_fa2017[enrolled_fa2017$Enrollment == max(enrolled_fa2017$Enrollment), ]
```

What happened here?

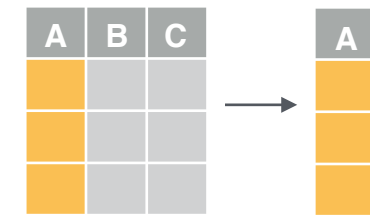
Data Wrangling

Manipulating raw data through transformations to obtain a useful format

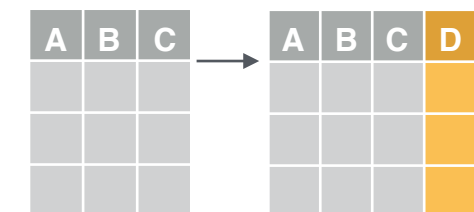
Underlying Grammar

... phrasing of questions using **verbs** ...

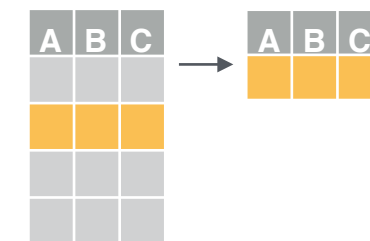
select: Retrieve a variable



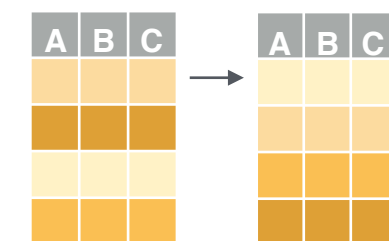
mutate: Add a variable to the data



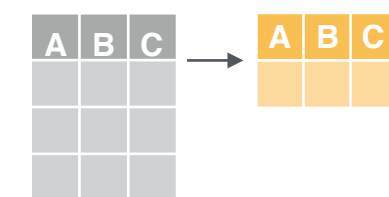
filter: Extracts cases based on values



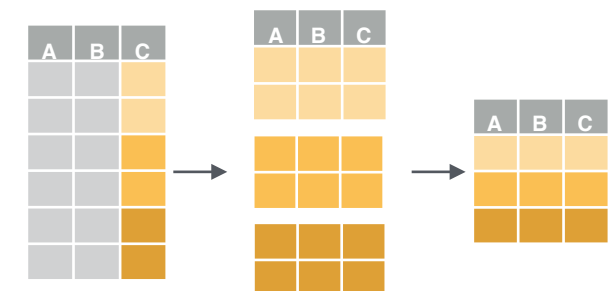
arrange: Change the order of the data



summarise: Reduce multiple values to statistics



group_by: Split the data by trait



dplyr



... grammar for manipulating data ...

```
install.packages("dplyr")  
library("dplyr")
```

Function	Description
<code>filter(.data, ...)</code>	Extracts cases based on values
<code>select(.data, ...)</code>	Include or exclude variables (var / -var)
<code>arrange(.data, ...)</code>	Change the order of the data
<code>mutate(.data, ...)</code>	Add new variables to the data
<code>summarise(.data, ...)</code>	Reduce multiple values to statistics
<code>group_by(data, ...)</code>	Split the data by trait

Filter Data

... subset by **men** ...

Dataset

No quotes on variable!
Non-standard evaluation (NSE)



```
enrolled_fa2017_men = filter(enrolled_fa2017, Gender == "Men")
```

Year	Gender	Enrolled
Undergrad	Men	18,345
Undergrad	Women	15,267
Undergrad	Unknown	12
Professional	Men	352
Professional	Women	640
Professional	Unknown	0
Graduate	Men	7,173
Graduate	Women	6,028
Graduate	Unknown	9

enrolled_fa2017

9 x 3

Year	Gender	Enrolled
Undergrad	Men	18,345
Professional	Men	352
Graduate	Men	7,173

3 x 3

enrolled_fa2017_men

Filter Data

... subset by **women** ...

```
enrolled_fa2017_women = filter(enrolled_fa2017, Gender == "Women") # dplyr  
enrolled_fa2017_women = enrolled_fa2017[enrolled_fa2017$Gender == "Women", ] # base R
```

Year	Gender	Enrolled
Undergrad	Men	18,345
Undergrad	Women	15,267
Undergrad	Unknown	12
Professional	Men	352
Professional	Women	640
Professional	Unknown	0
Graduate	Men	7,173
Graduate	Women	6,028
Graduate	Unknown	9

enrolled_fa2017

9 x 3

Year	Gender	Enrolled
Undergrad	Women	15,267
Professional	Women	640
Graduate	Women	6,028

3 x 3

enrolled_fa2017_women

Select Variables

... retrieve year and enrolled ...

```
enrolled_fa2017_women_info = select(enrolled_fa2017_women,  
                                     Year, Enrolled)
```

Year	Gender	Enrolled
Undergrad	Women	15,267
Professional	Women	640
Graduate	Women	6,028

3 x 3

enrolled_fa2017_women

Year	Enrolled
Undergrad	15,267
Professional	640
Graduate	6,028

3 x 2

enrolled_fa2017_women_info

Arrange Data

... changing order of data ...

```
enrolled_fa2017_women_ordered = arrange(enrolled_fa2017_women_info,  
                                           Enrolled)
```

Year	Enrolled
Undergrad	15,267
Professional	640
Graduate	6,028

3 x 2

enrolled_fa2017_women_info

Year	Enrolled
Professional	640
Graduate	6,028
Undergrad	15,267

3 x 2

enrolled_fa2017_ordered_women

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

Arranging Data

... descending order ...

```
enrolled_fa2017_women_ordered =  
  arrange(enrolled_fa2017_women_info, desc(Enrolled))
```

Year	Enrolled
Undergrad	15,267
Professional	640
Graduate	6,028

3 x 2

enrolled_fa2017_women_info

Year	Enrolled
Undergrad	15,267
Graduate	6,028
Professional	640

3 x 2

enrolled_fa2017_ordered_women

Source: http://www.dmi.illinois.edu/stuenr/abstracts/FA17_ten.htm

Summarise

... figuring out total enrollment across years ...

```
enrolled_fa2017_total_women = summarise(enrolled_fa2017_women_info,  
                                         Total_Enrolled = sum(Enrolled))
```

Year	Enrolled
Undergrad	15,267
Professional	640
Graduate	6,028

3 x 2

enrolled_fa2017_women_info

Total_Enrolled
21,935

1 x 1

enrolled_fa2017_total_women

Mutating Data

... padding the women enrollment ...

```
enrolled_fa2017_women_add = mutate(enrolled_fa2017_women_info,  
                                     Additional = Enrolled + 550)
```

Year	Enrolled
Undergrad	15,267
Professional	640
Graduate	6,028

3 x 2

enrolled_fa2017_women_info

Year	Enrolled	Additional
Undergrad	15,267	15,817
Professional	640	1190
Graduate	6,028	6,578

3 x 3

enrolled_fa2017_women_add

Previously

Definition:


Piping is the act of taking one value and immediately placing it into another function to form a flow of results.

Left Function
Transmitting function result
`rnorm(10)`

Pipe Operator
Facilitate moving left result
to the function on right

Right Function
Receiving function result in
first parameter
`abs(rnorm(10))`

`rnorm(10) %>% abs()`



`%>%` is read as "and, then"

dplyr with Pipes

... piping together different chunks of code ...

```
enrolled_fa2017_total_women =  
  enrolled_fa2017 %>% # Take the enrollment data and, then  
  filter(Gender == "Women") %>% # Retrieve all Women data and, then  
  select(Year, Enrolled) %>% # Take Year and Enrolled variables and, then  
  arrange(Enrolled) %>% # Order Enrolled in Ascending order and, then  
  summarise(Total_Enrolled = sum(Enrolled)) # Get total women enrollment
```

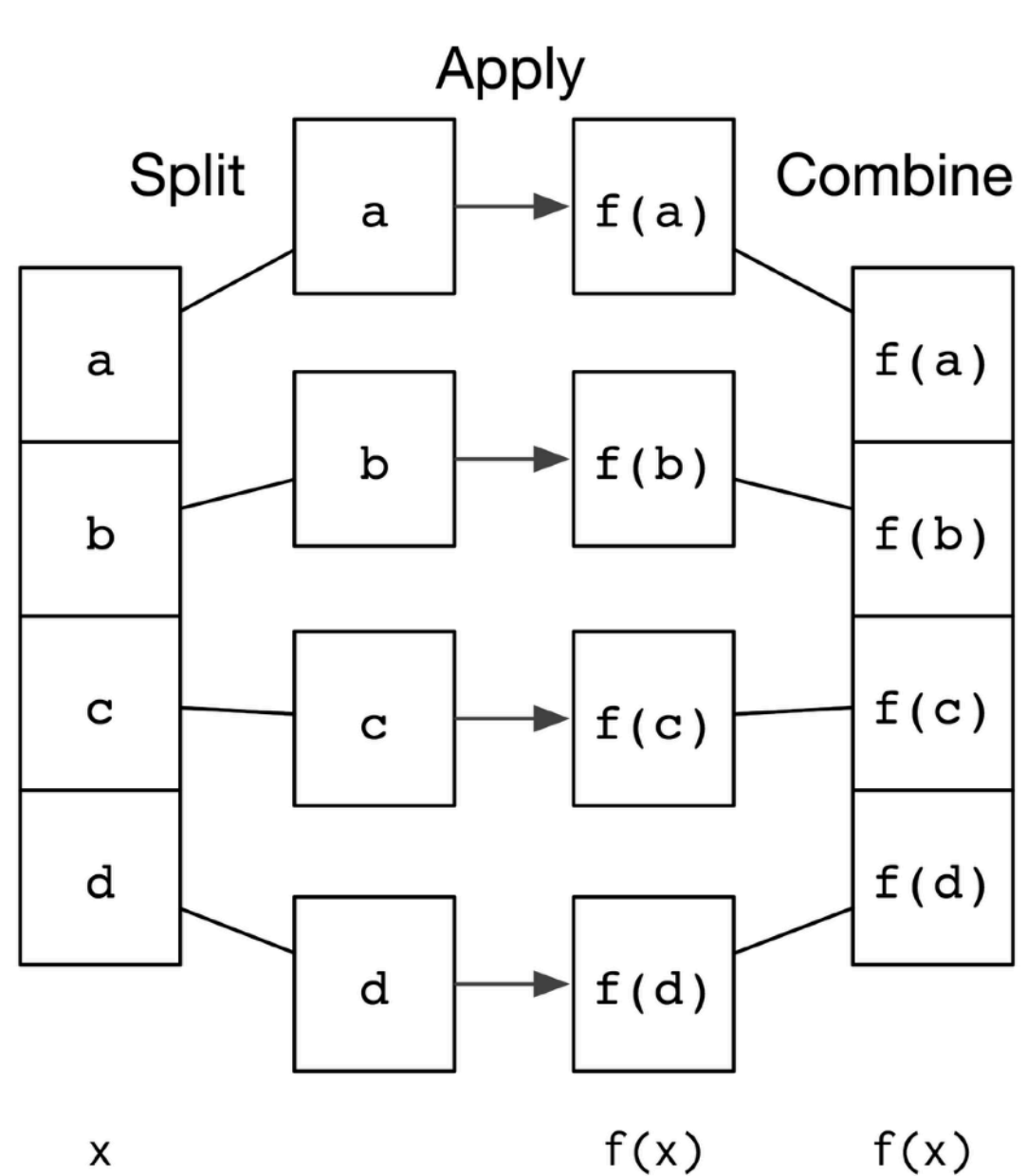
Your Turn

1. Select the Sepal.Length and Petal.Length variables in the **iris** data set
2. Retrieve all of the virginica **Species** observations from **iris**

Split-Apply-Combine

Split-Apply-Combine

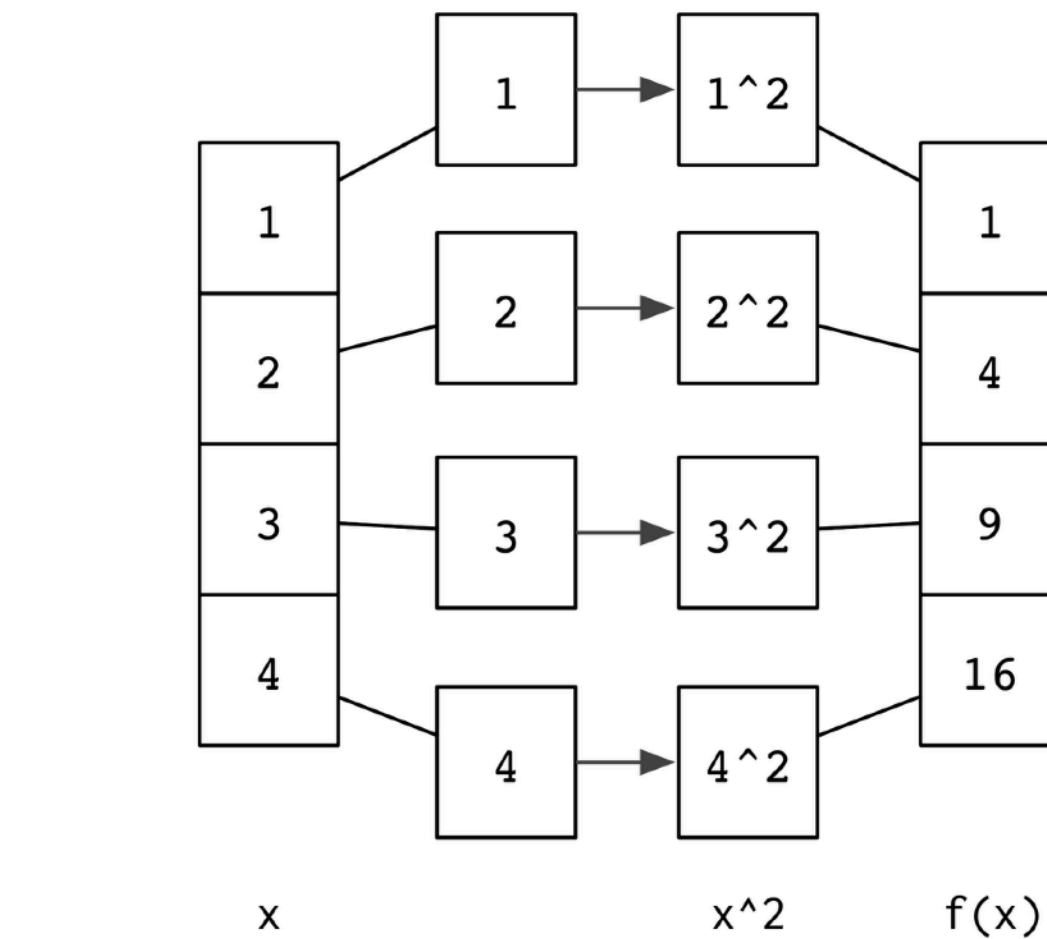
... overview ...



1. **Split** Data into pieces, 2. **Apply** function to each piece, and 3. **Combine** result

Vectorization

... in the split-apply-combine framework ...



```
x = 1L:4L  
(y = x^2)  
# [1] 1 4 9 16
```


Split-Apply-Combine

AKA

MapReduce

Summarise by Group

... summary statistics broken down by groups ...

```
enrolled_fa2017_grouped = group_by(enrolled_fa2017, Gender)  
enrolled_fa2017_gender = summarise(enrolled_fa2017_grouped,  
                                     Total_Enrollment = sum(Enrolled))
```

Year	Gender	Enrolled
Undergrad	Men	18,345
Undergrad	Women	15,267
Undergrad	Unknown	12
Professional	Men	352
Professional	Women	640
Professional	Unknown	0
Graduate	Men	7,173
Graduate	Women	6,028
Graduate	Unknown	9

9 x 3

enrolled_fa2017

Gender	Total_Enrollment
Men	25,870
Women	21,935
Unknown	21

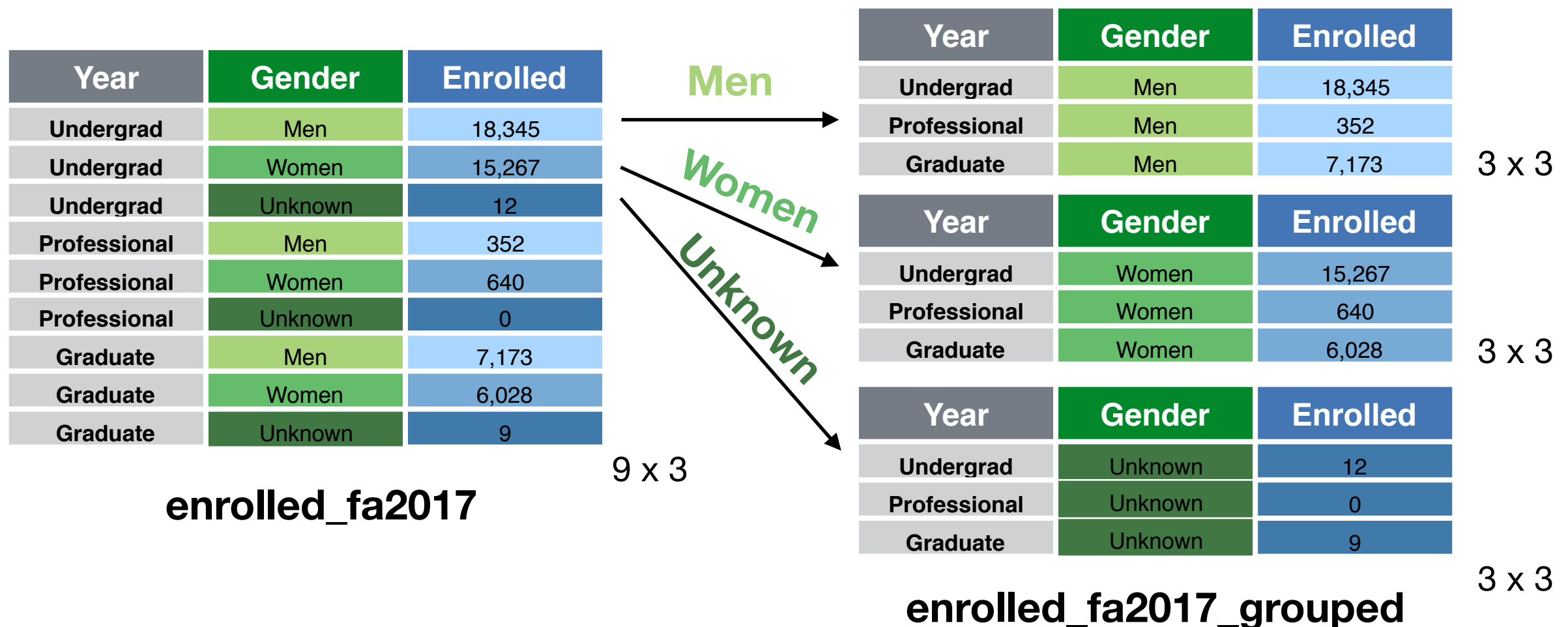
3 x 2

enrolled_fa2017_gender

Split Step

SPLIT by Gender

```
enrolled_fa2017_grouped = group_by(enrolled_fa2017 , Gender)
```



Apply Step: Using a Function

APPLY **sum** on **Enrolled**

```
summarise(enrolled_fa2017_grouped, Total_Enrollment = sum(Enrolled))
```

Year	Gender	Enrolled		Total Enrollment
Undergrad	Men	18,345	→ sum	18,345 + 352 + 7,173 = 25,870
Professional	Men	352		
Graduate	Men	7,173		

3 x 3

Year	Gender	Enrolled		Total Enrollment
Undergrad	Women	15,267	→ sum	15,267 + 640 + 6,028 = 21,935
Professional	Women	640		
Graduate	Women	6,028		

3 x 3

Year	Gender	Enrolled		Total Enrollment
Undergrad	Unknown	12	→ sum	12 + 0 + 9 = 21
Professional	Unknown	0		
Graduate	Unknown	9		

3 x 3

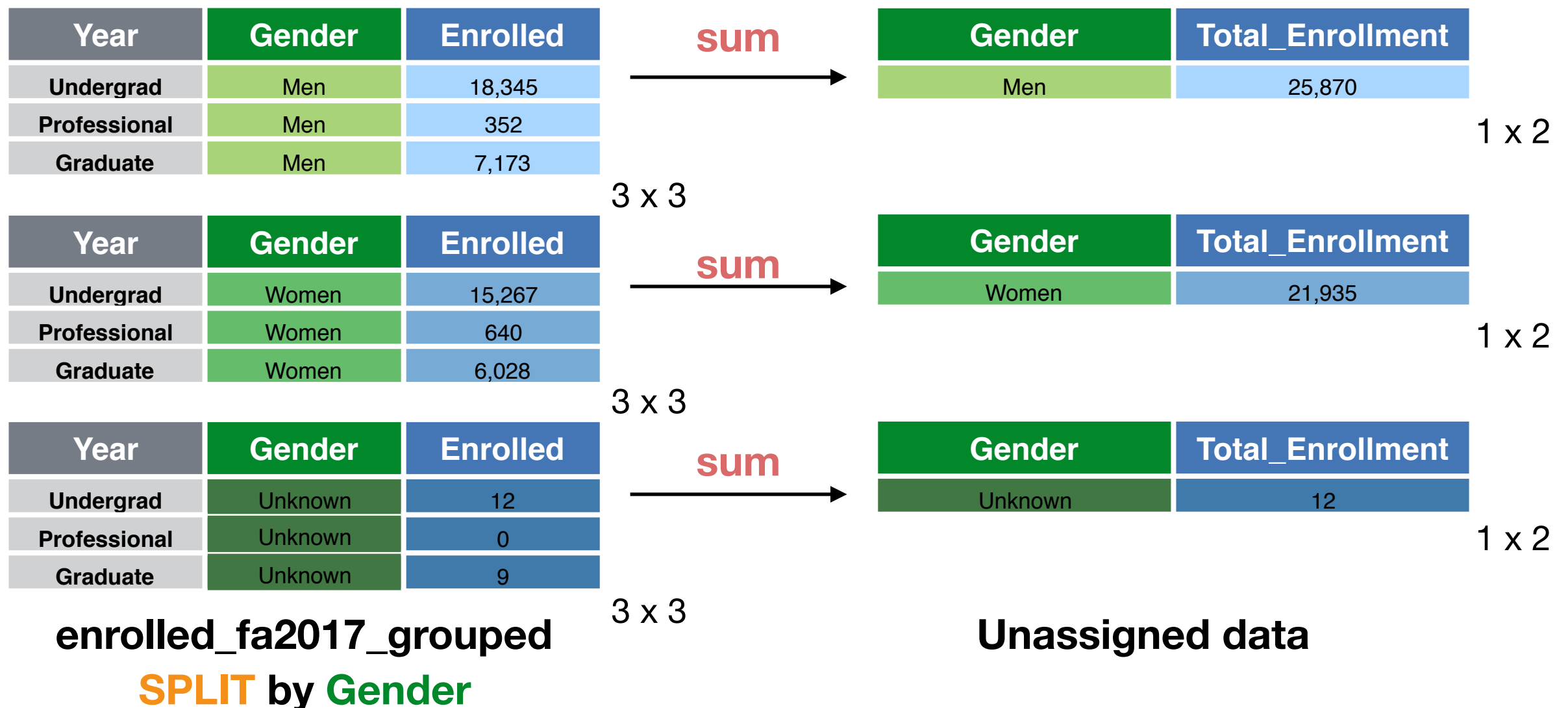
enrolled_fa2017_grouped
SPLIT by **Gender**

Unassigned data

Apply Step: Match to Group

APPLY **sum** on **Enrolled**

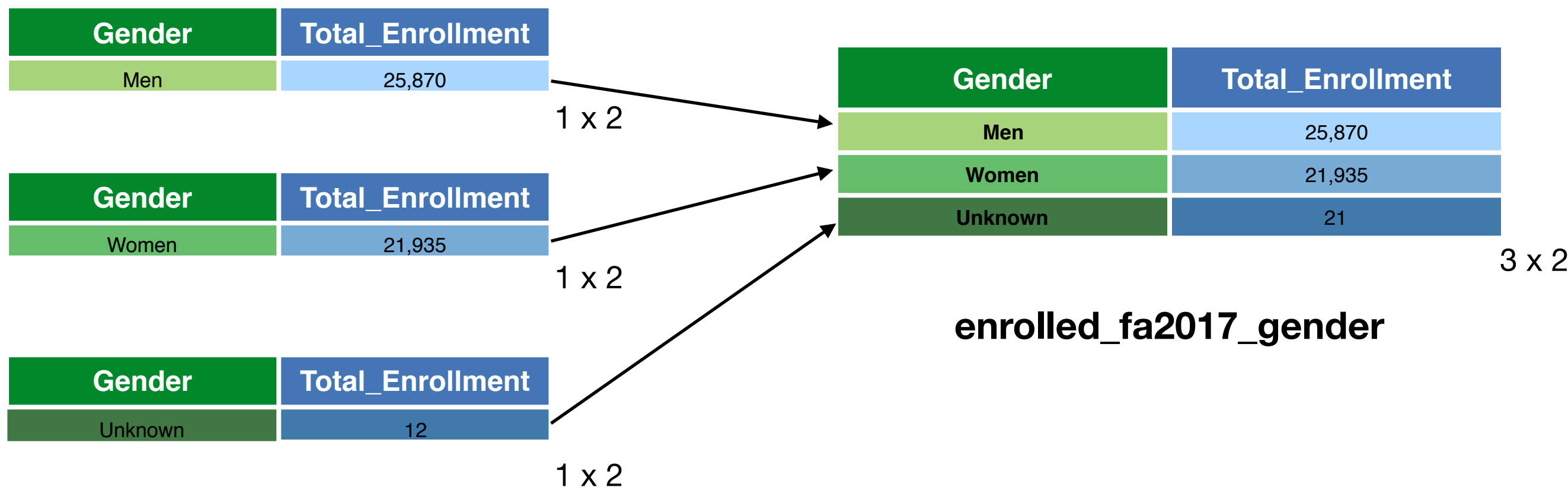
summarise(enrolled_fa2017_grouped, **Total_Enrollment** = **sum**(**Enrolled**))



Combine Step

COMBINE Total_Enrollment by Gender

```
enrolled_fa2017_gender = summarise(enrolled_fa2017_grouped,  
                                   Total_Enrollment = sum(Enrolled))
```



APPLY **sum** on **Enrolled**

Your Turn

Provide the *mean, maximum, minimum* of the Sepal.Length for each of species of **iris** alongside a **count**.

Recap

- **Grammar of Data**

- Pose question about the data
- Answer the questions through **five** verbs: select, filter, mutate, arrange, and summarise

- **Split-Apply-Combine**

- **Split** Data into pieces
- **Apply** function to each piece, and
- **Combine** result

Resources

Cheatsheet

... dplyr cheat sheet ...

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**

&



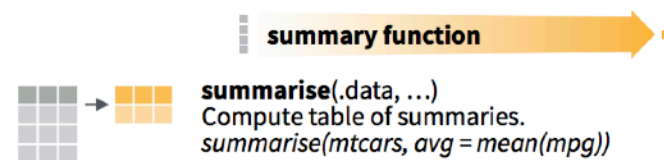
Each **observation**, or **case**, is in its own **row**



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

Summarise Cases

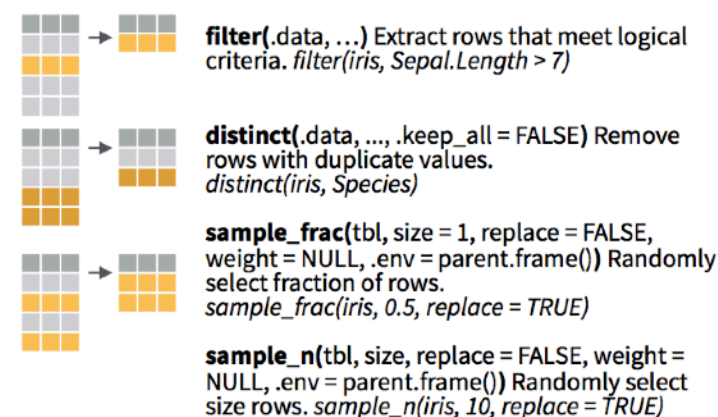
These apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).



Manipulate Cases

EXTRACT CASES

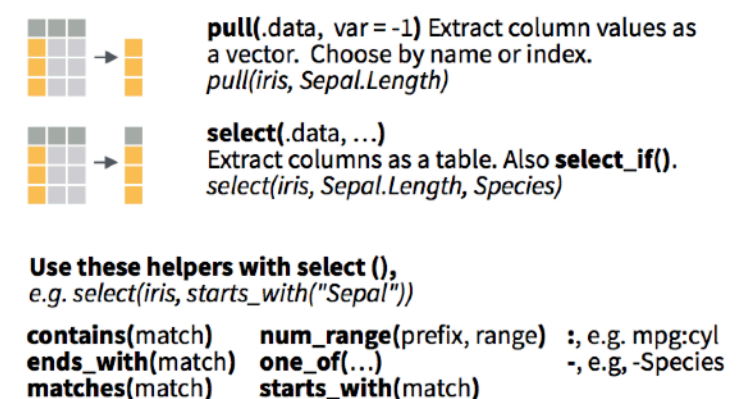
Row functions return a subset of rows as a new table.



Manipulate Variables

EXTRACT VARIABLES

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



<https://github.com/rstudio/cheatsheets/raw/master/data-transformation.pdf>

Acknowledgements

Acknowledgements

- Style of the RStudio Cheatsheet for Data Transformations
- [The Split-Apply-Combine Strategy for Data Analytics](#) by Hadley Wickham

This work is licensed under the
Creative Commons
Attribution-NonCommercial-
ShareAlike 4.0 International
License

