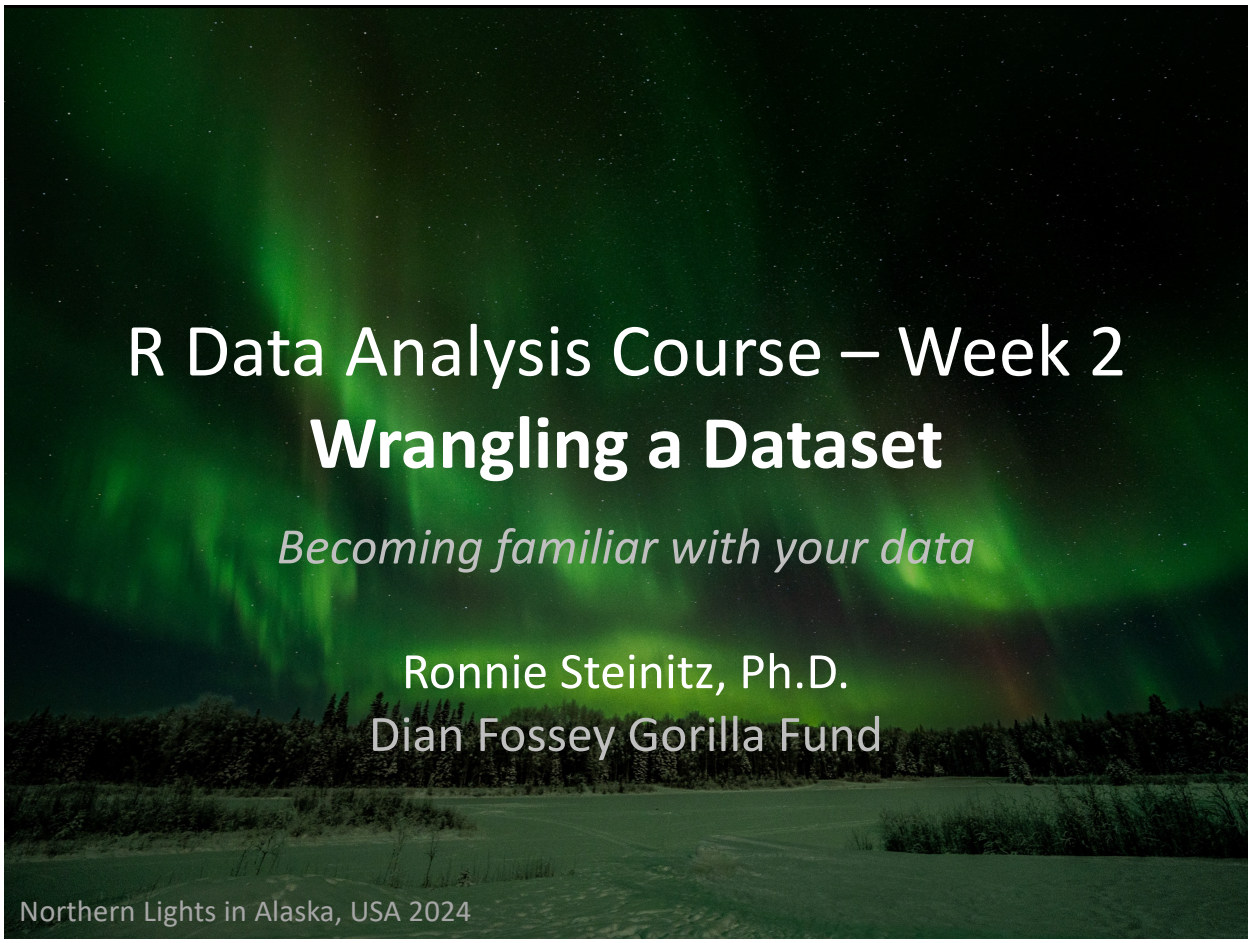


The Pipe Operator %>%

[Ctrl + Shift + M]

- Reads as "*and then*" (take '`data`,' and then, …)
- Sends the result of one function into the next function
- Makes code easier to read, like steps in a sentence
- Replaces nested functions with clear, left-to-right flow

Instead of **nesting functions**; difficult to understand:

*(task 4(task 3(task 2(task 1(**data**)))))*

Pipes allow us to read like a sentence; logical order:

Take *data %>%*

then do *task 1 %>%*

then do *task 2 %>%*

then do *task 3 %>%*

then do *task 4*

Instead of **nesting functions**; difficult to understand:

*(bake(stir_well(add_egg(add_water(take **flour**)))))*

Pipes allow us to read like a sentence; logical order:

Take ***flour*** *%>%*

then add_water *%>%*

then add_egg *%>%*

then stir_well *%>%*

then bake_into_bread

Nesting:

*arrange(summarise(mutate(filter(**data**, mass > 20), ratio = mass/height), average_ratio = mean(ratio)), average_ratio)*

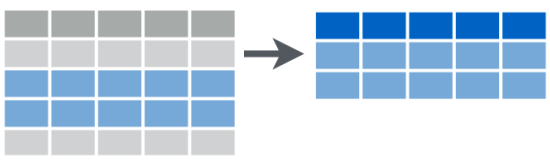
Pipe:

```
data %>%
    filter(height > 20) %>%
    mutate(ratio = mass/height) %>%
    summarise(average_ratio = mean(ratio)) %>%
    arrange(average_ratio)
```

dplyr::**filter( )**

- Conditional subset of observations
  - *Observations where **X** is **greater** than **10***
    - *data %>% filter(**X > 10**)*
  - *Observations only from the city of Kigali*
    - *data %>% filter(**city == Kigali**)*

Subset Observations (Rows)



dplyr::**select( )**

- Subset columns
  - *The **name** and **group** of the individual, and the **date of observation***
    - *data %>% select(**group**, **name**, **date**)*
  - *Exclude **notes***
    - *data %>% select(**-notes**)*

Subset Variables (Columns)

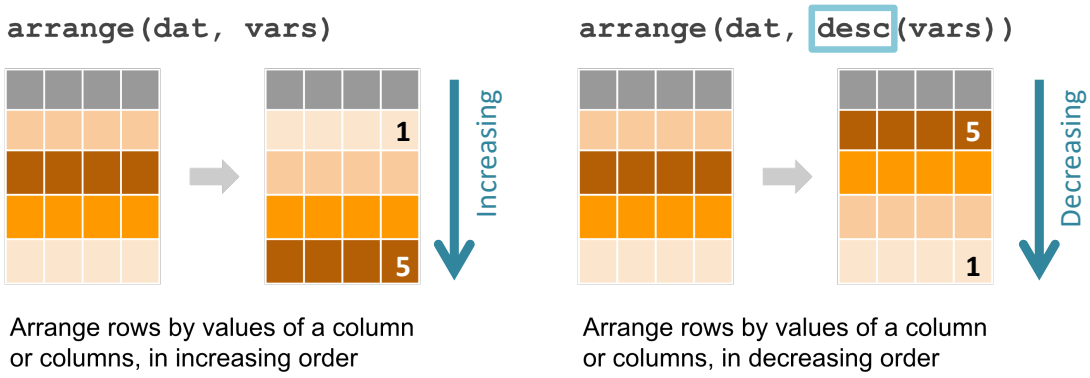
dplyr::**mutate( )**

- Create or transform variables
  - *Convert weight in **grams** to **kilograms***
  - *data %>% mutate(**weight_kg** = **weight_g** / 1000)*

Make New Variables

dplyr::**arrange( )**

- Reorder rows by values in a column
  - *Sort all individuals, from **youngest** to **oldest***
  - *data %>% arrange(**age**) # increasing order 1 →100*
  - *data %>% arrange(desc(**age**)) # decreasing 100 →1*



Arrange rows by values of a column or columns, in increasing order

Arrange rows by values of a column or columns, in decreasing order

Week 2 - Skills Application

Working on **your own** dataset**:** create a Markdown file which you will be using continuously and adding to each week. Name it: ***Lastname_Firstname_Data.Rmd*** (in your working directory)

Working on **palmer_penguins_raw** dataset**:** create a Markdown file just for this week. Name it: ***Lastname_Firstname_Week2.Rmd*** (in Week 2)

Create new markdown file. Load packages. Import the dataset into the environment and save as an object (e.g., `data`).

1. Identify which variables have **missing values**.
2. Check for **misclassified columns** (e.g., text stored as factors, numbers stores as text). Get summary statistics for two variables of different classes.
3. **Convert** one column to the correct type.
4. Use `**select()**` to keep only 3 columns of your choice.
5. Use `**filter()**` to create a subset with one species of penguin.
6. Create a new variable using `**mutate()**` (for example, converting mm → cm).
7. Make a **histogram** of a numeric variable from your subset. **Add a title** and **axis labels** to your plot.
8. **Knit** the entire script into an HTML file; **save** to your working directory; **upload** to:

   Google Drive > *R Course Materials for Students* > *Submissions*