Module_1_intro_dplyr

Elyse, Jeanne & Sheila

Table of contents

Welcome & Setup

Welcome to our short course! Today we'll tackle a common pain point in R: **what happens when your data is just too big?**

But first, let's get everyone set up and comfortable.

Learning Objectives (Module 1)

By the end of this module, you will be able to:

- Master the 6 core dplyr verbs: `filter()`, `select()`, `arrange()`, `mutate()`, `group_by()`, and `summarise()`
- Build multi-step data transformation pipelines using the pipe operator (`|>`)
- Apply the "filter early, select early" optimization strategy
- Construct grouped summaries and aggregations
- Understand the analytical thinking process that scales to big data
- Practice troubleshooting common dplyr errors

The Tools We'll Use

The tools we're introducing today **each play a specific role** in helping us work with larger-than-memory datasets — and we can combine them to build scalable workflows.

Tool	Purpose
`arrow`	(Columnar storage + lazy reading) → Efficiently reads large datasets **without loading the whole file into memory.** Supports fast filtering and streaming from disk.
`DBI`	(Database connection interface) → Provides a **common language to connect R to databases.** DuckDB uses it to talk to R.

Tool	Purpose
`duckdb`	(Fast in-process SQL database) → Allows you to **query large datasets using SQL syntax inside R.** Works especially well for joins, aggregations, and window functions.
`dplyr`	(User-friendly data wrangling) → Offers an **intuitive, readable grammar for filtering, summarizing, and transforming data.** It's our main pipeline tool.
`dbplyr`	(Bridge between dplyr and databases) → **Translates dplyr code into SQL automatically.** Lets you use dplyr pipelines on database tables (like DuckDB) without writing raw SQL.

-

Load Your Tools

Let's load the packages we'll need today.

```
# Install and load required packages
required_packages <- c("tidyverse", "arrow", "duckdb","DBI","dbplyr")

# Install missing packages
for (pkg in required_packages) {
  if (!requireNamespace(pkg, quietly = TRUE)) {
    install.packages(pkg)
  }
}

# Load all packages
for (pkg in required_packages) {
  library(pkg, character.only = TRUE)
}
```

```
Warning: package 'tidyverse' was built under R version 4.3.3
```

```
Warning: package 'tibble' was built under R version 4.3.3
```

```
Warning: package 'tidyr' was built under R version 4.3.3
```

```
Warning: package 'readr' was built under R version 4.3.3

Warning: package 'purrr' was built under R version 4.3.3

Warning: package 'dplyr' was built under R version 4.3.3

Warning: package 'stringr' was built under R version 4.3.3

Warning: package 'forcats' was built under R version 4.3.3

Warning: package 'lubridate' was built under R version 4.3.3

-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor

Warning: package 'arrow' was built under R version 4.3.3


Attaching package: 'arrow'

The following object is masked from 'package:lubridate':

    duration

The following object is masked from 'package:utils':

    timestamp

Warning: package 'duckdb' was built under R version 4.3.3

Loading required package: DBI
```

```
Warning: package 'DBI' was built under R version 4.3.3

Warning: package 'dbplyr' was built under R version 4.3.3


Attaching package: 'dbplyr'

The following objects are masked from 'package:dplyr':

    ident, sql
```

For this shortcourse we will be using the Seattle Library Checkouts Dataset if you didn;t do it in the pre-work then please do it now. We will talk about this in the Module 2- Arrow.

Step 1: Create a Directory

First, let's create a special folder to store our data:

```
# Create a "data" directory if it doesn't exist already
# Using showWarnings = FALSE to suppress warning if directory already exists

dir.create("data", showWarnings = FALSE)
```

Step 2: Download the Dataset

Now for the fun part! We'll download the Seattle Library dataset (9GB).

Important: This is a 9GB file, so:

- Make sure you have enough disk space

```
# Download Seattle library checkout dataset:

# 1. Fetch data from AWS S3 bucket URL
# 2. Save to local data directory
# 3. Use resume = TRUE to allow continuing interrupted downloads

curl::multi_download("https://r4ds.s3.us-west-2.amazonaws.com/seattle-library-checkouts.csv"
```

What Makes Big Data Big?

Tip: You might see errors like "cannot allocate vector of size…"

Example:

```
# simulation
big_data_test <- tibble(x = rnorm(50000000))

big_data_test
```

```
# A tibble: 50,000,000 x 1
        x
    <dbl>
 1  0.237
 2 -0.195
 3  1.57
 4 -0.192
 5  0.521
 6 -0.307
 7 -0.478
 8 -0.751
 9 -0.280
10  0.877
# i 49,999,990 more rows
```

```
# Attempt to simulate a "too large" allocation
big_data_test <- tibble(x = rnorm(1e9))  # 1 billion rows
```

Core Concepts: The 3 Vs

The Three V's Framework

Volume: The size of your data exceeds your computer's RAM

- Example: A 5GB CSV file on a laptop with 8GB RAM

Velocity: Data is generated faster than you can process it

- Example: Real-time sensor data, streaming transactions

Variety: Different data types and structures

- Example: Combining CSV files, JSON logs, and database tables

Interactive discussion:

- What's the largest dataset you've tried to analyze in R?
- When has R crashed on you? What error messages did you see?
- How do you currently handle data that's "too big"?

Common strategies people use (and their limitations):

- **Sampling**: Loses information and representativeness
- **Chunking**: Complex to manage and prone to errors
- **Giving up**: Not a solution!
- **Today's approach**: Use the right tools for the job

B. dplyr Refresher

Why this refresher?

Before we dive into `arrow` and **bigger-than-memory data**, let's quickly revisit `dplyr`, the backbone of tidy data manipulation in R.

Experience check:

- Daily dplyr users (you'll be our helpers!)
- Occasional users (perfect timing for a refresher!)
- New to dplyr (you're in for a treat!)

The dplyr Mindset: Think in Verbs

Traditional R vs. dplyr Approach

Traditional R thinking:

```
# Multiple objects, hard to follow
subset_data <- subset(starwars, species == "Human")
selected_data <- subset_data[, c("name", "height", "mass")]
ordered_data <- selected_data[order(selected_data$height, decreasing = TRUE), ]
```

dplyr thinking:

```r
# One pipeline, easy to read
starwars |>
  filter(species == "Human") |>
  select(name, height, mass) |>
  arrange(desc(height))
```

```
# A tibble: 35 x 3
   name                height  mass
   <chr>                <int> <dbl>
 1 Darth Vader            202 136
 2 Qui-Gon Jinn           193  89
 3 Dooku                  193  80
 4 Bail Prestor Organa    191  NA
 5 Anakin Skywalker       188  84
 6 Mace Windu             188  84
 7 Raymus Antilles        188  79
 8 Padmé Amidala          185  45
 9 Biggs Darklighter      183  84
10 Boba Fett              183  78.2
# i 25 more rows
```

Before we scale up to big data, let's review the `dplyr` verbs that will be our building blocks.

Rule of thumb:

- Small datasets (< 1MB): Either approach works
- Medium datasets (1MB - 100MB): dplyr is cleaner
- Large datasets (100MB+): dplyr pipeline thinking is essential

Meet Your Practice Dataset: `starwars`

Built in dataset perfect for learning

Use Case Dataset: `starwars`

We'll work with the built-in `starwars` tibble.

```r
#check out the first 5 rows
starwars |>
  head()
```

```
# A tibble: 6 x 14
  name      height  mass hair_color skin_color eye_color birth_year sex   gender
  <chr>      <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
1 Luke Sky~    172    77 blond      fair       blue            19   male  mascu~
2 C-3PO        167    75 <NA>       gold       yellow         112   none  mascu~
3 R2-D2         96    32 <NA>       white, bl~ red             33   none  mascu~
4 Darth Va~    202   136 none       white      yellow          41.9 male  mascu~
5 Leia Org~    150    49 brown      light      brown           19   fema~ femin~
6 Owen Lars    178   120 brown, gr~ light      blue            52   male  mascu~
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

filter(): Keep only rows that meet a condition

```
# Single condition
starwars |>
  filter(species == "Human")
```

```
# A tibble: 35 x 14
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Luke Sk~    172    77 blond      fair       blue            19   male  mascu~
 2 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
 3 Leia Or~    150    49 brown      light      brown           19   fema~ femin~
 4 Owen La~    178   120 brown, gr~ light      blue            52   male  mascu~
 5 Beru Wh~    165    75 brown      light      blue            47   fema~ femin~
 6 Biggs D~    183    84 black      light      brown           24   male  mascu~
 7 Obi-Wan~    182    77 auburn, w~ fair       blue-gray       57   male  mascu~
 8 Anakin ~    188    84 blond      fair       blue            41.9 male  mascu~
 9 Wilhuff~    180    NA auburn, g~ fair       blue            64   male  mascu~
10 Han Solo    180    80 brown      fair       brown           29   male  mascu~
# i 25 more rows
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

```
# Multiple conditions (AND)
# Find all humans from Tatooine
starwars |>
  filter(species == "Human", homeworld == "Tatooine")
```

```
# A tibble: 8 x 14
  name      height  mass hair_color skin_color eye_color birth_year sex   gender
  <chr>      <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
1 Luke Sky~    172    77 blond      fair       blue            19   male  mascu~
2 Darth Va~    202   136 none       white      yellow          41.9 male  mascu~
3 Owen Lars    178   120 brown, gr~ light      blue            52   male  mascu~
4 Beru Whi~    165    75 brown      light      blue            47   fema~ femin~
5 Biggs Da~    183    84 black      light      brown           24   male  mascu~
6 Anakin S~    188    84 blond      fair       blue            41.9 male  mascu~
7 Shmi Sky~    163    NA black      fair       brown           72   fema~ femin~
8 Cliegg L~    183    NA brown      fair       blue            82   male  mascu~
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

```
# Multiple conditions (OR)
starwars |>
  filter(species == "Human" | species == "Droid")
```

```
# A tibble: 41 x 14
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Luke Sk~    172    77 blond      fair       blue            19   male  mascu~
 2 C-3PO       167    75 <NA>       gold       yellow         112   none  mascu~
 3 R2-D2        96    32 <NA>       white, bl~ red             33   none  mascu~
 4 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
 5 Leia Or~    150    49 brown      light      brown           19   fema~ femin~
 6 Owen La~    178   120 brown, gr~ light      blue            52   male  mascu~
 7 Beru Wh~    165    75 brown      light      blue            47   fema~ femin~
 8 R5-D4        97    32 <NA>       white, red red             NA   none  mascu~
 9 Biggs D~    183    84 black      light      brown           24   male  mascu~
10 Obi-Wan~    182    77 auburn, w~ fair       blue-gray       57   male  mascu~
# i 31 more rows
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

```
# Numeric conditions
starwars |>
  filter(height > 180, mass < 100)
```

```
# A tibble: 22 x 14
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
```

```
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Biggs D~    183  84   black      light      brown           24   male  mascu~
 2 Obi-Wan~    182  77   auburn, w~ fair       blue-gray       57   male  mascu~
 3 Anakin ~    188  84   blond      fair       blue            41.9 male  mascu~
 4 Boba Fe~    183  78.2 black      fair       brown           31.5 male  mascu~
 5 Qui-Gon~    193  89   brown      fair       blue            92   male  mascu~
 6 Nute Gu~    191  90   none       mottled g~ red             NA   male  mascu~
 7 Padmé A~    185  45   brown      light      brown           46   fema~ femin~
 8 Jar Jar~    196  66   none       orange     orange          52   male  mascu~
 9 Roos Ta~    224  82   none       grey       orange          NA   male  mascu~
10 Mace Wi~    188  84   none       dark       brown           72   male  mascu~
# i 12 more rows
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

```
# Handle missing values
starwars |>
  filter(!is.na(height))
```

```
# A tibble: 81 x 14
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Luke Sk~    172    77 blond      fair       blue            19   male  mascu~
 2 C-3PO       167    75 <NA>       gold       yellow         112   none  mascu~
 3 R2-D2        96    32 <NA>       white, bl~ red             33   none  mascu~
 4 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
 5 Leia Or~    150    49 brown      light      brown           19   fema~ femin~
 6 Owen La~    178   120 brown, gr~ light      blue            52   male  mascu~
 7 Beru Wh~    165    75 brown      light      blue            47   fema~ femin~
 8 R5-D4        97    32 <NA>       white, red red             NA   none  mascu~
 9 Biggs D~    183    84 black      light      brown           24   male  mascu~
10 Obi-Wan~    182    77 auburn, w~ fair       blue-gray       57   male  mascu~
# i 71 more rows
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

Advanced `filter()`ing

```
# Using %in% for multiple values
starwars |>
  filter(homeworld %in% c("Tatooine", "Naboo", "Alderaan"))
```

```
# A tibble: 24 x 14
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Luke Sk~    172    77 blond      fair       blue            19   male  mascu~
 2 C-3PO       167    75 <NA>       gold       yellow         112   none  mascu~
 3 R2-D2        96    32 <NA>       white, bl~ red             33   none  mascu~
 4 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
 5 Leia Or~    150    49 brown      light      brown           19   fema~ femin~
 6 Owen La~    178   120 brown, gr~ light      blue            52   male  mascu~
 7 Beru Wh~    165    75 brown      light      blue            47   fema~ femin~
 8 R5-D4        97    32 <NA>       white, red red             NA   none  mascu~
 9 Biggs D~    183    84 black      light      brown           24   male  mascu~
10 Anakin ~    188    84 blond      fair       blue            41.9 male  mascu~
# i 14 more rows
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

```
# String matching
starwars |>
  filter(str_detect(name, "Skywalker"))
```

```
# A tibble: 3 x 14
  name      height  mass hair_color skin_color eye_color birth_year sex   gender
  <chr>      <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
1 Luke Sky~    172    77 blond      fair       blue            19   male  mascu~
2 Anakin S~    188    84 blond      fair       blue            41.9 male  mascu~
3 Shmi Sky~    163    NA black      fair       brown           72   fema~ femin~
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

```
# Complex logical conditions
starwars |>
  filter((species == "Human" & height > 175) | species == "Droid")
```

```
# A tibble: 25 x 14
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 C-3PO       167    75 <NA>       gold       yellow         112   none  mascu~
 2 R2-D2        96    32 <NA>       white, bl~ red             33   none  mascu~
 3 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
 4 Owen La~    178   120 brown, gr~ light      blue            52   male  mascu~
```

```
 5 R5-D4        97    32 <NA>       white, red red             NA   none  mascu~
 6 Biggs D~    183    84 black      light      brown           24   male  mascu~
 7 Obi-Wan~    182    77 auburn, w~ fair       blue-gray       57   male  mascu~
 8 Anakin ~    188    84 blond      fair       blue            41.9 male  mascu~
 9 Wilhuff~    180    NA auburn, g~ fair       blue            64   male  mascu~
10 Han Solo    180    80 brown      fair       brown           29   male  mascu~
# i 15 more rows
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

select(): Pick specific columns

```
# Select specific columns
starwars |>
  select(name, height, mass)
```

```
# A tibble: 87 x 3
   name               height  mass
   <chr>               <int> <dbl>
 1 Luke Skywalker        172    77
 2 C-3PO                 167    75
 3 R2-D2                  96    32
 4 Darth Vader           202   136
 5 Leia Organa           150    49
 6 Owen Lars             178   120
 7 Beru Whitesun Lars    165    75
 8 R5-D4                  97    32
 9 Biggs Darklighter     183    84
10 Obi-Wan Kenobi        182    77
# i 77 more rows
```

```
# Select ranges
starwars |>
  select(name:mass)
```

```
# A tibble: 87 x 3
   name               height  mass
   <chr>               <int> <dbl>
 1 Luke Skywalker        172    77
 2 C-3PO                 167    75
```

```
 3 R2-D2                  96    32
 4 Darth Vader           202   136
 5 Leia Organa           150    49
 6 Owen Lars             178   120
 7 Beru Whitesun Lars    165    75
 8 R5-D4                  97    32
 9 Biggs Darklighter     183    84
10 Obi-Wan Kenobi        182    77
# i 77 more rows
```

```
# Exclude columns
starwars |>
  select(-films, -vehicles, -starships)
```

```
# A tibble: 87 x 11
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Luke Sk~    172    77 blond      fair       blue            19   male  mascu~
 2 C-3PO       167    75 <NA>       gold       yellow         112   none  mascu~
 3 R2-D2        96    32 <NA>       white, bl~ red             33   none  mascu~
 4 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
 5 Leia Or~    150    49 brown      light      brown           19   fema~ femin~
 6 Owen La~    178   120 brown, gr~ light      blue            52   male  mascu~
 7 Beru Wh~    165    75 brown      light      blue            47   fema~ femin~
 8 R5-D4        97    32 <NA>       white, red red             NA   none  mascu~
 9 Biggs D~    183    84 black      light      brown           24   male  mascu~
10 Obi-Wan~    182    77 auburn, w~ fair       blue-gray       57   male  mascu~
# i 77 more rows
# i 2 more variables: homeworld <chr>, species <chr>
```

```
# Select by pattern
starwars |>
  select(starts_with("s"))  # species, skin_color, starships
```

```
# A tibble: 87 x 4
   skin_color  sex    species starships
   <chr>       <chr>  <chr>   <list>
 1 fair        male   Human   <chr [2]>
 2 gold        none   Droid   <chr [0]>
 3 white, blue none   Droid   <chr [0]>
 4 white       male   Human   <chr [1]>
```

```
 5 light       female Human   <chr [0]>
 6 light       male   Human   <chr [0]>
 7 light       female Human   <chr [0]>
 8 white, red  none   Droid   <chr [0]>
 9 light       male   Human   <chr [1]>
10 fair        male   Human   <chr [5]>
# i 77 more rows
```

```
starwars |>
  select(ends_with("color"))  # hair_color, skin_color, eye_color
```

```
# A tibble: 87 x 3
   hair_color    skin_color  eye_color
   <chr>         <chr>       <chr>
 1 blond         fair        blue
 2 <NA>          gold        yellow
 3 <NA>          white, blue red
 4 none          white       yellow
 5 brown         light       brown
 6 brown, grey   light       blue
 7 brown         light       blue
 8 <NA>          white, red  red
 9 black         light       brown
10 auburn, white fair        blue-gray
# i 77 more rows
```

```
starwars |>
  select(contains("_"))  # hair_color, skin_color, eye_color, birth_year
```

```
# A tibble: 87 x 4
   hair_color    skin_color  eye_color birth_year
   <chr>         <chr>       <chr>          <dbl>
 1 blond         fair        blue            19
 2 <NA>          gold        yellow         112
 3 <NA>          white, blue red             33
 4 none          white       yellow          41.9
 5 brown         light       brown           19
 6 brown, grey   light       blue            52
 7 brown         light       blue            47
 8 <NA>          white, red  red             NA
 9 black         light       brown           24
```

```
10 auburn, white fair        blue-gray       57
# i 77 more rows
```

Advanced Select() ing

```
# Rename while selecting
starwars |>
  select(character_name = name, height_cm = height)
```

```
# A tibble: 87 x 2
   character_name     height_cm
   <chr>                  <int>
 1 Luke Skywalker           172
 2 C-3PO                    167
 3 R2-D2                     96
 4 Darth Vader              202
 5 Leia Organa              150
 6 Owen Lars                178
 7 Beru Whitesun Lars       165
 8 R5-D4                     97
 9 Biggs Darklighter        183
10 Obi-Wan Kenobi           182
# i 77 more rows
```

```
# Select and reorder
starwars |>
  select(name, species, everything())  # name and species first, then everything else
```

```
# A tibble: 87 x 14
   name    species height  mass hair_color skin_color eye_color birth_year sex
   <chr>   <chr>    <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr>
 1 Luke S~ Human      172    77 blond      fair       blue            19   male
 2 C-3PO   Droid      167    75 <NA>       gold       yellow         112   none
 3 R2-D2   Droid       96    32 <NA>       white, bl~ red             33   none
 4 Darth ~ Human      202   136 none       white      yellow          41.9 male
 5 Leia O~ Human      150    49 brown      light      brown           19   fema~
 6 Owen L~ Human      178   120 brown, gr~ light      blue            52   male
 7 Beru W~ Human      165    75 brown      light      blue            47   fema~
 8 R5-D4   Droid       97    32 <NA>       white, red red             NA   none
 9 Biggs ~ Human      183    84 black      light      brown           24   male
10 Obi-Wa~ Human      182    77 auburn, w~ fair       blue-gray       57   male
```

```
# i 77 more rows
# i 5 more variables: gender <chr>, homeworld <chr>, films <list>,
#   vehicles <list>, starships <list>
```

arrange(): Sort rows

```
# Sort ascending (default)
starwars |>
  arrange(height)
```

```
# A tibble: 87 x 14
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Yoda         66    17 white      green      brown            896 male  mascu~
 2 Ratts T~     79    15 none       grey, blue unknown           NA male  mascu~
 3 Wicket ~     88    20 brown      brown      brown              8 male  mascu~
 4 Dud Bolt     94    45 none       blue, grey yellow            NA male  mascu~
 5 R2-D2        96    32 <NA>       white, bl~ red               33 none  mascu~
 6 R4-P17       96    NA none       silver, r~ red, blue         NA none  femin~
 7 R5-D4        97    32 <NA>       white, red red               NA none  mascu~
 8 Sebulba     112    40 none       grey, red  orange            NA male  mascu~
 9 Gasgano     122    NA none       white, bl~ black             NA male  mascu~
10 Watto       137    NA black      blue, grey yellow            NA male  mascu~
# i 77 more rows
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

```
# Sort descending
starwars |>
  arrange(desc(height))
```

```
# A tibble: 87 x 14
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Yarael ~    264    NA none       white      yellow            NA male  mascu~
 2 Tarfful     234   136 brown      brown      blue              NA male  mascu~
 3 Lama Su     229    88 none       grey       black             NA male  mascu~
 4 Chewbac~    228   112 brown      unknown    blue             200 male  mascu~
 5 Roos Ta~    224    82 none       grey       orange            NA male  mascu~
 6 Grievous    216   159 none       brown, wh~ green, y~         NA male  mascu~
```

```
 7 Taun We     213    NA none       grey       black           NA   fema~ femin~
 8 Rugor N~    206    NA none       green      orange          NA   male  mascu~
 9 Tion Me~    206    80 none       grey       black           NA   male  mascu~
10 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
# i 77 more rows
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

```
# Multiple sort columns
starwars |>
  arrange(species, desc(height))
```

```
# A tibble: 87 x 14
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Ratts T~     79    15 none       grey, blue unknown           NA male  mascu~
 2 Dexter ~    198   102 none       brown      yellow            NA male  mascu~
 3 Ki-Adi-~    198    82 white      pale       yellow            92 male  mascu~
 4 Mas Ame~    196    NA none       blue       blue              NA male  mascu~
 5 Zam Wes~    168    55 blonde     fair, gre~ yellow            NA fema~ femin~
 6 IG-88       200   140 none       metal      red               15 none  mascu~
 7 C-3PO       167    75 <NA>       gold       yellow           112 none  mascu~
 8 R5-D4        97    32 <NA>       white, red red               NA none  mascu~
 9 R2-D2        96    32 <NA>       white, bl~ red               33 none  mascu~
10 R4-P17       96    NA none       silver, r~ red, blue         NA none  femin~
# i 77 more rows
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

```
# Handle missing values
starwars |>
  arrange(desc(height), na.last = TRUE)
```

```
# A tibble: 87 x 14
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Yarael ~    264    NA none       white      yellow            NA male  mascu~
 2 Tarfful     234   136 brown      brown      blue              NA male  mascu~
 3 Lama Su     229    88 none       grey       black             NA male  mascu~
 4 Chewbac~    228   112 brown      unknown    blue             200 male  mascu~
 5 Roos Ta~    224    82 none       grey       orange            NA male  mascu~
```

```
 6 Grievous    216   159 none       brown, wh~ green, y~       NA   male  mascu~
 7 Taun We     213    NA none       grey       black           NA   fema~ femin~
 8 Rugor N~    206    NA none       green      orange          NA   male  mascu~
 9 Tion Me~    206    80 none       grey       black           NA   male  mascu~
10 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
# i 77 more rows
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

mutate(): Create or Modify names

```
# Create new column
starwars |>
  mutate(height_m = height / 100)
```

```
# A tibble: 87 x 15
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Luke Sk~    172    77 blond      fair       blue            19   male  mascu~
 2 C-3PO       167    75 <NA>       gold       yellow         112   none  mascu~
 3 R2-D2        96    32 <NA>       white, bl~ red             33   none  mascu~
 4 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
 5 Leia Or~    150    49 brown      light      brown           19   fema~ femin~
 6 Owen La~    178   120 brown, gr~ light      blue            52   male  mascu~
 7 Beru Wh~    165    75 brown      light      blue            47   fema~ femin~
 8 R5-D4        97    32 <NA>       white, red red             NA   none  mascu~
 9 Biggs D~    183    84 black      light      brown           24   male  mascu~
10 Obi-Wan~    182    77 auburn, w~ fair       blue-gray       57   male  mascu~
# i 77 more rows
# i 6 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>, height_m <dbl>
```

```
# Multiple new columns
starwars |>
  mutate(
    height_m = height / 100,
    bmi = mass / (height_m^2)
  )
```

```
# A tibble: 87 x 16
```

```
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Luke Sk~    172    77 blond      fair       blue            19   male  mascu~
 2 C-3PO       167    75 <NA>       gold       yellow         112   none  mascu~
 3 R2-D2        96    32 <NA>       white, bl~ red             33   none  mascu~
 4 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
 5 Leia Or~    150    49 brown      light      brown           19   fema~ femin~
 6 Owen La~    178   120 brown, gr~ light      blue            52   male  mascu~
 7 Beru Wh~    165    75 brown      light      blue            47   fema~ femin~
 8 R5-D4        97    32 <NA>       white, red red             NA   none  mascu~
 9 Biggs D~    183    84 black      light      brown           24   male  mascu~
10 Obi-Wan~    182    77 auburn, w~ fair       blue-gray       57   male  mascu~
# i 77 more rows
# i 7 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>, height_m <dbl>, bmi <dbl>
```

```r
# Modify existing column
starwars |>
  mutate(name = str_to_upper(name))
```

```
# A tibble: 87 x 14
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 LUKE SK~    172    77 blond      fair       blue            19   male  mascu~
 2 C-3PO       167    75 <NA>       gold       yellow         112   none  mascu~
 3 R2-D2        96    32 <NA>       white, bl~ red             33   none  mascu~
 4 DARTH V~    202   136 none       white      yellow          41.9 male  mascu~
 5 LEIA OR~    150    49 brown      light      brown           19   fema~ femin~
 6 OWEN LA~    178   120 brown, gr~ light      blue            52   male  mascu~
 7 BERU WH~    165    75 brown      light      blue            47   fema~ femin~
 8 R5-D4        97    32 <NA>       white, red red             NA   none  mascu~
 9 BIGGS D~    183    84 black      light      brown           24   male  mascu~
10 OBI-WAN~    182    77 auburn, w~ fair       blue-gray       57   male  mascu~
# i 77 more rows
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

Advanced `mutate()` ions

```r
# Conditional mutations
starwars |>
```

```
  mutate(
    size_category = case_when(
      height < 100 ~ "Very Short",
      height < 150 ~ "Short",
      height < 180 ~ "Average",
      height >= 180 ~ "Tall",
      TRUE ~ "Unknown"
    )
  )
```

```
# A tibble: 87 x 15
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Luke Sk~    172    77 blond      fair       blue            19   male  mascu~
 2 C-3PO       167    75 <NA>       gold       yellow         112   none  mascu~
 3 R2-D2        96    32 <NA>       white, bl~ red             33   none  mascu~
 4 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
 5 Leia Or~    150    49 brown      light      brown           19   fema~ femin~
 6 Owen La~    178   120 brown, gr~ light      blue            52   male  mascu~
 7 Beru Wh~    165    75 brown      light      blue            47   fema~ femin~
 8 R5-D4        97    32 <NA>       white, red red             NA   none  mascu~
 9 Biggs D~    183    84 black      light      brown           24   male  mascu~
10 Obi-Wan~    182    77 auburn, w~ fair       blue-gray       57   male  mascu~
# i 77 more rows
# i 6 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>, size_category <chr>
```

```
# Using ifelse for simple conditions
starwars |>
  mutate(is_tall = ifelse(height > 180, "Tall", "Not Tall"))
```

```
# A tibble: 87 x 15
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Luke Sk~    172    77 blond      fair       blue            19   male  mascu~
 2 C-3PO       167    75 <NA>       gold       yellow         112   none  mascu~
 3 R2-D2        96    32 <NA>       white, bl~ red             33   none  mascu~
 4 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
 5 Leia Or~    150    49 brown      light      brown           19   fema~ femin~
 6 Owen La~    178   120 brown, gr~ light      blue            52   male  mascu~
 7 Beru Wh~    165    75 brown      light      blue            47   fema~ femin~
```

```
 8 R5-D4        97    32 <NA>       white, red red             NA   none  mascu~
 9 Biggs D~    183    84 black      light      brown           24   male  mascu~
10 Obi-Wan~    182    77 auburn, w~ fair       blue-gray       57   male  mascu~
# i 77 more rows
# i 6 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>, is_tall <chr>
```

group_by()

```
# Group by single variable
starwars |>
  group_by(species)
```

```
# A tibble: 87 x 14
# Groups:   species [38]
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Luke Sk~    172    77 blond      fair       blue            19   male  mascu~
 2 C-3PO       167    75 <NA>       gold       yellow         112   none  mascu~
 3 R2-D2        96    32 <NA>       white, bl~ red             33   none  mascu~
 4 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
 5 Leia Or~    150    49 brown      light      brown           19   fema~ femin~
 6 Owen La~    178   120 brown, gr~ light      blue            52   male  mascu~
 7 Beru Wh~    165    75 brown      light      blue            47   fema~ femin~
 8 R5-D4        97    32 <NA>       white, red red             NA   none  mascu~
 9 Biggs D~    183    84 black      light      brown           24   male  mascu~
10 Obi-Wan~    182    77 auburn, w~ fair       blue-gray       57   male  mascu~
# i 77 more rows
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

```
# Group by multiple variables
starwars |>
  group_by(species, homeworld)
```

```
# A tibble: 87 x 14
# Groups:   species, homeworld [57]
   name     height  mass hair_color skin_color eye_color birth_year sex   gender
   <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
 1 Luke Sk~    172    77 blond      fair       blue            19   male  mascu~
```

```
 2 C-3PO       167    75 <NA>       gold       yellow         112   none  mascu~
 3 R2-D2        96    32 <NA>       white, bl~ red             33   none  mascu~
 4 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
 5 Leia Or~    150    49 brown      light      brown           19   fema~ femin~
 6 Owen La~    178   120 brown, gr~ light      blue            52   male  mascu~
 7 Beru Wh~    165    75 brown      light      blue            47   fema~ femin~
 8 R5-D4        97    32 <NA>       white, red red             NA   none  mascu~
 9 Biggs D~    183    84 black      light      brown           24   male  mascu~
10 Obi-Wan~    182    77 auburn, w~ fair       blue-gray       57   male  mascu~
# i 77 more rows
# i 5 more variables: homeworld <chr>, species <chr>, films <list>,
#   vehicles <list>, starships <list>
```

summarise()

```
# Single summary
starwars |>
  group_by(species) |>
  summarise(avg_height = mean(height, na.rm = TRUE))
```

```
# A tibble: 38 x 2
   species   avg_height
   <chr>          <dbl>
 1 Aleena          79
 2 Besalisk       198
 3 Cerean         198
 4 Chagrian       196
 5 Clawdite       168
 6 Droid          131.
 7 Dug            112
 8 Ewok            88
 9 Geonosian      183
10 Gungan         209.
# i 28 more rows
```

```
# Multiple summaries
starwars |>
  group_by(species) |>
  summarise(
    count = n(),
```

```
    avg_height = mean(height, na.rm = TRUE),
    max_mass = max(mass, na.rm = TRUE),
    .groups = "drop"  # Ungroup after summarising
  )
```

```
Warning: There were 6 warnings in `summarise()`.
The first warning was:
i In argument: `max_mass = max(mass, na.rm = TRUE)`.
i In group 4: `species = "Chagrian"`.
Caused by warning in `max()`:
! no non-missing arguments to max; returning -Inf
i Run `dplyr::last_dplyr_warnings()` to see the 5 remaining warnings.
```

```
# A tibble: 38 x 4
   species   count avg_height max_mass
   <chr>     <int>      <dbl>    <dbl>
 1 Aleena        1       79         15
 2 Besalisk      1      198        102
 3 Cerean        1      198         82
 4 Chagrian      1      196       -Inf
 5 Clawdite      1      168         55
 6 Droid         6      131.       140
 7 Dug           1      112         40
 8 Ewok          1       88         20
 9 Geonosian     1      183         80
10 Gungan        3      209.        82
# i 28 more rows
```

Advanced Summarize

```
# Conditional summaries
starwars |>
  group_by(species) |>
  summarise(
    count = n(),
    humans_count = sum(species == "Human", na.rm = TRUE),
    avg_height = mean(height, na.rm = TRUE),
    height_range = max(height, na.rm = TRUE) - min(height, na.rm = TRUE),
    .groups = "drop"
  )
```

```
# A tibble: 38 x 5
   species   count humans_count avg_height height_range
   <chr>     <int>        <int>      <dbl>        <int>
 1 Aleena        1            0         79            0
 2 Besalisk      1            0        198            0
 3 Cerean        1            0        198            0
 4 Chagrian      1            0        196            0
 5 Clawdite      1            0        168            0
 6 Droid         6            0        131.         104
 7 Dug           1            0        112            0
 8 Ewok          1            0         88            0
 9 Geonosian     1            0        183            0
10 Gungan        3            0        209.          28
# i 28 more rows
```

Why this matters for arrow

Each of these verbs has an equivalent when working with **arrow-backed data**, allowing you to **scale from local tibbles to massive datasets** without changing your `dplyr` workflow.

Next up: we'll see how to **read and query large datasets** using `arrow`, keeping the same grammar you already know.

Typical `dplyr` Workflow (Local or Scalable)

Here's a general-purpose `dplyr` pipeline workflow that applies to most tidyverse-style data tasks — whether you're working with small data (`tibble`), big local files (`arrow`), or SQL-like queries (`duckdb`):

Step-by-Step Pattern:

1. **Read or Connect to the Data**
    - `read_csv()`, `read_parquet()` for files (we will see this in the next module)
    - `open_dataset()` (arrow)
    - `dbConnect()` + `tbl()` (duckdb)
2. **Initial Filtering**

- `filter()` to narrow rows of interest early

3. **Select Columns**

- `select()` to reduce memory footprint and focus

4. **Mutate or Transform**

- `mutate()` to derive new columns (e.g., unit conversions, parsing

5. **Group and Summarise**

- `group_by()` + `summarise()` for aggregate

6. **Arrange or Rank**

- `arrange()` or `mutate(rank = ...)` to sort results

7. **Join or Bind**

- `left_join()`, `bind_rows()` as needed

8. **Collect to Memory (next session)**

- `collect()` for `arrow` or `duckdb` workflows when you're ready to compute

9. **Visualize or Write Out**

- `ggplot()`, `write_csv()`, or store to `.parquet`, `.csv`, `.duckdb`, etc.

Building Pipelines: The dplyr Way Here's the general pattern we'll use throughout the workshop:

```
data |>                          # 1. Start with data
  filter(condition) |>           # 2. Filter early (reduce rows)
  select(relevant_columns) |>    # 3. Select early (reduce columns)
  mutate(new_variables) |>       # 4. Transform as needed
  group_by(grouping_vars) |>     # 5. Group for summaries
  summarise(summary_stats) |>    # 6. Calculate aggregates
  arrange(sorting_column)        # 7. Sort results
```

Common Mistakes and Solutions

1. Forgetting the Pipe

```r
#  Wrong - breaks the pipeline
starwars |>
  filter(species == "Human")
select(name, height)  # This won't work!


#  Correct s
tarwars |>
  filter(species == "Human") |>
  select(name, height)
```

2. Not Handling Missing Values

```r
#  Will return NA
starwars |>
  summarise(avg_height = mean(height))

#  Handle NAs explicitly s
tarwars |>
  summarise(avg_height = mean(height, na.rm = TRUE))
```

3. Forgetting to Ungroup

```r
#  Leaves data grouped (can cause issues later)
grouped_data <- starwars |>
  group_by(species) |>
  summarise(count = n())
#  Always ungroup when done
clean_data <- starwars |>
  group_by(species) |>
  summarise(count = n(), .groups = "drop")
```

4. Incorrect Logical Operators

```r
#  Wrong - this is assignment, not comparison
starwars |>
```

```
  filter(species = "Human")

#  Correct - use == for comparison
starwars |>
  filter(species == "Human")

#  Wrong - can't use && in filter
starwars |>
  filter(height > 180 && mass < 100)

#  Correct - use & or separate conditions
starwars |>
  filter(height > 180 & mass < 100) # OR starwars |> filter(height > 180, mass < 100)
```

Try It Yourself: Challenges

Now it's your turn to practice! Work with a partner and help each other or take a brain break!

Beginner Challenge (5-10 minutes)

Goal: Practice the basic dplyr verbs with the `starwars` dataset.

Your mission: Find the tallest character from each homeworld.

Hints:

1. Start with `starwars`
2. Remove rows where `height` is missing
3. Group by `homeworld`
4. Find the maximum height in each group
5. Keep only the name, homeworld, and height columns

```
# Your code here - try before looking at the solution!

# Solution with detailed steps
tallest_by_homeworld <- starwars |>
  # Step 1: Remove rows where height is missing
  filter(!is.na(height)) |>
```

```r
  # Step 2: Group by homeworld to analyze each separately
  group_by(homeworld) |>

  # Step 3: For each homeworld, find the character with max height
  filter(height == max(height)) |>  # Keep only the tallest in each group

  # Step 4: Select only the columns we care about
  select(name, homeworld, height) |>

  # Step 5: Sort by height for easier reading
  arrange(desc(height)) |>

  # Step 6: Remove grouping for clean output
  ungroup()

# Display the results
tallest_by_homeworld
```

```
# A tibble: 49 x 3
   name            homeworld height
   <chr>           <chr>      <int>
 1 Yarael Poof     Quermia      264
 2 Tarfful         Kashyyyk     234
 3 Lama Su         Kamino       229
 4 Roos Tarpals    Naboo        224
 5 Grievous        Kalee        216
 6 Tion Medon      Utapau       206
 7 Darth Vader     Tatooine     202
 8 IG-88           <NA>         200
 9 Ki-Adi-Mundi    Cerea        198
10 Dexter Jettster Ojom         198
# i 39 more rows
```

Stretch goal: Can you also find the shortest character from each homeworld?

```r
# Find the shortest character from each homeworld
shortest_by_homeworld <- starwars |>
  filter(!is.na(height)) |>
  group_by(homeworld) |>
  filter(height == min(height)) |>  # Change max to min
  select(name, homeworld, height) |>
```

```
  arrange(height) |>  # Sort ascending instead of descending
  ungroup()

shortest_by_homeworld
```

```
# A tibble: 49 x 3
   name                  homeworld   height
   <chr>                 <chr>        <int>
 1 Yoda                  <NA>            66
 2 Ratts Tyerel          Aleen Minor     79
 3 Wicket Systri Warrick Endor           88
 4 Dud Bolt              Vulpter         94
 5 R2-D2                 Naboo           96
 6 R5-D4                 Tatooine        97
 7 Sebulba               Malastare      112
 8 Gasgano               Troiken        122
 9 Watto                 Toydaria       137
10 Leia Organa           Alderaan       150
# i 39 more rows
```

Intermediate Challenge (10-15 minutes)

Goal: Build a more complex analytical pipeline.

Your mission: Create a summary report of species diversity across different homeworlds.

Requirements:

1. Count how many different species live on each homeworld
2. Count the total number of characters from each homeworld
3. Calculate the "diversity ratio" (species count / character count)
4. Include the most common species on each homeworld
5. Sort by diversity ratio (most diverse first)

```
# Your code here - try before looking at the solution!

# Step 1: Create the main diversity summary
homeworld_diversity <- starwars |>
  # Remove rows with missing homeworld or species
  filter(!is.na(homeworld), !is.na(species)) |>
```

```r
  # Group by homeworld to analyze each separately
  group_by(homeworld) |>

  # Calculate diversity metrics
  summarise(
    # Count unique species
    species_count = n_distinct(species),

    # Count total characters
    character_count = n(),

    # Calculate diversity ratio
    diversity_ratio = species_count / character_count,

    # Find most common species (first alphabetically if tied)
    most_common_species = names(sort(table(species), decreasing = TRUE))[1],

    # Keep groups for potential further operations
    .groups = "keep"
  ) |>

  # Remove grouping and sort by diversity ratio
  ungroup() |>
  arrange(desc(diversity_ratio))

# Display results
homeworld_diversity
```

```
# A tibble: 46 x 5
   homeworld   species_count character_count diversity_ratio most_common_species
   <chr>               <int>           <int>           <dbl> <chr>
 1 Aleen Minor             1               1               1 Aleena
 2 Bespin                  1               1               1 Human
 3 Cato Neimo~             1               1               1 Neimodian
 4 Cerea                   1               1               1 Cerean
 5 Champala                1               1               1 Chagrian
 6 Chandrila               1               1               1 Human
 7 Concord Da~             1               1               1 Human
 8 Dathomir                1               1               1 Zabrak
 9 Dorin                   1               1               1 Kel Dor
10 Endor                   1               1               1 Ewok
```

```
# i 36 more rows
```

Stretch goals:

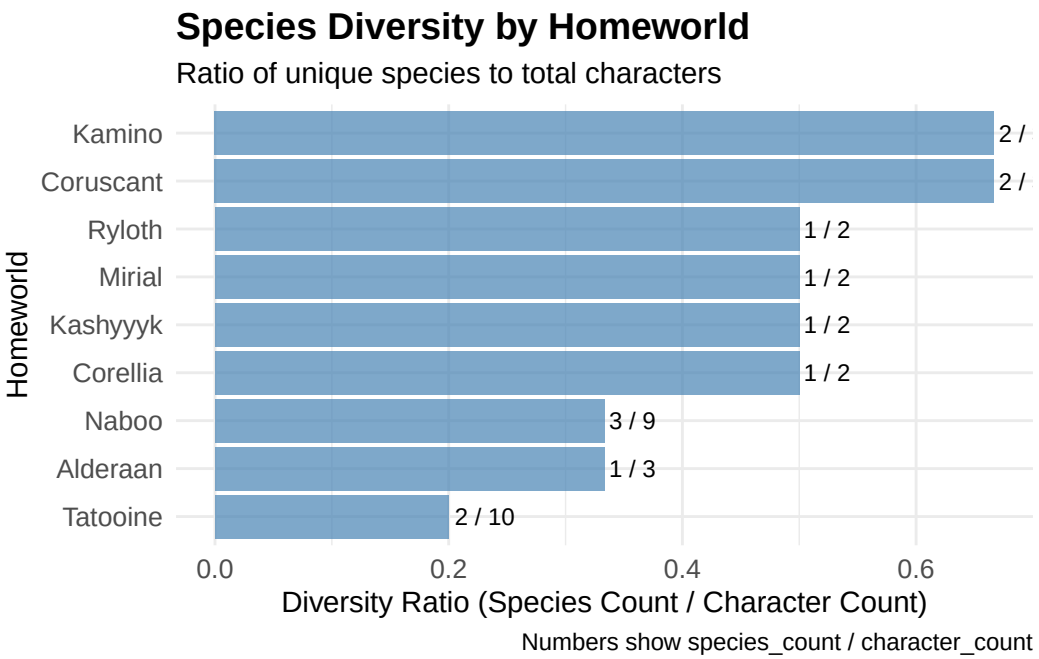
- Add average height and mass by homeworld

```
# Enhanced summary with physical characteristics
enhanced_diversity <- starwars |>
  filter(!is.na(homeworld), !is.na(species)) |>
  group_by(homeworld) |>
  summarise(
    species_count = n_distinct(species),
    character_count = n(),
    diversity_ratio = round(species_count / character_count, 3),

    # Physical characteristics
    avg_height = round(mean(height, na.rm = TRUE), 1),
    avg_mass = round(mean(mass, na.rm = TRUE), 1),

    # Most common species
    most_common_species = first(species[which.max(table(species))]),

    .groups = "drop"
  ) |>
  filter(character_count > 1) |>
  arrange(desc(diversity_ratio))

enhanced_diversity
```

```
# A tibble: 9 x 7
  homeworld species_count character_count diversity_ratio avg_height avg_mass
  <chr>             <int>           <int>           <dbl>      <dbl>    <dbl>
1 Coruscant             2               3           0.667       174.     50
2 Kamino                2               3           0.667       208.     83.1
3 Corellia              1               2           0.5         175      78.5
4 Kashyyyk              1               2           0.5         231     124
5 Mirial                1               2           0.5         168      53.1
6 Ryloth                1               2           0.5         179      55
7 Alderaan              1               3           0.333       176.     64
8 Naboo                 3               9           0.333       179.     60
9 Tatooine              2              10           0.2         170.     85.4
# i 1 more variable: most_common_species <chr>
```

- Create a visualization of your results

```
# Visualize diversity across homeworlds
diversity_plot <- homeworld_diversity |>
  filter(character_count > 1) |>  # Only homeworlds with multiple characters
  ggplot(aes(x = reorder(homeworld, diversity_ratio), y = diversity_ratio)) +
  geom_col(fill = "steelblue", alpha = 0.7) +
  geom_text(aes(label = paste(species_count, "/", character_count)),
            hjust = -0.1, size = 3) +
  coord_flip() +
  labs(
    title = "Species Diversity by Homeworld",
    subtitle = "Ratio of unique species to total characters",
    x = "Homeworld",
    y = "Diversity Ratio (Species Count / Character Count)",
    caption = "Numbers show species_count / character_count"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 10)
  )

diversity_plot
```



Why These Challenges Matter

These exercises demonstrate the **exact same thinking process** you'll use with big data:

1. **Filter early** to reduce data size
2. **Select** only what you need
3. **Group and summarize** to aggregate information
4. **Arrange** to present results clearly

The only difference with big data is that we'll add:

- `open_dataset()` instead of using built-in data
- `collect()` at the end to bring results into memory
- `show_query()` to see what's happening behind the scenes

What's Coming Next

In **Module 2**, we'll take these exact same dplyr skills and apply them to:

- A 9GB CSV file (40+ million rows)
- Converting CSV to Parquet for 5x speed improvements
- Processing data that's too large to fit in memory
- Using `arrow` for lazy evaluation and streaming

The promise: Same `dplyr` syntax you just practiced, but on datasets that are 100x larger!

NSF Acknowledgement: This material is based upon work supported by the National Science Foundation under Grant #DGE-2222148. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.