

2 Data Import and Wrangling

Mohan Khanal

2026-01-08

Table of contents

1	Chapter 2: Data Import and Data Wrangling	2
1.1	Agenda	2
1.2	Part 1: Reading Different Types of Data (30 mins)	2
1.2.1	CSV File Example: Iris Dataset	2
1.2.2	Understanding the Difference: read.csv() vs read_csv()	3
1.2.3	Quick Data Exploration	4
1.3	Part 2: Data Cleaning and Manipulation using dplyr (40 mins)	4
1.3.1	Load Required Packages	4
1.3.2	The Pipe Operator: %>%	5
1.3.3	Basic Wrangling Examples	6
1.3.4	Combining Multiple Operations	13
1.3.5	Additional Useful Functions	14
1.4	Part 3: Handling Missing Data (30 mins)	15
1.4.1	Checking Missing Values	15
1.4.2	Creating Sample Data with Missing Values	16
1.4.3	Removing Missing Values	17
1.4.4	Replacing Missing Values	17
1.4.5	Handling NAs During Analysis	19
1.5	Part 4: Practice and Q&A (20 mins)	20
1.5.1	Practice Tasks (based on iris data)	20
1.6	Further Resources (for Data Import & Wrangling)	23

1 Chapter 2: Data Import and Data Wrangling

1.1 Agenda

- **Part 1 (30 mins)** – Data Import and Data Wrangling (Reading Different Types of Data)
 - **Part 2 (40 mins)** – Data Cleaning and Manipulation using `dplyr`
 - **Part 3 (30 mins)** – Handling Missing Data
 - **Part 4 (20 mins)** – Practice and Q&A
-

1.2 Part 1: Reading Different Types of Data (30 mins)

We can also check our working directory using the command `getwd()`.

1.2.1 CSV File Example: Iris Dataset

```
# We can import csv file using the base library using the command read.csv
iris_data <- read.csv("../data/Iris.csv")

# Or we can use readr library using the command read_csv to import csv
library(readr)
iris_data_r <- read_csv("../data/Iris.csv")
```

Rows: 150 Columns: 6

-- Column specification -----

Delimiter: ","

chr (1): Species

dbl (5): Id, SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm

i Use ``spec()`` to retrieve the full column specification for this data.

i Specify the column types or set ``show_col_types = FALSE`` to quiet this message.

```
# View column names
names(iris_data)
```

```
[1] "Id"          "SepalLengthCm" "SepalWidthCm"  "PetalLengthCm"
[5] "PetalWidthCm" "Species"
```

```
# View first few rows
head(iris_data)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	1	5.1	3.5	1.4	0.2	Iris-setosa
2	2	4.9	3.0	1.4	0.2	Iris-setosa
3	3	4.7	3.2	1.3	0.2	Iris-setosa
4	4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	5.0	3.6	1.4	0.2	Iris-setosa
6	6	5.4	3.9	1.7	0.4	Iris-setosa

```
# Check structure
str(iris_data)
```

```
'data.frame':  150 obs. of  6 variables:
 $ Id          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ SepalLengthCm: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ SepalWidthCm : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ PetalLengthCm: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ PetalWidthCm : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species      : chr  "Iris-setosa" "Iris-setosa" "Iris-setosa" "Iris-
setosa" ...
```

1.2.2 Understanding the Difference: read.csv() vs read_csv()

```
# Base R: read.csv()
class(iris_data)      # Returns "data.frame"
```

```
[1] "data.frame"
```

```
# readr: read_csv()
class(iris_data_r)    # Returns "spec_tbl_df" "tbl_df" "tbl" "data.frame"
```

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

```
# Both work, but read_csv() is generally faster and has better defaults
```

1.2.3 Quick Data Exploration

```
# Dimensions  
dim(iris_data)
```

```
[1] 150  6
```

```
# Summary statistics  
summary(iris_data)
```

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm
Min. : 1.00	Min. :4.300	Min. :2.000	Min. :1.000
1st Qu.: 38.25	1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600
Median : 75.50	Median :5.800	Median :3.000	Median :4.350
Mean : 75.50	Mean :5.843	Mean :3.054	Mean :3.759
3rd Qu.:112.75	3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100
Max. :150.00	Max. :7.900	Max. :4.400	Max. :6.900
PetalWidthCm	Species		
Min. :0.100	Length:150		
1st Qu.:0.300	Class :character		
Median :1.300	Mode :character		
Mean :1.199			
3rd Qu.:1.800			
Max. :2.500			

```
# View in RStudio  
# View(iris_data)
```

1.3 Part 2: Data Cleaning and Manipulation using dplyr (40 mins)

1.3.1 Load Required Packages

```
library(dplyr)
```

```
Attaching package: 'dplyr'
```

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

`filter, lag`

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

`intersect, setdiff, setequal, union`

1.3.2 The Pipe Operator: %>%

The pipe operator %>% makes code more readable by chaining operations together.

```
# Without pipe (hard to read)
head(filter(iris_data, SepalLengthCm > 6))
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	51	7.0	3.2	4.7	1.4	Iris-versicolor
2	52	6.4	3.2	4.5	1.5	Iris-versicolor
3	53	6.9	3.1	4.9	1.5	Iris-versicolor
4	55	6.5	2.8	4.6	1.5	Iris-versicolor
5	57	6.3	3.3	4.7	1.6	Iris-versicolor
6	59	6.6	2.9	4.6	1.3	Iris-versicolor

```
# With pipe (easier to read)
iris_data %>%
  filter(SepalLengthCm > 6) %>%
  head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	51	7.0	3.2	4.7	1.4	Iris-versicolor
2	52	6.4	3.2	4.5	1.5	Iris-versicolor
3	53	6.9	3.1	4.9	1.5	Iris-versicolor
4	55	6.5	2.8	4.6	1.5	Iris-versicolor
5	57	6.3	3.3	4.7	1.6	Iris-versicolor
6	59	6.6	2.9	4.6	1.3	Iris-versicolor

Keyboard shortcut: Ctrl + Shift + M (Windows) or Cmd + Shift + M (Mac)

1.3.3 Basic Wrangling Examples

1.3.3.1 Selecting Columns with `select()`

```
# Select specific columns
iris_data %>%
  select(Species, SepalLengthCm, PetalLengthCm) %>%
  head()
```

	Species	SepalLengthCm	PetalLengthCm
1	Iris-setosa	5.1	1.4
2	Iris-setosa	4.9	1.4
3	Iris-setosa	4.7	1.3
4	Iris-setosa	4.6	1.5
5	Iris-setosa	5.0	1.4
6	Iris-setosa	5.4	1.7

```
# Select columns using patterns
iris_data %>%
  select(starts_with("Petal")) %>%
  head()
```

	PetalLengthCm	PetalWidthCm
1	1.4	0.2
2	1.4	0.2
3	1.3	0.2
4	1.5	0.2
5	1.4	0.2
6	1.7	0.4

```
iris_data %>%
  select(ends_with("Cm")) %>%
  head()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4

```
# Exclude columns with minus sign
iris_data %>%
  select(-Id) %>%
  head()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa

1.3.3.2 Filtering Rows with filter()

```
# Filter for setosa species only
iris_data %>%
  filter(Species == "Iris-setosa") %>%
  head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	1	5.1	3.5	1.4	0.2	Iris-setosa
2	2	4.9	3.0	1.4	0.2	Iris-setosa
3	3	4.7	3.2	1.3	0.2	Iris-setosa
4	4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	5.0	3.6	1.4	0.2	Iris-setosa
6	6	5.4	3.9	1.7	0.4	Iris-setosa

```
# Filter flowers with long petals
iris_data %>%
  filter(PetalLengthCm > 5) %>%
  head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	84	6.0	2.7	5.1	1.6	Iris-versicolor
2	101	6.3	3.3	6.0	2.5	Iris-virginica
3	102	5.8	2.7	5.1	1.9	Iris-virginica
4	103	7.1	3.0	5.9	2.1	Iris-virginica
5	104	6.3	2.9	5.6	1.8	Iris-virginica
6	105	6.5	3.0	5.8	2.2	Iris-virginica

```
# Multiple conditions with AND (&)
iris_data %>%
  filter(Species == "Iris-virginica" & SepalLengthCm > 6.5) %>%
  head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	103	7.1	3.0	5.9	2.1	Iris-virginica
2	106	7.6	3.0	6.6	2.1	Iris-virginica
3	108	7.3	2.9	6.3	1.8	Iris-virginica
4	109	6.7	2.5	5.8	1.8	Iris-virginica
5	110	7.2	3.6	6.1	2.5	Iris-virginica
6	113	6.8	3.0	5.5	2.1	Iris-virginica

```
# Multiple conditions with OR (|)
iris_data %>%
  filter(PetalLengthCm > 6 | PetalWidthCm > 2) %>%
  head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	101	6.3	3.3	6.0	2.5	Iris-virginica
2	103	7.1	3.0	5.9	2.1	Iris-virginica
3	105	6.5	3.0	5.8	2.2	Iris-virginica
4	106	7.6	3.0	6.6	2.1	Iris-virginica
5	108	7.3	2.9	6.3	1.8	Iris-virginica
6	110	7.2	3.6	6.1	2.5	Iris-virginica

```
# Using %in% operator for multiple values
iris_data %>%
  filter(Species %in% c("Iris-setosa", "Iris-versicolor")) %>%
  head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	1	5.1	3.5	1.4	0.2	Iris-setosa
2	2	4.9	3.0	1.4	0.2	Iris-setosa
3	3	4.7	3.2	1.3	0.2	Iris-setosa
4	4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	5.0	3.6	1.4	0.2	Iris-setosa
6	6	5.4	3.9	1.7	0.4	Iris-setosa

1.3.3.3 Creating New Variables with mutate()

```
# Calculate areas
iris_data <- iris_data %>%
  mutate(
    SepalArea = SepalLengthCm * SepalWidthCm,
    PetalArea = PetalLengthCm * PetalWidthCm
  )

head(iris_data)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	1	5.1	3.5	1.4	0.2	Iris-setosa
2	2	4.9	3.0	1.4	0.2	Iris-setosa
3	3	4.7	3.2	1.3	0.2	Iris-setosa
4	4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	5.0	3.6	1.4	0.2	Iris-setosa
6	6	5.4	3.9	1.7	0.4	Iris-setosa

	SepalArea	PetalArea
1	17.85	0.28
2	14.70	0.28
3	15.04	0.26
4	14.26	0.30
5	18.00	0.28
6	21.06	0.68

```
# Create categorical variables
iris_data <- iris_data %>%
  mutate(
    SizeCategory = ifelse(PetalLengthCm > 4, "Large", "Small"),
    SepalRatio = SepalLengthCm / SepalWidthCm
  )

head(iris_data)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	1	5.1	3.5	1.4	0.2	Iris-setosa
2	2	4.9	3.0	1.4	0.2	Iris-setosa
3	3	4.7	3.2	1.3	0.2	Iris-setosa
4	4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	5.0	3.6	1.4	0.2	Iris-setosa

6	6	5.4	3.9	1.7	0.4	Iris-setosa
	SepalArea	PetalArea	SizeCategory	SepalRatio		
1	17.85	0.28	Small	1.457143		
2	14.70	0.28	Small	1.633333		
3	15.04	0.26	Small	1.468750		
4	14.26	0.30	Small	1.483871		
5	18.00	0.28	Small	1.388889		
6	21.06	0.68	Small	1.384615		

```
# Using case_when() for multiple conditions
iris_data <- iris_data %>%
  mutate(
    PetalSize = case_when(
      PetalLengthCm < 2 ~ "Small",
      PetalLengthCm < 5 ~ "Medium",
      TRUE ~ "Large"
    )
  )
head(iris_data)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	1	5.1	3.5	1.4	0.2	Iris-setosa
2	2	4.9	3.0	1.4	0.2	Iris-setosa
3	3	4.7	3.2	1.3	0.2	Iris-setosa
4	4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	5.0	3.6	1.4	0.2	Iris-setosa
6	6	5.4	3.9	1.7	0.4	Iris-setosa

	SepalArea	PetalArea	SizeCategory	SepalRatio	PetalSize
1	17.85	0.28	Small	1.457143	Small
2	14.70	0.28	Small	1.633333	Small
3	15.04	0.26	Small	1.468750	Small
4	14.26	0.30	Small	1.483871	Small
5	18.00	0.28	Small	1.388889	Small
6	21.06	0.68	Small	1.384615	Small

1.3.3.4 Sorting Rows with arrange()

```
# Sort by Petal Length (ascending)
iris_data %>%
  arrange(PetalLengthCm) %>%
  head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	23	4.6	3.6	1.0	0.2	Iris-setosa
2	14	4.3	3.0	1.1	0.1	Iris-setosa
3	15	5.8	4.0	1.2	0.2	Iris-setosa
4	36	5.0	3.2	1.2	0.2	Iris-setosa
5	3	4.7	3.2	1.3	0.2	Iris-setosa
6	17	5.4	3.9	1.3	0.4	Iris-setosa

	SepalArea	PetalArea	SizeCategory	SepalRatio	PetalSize
1	16.56	0.20	Small	1.277778	Small
2	12.90	0.11	Small	1.433333	Small
3	23.20	0.24	Small	1.450000	Small
4	16.00	0.24	Small	1.562500	Small
5	15.04	0.26	Small	1.468750	Small
6	21.06	0.52	Small	1.384615	Small

```
# Sort by Petal Length (descending)
iris_data %>%
  arrange(desc(PetalLengthCm)) %>%
  head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	119	7.7	2.6	6.9	2.3	Iris-virginica
2	118	7.7	3.8	6.7	2.2	Iris-virginica
3	123	7.7	2.8	6.7	2.0	Iris-virginica
4	106	7.6	3.0	6.6	2.1	Iris-virginica
5	132	7.9	3.8	6.4	2.0	Iris-virginica
6	108	7.3	2.9	6.3	1.8	Iris-virginica

	SepalArea	PetalArea	SizeCategory	SepalRatio	PetalSize
1	20.02	15.87	Large	2.961538	Large
2	29.26	14.74	Large	2.026316	Large
3	21.56	13.40	Large	2.750000	Large
4	22.80	13.86	Large	2.533333	Large
5	30.02	12.80	Large	2.078947	Large
6	21.17	11.34	Large	2.517241	Large

```
# Sort by multiple columns
iris_data %>%
  arrange(Species, desc(SepalLengthCm)) %>%
  head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
--	----	---------------	--------------	---------------	--------------	---------

1	15	5.8	4.0	1.2	0.2	Iris-setosa
2	16	5.7	4.4	1.5	0.4	Iris-setosa
3	19	5.7	3.8	1.7	0.3	Iris-setosa
4	34	5.5	4.2	1.4	0.2	Iris-setosa
5	37	5.5	3.5	1.3	0.2	Iris-setosa
6	6	5.4	3.9	1.7	0.4	Iris-setosa

	SepalArea	PetalArea	SizeCategory	SepalRatio	PetalSize
1	23.20	0.24	Small	1.450000	Small
2	25.08	0.60	Small	1.295455	Small
3	21.66	0.51	Small	1.500000	Small
4	23.10	0.28	Small	1.309524	Small
5	19.25	0.26	Small	1.571429	Small
6	21.06	0.68	Small	1.384615	Small

1.3.3.5 Summarizing with group_by() and summarize()

```
# Overall summary
iris_data %>%
  summarize(
    avg_sepal_length = mean(SepalLengthCm, na.rm = TRUE),
    avg_petal_length = mean(PetalLengthCm, na.rm = TRUE),
    total_flowers = n()
  )
```

	avg_sepal_length	avg_petal_length	total_flowers
1	5.843333	3.758667	150

```
# Summary by species
iris_data %>%
  group_by(Species) %>%
  summarize(
    avg_sepal_length = mean(SepalLengthCm, na.rm = TRUE),
    avg_sepal_width = mean(SepalWidthCm, na.rm = TRUE),
    avg_petal_length = mean(PetalLengthCm, na.rm = TRUE),
    avg_petal_width = mean(PetalWidthCm, na.rm = TRUE),
    count = n()
  )
```

```
# A tibble: 3 x 6
  Species      avg_sepal_length avg_sepal_width avg_petal_length avg_petal_width
  <chr>          <dbl>          <dbl>          <dbl>          <dbl>
```

1	Iris-setosa	5.01	3.42	1.46	0.244
2	Iris-versic~	5.94	2.77	4.26	1.33
3	Iris-virgin~	6.59	2.97	5.55	2.03

i 1 more variable: count <int>

```
# Summary by multiple groups
iris_data %>%
  group_by(Species, SizeCategory) %>%
  summarize(
    avg_sepal_length = mean(SepalLengthCm, na.rm = TRUE),
    count = n(),
    .groups = 'drop'
  )
```

A tibble: 4 x 4

	Species	SizeCategory	avg_sepal_length	count
	<chr>	<chr>	<dbl>	<int>
1	Iris-setosa	Small	5.01	50
2	Iris-versicolor	Large	6.15	34
3	Iris-versicolor	Small	5.49	16
4	Iris-virginica	Large	6.59	50

1.3.4 Combining Multiple Operations

```
# Complex pipeline: Find large virginica flowers and their stats
iris_data %>%
  filter(Species == "Iris-virginica", PetalLengthCm > 6) %>%
  select(Species, SepalLengthCm, PetalLengthCm, SepalArea, PetalArea) %>%
  arrange(desc(PetalArea)) %>%
  head(10)
```

	Species	SepalLengthCm	PetalLengthCm	SepalArea	PetalArea
1	Iris-virginica	7.7	6.9	20.02	15.87
2	Iris-virginica	7.2	6.1	25.92	15.25
3	Iris-virginica	7.7	6.7	29.26	14.74
4	Iris-virginica	7.7	6.1	23.10	14.03
5	Iris-virginica	7.6	6.6	22.80	13.86
6	Iris-virginica	7.7	6.7	21.56	13.40
7	Iris-virginica	7.9	6.4	30.02	12.80
8	Iris-virginica	7.4	6.1	20.72	11.59
9	Iris-virginica	7.3	6.3	21.17	11.34

1.3.5 Additional Useful Functions

1.3.5.1 Count rows with count()

```
# Count flowers by species
iris_data %>%
  count(Species)
```

```
      Species  n
1  Iris-setosa 50
2 Iris-versicolor 50
3  Iris-virginica 50
```

```
# Count with sorting
iris_data %>%
  count(Species, sort = TRUE)
```

```
      Species  n
1  Iris-setosa 50
2 Iris-versicolor 50
3  Iris-virginica 50
```

```
# Count by multiple variables
iris_data %>%
  count(Species, SizeCategory)
```

```
      Species SizeCategory  n
1  Iris-setosa      Small  50
2 Iris-versicolor      Large 34
3 Iris-versicolor      Small 16
4  Iris-virginica      Large 50
```

1.3.5.2 Get unique values with distinct()

```
# Get unique species
iris_data %>%
  distinct(Species)
```

```

      Species
1    Iris-setosa
2 Iris-versicolor
3  Iris-virginica

```

```

# Get unique combinations
iris_data %>%
  distinct(Species, PetalSize)

```

```

      Species PetalSize
1    Iris-setosa    Small
2 Iris-versicolor  Medium
3 Iris-versicolor   Large
4  Iris-virginica   Large
5  Iris-virginica  Medium

```

1.4 Part 3: Handling Missing Data (30 mins)

1.4.1 Checking Missing Values

```

# Load nanian for missing data visualization
library(nanian)

# Check for missing values
miss_var_summary(iris_data)

```

```

# A tibble: 11 x 3
  variable      n_miss pct_miss
  <chr>         <int>   <num>
1 Id             0         0
2 SepalLengthCm  0         0
3 SepalWidthCm   0         0
4 PetalLengthCm  0         0
5 PetalWidthCm   0         0
6 Species        0         0
7 SepalArea      0         0
8 PetalArea      0         0

```

```

 9 SizeCategory      0      0
10 SepalRatio        0      0
11 PetalSize         0      0

```

```

# Check if any values are missing
any(is.na(iris_data))

```

```
[1] FALSE
```

```

# Count NAs in each column
colSums(is.na(iris_data))

```

```

      Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm
      0              0              0              0              0
Species      SepalArea      PetalArea SizeCategory      SepalRatio
      0              0              0              0              0
PetalSize
      0

```

1.4.2 Creating Sample Data with Missing Values

For demonstration, let's introduce some missing values:

```

# Create a copy with missing values
iris_missing <- iris_data
set.seed(123) # For reproducibility

# Randomly introduce NAs
iris_missing$SepalLengthCm[sample(1:nrow(iris_missing), 10)] <- NA
iris_missing$PetalWidthCm[sample(1:nrow(iris_missing), 15)] <- NA

# Check missing data
miss_var_summary(iris_missing)

```

```

# A tibble: 11 x 3
  variable      n_miss pct_miss
  <chr>         <int>   <num>
1 PetalWidthCm     15     10
2 SepalLengthCm    10     6.67
3 Id               0      0

```


4	SepalWidthCm	0	0
5	PetalLengthCm	0	0
6	Species	0	0
7	SepalArea	0	0
8	PetalArea	0	0
9	SizeCategory	0	0
10	SepalRatio	0	0
11	PetalSize	0	0

1.4.3 Removing Missing Values

```
# Remove rows with any missing values
iris_complete <- na.omit(iris_missing)

cat("Original rows:", nrow(iris_missing), "\n")
```

Original rows: 150

```
cat("After removing NAs:", nrow(iris_complete), "\n")
```

After removing NAs: 127

```
# Remove only if specific column has NA
iris_filtered <- iris_missing %>%
  filter(!is.na(SepalLengthCm))

cat("After filtering SepalLengthCm:", nrow(iris_filtered), "\n")
```

After filtering SepalLengthCm: 140

1.4.4 Replacing Missing Values

```
# Replace NAs with mean
iris_mean <- iris_missing %>%
  mutate(
    SepalLengthCm = ifelse(is.na(SepalLengthCm),
                          mean(SepalLengthCm, na.rm = TRUE),
```

```

        SepalLengthCm),
    PetalWidthCm = ifelse(is.na(PetalWidthCm),
                          mean(PetalWidthCm, na.rm = TRUE),
                          PetalWidthCm)
  )

# Verify
miss_var_summary(iris_mean)

# A tibble: 11 x 3
  variable      n_miss pct_miss
  <chr>         <int>    <num>
1 Id             0         0
2 SepalLengthCm  0         0
3 SepalWidthCm   0         0
4 PetalLengthCm  0         0
5 PetalWidthCm   0         0
6 Species        0         0
7 SepalArea      0         0
8 PetalArea      0         0
9 SizeCategory   0         0
10 SepalRatio     0         0
11 PetalSize      0         0

# Replace with median (more robust to outliers)
iris_median <- iris_missing %>%
  mutate(
    SepalLengthCm = ifelse(is.na(SepalLengthCm),
                          median(SepalLengthCm, na.rm = TRUE),
                          SepalLengthCm),
    PetalWidthCm = ifelse(is.na(PetalWidthCm),
                          median(PetalWidthCm, na.rm = TRUE),
                          PetalWidthCm)
  )

# Replace with species-specific mean (group imputation)
iris_group_mean <- iris_missing %>%
  group_by(Species) %>%
  mutate(
    SepalLengthCm = ifelse(is.na(SepalLengthCm),
                          mean(SepalLengthCm, na.rm = TRUE),

```

```

        SepalLengthCm),
    PetalWidthCm = ifelse(is.na(PetalWidthCm),
        mean(PetalWidthCm, na.rm = TRUE),
        PetalWidthCm)
) %>%
ungroup()

miss_var_summary(iris_group_mean)

```

```

# A tibble: 11 x 3
  variable      n_miss pct_miss
  <chr>         <int>    <num>
1 Id             0         0
2 SepalLengthCm  0         0
3 SepalWidthCm   0         0
4 PetalLengthCm  0         0
5 PetalWidthCm   0         0
6 Species        0         0
7 SepalArea      0         0
8 PetalArea      0         0
9 SizeCategory   0         0
10 SepalRatio     0         0
11 PetalSize      0         0

```

1.4.5 Handling NAs During Analysis

```

# Without na.rm - returns NA
mean(iris_missing$SepalLengthCm)

```

```
[1] NA
```

```

# With na.rm = TRUE - ignores NAs
mean(iris_missing$SepalLengthCm, na.rm = TRUE)

```

```
[1] 5.855714
```

```
# Summary with NA handling
iris_missing %>%
  summarize(
    mean_sepal = mean(SepalLengthCm, na.rm = TRUE),
    median_sepal = median(SepalLengthCm, na.rm = TRUE),
    count_non_na = sum(!is.na(SepalLengthCm)),
    count_na = sum(is.na(SepalLengthCm))
  )
```

```
mean_sepal median_sepal count_non_na count_na
1 5.855714 5.8 140 10
```

1.5 Part 4: Practice and Q&A (20 mins)

1.5.1 Practice Tasks (based on iris data)

1.5.1.1 Task 1: Import and Explore

- Import the iris dataset
- Display the first 10 rows
- Check the number of rows and columns

```
# Solution
iris_practice <- read.csv("../data/Iris.csv")
head(iris_practice, 10)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	1	5.1	3.5	1.4	0.2	Iris-setosa
2	2	4.9	3.0	1.4	0.2	Iris-setosa
3	3	4.7	3.2	1.3	0.2	Iris-setosa
4	4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	5.0	3.6	1.4	0.2	Iris-setosa
6	6	5.4	3.9	1.7	0.4	Iris-setosa
7	7	4.6	3.4	1.4	0.3	Iris-setosa
8	8	5.0	3.4	1.5	0.2	Iris-setosa
9	9	4.4	2.9	1.4	0.2	Iris-setosa
10	10	4.9	3.1	1.5	0.1	Iris-setosa

```
dim(iris_practice)
```

```
[1] 150   6
```

1.5.1.2 Task 2: Filter and Select

- Select only Iris-virginica flowers with PetalLengthCm > 6
- Show only Species, SepalLengthCm, and PetalLengthCm columns

```
# Solution
iris_practice %>%
  filter(Species == "Iris-virginica", PetalLengthCm > 6) %>%
  select(Species, SepalLengthCm, PetalLengthCm)
```

	Species	SepalLengthCm	PetalLengthCm
1	Iris-virginica	7.6	6.6
2	Iris-virginica	7.3	6.3
3	Iris-virginica	7.2	6.1
4	Iris-virginica	7.7	6.7
5	Iris-virginica	7.7	6.9
6	Iris-virginica	7.7	6.7
7	Iris-virginica	7.4	6.1
8	Iris-virginica	7.9	6.4
9	Iris-virginica	7.7	6.1

1.5.1.3 Task 3: Create New Variables

- Create a variable for PetalRatio = PetalLengthCm / PetalWidthCm
- Create a variable for SepalRatio = SepalLengthCm / SepalWidthCm

```
# Solution
iris_practice <- iris_practice %>%
  mutate(
    PetalRatio = PetalLengthCm / PetalWidthCm,
    SepalRatio = SepalLengthCm / SepalWidthCm
  )

head(iris_practice)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	1	5.1	3.5	1.4	0.2	Iris-setosa
2	2	4.9	3.0	1.4	0.2	Iris-setosa
3	3	4.7	3.2	1.3	0.2	Iris-setosa
4	4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	5.0	3.6	1.4	0.2	Iris-setosa
6	6	5.4	3.9	1.7	0.4	Iris-setosa

	PetalRatio	SepalRatio
1	7.00	1.457143
2	7.00	1.633333
3	6.50	1.468750
4	7.50	1.483871
5	7.00	1.388889
6	4.25	1.384615

1.5.1.4 Task 4: Group and Summarize

- Group by Species
- Calculate average SepalLengthCm, PetalLengthCm, and count per species

```
# Solution
iris_practice %>%
  group_by(Species) %>%
  summarize(
    avg_sepal_length = mean(SepalLengthCm, na.rm = TRUE),
    avg_petal_length = mean(PetalLengthCm, na.rm = TRUE),
    count = n()
  )
```

```
# A tibble: 3 x 4
  Species      avg_sepal_length avg_petal_length count
  <chr>          <dbl>          <dbl> <int>
1 Iris-setosa      5.01            1.46     50
2 Iris-versicolor  5.94            4.26     50
3 Iris-virginica   6.59            5.55     50
```

1.5.1.5 Task 5: Complex Pipeline

- Filter for flowers with SepalLengthCm > 6
- Create a new variable TotalLength = SepalLengthCm + PetalLengthCm
- Group by Species

- Calculate average TotalLength
- Sort by average TotalLength descending

```
# Solution
iris_practice %>%
  filter(SepalLengthCm > 6) %>%
  mutate(TotalLength = SepalLengthCm + PetalLengthCm) %>%
  group_by(Species) %>%
  summarize(
    avg_total_length = mean(TotalLength, na.rm = TRUE),
    count = n()
  ) %>%
  arrange(desc(avg_total_length))
```

```
# A tibble: 2 x 3
  Species      avg_total_length count
  <chr>          <dbl>   <int>
1 Iris-virginica      12.5     41
2 Iris-versicolor     11.0     20
```

1.6 Further Resources (for Data Import & Wrangling)

- Tidyverse Documentation: <https://www.tidyverse.org/packages/>
- Importing Data in R: <https://r4ds.hadley.nz/data-import.html>
- Data Transformation Cheatsheet: <https://posit.co/resources/cheatsheets/>