

2 Data Import and Wrangling

Mohan Khanal

2025-07-30

Table of contents

1	Chapter 2: Data Import and Data Wrangling	1
1.1	Agenda	1
1.2	Part 1: Reading Different Types of Data (30 mins)	2
1.2.1	CSV File Example: Car Dataset	2
1.3	Part 2: Data Cleaning and Manipulation using <code>dplyr</code> (40 mins)	2
1.3.1	Load Required Packages	2
1.3.2	Basic Wrangling Examples	3
1.4	Part 3: Handling Missing Data (30 mins)	4
1.4.1	Checking Missing Values	4
1.4.2	Removing or Replacing NAs	4
1.5	Part 4: Practice and Q&A (20 mins)	5
1.5.1	Practice Tasks (based on car data)	5
1.6	Further Resources (for Data Import & Wrangling)	6

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: Car Dataset

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

# or we can use readr library using the command read_csv to import csv

library(readr)
rcar_data <- read_csv("../data/car_data.csv")
```

Rows: 301 Columns: 9

-- Column specification -----

Delimiter: ","

chr (4): Car_Name, Fuel_Type, Seller_Type, Transmission

dbl (5): Year, Selling_Price, Present_Price, Kms_Driven, Owner

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.

```
names(car_data)
```

```
[1] "Car_Name"      "Year"          "Selling_Price" "Present_Price"
[5] "Kms_Driven"    "Fuel_Type"     "Seller_Type"   "Transmission"
[9] "Owner"
```

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 Basic Wrangling Examples

1.3.2.1 Selecting and Filtering

```
# Select relevant columns and filter cars after 2015
car_data_filtered <- car_data %>%
  select(Car_Name, Year, Selling_Price, Fuel_Type) %>%
  filter(Year > 2015)
```

1.3.2.2 Creating New Variables with mutate()

```
# Calculate car age and classify it
car_data <- car_data %>%
  mutate(
    Car_Age = 2025 - Year,
    Age_Group = ifelse(Car_Age <= 5, "New", "Old")
  )
```

1.3.2.3 Sorting Rows with arrange()

```
# Sort by Selling Price
car_data_sorted <- car_data %>% arrange(desc(Selling_Price))
```

1.3.2.4 Summarizing with group_by() and summarize()

```
# Average selling price by fuel type
car_data %>%
  group_by(Fuel_Type) %>%
  summarize(avg_price = mean(Selling_Price, na.rm = TRUE))
```

```
# A tibble: 3 x 2
  Fuel_Type avg_price
  <chr>      <dbl>
1 CNG        3.1
2 Diesel    10.3
3 Petrol     3.26
```

1.4 Part 3: Handling Missing Data (30 mins)

1.4.1 Checking Missing Values

```
library(naniar)
miss_var_summary(car_data)
```

```
# A tibble: 11 x 3
  variable      n_miss pct_miss
  <chr>         <int>   <num>
1 Car_Name         0         0
2 Year             0         0
3 Selling_Price    0         0
4 Present_Price    0         0
5 Kms_Driven       0         0
6 Fuel_Type        0         0
7 Seller_Type      0         0
8 Transmission     0         0
9 Owner            0         0
10 Car_Age          0         0
11 Age_Group        0         0
```

1.4.2 Removing or Replacing NAs

```
# Remove rows with any missing values
car_data_clean <- na.omit(car_data)

# Replace missing Selling_Price with 0 (if any)
car_data <- car_data %>%
```

```
mutate(Selling_Price = ifelse(is.na(Selling_Price), 0, Selling_Price))

head(car_data)
```

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type
1	ritz	2014	3.35	5.59	27000	Petrol
2	sx4	2013	4.75	9.54	43000	Diesel
3	ciaz	2017	7.25	9.85	6900	Petrol
4	wagon r	2011	2.85	4.15	5200	Petrol
5	swift	2014	4.60	6.87	42450	Diesel
6	vitara brezza	2018	9.25	9.83	2071	Diesel

	Seller_Type	Transmission	Owner	Car_Age	Age_Group
1	Dealer	Manual	0	11	Old
2	Dealer	Manual	0	12	Old
3	Dealer	Manual	0	8	Old
4	Dealer	Manual	0	14	Old
5	Dealer	Manual	0	11	Old
6	Dealer	Manual	0	7	Old

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

1.5.1 Practice Tasks (based on car data)

- Import the car dataset
- Select cars with Present_Price > 10
- Create a variable to calculate depreciation = Present_Price - Selling_Price
- Group by Transmission type and calculate average depreciation

```
car_data %>%
  mutate(depreciation = Present_Price - Selling_Price) %>%
  group_by(Transmission) %>%
  summarize(avg_depreciation = mean(depreciation, na.rm = TRUE))
```

```
# A tibble: 2 x 2
  Transmission avg_depreciation
  <chr>          <dbl>
1 Automatic      5.90
2 Manual         2.52
```

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/>