Lab 3: Data wrangling

Not graded, just practice

Katie Schuler

2024-09-12

Materials from lab

- Brittany's slides
- Wesley's slides

1 Tidy

1.1 Tidyverse

1. What is the relationship between tidyverse and readr?
    - (A) tidyverse is a package in the readr family of packages
    - (B) readr is a package in the tidyverse family of packages
    - (C) tidyverse and readr are two unrelated packages
    - (D) tidyverse and reader are two names for the same package
2. In the tidyverse, what does "tidy data" refer to?
    - (A) any data we load into the tidyverse
    - (B) a dataset with no missing values
    - (C) a standard way to organize a dataset
    - (D) the process of cleaning a dataset

3. What is the purpose of the `purrr` package?

   - (A) Data visualization
   - (B) Data wrangling
   - (C) Data importing
   - (D) Functional programming
   - (E) All of the above

4. What is the primary purpose of the `readr` package?

   - (A) Data visualization
   - (B) Data wrangling
   - (C) Data importing
   - (D) Functional programming
   - (E) All of the above

5. Which of the following returned this message?

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.2     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.5.0     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
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
```

   - (A) `library(tidyverse)`
   - (B) `family(tidyverse)`
   - (C) `library.collection(tidyverse)`
   - (D) `library(tidyverse, report=TRUE)`

1.2 purrr

Suppose we have the following tibble, stored with the variable `df`.

```
# A tibble: 4 x 3
      x     y     z
  <int> <int> <int>
1     1     5     9
2     2     6    10
3     3     7    11
4     4     8    12
```

1. What will `map(df, mean)` return?
    - (A) the mean of each row
    - (B) the mean of each column
    - (C) the mean of all values
    - (D) Error: cannot compute mean of type integer
2. Suppose we wanted to coerce each column in the previous tibble to the data type `double` with one line of code. Fill in the two arguments to `map` that would accomplish this:

- map(_____, _____)

1.3 Tibbles

Suppose we run the following code block and create 3 tibbles:

```
# create tibble tib
tib <- tibble(x = 1:2, y = c("a", "b"))

# create tibble x
x <- tribble(
    ~x, ~y,
    2, 3,
    4, 5
)

# create tibble tibby
tibby <- tibble(
```

```
    age = c(1, 2, 3, 5),
    name = c("dory", "hazel", "graham", "joan"),
    alt_name = c("dolores", NA, NA, "joanie")
)
```

1. What will `is.data.frame(tib)` return?
   - (A) True
   - (B) False
2. What will `typeof(tib)` return?
   - (A) double
   - (B) character
   - (C) 'double' • 'character'
   - (D) list
   - (E) tibble
   - (F) data.frame
3. What will `is_tibble(x)` return?
   - (A) True
   - (B) False
4. Which of the following would convert a dataframe called `df` to a tibble? (note that `df` is not defined above, consider any artibrary dataframe)
   - (A) `as_tibble(df)`
   - (B) `as.data.frame(df, tibble)`
   - (C) `tribble(df)`
   - (D) `df %>% as_tibble()`
   - (E) `df |> as_tibble()`
5. What will `tibby$a` return?

- (A) a warning and the value NULL
- (B) `age` via partial matching
- (C) `age` and `alt_name` via partial matching
- (D) `hazel`, `graham`, `joan`, and `joanie` via partial matching
- (E) an empty vector

2 Import

The questions below refer to this dataset borrowed from R4DS and available at the url https://pos.it/r4ds-students-csv.

```
Student ID,Full Name,favourite.food,mealPlan,AGE
1,Sunil Huffmann,Strawberry yoghurt,Lunch only,4
2,Barclay Lynn,French fries,Lunch only,5
3,Jayendra Lyne,N/A,Breakfast and lunch,7
4,Leon Rossini,Anchovies,Lunch only,
5,Chidiegwu Dunkel,Pizza,Breakfast and lunch,five
6,Güvenç Attila,Ice cream,Lunch only,6
```

Table 8.1 shows a representation of the same data as a table.

Table 8.1: Data from the students.csv file as a table.

Student ID	Full Name	favourite.food	mealPlan	AGE
1	Sunil Huffmann	Strawberry yoghurt	Lunch only	4
2	Barclay Lynn	French fries	Lunch only	5
3	Jayendra Lyne	N/A	Breakfast and lunch	7
4	Leon Rossini	Anchovies	Lunch only	NA
5	Chidiegwu Dunkel	Pizza	Breakfast and lunch	five
6	Güvenç Attila	Ice cream	Lunch only	6

1. What does the csv in `read_csv()` stand for? Fill in the blank.

- _____ separated values

2. Suppose we attempt to import the csv file given above with the code below. What will be the result?

```
data <- read_csv("https://pos.it/r4ds-students-csv",
    col_types = list(AGE = col_double())
)
```

- (A) imports with no errors or warnings
- (B) fails to import, throws error
- (C) imports, but with a warning that there are parsing issues
- (D) imports, but changes the column name to `age`

3. Suppose we import the dataset given above and name it `data`. What will `is.na(data[3,3])` return?
   - (A) True
   - (B) False
4. Suppose we import the dataset given above and name it `data`. Which of the following would return the first column?
   - (A) `data[1]`
   - (B) `data[[1]]`
   - (C) `data[[Student ID]]`
   - (D) `data$`Student ID``
5. True or false, assuming the same dataset the following code would rename the `Student ID` column to `student_id`?

   ```
   data %>% rename(student_id = `Student ID`)
   ```

   - (A) True
   - (B) False
6. True or false, we can use a `read_*()` function from `readr` to import a google sheet.
   - (A) True
   - (B) False

3 Transform

1. Which of the following `dplyr` functions retuns a data frame?
    - (A) `select()`
    - (B) `mutate()`
    - (C) `filter()`
    - (D) `rename()`
    - (E) None of the above
2. Which of the following `dplyr` functions takes a number as their first argument?
    - (A) `select()`
    - (B) `mutate()`
    - (C) `filter()`
    - (D) `rename()`
    - (E) None of the above
3. True or false, the following code blocks are equivalent.

```
# option 1
ratings %>% select(Word, Frequency) %>% glimpse()

# option 2
glimpse(select(ratings, Word, Frequency))
```

    - (A) True
    - (B) False

4. True or false, the following code options are equivalent

```
# option 1
ratings %>%
    select(Word:Class) %>%
    mutate(Length/Frequency, .after = Class)
```

```
# option 2
ratings %>%
    select(Word:Class) %>%
    mutate(Length/Frequency)
```

- (A) True
- (B) False

5. Recall that there are two possible values in the `Class` variable in the `ratings` dataset: "animal" or "plant". How many rows would be in the data frame returned by the following code block?

```
ratings %>% group_by(Class) %>% summarise(n = n())
```

- (A) 0
- (B) 2
- (C) 4
- (D) 81

6. Given the code block in the previous question, what will `n()` do?

- (A) summarize all classes including the letter n
- (B) count the number of rows per Class
- (C) adds the string `n` before each value of Class
- (D) error: missing arguments to n()

7. True or false, the following code blocks will return the same dataframe

```
# code block 1
ratings %>% select(complexity = Complex)


# code block 2
ratings %>% rename(complexity = Complex)
```

- (A) True
- (B) False

8. Which of the following code blocks will return a dataframe including only the rows in `ratings` for which the Class value is "animal"?

```
# code block a
ratings %>% filter(Class = "animal")

# code block b
ratings %>% filter(Class == "animal")
```

- (A) a
- (B) b
- (C) both a and b

9. By default the `arrange()` function arranges the rows in ascending order. Which of the following code blocks would arrange the Frequency variable in descending order?

```
# code block a
ratings %>% arrange(Frequency, order = "descending")

# code block b
ratings %>% arrange(Frequency, order = "reverse")

# code block c
ratings %>% arrange(desc(Frequency))
```

- (A) a
- (B) b
- (C) c
- (D) a and b
- (E) a and c

10. Which of the following code blocks could be used to return the mean frequency by class?

```
# code block a
ratings %>% group_by(Class) %>% summarise( mean = mean(Frequency) )

# code block b
ratings %>% summarise(
    mean = mean(Frequency), .by = c(Class) )
```

```
# code block c
ratings %>% mean(Frequency) %>% group_by(Class)
```

- (A) a only
- (B) b only
- (C) c only
- (D) a and b
- (E) b and c
- (F) a, b, and c