Prep2S24

BilalTariq

2024-02-08

Reminder: Prep assignments are to be completed individually. Upload a final copy of the .Qmd and renamed .pdf to your private repo, and submit the renamed pdf to Gradescope before Sunday, Feb. 11th at midnight (11:59 pm is what Gradescope shows).

Reading

The associated reading for the week is Chapter 4, Chapter 5, Chapter 6 (skip 6.4) and Sections 8.3 and 8.4. This reading explores major functions in wrangling data, including reshaping data. There are many commands here to learn about - do your best to develop a sense of what they each do, and we will build on that by using them for the rest of the semester. You don't need to memorize them all.

Remember, I recommend you code along with the book examples. You can try out the code yourself - just be sure to load the mdsr package and any other packages referenced. You can get the code in R script files (basically, files of just R code, not like a .Rmd or .Qmd) from the book website.

1 - Some basics

Many different data wrangling commands are covered in these chapters. Identify the command you'd use for each of the operations below.

> part a - Add the average of 3 variables to the data set as a new variable.

Solution: Use mutate()

> part b - Keep only 4 columns of a data frame in a new data set.

Solution: We would use the select() function.

> part c - Choose observations that match a particular category of a categorical variable to keep in a new data set.

Solution: group_by() and summarize()

> part d - Combine two data sets based on common variables where all rows from the first are returned, along with any matches in the second.

Solution: Not sure about this

2 - NYC Flights

In Section 5.1, the `flights` and `airlines` tables within the `nycflights13` package are joined together.

> part a - Recreate the `flights_joined` dataset from Section 5.1, being sure to *glimpse* the data in the Console (or via the code chunk) to verify the join worked.

Solution:

```
flights_joined <- flights %>%
  inner_join(airlines, by = c("carrier" = "carrier"))
glimpse(flights_joined)
```

```
Rows: 336,776
Columns: 20
$ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2~
$ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
$ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
$ dep_time       <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 558, ~
$ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 600, 600, ~
$ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2, -1~
$ arr_time       <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 753, 849,~
$ sched_arr_time <int> 819, 830, 850, 1022, 837, 728, 854, 723, 846, 745, 851,~
$ arr_delay      <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7, -1~
$ carrier        <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6", "~
$ flight         <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301, 4~
$ tailnum        <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N394~
$ origin         <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LGA",~
$ dest           <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IAD",~
$ air_time       <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138, 149, 1~
$ distance       <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 944, 733, ~
$ hour           <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6, 6~
$ minute         <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, 0, 59, 0~
$ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0~
$ name           <chr> "United Air Lines Inc.", "United Air Lines Inc.", "Amer~
```

> part b - Now, starting from `flights_joined`, create a new dataset `flights_short` that does the following:

- creates a new variable, `distance_km`, which is distance in kilometers (note that 1 mile is about 1.6 kilometers)
- keeps only the variables: `name`, `flight`, `arr_delay`, and `distance_km` and

- keeps only observations where the distance is less than 480 kilometers (300 miles).

Solution:

```
flights_short <- flights_joined %>%
    mutate(distance_km = distance*1.6) %>%
    select(name,flight,arr_delay,distance_km) %>%
    filter(distance_km < 480)

  glimpse(flights_short)
```

```
Rows: 51,287
Columns: 4
$ name        <chr> "ExpressJet Airlines Inc.", "JetBlue Airways", "Southwest ~
$ flight      <int> 5708, 1806, 4646, 4144, 1002, 20, 44, 1172, 1838, 27, 4418~
$ arr_delay   <dbl> -14, -4, -19, 12, -10, -1, 4, -19, -22, -14, -13, 851, -7,~
$ distance_km <dbl> 366.4, 299.2, 296.0, 339.2, 299.2, 422.4, 334.4, 320.0, 29~
```

> part c - Using the functions introduced in Section 4.1.4, compute the number of flights (call this `N`), the average arrival delay (call this `avg_arr_delay`), and the average distance in kilometers (call this `avg_dist_km`) among these flights with distances less than 480 km (i.e. working off of `flights_short`), grouping by the carrier name. Sort the results in descending order based on `avg_arr_delay`. Save the results in a tibble object called `delay_summary`, and display the table.

Solution:

```
delay_summary <- flights_short %>%
  group_by(name) %>%

summarize(
    N = n(),
    avg_arr_delay = mean(arr_delay),
    avg_dist_km = mean(distance_km),
  )%>%
    arrange(desc(avg_arr_delay))

delay_summary
```

```
# A tibble: 11 x 4
   name                          N avg_arr_delay avg_dist_km
   <chr>                     <int>         <dbl>       <dbl>
```

```
 1 SkyWest Airlines Inc.        1              3        366.
 2 American Airlines Inc.    1455             NA        299.
 3 Delta Air Lines Inc.      1214             NA        325.
 4 Endeavor Air Inc.         5779             NA        319.
 5 Envoy Air                 2924             NA        350.
 6 ExpressJet Airlines Inc. 15588             NA        372.
 7 JetBlue Airways          10813             NA        360.
 8 Mesa Airlines Inc.         319             NA        361.
 9 Southwest Airlines Co.     208             NA        272.
10 US Airways Inc.           9633             NA        309.
11 United Air Lines Inc.     3353             NA        320.
```

part d - Rename the four columns in the `delay_summary` data table to `Airline`, `"Total flights under 480 km"`, `"Average arrival delay (mins)"` and `"Average distance (km)"`, respectively, then use `kable(booktabs = TRUE, digits = 0)` to make the final table output in the pdf close to publication quality.

Solution:

```
Airline <- delay_summary %>%
  rename(
        Airline = name,
        "Total flights under 480 km" = N,
        "Average arrival delay (mins)" = avg_arr_delay,
        "Average distance (km)" = avg_dist_km
    )

kable(delay_summary, booktabs = TRUE, digits = 1)
```

name	N	avg_arr_delay	avg_dist_km
SkyWest Airlines Inc.	1	3	366.4
American Airlines Inc.	1455	NA	299.2
Delta Air Lines Inc.	1214	NA	324.8
Endeavor Air Inc.	5779	NA	318.7
Envoy Air	2924	NA	350.4
ExpressJet Airlines Inc.	15588	NA	372.3
JetBlue Airways	10813	NA	359.6
Mesa Airlines Inc.	319	NA	361.1
Southwest Airlines Co.	208	NA	272.2
US Airways Inc.	9633	NA	308.6
United Air Lines Inc.	3353	NA	319.7

```
Airline
```

```
# A tibble: 11 x 4
   Airline  Total flights under ~1 Average arrival dela~2 Average distance (km~3
   <chr>                     <int>                  <dbl>                  <dbl>
 1 SkyWest~                      1                      3                   366.
 2 America~                   1455                     NA                   299.
 3 Delta A~                   1214                     NA                   325.
 4 Endeavo~                   5779                     NA                   319.
 5 Envoy A~                   2924                     NA                   350.
 6 Express~                  15588                     NA                   372.
 7 JetBlue~                  10813                     NA                   360.
 8 Mesa Ai~                    319                     NA                   361.
 9 Southwe~                    208                     NA                   272.
10 US Airw~                   9633                     NA                   309.
11 United ~                   3353                     NA                   320.
# i abbreviated names: 1: `Total flights under 480 km`,
#   2: `Average arrival delay (mins)`, 3: `Average distance (km)`
```

3 - Baby names - Variant of 6.2.5 example

> part a - Working with the `babynames` data in the **babynames** package, create a dataset `recent_names` that only includes years 2003 to 2017 (giving us the most recent 15 years of data).

Solution:

```
recent_names <- babynames %>%
 filter(year >= 2003 & year <= 2017)
glimpse(recent_names)
```

```
Rows: 501,324
Columns: 5
$ year <dbl> 2003, 2003, 2003, 2003, 2003, 2003, 2003, 2003, 2003, 2003, 2003,~
$ sex  <chr> "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", ~
$ name <chr> "Emily", "Emma", "Madison", "Hannah", "Olivia", "Abigail", "Alexi~
$ n    <int> 25688, 22704, 20197, 17634, 16146, 15925, 14866, 14513, 14102, 13~
$ prop <dbl> 0.01280938, 0.01132140, 0.01007128, 0.00879323, 0.00805124, 0.007~
```

> part b - Following the code presented in Section 6.2.5, create a dataset called `recentnames_summary` that summarizes the total number of people in recent history (years 2003 to 2017) with each name, grouped by sex.

Solution:

```
recentnames_summary <- recent_names %>%

  group_by(name, sex) %>%
  summarize(total = sum(n))
```

```
`summarise()` has grouped output by 'name'. You can override using the
`.groups` argument.
```

> part c - Now, following the third and fourth code chunks presented in Section 6.2.5, reshape or *pivot* the summary data from *long* format to *wide* format. Only keep observations where more than 8,000 babies have been named in each sex (`M` and `F`), and find the smaller of the two ratios `M / F` and `F / M` to identify the top three sex-balanced names (and only the top three!). Save the wide data as `recentnames_balanced_wide`. Display the table.

Solution:

```
recentnames_summary %>%
   pivot_wider(
    names_from = sex,
    values_from = total
  )
```

```
# A tibble: 63,515 x 3
# Groups:   name [63,515]
   name          M     F
   <chr>     <int> <int>
 1 Aaban       107    NA
 2 Aabha        NA    35
 3 Aabid        10    NA
 4 Aabir         5    NA
 5 Aabriella    NA    32
 6 Aada         NA     5
 7 Aadam       185    NA
 8 Aadan       130    NA
 9 Aadarsh     177    NA
10 Aaden      4633     5
# i 63,505 more rows
```

part d - Finally, use `pivot_longer()` to put the dataset back into *long* form. Call this dataset `recentnames_balanced` and display the table.

Solution:

4 - Ethics

Each subsection of Section 8.4 discusses an ethical scenario and ends with one or more questions. Consider the subsection 8.4.6 on "Reproducible spreadsheet analysis".

Write two or three sentences reflecting on how using RMarkdown would help avoid some of the issues described in this scenario, or at least make them easier to spot.

Solution: Since Rmarkdown is code based rather than a visual spreadsheet, to spot ethical errors would be easier as the code is more concise and easier to follow. This makes it easier to then spot any biases or cherry picking of data.