Workbook #3 - Data Transformation

Aldrich Wang

November 5, 2022

Contents

Introduction 1
Penguins 2
Session Info 4

Introduction

Congratulations on completing our first R workbook exercise! Hopefully, you guys enjoyed the process of coding, googling, reaching out for help, debugging, googling, reaching out for help, debugging, ..., and eventually creating some amazing output that you want. It is always possible to encounter different bugs and issues while programming, so just don't panic at all!

We have spent two entire weeks on the introduction of data frames and data visualization in R, so the next step is to learn some data transformation. **Why are we doing data transformation?** For almost all the time, your data frame is not in a usable format and this is your job to structure it into a format that you can make full use of a data frame. In this workbook, we will use another data frame provided by R itself, learn useful functions in data transformation, and create something interesting.

As always, we need to create a chunk of code to load packages first.

```
#install.packages("tidyverse")
#install.packages("palmerpenguins") # this contains "penguins"
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
## -- Conflicts --------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(palmerpenguins)
```

```
## Warning: package 'palmerpenguins' was built under R version 4.1.2
```

Penguins

After importing the data frame, we can view the displayed data frame to get a sense of what this actually looks like.

```
penguins
```

```
## # A tibble: 344 x 8
##    species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##    <fct>   <fct>              <dbl>         <dbl>             <int>       <int>
##  1 Adelie  Torgersen           39.1          18.7               181        3750
##  2 Adelie  Torgersen           39.5          17.4               186        3800
##  3 Adelie  Torgersen           40.3          18                 195        3250
##  4 Adelie  Torgersen           NA            NA                  NA          NA
##  5 Adelie  Torgersen           36.7          19.3               193        3450
##  6 Adelie  Torgersen           39.3          20.6               190        3650
##  7 Adelie  Torgersen           38.9          17.8               181        3625
##  8 Adelie  Torgersen           39.2          19.6               195        4675
##  9 Adelie  Torgersen           34.1          18.1               193        3475
## 10 Adelie  Torgersen           42            20.2               190        4250
## # ... with 334 more rows, and 2 more variables: sex <fct>, year <int>
```

```
unique(penguins$species) # retrieve different species of penguins
```

```
## [1] Adelie    Gentoo    Chinstrap
## Levels: Adelie Chinstrap Gentoo
```

From the above, there are three species of penguins - Adelie, Gentoo, and Chinstrap. *Here is our first question - what is the average body mass of a Gentoo penguin?*

How to approach this problem? Two essential steps that we are going to consider: (1) select the rows that are observations of Gentoo penguins, and (2) calculate the average body mass of Gentoo penguins. Here is our example code: (Explanations: The first line of code uses the filter function to filter out only Gentoo penguins and assigns the filtered data frame a new name - gentoo_penguins. The second line of code helps calculate the average mass of Gentoo penguins.)

```
gentoo_penguins <- filter(.data = penguins, species == "Gentoo")
summarize(.data = gentoo_penguins, avg_mass = mean(body_mass_g, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   avg_mass
##      <dbl>
## 1    5076.
```

```
gentoo_penguins
```

```
## # A tibble: 124 x 8
##    species island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##    <fct>   <fct>           <dbl>         <dbl>             <int>       <int>
##  1 Gentoo  Biscoe           46.1          13.2               211        4500
##  2 Gentoo  Biscoe           50            16.3               230        5700
##  3 Gentoo  Biscoe           48.7          14.1               210        4450
##  4 Gentoo  Biscoe           50            15.2               218        5700
##  5 Gentoo  Biscoe           47.6          14.5               215        5400
##  6 Gentoo  Biscoe           46.5          13.5               210        4550
##  7 Gentoo  Biscoe           45.4          14.6               211        4800
##  8 Gentoo  Biscoe           46.7          15.3               219        5200
##  9 Gentoo  Biscoe           43.3          13.4               209        4400
## 10 Gentoo  Biscoe           46.8          15.4               215        5150
## # ... with 114 more rows, and 2 more variables: sex <fct>, year <int>
```

We have utilized two important functions here - **filter** and **summarize**. The filter function helps us to filter out useful observations of the data frame, and the summarize function can summarize the data frame into one neat value/factor.

Here is our first (very simple) exercise. *What is the average body mass of an Adelie penguin?*

Let's level the two previous questions a little. *What is the average body mass of a penguin for each species?* You could definitely use the filter function to obtain the average mass of each of the three species respectively. But this method tends to be extremely tedious if there are thousands of species of penguins, doesn't it? We can use the **group_by** function here when we focus on one variable of the data frame. The sample code is shown below:

```
species_penguins <- group_by(penguins, species)
summarize(.data = species_penguins, avg_mass = mean(body_mass_g, na.rm = TRUE)) # argument "na.rm = TRU
```

```
## # A tibble: 3 x 2
##   species   avg_mass
##   <fct>        <dbl>
## 1 Adelie       3701.
## 2 Chinstrap    3733.
## 3 Gentoo       5076.
```

Great! So here we have accomplished the task of computing the average body mass of each species. Importantly, we have learned several functions in data transformation - filter, summarize, and group_by. Then, the pipe operator, %>%, will be introduced to help us link a sequence of code. Suppose we are asked the same question, but this time we are going to employ the pipe operator. Here is the sample code with piping: (Note: the pipe operator does not change the result that we get by any means, it is just supposed to make the code neater!)

```
avg_mass_penguins <- penguins %>%
  group_by(species) %>%
  summarize(avg_mass = mean(body_mass_g, na.rm = TRUE))

avg_mass_penguins
```

```
## # A tibble: 3 x 2
##   species   avg_mass
##   <fct>        <dbl>
## 1 Adelie       3701.
## 2 Chinstrap    3733.
## 3 Gentoo       5076.
```

Now, the floor is yours! Here is a somehow challenging problem - *can you compute the average bill length for each Gentoo penguin by sex?* It doesn't really matter whether you are going to employ the pipe operator.

Another question! *Can you compute both the average bill length and depth for each Chinstrap penguin by sex?* Try not to print your results in two separate tables, put them in one.

If you have completed both questions above, feel free to play around with this data frame using the following chunk of space. Here is one perspective that you might want to try: can you employ data visualization techniques on this data frame? (Maybe try to plot a boxplot to showcase the distribution of average bill length of Chinstrap penguins.)

Session Info

```
sessioninfo::session_info()
```

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.1.1 (2021-08-10)
##  os       macOS Big Sur 10.16
##  system   x86_64, darwin17.0
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       America/Chicago
##  date     2022-11-05
##  pandoc   2.14.0.3 @ /Applications/RStudio.app/Contents/MacOS/pandoc/ (via rmarkdown)
##
## - Packages -------------------------------------------------------------------
##  package        * version date (UTC) lib source
##  assertthat       0.2.1   2019-03-21 [1] CRAN (R 4.1.0)
##  backports        1.4.1   2021-12-13 [1] CRAN (R 4.1.0)
##  broom            0.7.12  2022-01-28 [1] CRAN (R 4.1.2)
##  cellranger       1.1.0   2016-07-27 [1] CRAN (R 4.1.0)
##  cli              3.4.1   2022-09-23 [1] CRAN (R 4.1.2)
##  colorspace       2.0-3   2022-02-21 [1] CRAN (R 4.1.2)
##  crayon           1.5.1   2022-03-26 [1] CRAN (R 4.1.2)
##  DBI              1.1.2   2021-12-20 [1] CRAN (R 4.1.0)
```

```
##  dbplyr           2.1.1   2021-04-06 [1] CRAN (R 4.1.0)
##  digest           0.6.29  2021-12-01 [1] CRAN (R 4.1.0)
##  dplyr          * 1.0.8   2022-02-08 [1] CRAN (R 4.1.2)
##  ellipsis         0.3.2   2021-04-29 [1] CRAN (R 4.1.0)
##  evaluate         0.15    2022-02-18 [1] CRAN (R 4.1.2)
##  fansi            1.0.3   2022-03-24 [1] CRAN (R 4.1.2)
##  fastmap          1.1.0   2021-01-25 [1] CRAN (R 4.1.0)
##  forcats        * 0.5.1   2021-01-27 [1] CRAN (R 4.1.0)
##  fs               1.5.2   2021-12-08 [1] CRAN (R 4.1.0)
##  gargle           1.2.0   2021-07-02 [1] CRAN (R 4.1.0)
##  generics         0.1.2   2022-01-31 [1] CRAN (R 4.1.2)
##  ggplot2        * 3.3.6   2022-05-03 [1] CRAN (R 4.1.2)
##  glue             1.6.2   2022-02-24 [1] CRAN (R 4.1.2)
##  googledrive      2.0.0   2021-07-08 [1] CRAN (R 4.1.0)
##  googlesheets4    1.0.0   2021-07-21 [1] CRAN (R 4.1.0)
##  gtable           0.3.0   2019-03-25 [1] CRAN (R 4.1.0)
##  haven            2.4.3   2021-08-04 [1] CRAN (R 4.1.0)
##  hms              1.1.1   2021-09-26 [1] CRAN (R 4.1.0)
##  htmltools        0.5.2   2021-08-25 [1] CRAN (R 4.1.0)
##  httr             1.4.2   2020-07-20 [1] CRAN (R 4.1.0)
##  jsonlite         1.8.0   2022-02-22 [1] CRAN (R 4.1.2)
##  knitr            1.38    2022-03-25 [1] CRAN (R 4.1.2)
##  lifecycle        1.0.3   2022-10-07 [1] CRAN (R 4.1.2)
##  lubridate        1.8.0   2021-10-07 [1] CRAN (R 4.1.0)
##  magrittr         2.0.2   2022-01-26 [1] CRAN (R 4.1.2)
##  modelr           0.1.8   2020-05-19 [1] CRAN (R 4.1.0)
##  munsell          0.5.0   2018-06-12 [1] CRAN (R 4.1.0)
##  palmerpenguins * 0.1.1   2022-08-15 [1] CRAN (R 4.1.2)
##  pillar           1.7.0   2022-02-01 [1] CRAN (R 4.1.2)
##  pkgconfig        2.0.3   2019-09-22 [1] CRAN (R 4.1.0)
##  purrr          * 0.3.4   2020-04-17 [1] CRAN (R 4.1.0)
##  R6               2.5.1   2021-08-19 [1] CRAN (R 4.1.0)
##  Rcpp             1.0.8.3 2022-03-17 [1] CRAN (R 4.1.2)
##  readr          * 2.1.2   2022-01-30 [1] CRAN (R 4.1.2)
##  readxl           1.3.1   2019-03-13 [1] CRAN (R 4.1.0)
##  reprex           2.0.1   2021-08-05 [1] CRAN (R 4.1.0)
##  rlang            1.0.6   2022-09-24 [1] CRAN (R 4.1.2)
##  rmarkdown        2.13    2022-03-10 [1] CRAN (R 4.1.2)
##  rstudioapi       0.13    2020-11-12 [1] CRAN (R 4.1.0)
##  rvest            1.0.2   2021-10-16 [1] CRAN (R 4.1.0)
##  scales           1.1.1   2020-05-11 [1] CRAN (R 4.1.0)
##  sessioninfo      1.2.2   2021-12-06 [1] CRAN (R 4.1.0)
##  stringi          1.7.6   2021-11-29 [1] CRAN (R 4.1.0)
##  stringr        * 1.4.0   2019-02-10 [1] CRAN (R 4.1.0)
##  tibble         * 3.1.8   2022-07-22 [1] CRAN (R 4.1.2)
##  tidyr          * 1.2.0   2022-02-01 [1] CRAN (R 4.1.2)
##  tidyselect       1.1.2   2022-02-21 [1] CRAN (R 4.1.2)
##  tidyverse      * 1.3.2   2022-07-18 [1] CRAN (R 4.1.2)
##  tzdb             0.3.0   2022-03-28 [1] CRAN (R 4.1.2)
##  utf8             1.2.2   2021-07-24 [1] CRAN (R 4.1.0)
##  vctrs            0.5.0   2022-10-22 [1] CRAN (R 4.1.2)
##  withr            2.5.0   2022-03-03 [1] CRAN (R 4.1.2)
##  xfun             0.30    2022-03-02 [1] CRAN (R 4.1.2)
##  xml2             1.3.3   2021-11-30 [1] CRAN (R 4.1.0)
```

```
##  yaml             2.3.5   2022-02-21 [1] CRAN (R 4.1.2)
##
##  [1] /Library/Frameworks/R.framework/Versions/4.1/Resources/library
##
## ------------------------------------------------------------------------------
```