Introduction to R and the data.table package for Data Cleaning and Manipulation

2023-12-04

Table of contents

1 Introduction

This document supplements the suggested guides to getting started using R, tidyverse and data.table for data cleaning and manipulation in the RA guide. It can be used as a reference for how we typically use R to clean and manipulate data.

I recommend that you try to run the code in this document in your own R script. Remember to set the working directory.

2 Tasks

2.1 Simulate data

We start by loading the tidyverse package to get access to piping and plotting functions. We then load the data.table package and simulate a dataset with firms observed in a given year. The data includes information on revenue, the number of employees, and whether the firm is located in a city or not. We also introduce some missing values in the revenue variable.

```
# (Install and) load packages
pacman::p_load(
   tidyverse, # for piping and plotting
   data.table, #for data wrangling
   modelsummary # for summary statistics tables
   )

# Simulate some data
set.seed(123) # for reproducibility
n <- 1000 # number of firms
firm_data <- data.table(
  firm_id = 1:n,
  revenue = exp(rnorm(n, mean=8.8, sd=2)), # random revenue that is log-normal distributed
  employees = sample(20:1000, n, replace = TRUE), # random number of employees
  is_city = sample(c(TRUE, FALSE), n, replace = TRUE) # city or non-city
)

# Introduce missing values
firm_data[sample(n, 20), revenue := NA] # randomly assign NAs to revenue

# View the head and tail of the data.table containing the simulated data
firm_data
```

```
     firm_id     revenue employees is_city
  1:       1          NA       244   FALSE
  2:       2   4186.6033       274    TRUE
  3:       3 149853.9794       580    TRUE
  4:       4   7638.9603       965   FALSE
  5:       5   8591.9025       573    TRUE
 ---
996:     996   5541.6613        29   FALSE
997:     997  56445.5691       474   FALSE
998:     998    444.8776       616   FALSE
999:     999   2332.6649       900   FALSE
```

1000: 1000 4030.3909 979 FALSE

2.2 Task: Replace missing values with mean

Objective: Impute missing values with mean

```
# show number of missing values
firm_data[, sum(is.na(revenue))]
```

[1] 20

```
# impute missing revenue with mean from remaining firms (simple imputation)
nonna_mean_rev = mean(firm_data[!is.na(revenue), revenue])
nonna_mean_rev
```

[1] 48666.74

```
firm_data[is.na(revenue), revenue := nonna_mean_rev] # impute missing revenue with mean

# show that update worked - no more missing values
firm_data[, sum(is.na(revenue))]
```

[1] 0

2.3 Task: Filtering Data

Objective: Filter firms based on certain criteria.

```
# Filtering firms with more than 50 employees
large_firms <- firm_data[employees > 50]

# Removing firms with anomalously high revenue (potential outliers)
firm_data <- firm_data[revenue < 200000]
```

2.4 Task: Advanced Filtering

Objective: Use complex conditions to filter data.

```
# Firms with revenue greater than the median and located in a city
high_revenue_city_firms <- firm_data[revenue > median(revenue) & is_city]
high_revenue_city_firms
```

```
   firm_id    revenue employees is_city
1:       3 149853.979       580    TRUE
2:       5   8591.903       573    TRUE
3:       7  16677.777       670    TRUE
4:      12  13624.537       132    TRUE
5:      17  17956.383       242    TRUE
```

```
 ---
231:    978   7783.393       65    TRUE
232:    980   8891.352      182    TRUE
233:    988  32024.843      283    TRUE
234:    991  25630.154       87    TRUE
235:    993  16467.691      830    TRUE
```

2.5 Task: Create new variables showing size and median revenue within size-group

```
# Create a new indicator variable for large firms (empl > 250)
firm_data[, large := fifelse(employees > 250, "Large", "Small")]

# Create a variable with median revenue by firm-size group
firm_data[, median_revenue := median(revenue), by = large]

firm_data
```

```
     firm_id     revenue employees is_city large median_revenue
  1:       1  48666.7396       244   FALSE Small       8762.142
  2:       2   4186.6033       274    TRUE Large       5752.466
  3:       3 149853.9794       580    TRUE Large       5752.466
  4:       4   7638.9603       965   FALSE Large       5752.466
  5:       5   8591.9025       573    TRUE Large       5752.466
 ---
948:     996   5541.6613        29   FALSE Small       8762.142
949:     997  56445.5691       474   FALSE Large       5752.466
950:     998    444.8776       616   FALSE Large       5752.466
951:     999   2332.6649       900   FALSE Large       5752.466
952:    1000   4030.3909       979   FALSE Large       5752.466
```

2.6 Task: Aggregation and Grouped Operations

Objective: Calculate summary statistics for different groups.

```
# Average revenue for firms in the city vs. outside the city
avg_revenue_by_location <- firm_data[, .(average_revenue = mean(revenue)), by = .(is_city)]
avg_revenue_by_location
```

```
   is_city average_revenue
1:   FALSE       20657.96
2:    TRUE       19286.00
```

```
# Counting the number of firms in each category
firm_count_by_location <- firm_data[, .N, by = .(is_city)]
firm_count_by_location
```

```
   is_city   N
1:   FALSE 487
2:    TRUE 465
```

2.7 Task: Create crosstables

Objective: Create a contingency table showing the number of large/small firms by city/not city.

```
## simple table
firm_data[, table(large, is_city)]
```

```
       is_city
large   FALSE TRUE
  Large   376  351
  Small   111  114
```

```
## datasummary table
datasummary_crosstab(
   large ~ is_city,
   data = firm_data,
   output = "markdown"
   )
```

large		FALSE	TRUE	All
Large	N	376	351	727
	% row	51.7	48.3	100.0
Small	N	111	114	225
	% row	49.3	50.7	100.0
All	N	487	465	952
	% row	51.2	48.8	100.0

```
## Adding another variable
datasummary_crosstab(
   large * is_city ~ median_revenue,
   data = firm_data,
   output = "markdown"
)
```

large	is_city		5752.46551871682	8762.14234816756	All
Large	FALSE	N	376	0	376
		% row	100.0	0.0	100.0
	TRUE	N	351	0	351
		% row	100.0	0.0	100.0
Small	FALSE	N	0	111	111
		% row	0.0	100.0	100.0
	TRUE	N	0	114	114
		% row	0.0	100.0	100.0
	All	N	727	225	952
		% row	76.4	23.6	100.0

2.8 Task: Create correlation matrix for revenue and employees

Objective: Create a correlation matrix for revenue and employees.

```
# simple
cor(firm_data[, .(revenue, employees)], use = "complete.obs")
```

```
            revenue   employees
revenue    1.00000000 -0.06038354
employees -0.06038354  1.00000000
```

```
# prettier and possible to save to latex table
datasummary_correlation(
  data = firm_data[, .(revenue, employees)],
  format = "markdown"
  )
```

	revenue	employees
revenue	1	.
employees	−0.06	1

```
# Redo for all numeric variables
datasummary_correlation(
  data = firm_data,
  format = "markdown"
  )
```

	firm_id	revenue	employees	median_revenue
firm_id	1	.	.	.
revenue	0.01	1	.	.
employees	−0.03	−0.06	1	.
median_revenue	0.03	0.07	−0.75	1

2.9 Task: Create summary statistics table with modelsummary (by city/not city)

Objective: Create a summary statistics table with modelsummary (by city/not city).

```
pacman::p_load(modelsummary)

## markdown format
datasummary(
  revenue + employees ~ Mean + SD + Median + NUnique,
  firm_data,
  group = "is_city",
  output = "markdown",
  )
```

	Mean	Std.Dev.	Median	Unique N
Revenue	19 987.84	33 250.57	6267.03	933
Total employees	528.16	288.57	557.00	612

Size		Mean	Std.Dev.	Median	Unique N
Large	Revenue	18 727.53	31 610.04	5752.47	714
	Total employees	648.96	214.24	658.00	464
Small	Revenue	24 060.02	37 862.81	8762.14	220
	Total employees	137.85	66.38	147.00	148

	Mean	SD	Median	NUnique
revenue	19987.84	33250.57	6267.03	933
employees	528.16	288.57	557.00	612

```
## add names to variables
datasummary(
   (`Revenue` = revenue) + (`Total employees` = employees) ~ Mean + (`Std.Dev.`= SD) + Median + (`Unique N`= NUnique),
   firm_data,
   group = "is_city",
   output = "markdown"
   )
```

	Mean	Std.Dev.	Median	Unique N
Revenue	19987.84	33250.57	6267.03	933
Total employees	528.16	288.57	557.00	612

```
## latex format
tab = datasummary(
   (`Revenue` = revenue) + (`Total employees` = employees) ~ Mean + (`Std.Dev.`= SD) + Median + (`Unique N`= NUnique),
   firm_data,
   group = "is_city",
   output = "latex"
   )
tab
```

```
## Summary statistics by firm size group
tab = datasummary(
   (Size = large) * ((`Revenue` = revenue) + (`Total employees` = employees)) ~
      Mean + (`Std.Dev.`= SD) + Median + (`Unique N`= NUnique),
   firm_data,
   group = "is_city",
   output = "latex"
   )
tab
```

Table 7: Summary statistics by firm size group

Size		Mean	Std.Dev.	Median	Unique N
Large	Revenue	18 727.53	31 610.04	5752.47	714
	Total employees	648.96	214.24	658.00	464
Small	Revenue	24 060.02	37 862.81	8762.14	220
	Total employees	137.85	66.38	147.00	148

Note: The table is constructed from the full sample of firms.

Table 8: Summary statistics by firm size group

Size		Mean	Std.Dev.	Median	Unique N
Large	Revenue	18 727.53	31 610.04	5752.47	714
	Total employees	648.96	214.24	658.00	464
Small	Revenue	24 060.02	37 862.81	8762.14	220
	Total employees	137.85	66.38	147.00	148

Note: The table is constructed from the full sample of firms.

```
## add footnote using kableExtra package
pacman::p_load(kableExtra)

tab = datasummary(
   (Size = large) * ((`Revenue` = revenue) + (`Total employees` = employees)) ~
      Mean + (`Std.Dev.`= SD) + Median + (`Unique N`= NUnique),
   firm_data,
   group = "is_city",
   output = "latex",
   title = "Summary statistics by firm size group"
   ) %>%
   footnote(
      threeparttable = T, # add threeparttable environment to make the table footnote look great
      general = "Note: The table is constructed from the full sample of firms.",
      general_title = ""
      )
tab
```

```
## write table to file and include in a markdown (or LaTeX) document
tab %>% writeLines("summary_statistics.tex")
```

2.10 Task: Joining Data

Objective: Merge the firm data with another dataset, such as industry classification.

Note: When data.table is loaded, the merge() function is overwritten. This means that we can use the merge() function from data.table instead of the merge() function from base R. The merge() function from data.table is faster and more flexible than the merge() function from base R.

```
# Simulating an industry classification dataset
industry_data <- data.table(
  firm_id = 1:n,
  industry = sample(c("Tech", "Retail", "Manufacturing"), n, replace = TRUE)
  )

# left join industry data onto firm_data, keeping all rows in firm_data
firm_data_left <- merge(firm_data, industry_data, on = "firm_id", all.x = T)
firm_data_left
```

```
    firm_id     revenue employees is_city large median_revenue      industry
 1:       1  48666.7396       244   FALSE Small       8762.142        Retail
 2:       2   4186.6033       274    TRUE Large       5752.466 Manufacturing
 3:       3 149853.9794       580    TRUE Large       5752.466          Tech
 4:       4   7638.9603       965   FALSE Large       5752.466 Manufacturing
 5:       5   8591.9025       573    TRUE Large       5752.466          Tech
 ---
948:     996   5541.6613        29   FALSE Small       8762.142 Manufacturing
949:     997  56445.5691       474   FALSE Large       5752.466        Retail
950:     998    444.8776       616   FALSE Large       5752.466 Manufacturing
951:     999   2332.6649       900   FALSE Large       5752.466 Manufacturing
952:    1000   4030.3909       979   FALSE Large       5752.466 Manufacturing
```

```
# inner join industry data onto firm_data, keeping all shared rows
firm_data_inner <- merge(firm_data, industry_data, on = "firm_id")
firm_data_inner
```

```
    firm_id     revenue employees is_city large median_revenue      industry
 1:       1  48666.7396       244   FALSE Small       8762.142        Retail
 2:       2   4186.6033       274    TRUE Large       5752.466 Manufacturing
 3:       3 149853.9794       580    TRUE Large       5752.466          Tech
 4:       4   7638.9603       965   FALSE Large       5752.466 Manufacturing
 5:       5   8591.9025       573    TRUE Large       5752.466          Tech
 ---
948:     996   5541.6613        29   FALSE Small       8762.142 Manufacturing
949:     997  56445.5691       474   FALSE Large       5752.466        Retail
950:     998    444.8776       616   FALSE Large       5752.466 Manufacturing
951:     999   2332.6649       900   FALSE Large       5752.466 Manufacturing
952:    1000   4030.3909       979   FALSE Large       5752.466 Manufacturing
```

```
# right join firm_data onto industry data, keeping all rows in firm_data
firm_data_right <- merge(firm_data, industry_data, on = "firm_id", all.y = T)
firm_data_right
```

```
      firm_id     revenue employees is_city large median_revenue      industry
   1:       1  48666.7396       244   FALSE Small       8762.142        Retail
   2:       2   4186.6033       274    TRUE Large       5752.466 Manufacturing
   3:       3 149853.9794       580    TRUE Large       5752.466          Tech
   4:       4   7638.9603       965   FALSE Large       5752.466 Manufacturing
   5:       5   8591.9025       573    TRUE Large       5752.466          Tech
  ---
 996:     996   5541.6613        29   FALSE Small       8762.142 Manufacturing
 997:     997  56445.5691       474   FALSE Large       5752.466        Retail
 998:     998    444.8776       616   FALSE Large       5752.466 Manufacturing
 999:     999   2332.6649       900   FALSE Large       5752.466 Manufacturing
1000:    1000   4030.3909       979   FALSE Large       5752.466 Manufacturing
```

```r
firm_data = merge(firm_data, industry_data, on = "firm_id", all.x = T)
```

2.11 Task: Data Export

Objective: Export the cleaned and manipulated data to a CSV and a parquet file.

```r
# Writing the final data to a CSV and parquet file
library(rio) # package to import() and export() almost any filetype
export(firm_data, file = "cleaned_firm_data.csv")
export(firm_data, file = "cleaned_firm_data.parquet")
```