Tips for Handling Missing Values

Tips from R 4 Data Science, 2nd ed.

Alier Reng

Saturday, October 21, 2023

Table of contents

1 Using `coalesce()` from `dplyr` to replace `NAs` with 0

Some times missing values represent some fixed and known value, most commonly 0. 19.2.2 Fixed values

```
library(tidyverse)
# R 4 Data Science (2ed)
# 19.2.2 Fixed values - using coalesce() from dplyr to replace NAs with 0
student_grades <-

  tribble(
  ~student, ~english, ~math, ~kiswahili,
  "deng", 100, 67,78,
  "kuol", 94, 99, NA,
  "Ojuok", NA, 89, 75,
  "Gatwich", 57, 100, NA,
  "Ujang", 98, NA, 88
  )

# Replace NAs with 0
student_final_grades <-
  mutate(
    student_grades,
    # Use anonimous or lambda function
    across(where(is.numeric), \(x) coalesce(x, 0))
  )

# Display student final grades
knitr::kable(student_final_grades)
```

student	english	math	kiswahili
deng	100	67	78
kuol	94	99	0
Ojuok	0	89	75
Gatwich	57	100	0
Ujang	98	0	88

2 Extending our Missing Values Example

```r
# Compute totals
final_grades_with_totals <-

  student_final_grades |>
  rowwise() |>
  mutate(
    total = sum(c_across(where(is.numeric))),
    letter_grade = case_when(
      total / 3 >= 90 ~ "A",
      total / 3 >= 80 ~ "B",
      total / 3 >= 70 ~ "C",
      total / 3 >= 60 ~ "D",
      TRUE ~ "F"
    )
  )

# Display student final grades
knitr::kable(
  final_grades_with_totals, align = "c"
)
```

student	english	math	kiswahili	total	letter_grade
deng	100	67	78	245	B
kuol	94	99	0	193	D
Ojuok	0	89	75	164	F
Gatwich	57	100	0	157	F
Ujang	98	0	88	186	D

3 Using `complete()` & `fill()` Functions from `tidyr`

Here we are going demonstrate how to use `complete()` and `fill` functions from `tidyr`.

```
# Load data
salary <- vroom::vroom("data/salary_data.csv", show_col_types = FALSE)

# Inspect the first 7 rows
salary |> slice_head(n = 7)
```

```
# A tibble: 7 x 4
  Name    Year Month Salary
  <chr>  <dbl> <dbl> <chr>
1 Garang  2023     1 $1,500.00
2 <NA>    2023     2 $1,800.00
3 <NA>    2023     3 $1,000.00
4 <NA>    2023     4 $1,200.00
5 <NA>    2023     5 $1,600.00
6 <NA>    2023     6 $1,550.00
7 <NA>    2023     7 $1,040.00
```

Based on the output provided above, it's evident that the 'Name' column contains missing values (NAs). To address this issue, we can use the 'fill()' function from tidyr. The 'fill()' function offers a '.direction' option that can be set to 'down', 'up', 'downup', or 'updown' to handle missing values appropriately.

Furthermore, it's worth noting that some employees did receive payments for all 12 months in 2023. To account for this, we will utilize the 'complete()' function to fill in the missing months for those employees who did receive their paychecks throughout the year.

```
salary_tbl <- salary |>

  # Transform column names
  janitor::clean_names() |>

  # Fill the missing values downward
  fill(name, .direction = "down") |>

  # Add the missing months
  complete(name, year, month)

# View updated salary data
```

```
glimpse(salary_tbl)
```

```
Rows: 36
Columns: 4
$ name   <chr> "Atem", "Atem", "Atem", "Atem", "Atem", "Atem", "Atem", "Atem",~
$ year   <dbl> 2023, 2023, 2023, 2023, 2023, 2023, 2023, 2023, 2023, 2023, 202~
$ month  <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8, ~
$ salary <chr> "$2,500.00", NA, "$4,500.00", "$1,500.00", "$1,800.00", NA, "$2~
```

We now observe a total of 36 rows in the dataset, with each of these rows corresponding to the 12 months for every employee.

3.1 Bonus

Now, let's learn how to remove the dollar symbol '$' from the salary column using `parse_number` from `dplyr`.

```
# Our cleaned salary column
clean_salary <-

  mutate(
    salary_tbl,
    salary = parse_number(salary)
  ) |>

  # spread the data
  pivot_wider(
    names_from = month,
    values_from = salary
  )

# Display cleaned salary data
knitr::kable(
  clean_salary, align = "c"
)
```

name	year	1	2	3	4	5	6	7	8	9	10	11	12
Atem	2023	2500	NA	4500	1500	1800	NA	2500	1900	2000	1200	1380	3500
Deng	2023	6500	NA	1500	4000	2500	1259	600	1800	1200	1678	6000	NA

name	year	1	2	3	4	5	6	7	8	9	10	11	12
Garang	2023	1500	1800	1000	1200	1600	1550	1040	1160	1750	2000	3600	800

Happy Coding!