Tidyverse Practice: Data Preprocessing

Jongrak
2022-07-26

(Quiz by Kyusun 2022-07-24)

Load the package

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✓ ggplot2 3.3.6     ✓ purrr   0.3.4
## ✓ tibble  3.1.7     ✓ dplyr   1.0.9
## ✓ tidyr   1.2.0     ✓ stringr 1.4.0
## ✓ readr   2.1.2     ✓ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

Load the data

```
setwd("~/Library/Mobile Documents/com~apple~CloudDocs/Study/Data Science/2022-S/0722 Quiz")
reserve <- as_tibble(read_csv("reserve.csv"))
```

```
## Rows: 4030 Columns: 9
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): reserve_id, hotel_id, customer_id
## dbl  (2): people_num, total_price
## dttm (1): reserve_datetime
## date (2): checkin_date, checkout_date
## time (1): checkin_time
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Q1: Data overview - How many rows the data have?

```
#It has 4030 rows

reserve
```

```
## # A tibble: 4,030 × 9
##    reserve_id hotel_id customer_id reserve_datetime    checkin_date checkin_time
##    <chr>      <chr>    <chr>       <dttm>              <date>       <time>
##  1 r1         h_75     c_1         2016-03-06 13:09:42 2016-03-26   10:00
##  2 r2         h_219    c_1         2016-07-16 23:39:55 2016-07-20   11:30
##  3 r3         h_179    c_1         2016-09-24 10:03:17 2016-10-19   09:00
##  4 r4         h_214    c_1         2017-03-08 03:20:10 2017-03-29   11:00
##  5 r5         h_16     c_1         2017-09-05 19:50:37 2017-09-22   10:30
##  6 r6         h_241    c_1         2017-11-27 18:47:05 2017-12-04   12:00
##  7 r7         h_256    c_1         2017-12-29 10:38:36 2018-01-25   10:30
##  8 r8         h_241    c_1         2018-05-26 08:42:51 2018-06-08   10:00
##  9 r9         h_217    c_2         2016-03-05 13:31:06 2016-03-25   09:30
## 10 r10        h_240    c_2         2016-06-25 09:12:22 2016-07-14   11:00
## # … with 4,020 more rows, and 3 more variables: checkout_date <date>,
## #   people_num <dbl>, total_price <dbl>
```

The cheapest hotel

Which hotel is the cheapest one? (by day, by mean)

Q2. Select the columns(variables) we need and make a tibble using 'select()' function. (Assumption: The number of people is ignored. - ignore 'people_num')

```
reserve_tb <- select(reserve, hotel_id, checkin_date, checkout_date, total_price)

reserve_tb
```

```
## # A tibble: 4,030 × 4
##    hotel_id checkin_date checkout_date total_price
##    <chr>    <date>       <date>             <dbl>
##  1 h_75     2016-03-26   2016-03-29         97200
##  2 h_219    2016-07-20   2016-07-21         20600
##  3 h_179    2016-10-19   2016-10-22         33600
##  4 h_214    2017-03-29   2017-03-30        194400
##  5 h_16     2017-09-22   2017-09-23         68100
##  6 h_241    2017-12-04   2017-12-06         36000
##  7 h_256    2018-01-25   2018-01-28        103500
##  8 h_241    2018-06-08   2018-06-09          6000
##  9 h_217    2016-03-25   2016-03-27         68400
## 10 h_240    2016-07-14   2016-07-17        320400
## # … with 4,020 more rows
```

Q3. Find out 'price per day' and add this variable to that tibble using 'mutate()'.

```
reserve_tb2 <- reserve %>%
  mutate(num_date = as.numeric(checkout_date - checkin_date),
         price_per_day = total_price / num_date
                       )

reserve_tb2
```

```
## # A tibble: 4,030 × 11
##    reserve_id hotel_id customer_id reserve_datetime    checkin_date checkin_time
##    <chr>      <chr>    <chr>       <dttm>              <date>       <time>
##  1 r1         h_75     c_1         2016-03-06 13:09:42 2016-03-26   10:00
##  2 r2         h_219    c_1         2016-07-16 23:39:55 2016-07-20   11:30
##  3 r3         h_179    c_1         2016-09-24 10:03:17 2016-10-19   09:00
##  4 r4         h_214    c_1         2017-03-08 03:20:10 2017-03-29   11:00
##  5 r5         h_16     c_1         2017-09-05 19:50:37 2017-09-22   10:30
##  6 r6         h_241    c_1         2017-11-27 18:47:05 2017-12-04   12:00
##  7 r7         h_256    c_1         2017-12-29 10:38:36 2018-01-25   10:30
##  8 r8         h_241    c_1         2018-05-26 08:42:51 2018-06-08   10:00
##  9 r9         h_217    c_2         2016-03-05 13:31:06 2016-03-25   09:30
## 10 r10        h_240    c_2         2016-06-25 09:12:22 2016-07-14   11:00
## # … with 4,020 more rows, and 5 more variables: checkout_date <date>,
## #   people_num <dbl>, total_price <dbl>, num_date <dbl>, price_per_day <dbl>
```

Q4. Find the average price of each hotel and add this variable using 'group_by()' and 'summarize()'.

```
reserve_tb3 <- reserve_tb2 %>%
  group_by(hotel_id) %>%
  summarize(hotel_price_mean = mean(price_per_day, na.rm = TRUE))

reserve_tb3
```

```
## # A tibble: 300 × 2
##    hotel_id hotel_price_mean
##    <chr>               <dbl>
##  1 h_1                67860
##  2 h_10               14933.
##  3 h_100              12960
##  4 h_101              33765.
##  5 h_102              16615.
##  6 h_103              46980
##  7 h_104              84400
##  8 h_105              25407.
##  9 h_106              66600
## 10 h_107              56400
## # … with 290 more rows
```

Q5. Sort the average price of each hotel in ascending order using 'arrange()'. Which hotel is the cheapest one?

```
#The average price of hotel 'h_235' is cheapest among those hotels. it's 8750.

arrange(reserve_tb3, hotel_price_mean)
```

```
## # A tibble: 300 × 2
##    hotel_id hotel_price_mean
##    <chr>               <dbl>
##  1 h_235               8750 
##  2 h_35                9406.
##  3 h_197              10133.
##  4 h_44               10574.
##  5 h_224              10667.
##  6 h_74               10909.
##  7 h_15               11108.
##  8 h_41               11345.
##  9 h_24               11500 
## 10 h_50               11769.
## # … with 290 more rows
```

Q6. This time, considering the number of people ('people_num' variable), reorganize it on a per person. (using pipelines with '%>%')

```
#The average price of hotel 'h_35' is cheapest among those hotels. it's 3500

reserve_tb4 <- reserve %>%
  select(hotel_id, checkin_date, checkout_date, people_num, total_price) %>%
  mutate(num_date = as.numeric(checkout_date - checkin_date)) %>% #1 Day
  mutate(price_per_day = total_price / num_date / people_num) %>% #Price per 1 day an
d 1 person
  group_by(hotel_id) %>%
  summarize(hotel_price_mean = mean(price_per_day, na.rm = TRUE)) %>%
  arrange(hotel_price_mean)

reserve_tb4
```

```
## # A tibble: 300 × 2
##    hotel_id hotel_price_mean
##    <chr>               <dbl>
##  1 h_35                 3500
##  2 h_53                 3700
##  3 h_41                 3900
##  4 h_197                4000
##  5 h_224                4000
##  6 h_235                4200
##  7 h_15                 4300
##  8 h_24                 4600
##  9 h_13                 4700
## 10 h_100                4800
## # … with 290 more rows
```

Sales Comparison

Q. Find the summary of total reserves and sales of each hotel since June 2017.

```
library(lubridate)

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

reserve_tb5 <- reserve %>%
  select(hotel_id, reserve_datetime, checkout_date, checkin_date, total_price) %>%
  filter(reserve_datetime >= "2017-06-01") %>%
  group_by(hotel_id) %>%
  summarize(total_reserves = n(),
            total_sales = sum(total_price, na.rm = TRUE)) %>%
  arrange(total_reserves, total_sales)

reserve_tb5

## # A tibble: 291 × 3
##    hotel_id total_reserves total_sales
##    <chr>             <int>       <dbl>
##  1 h_24                  1        4600
##  2 h_76                  1        9200
##  3 h_229                 1       14800
##  4 h_108                 1       17200
##  5 h_102                 1       18000
##  6 h_265                 1       18600
##  7 h_122                 1       20100
##  8 h_208                 1       26700
##  9 h_18                  1       27800
## 10 h_29                  1       27900
## # … with 281 more rows
```