Cleaning and Exploring data using R

2022-08-09

Introduction

Importing data from outside of `R` using the `read_csv()` function.

Step 1: Load packages

Start by installing your required package and loading`tidyverse`.

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

Once a package is installed, we can load it by running the `library()` function with the package name inside the parentheses:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Step 2: Import data

In the chunk below, use the `read_csv()` function to import data from a .csv in the project folder called "hotel_bookings.csv" and save it as a data frame called `bookings_df`.

```
bookings_df <- read_csv("hotel_bookings.csv")
```

```
## Rows: 119390 Columns: 32
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (13): hotel, arrival_date_month, meal, country, market_segment, distrib...
## dbl  (18): is_canceled, lead_time, arrival_date_year, arrival_date_week_numb...
## date  (1): reservation_status_date
## 
## 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.
```

Step 3: Inspect & clean data

To preview the data use the `head()` function.

```
head(bookings_df)
```

```
## # A tibble: 6 x 32
##   hotel   is_ca~1 lead_~2 arriv~3 arriv~4 arriv~5 arriv~6 stays~7 stays~8 adults
##   <chr>     <dbl>   <dbl>   <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>  <dbl>
## 1 Resort~       0     342    2015 July         27       1       0       0      2
## 2 Resort~       0     737    2015 July         27       1       0       0      2
## 3 Resort~       0       7    2015 July         27       1       0       1      1
## 4 Resort~       0      13    2015 July         27       1       0       1      1
## 5 Resort~       0      14    2015 July         27       1       0       2      2
## 6 Resort~       0      14    2015 July         27       1       0       2      2
## # ... with 22 more variables: children <dbl>, babies <dbl>, meal <chr>,
## #   country <chr>, market_segment <chr>, distribution_channel <chr>,
## #   is_repeated_guest <dbl>, previous_cancellations <dbl>,
## #   previous_bookings_not_canceled <dbl>, reserved_room_type <chr>,
## #   assigned_room_type <chr>, booking_changes <dbl>, deposit_type <chr>,
## #   agent <chr>, company <chr>, days_in_waiting_list <dbl>,
## #   customer_type <chr>, adr <dbl>, required_car_parking_spaces <dbl>, ...
## # i Use `colnames()` to see all variable names
```

To summarize or preview your data frame use the `str()` and function will provide summaries of each column in your data arranged horizontally.

```
str(bookings_df)
```

```
## spec_tbl_df [119,390 x 32] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ hotel                         : chr [1:119390] "Resort Hotel" "Resort Hotel" "Resort Hotel" "Reso
##  $ is_canceled                   : num [1:119390] 0 0 0 0 0 0 0 0 1 1 ...
##  $ lead_time                     : num [1:119390] 342 737 7 13 14 14 0 9 85 75 ...
##  $ arrival_date_year             : num [1:119390] 2015 2015 2015 2015 2015 ...
##  $ arrival_date_month            : chr [1:119390] "July" "July" "July" "July" ...
##  $ arrival_date_week_number      : num [1:119390] 27 27 27 27 27 27 27 27 27 27 ...
##  $ arrival_date_day_of_month     : num [1:119390] 1 1 1 1 1 1 1 1 1 1 ...
##  $ stays_in_weekend_nights       : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ stays_in_week_nights          : num [1:119390] 0 0 1 1 2 2 2 2 3 3 ...
##  $ adults                        : num [1:119390] 2 2 1 1 2 2 2 2 2 2 ...
##  $ children                      : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ babies                        : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ meal                          : chr [1:119390] "BB" "BB" "BB" "BB" ...
##  $ country                       : chr [1:119390] "PRT" "PRT" "GBR" "GBR" ...
##  $ market_segment                : chr [1:119390] "Direct" "Direct" "Direct" "Corporate" ...
##  $ distribution_channel          : chr [1:119390] "Direct" "Direct" "Direct" "Corporate" ...
##  $ is_repeated_guest             : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ previous_cancellations        : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ previous_bookings_not_canceled: num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ reserved_room_type            : chr [1:119390] "C" "C" "A" "A" ...
##  $ assigned_room_type            : chr [1:119390] "C" "C" "C" "A" ...
##  $ booking_changes               : num [1:119390] 3 4 0 0 0 0 0 0 0 0 ...
##  $ deposit_type                  : chr [1:119390] "No Deposit" "No Deposit" "No Deposit" "No Deposit
##  $ agent                         : chr [1:119390] "NULL" "NULL" "NULL" "304" ...
##  $ company                       : chr [1:119390] "NULL" "NULL" "NULL" "NULL" ...
##  $ days_in_waiting_list          : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ customer_type                 : chr [1:119390] "Transient" "Transient" "Transient" "Transient" ..
##  $ adr                           : num [1:119390] 0 0 75 75 98 ...
##  $ required_car_parking_spaces   : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ total_of_special_requests     : num [1:119390] 0 0 0 0 1 1 0 1 1 0 ...
##  $ reservation_status            : chr [1:119390] "Check-Out" "Check-Out" "Check-Out" "Check-Out" ..
##  $ reservation_status_date       : Date[1:119390], format: "2015-07-01" "2015-07-01" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   hotel = col_character(),
##   ..   is_canceled = col_double(),
##   ..   lead_time = col_double(),
##   ..   arrival_date_year = col_double(),
##   ..   arrival_date_month = col_character(),
##   ..   arrival_date_week_number = col_double(),
##   ..   arrival_date_day_of_month = col_double(),
##   ..   stays_in_weekend_nights = col_double(),
##   ..   stays_in_week_nights = col_double(),
##   ..   adults = col_double(),
##   ..   children = col_double(),
##   ..   babies = col_double(),
##   ..   meal = col_character(),
##   ..   country = col_character(),
##   ..   market_segment = col_character(),
##   ..   distribution_channel = col_character(),
##   ..   is_repeated_guest = col_double(),
##   ..   previous_cancellations = col_double(),
##   ..   previous_bookings_not_canceled = col_double(),
##   ..   reserved_room_type = col_character(),
##   ..   assigned_room_type = col_character(),
##   ..   booking_changes = col_double(),
##   ..   deposit_type = col_character(),
##   ..   agent = col_character(),
##   ..   company = col_character(),
##   ..   days_in_waiting_list = col_double(),
##   ..   customer_type = col_character(),
##   ..   adr = col_double(),
##   ..   required_car_parking_spaces = col_double(),
##   ..   total_of_special_requests = col_double(),
##   ..   reservation_status = col_character(),
##   ..   reservation_status_date = col_date(format = "")
##   .. )
##  - attr(*, "problems")=<externalptr>
```

To find out what columns you have in your data frame, run the `colnames()` function.

```
colnames(bookings_df)
```

```
##  [1] "hotel"                          "is_canceled"
##  [3] "lead_time"                      "arrival_date_year"
##  [5] "arrival_date_month"             "arrival_date_week_number"
##  [7] "arrival_date_day_of_month"      "stays_in_weekend_nights"
##  [9] "stays_in_week_nights"           "adults"
## [11] "children"                       "babies"
## [13] "meal"                           "country"
## [15] "market_segment"                 "distribution_channel"
## [17] "is_repeated_guest"              "previous_cancellations"
## [19] "previous_bookings_not_canceled" "reserved_room_type"
## [21] "assigned_room_type"             "booking_changes"
## [23] "deposit_type"                   "agent"
## [25] "company"                        "days_in_waiting_list"
## [27] "customer_type"                  "adr"
## [29] "required_car_parking_spaces"    "total_of_special_requests"
## [31] "reservation_status"             "reservation_status_date"
```

To create another data frame using `bookings_df` that focuses on the average daily rate, which is referred to as `adr` in the data frame, and `adults`.

```
new_df <- select(bookings_df, `adr`, adults)
```

To create new variables in your data frame use the `mutate()` function. This will make changes to the data frame, but not to the original data set you imported. That source data will remain unchanged.

```
mutate(new_df, total = `adr` / adults)
```

```
## # A tibble: 119,390 x 3
##      adr adults total
##    <dbl>  <dbl> <dbl>
##  1     0       2   0
##  2     0       2   0
##  3    75       1  75
##  4    75       1  75
##  5    98       2  49
##  6    98       2  49
##  7   107       2  53.5
##  8   103       2  51.5
##  9    82       2  41
## 10   106.      2  52.8
## # ... with 119,380 more rows
## # i Use `print(n = ...)` to see more rows
```

Load packages

In order to start cleaning data, need to by install the required packages. `skimr`, and `janitor`.

```
install.packages("skimr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("janitor")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

Once a package is installed, you can load it by running the `library()` function with the package name inside the parentheses:

```
library(tidyverse)
library(skimr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

The `skimr` package, which has a number of functions for this purpose. For example, the `skim_without_charts()` function provides a detailed summary of the data.

```
skim_without_charts(bookings_df)
```

Table 1: Data summary

Name	bookings_df
Number of rows	119390
Number of columns	32
Column type frequency:	
character	13
Date	1
numeric	18
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
hotel	0	1	10	12	0	2	0
arrival_date_month	0	1	3	9	0	12	0
meal	0	1	2	9	0	5	0
country	0	1	2	4	0	178	0
market_segment	0	1	6	13	0	8	0
distribution_channel	0	1	3	9	0	5	0
reserved_room_type	0	1	1	1	0	10	0
assigned_room_type	0	1	1	1	0	12	0
deposit_type	0	1	10	10	0	3	0
agent	0	1	1	4	0	334	0
company	0	1	1	4	0	353	0
customer_type	0	1	5	15	0	4	0
reservation_status	0	1	7	9	0	3	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
reservation_status_date	0	1	2014-10-17	2017-09-14	2016-08-07	926

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
is_canceled	0	1	0.37	0.48	0.00	0.00	0.00	1	1
lead_time	0	1	104.01	106.86	0.00	18.00	69.00	160	737
arrival_date_year	0	1	2016.16	0.71	2015.00	2016.00	2016.00	2017	2017
arrival_date_week_number	0	1	27.17	13.61	1.00	16.00	28.00	38	53
arrival_date_day_of_month	0	1	15.80	8.78	1.00	8.00	16.00	23	31
stays_in_weekend_nights	0	1	0.93	1.00	0.00	0.00	1.00	2	19
stays_in_week_nights	0	1	2.50	1.91	0.00	1.00	2.00	3	50

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
adults	0	1	1.86	0.58	0.00	2.00	2.00	2	55
children	4	1	0.10	0.40	0.00	0.00	0.00	0	10
babies	0	1	0.01	0.10	0.00	0.00	0.00	0	10
is_repeated_guest	0	1	0.03	0.18	0.00	0.00	0.00	0	1
previous_cancellations	0	1	0.09	0.84	0.00	0.00	0.00	0	26
previous_bookings_not_canceled	0	1	0.14	1.50	0.00	0.00	0.00	0	72
booking_changes	0	1	0.22	0.65	0.00	0.00	0.00	0	21
days_in_waiting_list	0	1	2.32	17.59	0.00	0.00	0.00	0	391
adr	0	1	101.83	50.54	-6.38	69.29	94.58	126	5400
required_car_parking_spaces	0	1	0.06	0.25	0.00	0.00	0.00	0	8
total_of_special_requests	0	1	0.57	0.79	0.00	0.00	0.00	1	5

Cleaning your data

Now, let's say we are primarily interested in the following variables: 'hotel', 'is_canceled', and 'lead_time'. Create a new data frame with just those columns, calling it `trimmed_df` by adding the variable names to this code chunk:

```
trimmed_df <- bookings_df %>%
  select(hotel, is_canceled, lead_time)
```

Renaming a column

```
trimmed_df %>%
  select(hotel, is_canceled, lead_time) %>%
  rename(hotel_type = hotel)
```

```
## # A tibble: 119,390 x 3
##    hotel_type   is_canceled lead_time
##    <chr>              <dbl>     <dbl>
##  1 Resort Hotel           0       342
##  2 Resort Hotel           0       737
##  3 Resort Hotel           0         7
##  4 Resort Hotel           0        13
##  5 Resort Hotel           0        14
##  6 Resort Hotel           0        14
##  7 Resort Hotel           0         0
##  8 Resort Hotel           0         9
##  9 Resort Hotel           1        85
## 10 Resort Hotel           1        75
## # ... with 119,380 more rows
## # i Use `print(n = ...)` to see more rows
```

To either split or combine data in different columns. Combine the arrival month and year into one column using the unite() function:

```
example_df <- bookings_df %>%
  select(arrival_date_year, arrival_date_month) %>%
  unite(arrival_month_year, c("arrival_date_month", "arrival_date_year"), sep = " ")
```

Another way of doing things

we can also use the`mutate()` function to make changes to our columns. Let's say we wanted to create a new column that summed up all the adults, children, and babies on a reservation for the total number of people.

Modify the code chunk below to create that new column:

```
example_df <- bookings_df %>%
  mutate(guests = adults + children + babies)

head(example_df)
```

```
## # A tibble: 6 x 33
##   hotel   is_ca~1 lead_~2 arriv~3 arriv~4 arriv~5 arriv~6 stays~7 stays~8 adults
##   <chr>     <dbl>   <dbl>   <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>  <dbl>
## 1 Resort~       0     342    2015 July         27       1       0       0      2
## 2 Resort~       0     737    2015 July         27       1       0       0      2
## 3 Resort~       0       7    2015 July         27       1       0       1      1
## 4 Resort~       0      13    2015 July         27       1       0       1      1
## 5 Resort~       0      14    2015 July         27       1       0       2      2
## 6 Resort~       0      14    2015 July         27       1       0       2      2
## # ... with 23 more variables: children <dbl>, babies <dbl>, meal <chr>,
## #   country <chr>, market_segment <chr>, distribution_channel <chr>,
## #   is_repeated_guest <dbl>, previous_cancellations <dbl>,
## #   previous_bookings_not_canceled <dbl>, reserved_room_type <chr>,
## #   assigned_room_type <chr>, booking_changes <dbl>, deposit_type <chr>,
## #   agent <chr>, company <chr>, days_in_waiting_list <dbl>,
## #   customer_type <chr>, adr <dbl>, required_car_parking_spaces <dbl>, ...
## # i Use `colnames()` to see all variable names
```