Procesando Datos

Zabdiel Emilio Moreno Mendoza

2022-08-03

Limpieza de datos

Decidí utilizar R pues me facilita todas las operaciones a realizar como crear columnas pues no debo hacer mes por mes, con R puedo utilizar un script para hacerlo con todos los meses.

Utilice los datos de la hora de comienzo de viaje y de termino de viaje pues a diferencia de otros datos no presenta datos nulos, ademas nos da una idea de la cantidad de tiempo que dura un viaje.

Deje a un lado los datos del nombre de la estación de comienzo y fin, así como los datos de coordenadas geográficas. Solo me quede con:

- Ride_id
- Rideable_type
- Started_at
- Ended_at
- Start_station_id
- End_station_id
- Member_casual

Quedaron 12 conjuntos de datos como este:

```
    julio2021

## # A tibble: 822,410 × 9
##    ride_id       ridea…¹ started_at          ended_at            start…²
end_s…³
##    <chr>         <chr>   <dttm>              <dttm>              <chr>
<chr>
##  1 0A1B623926EF… docked… 2021-07-02 14:44:36 2021-07-02 15:19:58 13001
KA1504…
##  2 B2D5583A5A5E… classi… 2021-07-07 16:57:42 2021-07-07 17:16:09 17660
13432
##  3 6F264597DDBF… classi… 2021-07-25 11:30:55 2021-07-25 11:48:45 SL-012
KA1503…
##  4 379B58EAB20E… classi… 2021-07-08 22:08:30 2021-07-08 22:23:32 17660
13196
##  5 6615C1E4EB08… electr… 2021-07-28 16:08:06 2021-07-28 16:27:09 17660
13197
##  6 62DC2B32872F… electr… 2021-07-29 17:09:08 2021-07-29 17:15:00 17660
```

```
15655
##  7 4BBB6E80E6A2… classi… 2021-07-28 16:51:47 2021-07-28 17:03:45 17660
15655
##  8 22CA03D32C6B… classi… 2021-07-03 12:44:50 2021-07-03 12:52:55 13128
13303
##  9 61F0D07D1EEE… classi… 2021-07-02 18:18:22 2021-07-02 18:38:21 TA1307…
TA1309…
## 10 09B4551386A8… classi… 2021-07-29 21:54:05 2021-07-29 22:07:26 TA1307…
KA1504…
## # … with 822,400 more rows, 3 more variables: member_casual <chr>,
## #   ride_length <time>, day_of_week <chr>, and abbreviated variable names
## #   ¹rideable_type, ²start_station_id, ³end_station_id
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all
variable names
```

Se creo la columna “ride_lenght” que es la diferencia entre el fin del viaje y el comienzo.

```
julio2021 <- julio2021 %>% mutate(ride_length = hms::as_hms(ended_at -
started_at))
julio2021 %>% select(ride_id,ride_length)
```

```
## # A tibble: 822,410 × 2
##    ride_id          ride_length
##    <chr>            <time>
##  1 0A1B623926EF4E16 35'22"
##  2 B2D5583A5A5E76EE 18'27"
##  3 6F264597DDBF427A 17'50"
##  4 379B58EAB20E8AA5 15'02"
##  5 6615C1E4EB08E8FB 19'03"
##  6 62DC2B32872F9BA8 05'52"
##  7 4BBB6E80E6A2A16D 11'58"
##  8 22CA03D32C6BB094 08'05"
##  9 61F0D07D1EEE72EE 19'59"
## 10 09B4551386A8410E 13'21"
## # … with 822,400 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

Para comprobar la integridad de los datos decidí ver si la diferencia entre el comienzo del viaje y el fin del mismo era negativa.

```
filter(julio2021,ride_length<=0) %>% select(ride_id,ride_length)
```

```
## # A tibble: 82 × 2
##    ride_id          ride_length
##    <chr>            <time>
##  1 F0E6092560FF78ED  00'00"
##  2 E6C4F61273BC824D -00'04"
##  3 CCCF93E09080ADA5  00'00"
##  4 18FA8EAE88922DAF -00'05"
##  5 3C634446B054AB44 -00'01"
##  6 A8714397CF341E97  00'00"
```

```
##  7 FA4DAC61BD7C4DC8  00'00"
##  8 D22A7B1DAC07DBEC -00'01"
##  9 B8537F80ADD49E5B  00'00"
## 10 B9E0B20E23B2AB58  00'00"
## # … with 72 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

Me percaté de que sí eran negativas pero por muy poco tiempo, pensé en que probablemente esos viajes no habrían sido mas que personas cambiando la bicicleta de lugar en la misma estación, entonces verifique que la estación de inicio fuera la misma de la estación de termino.

```
filter(julio2021,ride_length<=0 & start_station_id == end_station_id) %>%
select(ride_id,ride_length)
```

```
## # A tibble: 28 × 2
##    ride_id          ride_length
##    <chr>            <time>
##  1 E6C4F61273BC824D -00'04"
##  2 CCCF93E09080ADA5  00'00"
##  3 18FA8EAE88922DAF -00'05"
##  4 3C634446B054AB44 -00'01"
##  5 A8714397CF341E97  00'00"
##  6 FA4DAC61BD7C4DC8  00'00"
##  7 D22A7B1DAC07DBEC -00'01"
##  8 A4F0CEEE2CACE722  00'00"
##  9 CBBDF121760541A8 -00'08"
## 10 9180A7126B192284 -00'04"
## # … with 18 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

Los números no coincidían, pero recordé que había nulos, entonces decidí contar el numero de nulos que cumplían con esas condiciones

```
totales <- filter(julio2021,ride_length<=0) %>%
  count()
nulos <- filter(julio2021,ride_length<=0 & (is.na(start_station_id) |
is.na(end_station_id))) %>%
  count()
noNulos <- filter(julio2021,ride_length<=0 & !(is.na(start_station_id) |
is.na(end_station_id)) & start_station_id == end_station_id) %>%
  count()

totales$n
```

```
## [1] 82
```

```
nulos$n
```

```
## [1] 54
```

```
noNulos$n
```

```
## [1] 28
```

```
noNulos$n + nulos$n == totales$n
```

```
## [1] TRUE
```

Listo, los datos coinciden, entonces son íntegros.

Por ultimo creamos la columna week_day que nos muestra el día de la semana en que se empezó el viaje.

```
julio2021 %>% mutate(day_of_week = wday(started_at,label = TRUE,abbr = FALSE)
) %>% select(ride_id,day_of_week)
```

```
## # A tibble: 822,410 × 2
##    ride_id          day_of_week
##    <chr>            <ord>
##  1 0A1B623926EF4E16 viernes
##  2 B2D5583A5A5E76EE miércoles
##  3 6F264597DDBF427A domingo
##  4 379B58EAB20E8AA5 jueves
##  5 6615C1E4EB08E8FB miércoles
##  6 62DC2B32872F9BA8 jueves
##  7 4BBB6E80E6A2A16D miércoles
##  8 22CA03D32C6BB094 sábado
##  9 61F0D07D1EEE72EE viernes
## 10 09B4551386A8410E jueves
## # … with 822,400 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

El script con los comandos utilizados tiene el nombre de “Procesar.R” y se encuentra en la misma carpeta que este informe.