Data Wrangling With R (Bikeshare Data)

Kristian Murimi

3/21/2022

```
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(janitor)

##
## Attaching package: 'janitor'

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

library(lubridate)

##
## Attaching package: 'lubridate'

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

Import Data

```
jan_2022 <- read_csv("capstone/202201-divvy-tripdata.csv")

## Rows: 103770 Columns: 13

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
```

```
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## 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.
```

```
dec_2021 <- read_csv("capstone/202112-divvy-tripdata.csv")
```

```
## Rows: 247540 Columns: 13

## -- Column specification ---------------------------------------------------
------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## 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.
```

```
nov_2021 <- read_csv("capstone/202111-divvy-tripdata.csv")
```

```
## Rows: 359978 Columns: 13

## -- Column specification ---------------------------------------------------
------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## 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.
```

```
oct_2021 <- read_csv("capstone/202110-divvy-tripdata.csv")
```

```
## Rows: 631226 Columns: 13

## -- Column specification ---------------------------------------------------
------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
```

```
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## 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.

sep_2021 <- read_csv("capstone/202109-divvy-tripdata.csv")

## Rows: 756147 Columns: 13

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## 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.

aug_2021 <- read_csv("capstone/202108-divvy-tripdata.csv")

## Rows: 804352 Columns: 13

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## 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.

jul_2021 <- read_csv("capstone/202107-divvy-tripdata.csv")

## Rows: 822410 Columns: 13

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
```

```
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
jun_2021 <- read_csv("capstone/202106-divvy-tripdata.csv")
```

```
## Rows: 729595 Columns: 13
```

```
## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
may_2021 <- read_csv("capstone/202105-divvy-tripdata.csv")
```

```
## Rows: 531633 Columns: 13
```

```
## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
apr_2021 <- read_csv("capstone/202104-divvy-tripdata.csv")
```

```
## Rows: 337230 Columns: 13
```

```
## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
```

```
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## 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.
```

```
mar_2021 <- read_csv("capstone/202103-divvy-tripdata.csv")
```

```
## Rows: 228496 Columns: 13

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## 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.
```

```
feb_2021 <- read_csv("capstone/202102-divvy-tripdata.csv")
```

```
## Rows: 49622 Columns: 13

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at

##
## 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.
```

Merging our data

We will compare our tables to make sure that the column names and coresponding data is the same across all the tables. This is important since the data cannot be merged if there are differences in the data.

```
compare_df_cols_same(jan_2022,dec_2021,nov_2021,oct_2021,sep_2021,aug_2021,
                     jul_2021,jun_2021,may_2021,apr_2021,mar_2021,feb_2021)
```

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

The code gives us a result of TRUE meaning that the data is row-bindable. We can therefore merge the tables into a single dataset.

```
tripdata <-
do.call("rbind",list(jan_2022,dec_2021,nov_2021,oct_2021,sep_2021,

aug_2021,jul_2021,jun_2021,may_2021,apr_2021,mar_2021,feb_2021))
```

Looking at our new table

Let us take a look at our new table to make sure our data is all there and in the correct data type.

```
colnames(tripdata)

##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"

str(tripdata)

## spec_tbl_df [5,601,999 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5601999] "C2F7DD78E82EC875"
"A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED419105406" ...
##  $ rideable_type     : chr [1:5601999] "electric_bike" "electric_bike"
"classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:5601999], format: "2022-01-13 11:59:47"
"2022-01-10 08:41:56" ...
##  $ ended_at          : POSIXct[1:5601999], format: "2022-01-13 12:02:44"
"2022-01-10 08:46:17" ...
##  $ start_station_name: chr [1:5601999] "Glenwood Ave & Touhy Ave"
"Glenwood Ave & Touhy Ave" "Sheffield Ave & Fullerton Ave" "Clark St & Bryn
Mawr Ave" ...
##  $ start_station_id  : chr [1:5601999] "525" "525" "TA1306000016"
"KA1504000151" ...
##  $ end_station_name  : chr [1:5601999] "Clark St & Touhy Ave" "Clark St &
Touhy Ave" "Greenview Ave & Fullerton Ave" "Paulina St & Montrose Ave" ...
##  $ end_station_id    : chr [1:5601999] "RP-007" "RP-007" "TA1307000001"
"TA1309000021" ...
##  $ start_lat         : num [1:5601999] 42 42 41.9 42 41.9 ...
##  $ start_lng         : num [1:5601999] -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num [1:5601999] 42 42 41.9 42 41.9 ...
##  $ end_lng           : num [1:5601999] -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr [1:5601999] "casual" "casual" "member" "casual"
...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
head(tripdata)
```

```
## # A tibble: 6 x 13
##   ride_id rideable_type started_at          ended_at
start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 C2F7DD~ electric_bike 2022-01-13 11:59:47 2022-01-13 12:02:44 Glenwood
Ave & ~
## 2 A6CF89~ electric_bike 2022-01-10 08:41:56 2022-01-10 08:46:17 Glenwood
Ave & ~
## 3 BD0F91~ classic_bike  2022-01-25 04:53:40 2022-01-25 04:58:01 Sheffield
Ave &~
## 4 CBB80E~ classic_bike  2022-01-04 00:18:04 2022-01-04 00:33:00 Clark St &
Bryn~
## 5 DDC963~ classic_bike  2022-01-20 01:31:10 2022-01-20 01:37:12 Michigan
Ave & ~
## 6 A39C6F~ classic_bike  2022-01-11 18:48:09 2022-01-11 18:51:31 Wood St &
Chica~
## # ... with 8 more variables: start_station_id <chr>, end_station_name
<chr>,
## #   end_station_id <chr>, start_lat <dbl>, start_lng <dbl>, end_lat <dbl>,
## #   end_lng <dbl>, member_casual <chr>
```

We can coclude that our data merge was successful and our data is now in a single table making it easier to work with.

Filtering our data

Now that all our data is in one table, we can filter our data to have only the columns we want to use.

```
trip_data <- tripdata[,-c(9,10:12)]
```

The above code has allowed us to get rid of unwanted columns and create a new table with only the columns we want.

```
colnames(trip_data)
```

```
## [1] "ride_id"            "rideable_type"      "started_at"
## [4] "ended_at"           "start_station_name" "start_station_id"
## [7] "end_station_name"   "end_station_id"     "member_casual"
```

Data transformation

Now we can add new columns that can give use more insight into our data from the columns we have.

We can get the duration of a trip using the start and end times from the table.

```
trip_data$trip_duration <- round(difftime(trip_data$ended_at,
                                           trip_data$started_at,units =
'mins'),2)
```

Let us make sure we worked it out correctly.

```
colnames(trip_data)

##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "member_casual"
## [10] "trip_duration"

head(trip_data)

## # A tibble: 6 x 10
##   ride_id rideable_type started_at          ended_at
start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 C2F7DD~ electric_bike 2022-01-13 11:59:47 2022-01-13 12:02:44 Glenwood
Ave & ~
## 2 A6CF89~ electric_bike 2022-01-10 08:41:56 2022-01-10 08:46:17 Glenwood
Ave & ~
## 3 BD0F91~ classic_bike  2022-01-25 04:53:40 2022-01-25 04:58:01 Sheffield
Ave &~
## 4 CBB80E~ classic_bike  2022-01-04 00:18:04 2022-01-04 00:33:00 Clark St &
Bryn~
## 5 DDC963~ classic_bike  2022-01-20 01:31:10 2022-01-20 01:37:12 Michigan
Ave & ~
## 6 A39C6F~ classic_bike  2022-01-11 18:48:09 2022-01-11 18:51:31 Wood St &
Chica~
## # ... with 5 more variables: start_station_id <chr>, end_station_name
<chr>,
## #   end_station_id <chr>, member_casual <chr>, trip_duration <drtn>
```

The new column has been successfully added. Let us check whether our new column has some null values.

```
sum(trip_data$trip_duration < 0)

## [1] 145
```

There are 145 null or invalid values in the trip_duration column. Let us remove them as we cannot get further information about those particular trips.

```
trip_data <- trip_data[!(trip_data$trip_duration < 0),]

sum(trip_data$trip_duration < 0)

## [1] 0
```

Now our column has no invalid or null values.

Let us add a column that gives us the day of the week each trip was taken.

```
trip_data$trip_day <- weekdays(trip_data$started_at)

head(trip_data)

## # A tibble: 6 x 11
##   ride_id rideable_type started_at          ended_at            
start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>          
## 1 C2F7DD~ electric_bike 2022-01-13 11:59:47 2022-01-13 12:02:44 Glenwood
Ave & ~
## 2 A6CF89~ electric_bike 2022-01-10 08:41:56 2022-01-10 08:46:17 Glenwood
Ave & ~
## 3 BD0F91~ classic_bike  2022-01-25 04:53:40 2022-01-25 04:58:01 Sheffield
Ave &~
## 4 CBB80E~ classic_bike  2022-01-04 00:18:04 2022-01-04 00:33:00 Clark St &
Bryn~
## 5 DDC963~ classic_bike  2022-01-20 01:31:10 2022-01-20 01:37:12 Michigan
Ave & ~
## 6 A39C6F~ classic_bike  2022-01-11 18:48:09 2022-01-11 18:51:31 Wood St &
Chica~
## # ... with 6 more variables: start_station_id <chr>, end_station_name
<chr>,
## #   end_station_id <chr>, member_casual <chr>, trip_duration <drtn>,
## #   trip_day <chr>
```

Let us also group the time of day each trip took place under a new column.

```
breaks <- hour(hm("00:00","6:00","12:00","18:00","23:59"))
labels <- c("Night","Morning","Afternoon","Evening")

trip_data$time_of_trip<- cut(x=hour(trip_data$started_at),
                             breaks = breaks, labels = labels,include.lowest
= TRUE)

head(trip_data)

## # A tibble: 6 x 12
##   ride_id rideable_type started_at          ended_at            
start_station_n~
```

```
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 C2F7DD~ electric_bike 2022-01-13 11:59:47 2022-01-13 12:02:44 Glenwood
Ave & ~
## 2 A6CF89~ electric_bike 2022-01-10 08:41:56 2022-01-10 08:46:17 Glenwood
Ave & ~
## 3 BD0F91~ classic_bike  2022-01-25 04:53:40 2022-01-25 04:58:01 Sheffield
Ave &~
## 4 CBB80E~ classic_bike  2022-01-04 00:18:04 2022-01-04 00:33:00 Clark St &
Bryn~
## 5 DDC963~ classic_bike  2022-01-20 01:31:10 2022-01-20 01:37:12 Michigan
Ave & ~
## 6 A39C6F~ classic_bike  2022-01-11 18:48:09 2022-01-11 18:51:31 Wood St &
Chica~
## # ... with 7 more variables: start_station_id <chr>, end_station_name
<chr>,
## #   end_station_id <chr>, member_casual <chr>, trip_duration <drtn>,
## #   trip_day <chr>, time_of_trip <fct>
```

Exporting data for visualization

With those new columns our data is ready for visualization. Let us export the data into a csv file.

```
write.csv(trip_data,"C:\\users\\krism\\documents\\trip_data.csv")
```