Q1 Attempt

yiren

05/10/2021

Previous

```
mytable <- read_csv("data/taxi_trips.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   `Trip ID` = col_character(),
##   `Taxi ID` = col_character(),
##   `Trip Start Timestamp` = col_character(),
##   `Trip End Timestamp` = col_character(),
##   `Payment Type` = col_character(),
##   Company = col_character(),
##   `Pickup Centroid Location` = col_character(),
##   `Dropoff Centroid  Location` = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
#loading useful packages
pacman::p_load(pacman, party, psych, rio, tidyverse)
p_load(janitor)

taxi_trips <- import("data/taxi_trips.csv") %>%
  as_tibble() %>%
  clean_names() %>% #from janitor package to remove space from names
  print()
```

```
## # A tibble: 3,889,032 x 23
##    trip_id   taxi_id   trip_start_time~ trip_end_timest~ trip_seconds trip_miles
##    <chr>     <chr>     <chr>            <chr>                  <int>      <dbl>
##  1 16c7456d~ 88d3be8c~ 01/01/2020 12:0~ 01/01/2020 12:0~          60       0
##  2 472eef1d~ 199fa05b~ 01/01/2020 12:0~ 01/01/2020 12:3~        1740       0
##  3 031a4d88~ aabecb47~ 01/01/2020 12:0~ 01/01/2020 12:1~         720       0.7
##  4 3c416c24~ ba106251~ 01/01/2020 12:0~ 01/01/2020 12:1~         720       0.8
##  5 3c0a2297~ 1f1970d8~ 01/01/2020 12:0~ 01/01/2020 12:0~         556       0.77
##  6 451beabd~ a572ffd4~ 01/01/2020 12:0~ 01/01/2020 12:1~         960       2.4
##  7 60da9b02~ 155ffe17~ 01/01/2020 12:0~ 01/01/2020 12:3~        1861       3.16
##  8 64bd0989~ e2d8418f~ 01/01/2020 12:0~ 01/01/2020 12:3~        1723       0.7
##  9 8e2fa4ad~ 249ef6f7~ 01/01/2020 12:0~ 01/01/2020 01:0~        3745       1.64
## 10 9323dfcf~ d5c4fbae~ 01/01/2020 12:0~ 01/01/2020 12:0~         300       1.1
## # ... with 3,889,022 more rows, and 17 more variables:
## #   pickup_census_tract <int64>, dropoff_census_tract <int64>,
## #   pickup_community_area <int>, dropoff_community_area <int>, fare <dbl>,
```

```
## #   tips <dbl>, tolls <dbl>, extras <dbl>, trip_total <dbl>,
## #   payment_type <chr>, company <chr>, pickup_centroid_latitude <dbl>,
## #   pickup_centroid_longitude <dbl>, pickup_centroid_location <chr>,
## #   dropoff_centroid_latitude <dbl>, dropoff_centroid_longitude <dbl>,
## #   dropoff_centroid_location <chr>
```

```
#Extracting random sample
sample <- taxi_trips %>%
  sample_n(10000)

#
# df1 <- sample %>%
#   mutate(pickup_community_area = as.numeric(pickup_community_area)) %>%
#   select(pickup_community_area, trip_total) %>%
#   mutate(trip_total, scale) %>% #standardize c("y", "z"), ~(scale(.) %>% as.vector)
#   group_by(pickup_community_area) %>%
#   arrange(pickup_community_area) %>%
#   summarise(total = sum(trip_total, na.rm=T)) %>%
#   #hist(df1$pickup_community_area, df1$total)
#   print()
#
# ?mutate_at

# min(df1[1], na.rm = T) # 1
#
# max(df1[1], na.rm = T) # 77
```

Dataset

```
head(sample)
```

```
## # A tibble: 6 x 23
##   trip_id   taxi_id    trip_start_time~ trip_end_timest~ trip_seconds trip_miles
##   <chr>     <chr>      <chr>            <chr>                  <int>      <dbl>
## 1 fbff1649~ f1fd560a9~ 10/14/2020 04:0~ 10/14/2020 04:1~        1121       7.1
## 2 86856071~ 5d7636d4b~ 02/03/2020 06:3~ 02/03/2020 06:4~         443       0.98
## 3 c382d22a~ f14528f8f~ 01/19/2020 07:4~ 01/19/2020 07:4~         357       1.93
## 4 9deb3196~ d794a17f8~ 02/07/2020 05:4~ 02/07/2020 06:0~         720       1.5
## 5 81f982f5~ dafccbebf~ 08/28/2020 08:3~ 08/28/2020 08:3~         600       2.7
## 6 d16cfee4~ e67435947~ 01/28/2020 09:0~ 01/28/2020 09:1~         300       0.5
## # ... with 17 more variables: pickup_census_tract <int64>,
## #   dropoff_census_tract <int64>, pickup_community_area <int>,
## #   dropoff_community_area <int>, fare <dbl>, tips <dbl>, tolls <dbl>,
## #   extras <dbl>, trip_total <dbl>, payment_type <chr>, company <chr>,
## #   pickup_centroid_latitude <dbl>, pickup_centroid_longitude <dbl>,
## #   pickup_centroid_location <chr>, dropoff_centroid_latitude <dbl>,
## #   dropoff_centroid_longitude <dbl>, dropoff_centroid_location <chr>
```

You can also embed plots, for example: ### 1. Direct Calc w/o Standardization

```
## # A tibble: 10,000 x 2
##    pickup_community_area trip_total
##                    <dbl>      <dbl>
##  1                    39      20.8
##  2                     8       9
```

```
##  3                    10       7.5
##  4                    32      10.5
##  5                    32      11.5
##  6                    28       7.5
##  7                    29       7.75
##  8                     6       4.25
##  9                    NA       7.25
## 10                    28      16.5
## # ... with 9,990 more rows
```

It shows community 76 is the one getting most revenue, however, when we look at the area data, community 76 O'Hare is obviously the largest community in Chicago.

2. Considering Area

Area data https://en.wikipedia.org/wiki/Community_areas_in_Chicago#cite_note-CMAP_Area-10

Geospatial Data https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Community-Areas-current-/cauq-8yn6

Data Cleaning

```
comm_area = read_csv("data/mywiki.csv")
```

```
## Parsed with column specification:
## cols(
##   No. = col_character(),
##   Name = col_character(),
##   Population = col_number(),
##   `Area(sqmi)` = col_double(),
##   `Area(km2)` = col_double(),
##   `Density(sqmi)` = col_number(),
##   `Density(km2)` = col_number()
## )
```

```
comm_area_1 = comm_area %>%
  select(No., `Area(km2)`) %>%
  rename(comm = No.) %>%
  mutate(comm = as.numeric(comm))
```

Join the two dataframes by the community code

```
revenue_area = merge(mytrip, comm_area_1)
revenue_area %>% mutate(standard_rev = total/`Area(km2)`) %>% arrange(desc(standard_rev))
```

```
##    comm       total Area(km2) standard_rev
## 1    32 11513000.98      4.27  2696253.157
## 2     8 12273633.21      7.10  1728680.734
## 3    76 15611519.07     34.55   451852.940
## 4    28  5557326.18     14.74   377023.486
## 5    56  3504205.84     10.96   319726.810
## 6    33  1337160.48      4.61   290056.503
## 7     6  1715605.76      8.08   212327.446
## 8    77   795889.60      4.51   176472.195
## 9    41   613786.31      4.17   147190.962
## 10    3   877167.68      6.01   145951.361
## 11    7  1072934.62      8.18   131165.601
## 12   39   292966.33      2.69   108909.416
```

```
## 13   35   429264.67      4.27   100530.368
## 14   36   150142.88      1.50   100095.253
## 15    1   460290.95      4.77    96497.055
## 16   38   385320.19      4.51    85436.849
## 17   43   448157.34      7.59    59045.763
## 18   34   137143.15      2.59    52951.023
## 19   24   618820.36     11.86    52177.096
## 20    2   476250.24      9.14    52106.153
## 21    4   314632.53      6.63    47455.887
## 22   14   235106.75      4.97    47305.181
## 23   37    85470.17      1.84    46451.179
## 24   16   381450.15      8.31    45902.545
## 25   42   232810.37      5.36    43434.771
## 26   44   279163.83      7.64    36539.768
## 27   21   186792.28      5.13    36411.750
## 28   11   210913.85      6.03    34977.421
## 29   22   314626.08      9.30    33830.761
## 30    5   175064.14      5.31    32968.765
## 31   40   103124.04      3.94    26173.614
## 32   45    81462.06      3.24    25142.611
## 33   69   228416.01      9.19    24854.843
## 34   27   121284.40      5.00    24256.880
## 35   48   101700.54      4.53    22450.450
## 36   71   217113.30      9.76    22245.215
## 37   47    33644.37      1.58    21293.905
## 38   68   152046.45      7.95    19125.340
## 39   15   194809.72     10.23    19042.983
## 40   13   122566.96      6.53    18769.825
## 41   73   132079.00      7.41    17824.426
## 42   46   147785.55      8.65    17085.035
## 43   60    91270.01      5.41    16870.612
## 44   49   207163.30     12.48    16599.623
## 45   26    51269.78      3.32    15442.705
## 46   50    83064.07      5.49    15130.067
## 47   59    52613.29      3.65    14414.600
## 48   53   132807.54      9.22    14404.289
## 49   29   114086.97      8.31    13728.877
## 50   20    38975.71      3.03    12863.271
## 51   67   104704.16      8.16    12831.392
## 52   25   237606.97     18.52    12829.750
## 53   23   119088.23      9.32    12777.707
## 54   10   142486.21     11.32    12587.121
## 55   31    94632.12      7.59    12468.000
## 56   75   101110.38      8.55    11825.775
## 57   19   104405.90     10.13    10306.604
## 58   66    92476.60      9.14    10117.790
## 59   12    76526.50      8.29     9231.182
## 60   62    26331.86      3.03     8690.383
## 61   61    98747.91     12.51     7893.518
## 62   18    19353.36      2.56     7559.906
## 63   70    94305.37     12.59     7490.498
## 64   72    54294.92      8.24     6589.189
## 65   17    55169.43      9.63     5728.913
## 66   58    39601.34      7.04     5625.190
```

```
## 67   52    39561.78      7.72     5124.583
## 68   63    29147.46      5.70     5113.589
## 69   30    53967.62     11.89     4538.908
## 70   65    34445.67      7.64     4508.596
## 71   54    39128.99      9.14     4281.071
## 72   57    20836.18      5.21     3999.267
## 73    9    11048.78      2.93     3770.915
## 74   51   104220.71     28.23     3691.842
## 75   64    23076.97      6.60     3496.511
## 76   55    25291.43     13.57     1863.775
## 77   74    10268.65      7.02     1462.771
```

just look at data

```
comm_area %>% arrange(desc(`Density(km2)`))
```

```
## # A tibble: 77 x 7
##    No.   Name            Population `Area(sqmi)` `Area(km2)` `Density(sqmi)`
##    <chr> <chr>                <dbl>        <dbl>       <dbl>           <dbl>
##  1 08    Near North Side     105481         2.74        7.1           38497.
##  2 06    Lake View           103050         3.12        8.08          33029.
##  3 77    Edgewater            56296         1.74        4.51          32354.
##  4 01    Rogers Park          55628         1.84        4.77          30233.
##  5 32    (The)  Loop[11]      42298         1.65        4.27          25635.
##  6 14    Albany Park          48396         1.92        4.97          25206.
##  7 03    Uptown               57182         2.32        6.01          24647.
##  8 07    Lincoln Park         70492         3.16        8.18          22308.
##  9 02    West Ridge           77122         3.53        9.14          21848.
## 10 20    Hermosa              24062         1.17        3.03          20566.
## # ... with 67 more rows, and 1 more variable: Density(km2) <dbl>
```

```
comm_area %>% arrange(desc(Population))
```

```
## # A tibble: 77 x 7
##    No.   Name            Population `Area(sqmi)` `Area(km2)` `Density(sqmi)`
##    <chr> <chr>                <dbl>        <dbl>       <dbl>           <dbl>
##  1 08    Near North Side     105481         2.74        7.1           38497.
##  2 06    Lake View           103050         3.12        8.08          33029.
##  3 25    Austin               96557         7.15       18.5           13504.
##  4 24    West Town            87781         4.58       11.9           19166.
##  5 19    Belmont Cragin       78116         3.91       10.1           19979.
##  6 02    West Ridge           77122         3.53        9.14          21848.
##  7 22    Logan Square         71665         3.59        9.3           19962.
##  8 30    South Lawndale       71399         4.59       11.9           15555.
##  9 07    Lincoln Park         70492         3.16        8.18          22308.
## 10 28    Near West Side       67881         5.69       14.7           11930.
## # ... with 67 more rows, and 1 more variable: Density(km2) <dbl>
```