Data Science Toolkit Case ReadMe

JaysMedina and NathanEstrada

1/20/2022

BarangayDensity.R

BarangayDensity.R processes the data from regionarea.csv and population.csv to derive each barangay's population density. To do this, we first assign the data into tables in R:

```
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
pop = read.csv("population.csv", stringsAsFactors = TRUE)
reg = read.csv("regionarea.csv", stringsAsFactors = TRUE)
```

Due to the regional and municipal factors, we have to set stringAsFactors to TRUE. Since the area data we have is based on regional data and nothing else, we assume that each barangay in a specific region receives equal area allocation (e.g, Barangay 1 Area = Region Area / Number of barangays in Region).

```
#counts frequency of regions in population.csv
freq = count(pop,'Region')

#attaches average land area per barangay to regional data
reg$average = reg$Area/freq$n
```

We use the count() function above to count how many instances of each region was used. Since each row is a specific barangay, the frequency of each Region value is the number of barangays for that region. That number is used to divide the area for each barangay from their respective region.

```
#outerjoin region and pop data
pop = merge(x=pop,y=reg,by=c("Region"), all= TRUE)

#computes for per brgy density
pop$density = pop$Population/pop$average
```

```
write.csv(top_n(pop,5,"density"),'brgypopdensity.csv')
```

After deriving the average area per region, it's most convenient to simply outer join reg and pop and compute each barangay's density within pop. Lastly, we output the top 5 most dense barangays in the country to a csv file named 'brgypopdensity.csv'

CityDensity.R

To compute for each city's population density, we must first aggregate the population for each city:

```
#sums up each city's population
citySum = aggregate(pop$Population,by=list(pop$CityProvince,pop$Region),FUN=sum)
```

As seen above, we cannot simply aggregate the city data due to some cities sharing the same name, despite existing in different regions. As such, we aggregate both by the CityProvince and Region factors.

```
#summarizes and merges duplicate pairs
regCity = pop %>%
  filter(complete.cases(.) & !duplicated(.)) %>%
  group_by(Region, CityProvince) %>%
  summarize(count = n())
```

```
## `summarise()` has grouped output by 'Region'. You can override using the `.groups` argument.
```

To meaningfully aggregate our data, we make a new dataframe called regCity where we pass the summarized and filtered Region to City pairs. If we look into the data, we can see that there's a column called "count", which is simpky the number of barangays for each city. However, that data will not be needed for this analysis.

```
#counts cities per region
cityCount = count(regCity,Region)
```

We also make a temporary holder for the number of cities per region so we can get the average area for each city. This data is then used to compute for the city land as shown below:

```
#computes for average city land per region
reg$cityLand = reg$Area/cityCount$n
```

From here, we simply have to combine the data into one dataframe for ease of processing. Here, we use regCity as our main dataframe as it has discretized each city already.

```
#pass city population info
regCity$cityPop = citySum$x

#outer join for city land
regCity = merge(x=regCity,y=reg,by=c("Region"), all= TRUE)
regCity$density = regCity$cityPop/regCity$cityLand

write.csv(top_n(regCity,5,density),'citypopdensity.csv')
```