Baltimore Mortality Data

Baltimore Open Data

The assignment link is here: https://classroom.github.com/a/6AiA-A2s

Open Baltimore is a resource for open data within Baltimore City. We are looking at the area-specific mortality for different age groups.

For example, the Mortality by Age (15-24 Years old):

> The number of deaths of persons between the ages of 25 and 44 per 10,000 persons within the area in a five year period. Source: Baltimore City Health Department Years Available: 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018

The National Health and Nutrition Examination Survey (NHANES) is a program of studies designed to assess the health and nutritional status of adults and children in the United States. The survey is unique in that it combines interviews and physical examinations. NHANES is a major program of the National Center for Health Statistics (NCHS). NCHS is part of the Centers for Disease Control and Prevention (CDC) and has the responsibility for producing vital and health statistics for the Nation.

Project

We want to demonstrate the skills to download data, and perform an exploratory analysis with some basic reporting. The expected output is a report to be given to a collaborator, but code should be demonstrated when necessary, but **not for all unnecessary code**.

This indicates we should have a statement of the purpose of the analysis/introduction, a description of the data (**not** using names from code, unless relevant), a description of the methods used (including any model formulations if models are used), the results of the analysis. The results section should **reference** figures or tables for the analysis and provide **insight** to what you are seeing. Tell the reader what you are seeing. Then a conclusion/discussion section should provide some distillation of the analysis and what was learned.

The Data

Purpose

We would like to show a few concepts:

1. Downloading and organizing complex data
2. Creating insightful plots and discerning how to create a report
3. Perform an analysis on real-world data

Tasks

1. Go to https://data.baltimorecity.gov/search?q=*. Click the 3 dots/ellipsis, and click `Export public catalog (CSV)`. The file should download `Open Baltimore.csv`. Save this to your data directory. Filter the data for those that have the word "mortality". Subset the columns of the data to show
2. Download the mortality data for all groups available: > Infant Mortality Rate Mortality by Age (1-14 Years old) Mortality by Age (15-24 Years old) Mortality by Age (25-44 Years old) Mortality by Age (45-64 Years old) Mortality by Age (65-84 Years old) Mortality by Age (85 and over)

Use the `url` column of the `Open Baltimore.csv`. If that does not work, you can also click `Full Details` on a data set and then go to `View API Resources`, and `Open in API Explorer` and get the query URL. For example, the URL for the Mortality by Age (85 and over) is:

https://services1.arcgis.com/mVFRs7NF4iFitgbY/arcgis/rest/services/Mort85_/FeatureServer/0/query?where

You must add the `/0/query?where=1%3D1&outFields=*&outSR=4326&f=json` to the URLs.

Download the json files to `data/json` using `curl::curl_download` or other download functions. Name the files based on the age, where it should be `mortality_infant.json`, `mortality_1-14.json`, etc.

Download the data for `Percent of Households with No Vehicles Available`, `Percent of Households with No Internet at Home`, and `Fast Food Outlet Density per 1,000 Residents`.

3. Read the data in using `jsonlite::read_json` or other JSON readers.

Extract the `attributes` element from the `features` element from the result. You will likely need to use `lapply` or `purrr::map` (or variants like `purrr:::map_df`).

For example, the result of `output$features[[1]]$attributes` from output for `25-44` year old groups should be something like

```
$OBJECTID
[1] 1

$CSA2010
[1] "Allendale/Irvington/S. Hilton"

$mort44_11
[1] 41.89017

$mort44_12
[1] 39.84674

$mort44_13
[1] 29.62963

$mort44_14
[1] 35.24904

$mort44_15
[1] 37.80332

$mort44_16
[1] 43.42273

$mort44_17
[1] 45.46616

$mort44_18
[1] 50.57471

$Shape__Area
[1] 63770462

$Shape__Length
[1] 38770.17
```

4. Create a plot for each of the age groups. The x-axis should be the year, and the y-axis should be the mortality rate. The lines and points should be colored by the age group, using a sequential palette. Put the legend *inside* the figure. The plot should be saved to `figures/mortality_over_time_by_age_group.png`.

5. Create distributions for the `Percent of Households with No Vehicles Available`, `Percent of Households with No Internet at Home`, and `Fast Food Outlet`

`Density per 1,000 Residents` (predictors/covariates). Create plots for each predictor versus mortality for all age groups, stratified by year. Discuss any trends you see.

6. Create maps of Baltimore mortality over time. We can create an `sf` feature below (where `res` is the output from `read_json`).

```
library(dplyr)
library(sf)
df = purrr:::map(res$features, function(r) {
  x = do.call(rbind, purrr::map(r$geometry$rings[[1]], unlist)) %>%
    as.data.frame()
  colnames(x) =  c("lon", "lat")
  x
})
out = purrr::map_df(res$features, function(x) {
  x$attributes %>% as.data.frame
})
names(df) = out$CSA2010
df = dplyr::bind_rows(df, .id = "CSA2010")

polygon <- df %>%
  group_by(CSA2010) %>%
  st_as_sf(coords = c("lon", "lat"), crs = 4326) %>%
  summarise(geometry = st_combine(geometry)) %>%
  st_cast("POLYGON")
```

Note, this is just for *one* mortality group. Repeat and combine with other mortality groups.

Task: Plot the mortality over time for all groups - this should be a grid where the rows the mortality groups and the column is the year of the estimates. For example, you can perform this for one column of the data:

```
library(ggplot2)
ggplot() +
  geom_sf(data = polygon, aes(fill = mort44_12)) +
  scale_y_continuous()
```

(Hint: you will need to reshape the data probably to fully do this)

7. (UNGRADED: Extra/fun/learning): Create one additional map using `tmap`. Create a leaflet map to include in `HTML`:

```
library(tmap)
tmap_mode("view")
tm_shape(polygon) + tm_fill("mort44_12", palette = sf.colors(5))
```

Notes

Do not use `setwd` in your scripts.

Make sure the figure aspect ratios are appropriate (get someone else to look it over!).

Any questions on this project should be posted as an issue on https://github.com/jh-adv-data-sci/adv_data_sci_2023/issues.