Georgia Tech Data Science and Analytics Bootcamp:

Repository for project:

The distribution of (auto) insurance companies, by county, in Georgia, USA relative to population size and number of reported car accidents.

Team C3:

Corey Devin Anderson, Erin Zheng, William Won Jung, Chuck Youngman

Languages: Python, R

IDEs: Jupyter Notebook (.ipynb), Base R

Output file formats: .csv, .png

We used Python with jupyter notebook and ‘pandas’ (plus additional modules, such as ‘numpy’, ‘requests’, and ‘json’) to read in and clean data sources acquired from .csv (census and accident). We also developed code for acquiring insurance agency data using Google Maps API, and component code for cleaning the data with pandas. Places searches were done initially for the first page of search results, which has a maximum of 20 records (places_cities_insurance.ipynb). We then developed code to do a recursive search of next_page_tokens for all places (places_cities_token.ipynb file) and then an analogue set of code for searching more efficiently along a systematic grid of points (places_grid_token.ipynb).

We used geopandas to build a GeoDataFrame for the county boundaries and merged it, by county, with the census, accident, and insurance agency data. The hot-spot analysis was done using the R package ‘spdep’, with the help of ‘gridExtra’, ‘maptools’, ‘rgdal’, ‘sp’, ‘spdep’, and ‘spatstat’ for handling and manipulating spatial objects and plots. Required shapefiles for the analysis and plots can be found in the files with the suffix shapefile.

Directory structure for the repository:

/accident_data

/census_data

/city_locations

/county_shapefile

/georgia_shapefile

/geopandas_code

/insurance_data

Project_1_Final.ppt

C3_Written_results_summary.GDOC

Detailed descriptions:

/accident_data

/accident_data/accidents_GA.ipynb

Shows the cleanup process of our accident data with over 4.2 million rows of data.

One major issue with accident data was with different capitalizations of three different counties and two counties being merged into one.

/accident_data/accident_GA_2017_2020_cleaned.csv

.csv file of all the accident data in Georgia from the years 2017 to 2020.

/census_data

/census_data/ga_census_reformatted.csv

.csv file of the census data for each county in Georgia.

/city_locations

/city_location/ga_cities_raw.csv

.csv with names of cities/towns/city-counties in Georgia from US census.

/city_location/ga_cities.csv

Reformatted data (from Excel) of the cities in Georgia to separate the name of the city and state into different columns.

/city_location/ga_citie_geo.csv

Data for the cities in Georgia after geocoding to obtain the latitude and longitude of each city.

/county_shapefile

/47f01f96-97ff-4424-87e1-b4dc85d67d92202043-1-1mic60v.toaw.shp

ESRI shapefile for county boundaries in Georgia (in lat/lon, WGS84).

/ga_shapefile

/ga_shapefile/Georgia_State_Boundary.shp

ESRI shapefile for the boundary of the state of Georgia.

/ga_shapefile/ga_grid.R

Component R code for mapping the search locations and boundaries.

/ga_shapefile/ga_grid.<name>.png

Output maps from R

/geopandas_code

/geopandas_code/county_ins_geopd.ipynb

Jupyter notebook file containing data for code for building the GeoDataFrame and merging the GeoDataFrame with other DataFrames (with data for census size and accidents).

/geopandas_code/all.csv

Output DataFrame from notebook after dropping the geometry column.

/geopandas_code/all_geo.csv

The GeoDataFrame output to .csv

/geopandas_code/<name>.png

Saved maps from /geopandas_code/county_ins_geopd.ipynb

/insurance_data

/insurance_data/Reverse_Geocode_County.ipynb

Reverse geocode coding for the insurance agencies to add a seperate column for their state and county.

(Note that because of unstable connection we got an error while running the code, but the code works fine with stable connection)

/insurance_data/agencies_df_grid.csv

.csv file of the insurance agencies in Georgia with their latitudes and longitudes using the grid method.

/insurance_data/agencies_df_grid_drop.csv

.csv file of the insurance agencies in Georgia with their latitudes and longitudes after dropping all duplicate locations.

/insurance_data/agencies_df_places_4_27_21.csv

.csv file of the insurance agencies in Georgia with their latitudes and longitudes using the Places API method.

/insurance_data/Insurance_agencies_master.csv

.csv file of the cleaned up data from agencies_df_places_4_27_21.csv to drop duplicates and added county and state for each location.

/insurance_data/insurance_agencies_even_cleaner.csv

.csv file by using our Insurance_agencies_master.csv file to clean up the data to remove all non-auto agencies and added an extra column to clean up the names of the agencies.

/insurance_data/insurance_by_county.csv

.csv file of the count of insurance agencies within each county. Separated into columns by the top 10 auto insurance agencies and a column for all auto insurances.

/insurance_data/graphs_matplotlib.ipynb

In this notebook you will be able to find all the plots and graphs of our data. This includes a bar graph, which shows the top 10 brick and mortar insurance agencies, and three scatterplots that show the relationship between each of our datasets.

/insurance_data/hotspot_analysis.R

R-code for hotspot analysis of insurance agency data.

/insurance_data/insurance_glm_preliminary.R

Code for scatter plots, histograms, and Quasi-Poisson GLM

/insurance_data/insurance_places_cleaning6.ipynb

Clean up process of our insurance agency data.

Renamed all agencies with different names from their company to match the name of their company.

Dropped insurance agencies that were not auto-insurance agencies.

Count of the top 10 insurance agencies in every county.

/insurance_data/insurance_cities_token.ipynb

Places API search for each insurance agency using the next page token to obtain additional counts of insurance_agencies.

/insurance_data/places_cities_insurance.ipynb

Places API search for each insurance agency within a 50 km radius of the cities in GA. First page of results only (max of 20).

/insurance_data/sorted_grid_df.csv

.csv file of our counties after reverse geocoding each location to add separate columns for state and county. This was using the data obtained from the grid method.