**PhD Data Analytics and Society**

**GEOG 5995 Programming for Social Sciences: Core Skills**

**Assignment part II**

**Home University ID: 8895377**

**Leeds University ID: 201190826**

**Dataset: Public Health England Fingertips Download – Pregnancy and Birth Profiles, UK. www.phe.org.uk**

Introduction

As part of a broader doctoral research project, I intend to access and use large administrative datasets associated with public health.

The data are large, and stored on a university supercomputer.

As a precursor to dealing with these files, I wish to investigate

Dataset

The script here is a basic start on building the tools I'll need to manipulate large datasets coming from public health authorities in England. The dataset used was downloaded as a .csv file from Public Health England and is a publicly available, non sensitive dataset.

Operations performed

1. Reads in a .csv file, parses and put into a pandas dataframe
2. Selects out two relevant sets of cases by question identifier number into two seperate pandas data frames
3. Selects out relevant boroughs within each of the variable data frames by reference to a list of borough names
4. Drop unnecessary columns from each frame
5. Prints a list of column headers
6. Rename a column in each of the data frames
7. Sets index column for each data frame, and then concatenates two data frames, and copies the index to a new column
8. Produces a scatter plot of data and saves to a .png and a time series trend for each borough as a series of subplots, using the group function, saves plot to a pdf. by calling methods defined in a separate script Methods.py.
9. Calculates mean and standard deviation, and prints these values on screen
10. Appends the two variables back together as a new data frame (one variable above the other)
11. Pivots data for both dataframes to unstack longitudinal data
12. Writes all of the manipulated data frames into a new excel file with each frame as a separate sheet so it can be sent to my supervisor who doesn't use python!

See the pdf document in the repository above for some basic UML to describe the methods defined within the module Methods.py and to define the variables contained within dataframes and how these dataframes inherit variables from each other.

Limitations

Future improvements

Appendix A: Code and Licencing

All code reproduced here is my own work – subject to the qualification that learning has been conducted through the many publicly available questions and answers on StackOverflow.com, and that the libraries Pandas, Numpy, Matplotlib and Seaborn, compiled by other Python users are used.

My code is open source and can be accessed, used and shared freely via a GitHub repository held online at <https://github.com/jendmurphy/GEOG5995_Assignment>

"""

GEOG5995\_201190826.py

"""

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np

import Methods

# Set up a list of all local authorities in the Greater Manchester Health partnership

gm = ['Bolton','Bury','Manchester','Oldham','Rochdale','Salford',

'Stockport','Tameside','Trafford','Wigan']

england = ['England']

# Read csv file into a panda dataframe

childhealthprofiledata = pd.read\_csv('Pregnancyandbirth.data.csv', encoding='cp1252')

# Select maternal smoker status variable into a a dataframe, select lbw into a separate dataframe

smoker = childhealthprofiledata.loc[childhealthprofiledata['Indicator ID']== 20301]

lbw = childhealthprofiledata.loc[childhealthprofiledata['Indicator ID']== 92531]

# Select only those boroughs within GM, referencing the list of GM boroughs.

gmsmoker = smoker.loc[smoker['Area Name'].isin(gm)]

gmlbw = lbw.loc[lbw['Area Name'].isin(gm)]

# Remove unnecessary columns

gmsmoker.drop(['Area Code','Value note','Recent Trend','Lower CI 99.8 limit','Upper CI 99.8 limit','Indicator ID','Parent Code','Parent Name','Area Type','Sex','Age','Category Type', 'Category','Compared to subnational parent value or percentiles'],axis=1, inplace=True)

gmlbw.drop(['Area Code','Value note','Recent Trend','Lower CI 99.8 limit','Upper CI 99.8 limit','Indicator ID','Parent Code','Parent Name','Area Type','Sex','Age','Category Type', 'Category','Compared to subnational parent value or percentiles'],axis=1, inplace=True)

# Two variables appended back together using df.append()

gmsmokerlbw = gmsmoker.append(gmlbw)

# Print a list of column headers

print(gmsmoker.columns.tolist()) # eyeball check of column headers

print(gmlbw.columns.tolist()) # eyeball check of column headers

# Stack of longitudinal data

gmsmokerpivot = gmsmoker.pivot(index = 'Area Name', columns = 'Time period')

gmlbwpivot = gmlbw.pivot(index = 'Area Name', columns = 'Time period')

# Rename "Value" column to a unique value in the two dataframes

gmsmoker.rename(columns = {'Value':'Smokervalue'}, inplace = True)

gmlbw.rename(columns = {'Value':'LBWvalue'}, inplace = True)

# Set index as area name for two dataframes

gmsmoker = gmsmoker.set\_index('Area Name')

gmlbw = gmlbw.set\_index('Area Name')

# Join selected columns from data frames together using concat function to allow plotting.

gmsmokerlbwconcat = pd.concat([gmsmoker['Smokervalue'], gmlbw[['LBWvalue','Time period']]], axis=1)

# Copy index to an additional column

gmsmokerlbwconcat['Area Name'] = gmsmokerlbwconcat.index

# Call plot functions

trendplot(gmsmokerlbwconcat,gmsmokerlbwconcat.index,'Time period','Smokervalue','Year',

'% of maternal smokers','Trend in maternal smoker rates',

'Trend in maternal smoker rates.pdf')

scatterplot("Smokervalue","LBWvalue",gmsmokerlbwconcat,"Area Name", 8,

"Correlation between Smoking and LBW","Correlation.png")

# Calculate mean and standard deviation using numpy

mean\_smk = np.mean(gmsmokerlbwconcat['Smokervalue'])

st\_dev\_smk = np.std(gmsmokerlbwconcat['Smokervalue'])

mean\_lbw = np.mean(gmsmokerlbwconcat['LBWvalue'])

st\_dev\_lbw = np.std(gmsmokerlbwconcat['LBWvalue'])

print(mean\_smk)

print(st\_dev\_smk)

print(mean\_lbw)

print(st\_dev\_lbw)

# Write manipulated data to a new Excel file for my supervisor who doesn't use Python

writer = pd.ExcelWriter('python\_gm\_borough\_output.xlsx')

gmsmokerpivot.to\_excel(writer,'gmsmokerpivot')

gmsmoker.to\_excel(writer,'gmsmoker')

gmlbwpivot.to\_excel(writer,'gmlbwpivot')

gmlbw.to\_excel(writer,'gmlbw')

gmsmokerlbw.to\_excel(writer,'gmsmokerlbw')

gmsmokerlbwconcat.to\_excel(writer,'gmsmokerlbwconcat')

writer.save()

"""

CODE ENDS

"""

"""

Methods.py

"""

import seaborn as sns

import matplotlib.pyplot as plt

def **scatterplot**(x,y,data,hue,size,scattertitle,fname2):

# Plot using Seaborn package, save figure as a pdf.

sns.pairplot(x\_vars=[x], y\_vars=[y], data = data, hue=hue, size=size)

plt.title(scattertitle)

plt.savefig(fname2)

def **trendplot**(data,group1,xvar,yvar,xlab,ylab,plottitle,fname):

# Plotting the trend in maternal smoker rates across different boroughs

fig, ax = plt.subplots()

for title, group in data.groupby(group1):

ax.plot(group[xvar], group[yvar], label = title)

plt.xlabel(xlab)

plt.ylabel(ylab)

plt.title(plottitle)

plt.legend(loc=1)

plt.savefig(fname)

"""

CODE ENDS

"""

Appendix B: Data Variables

The initial dataset contains 23 variables (columns) and 26,648 cases (rows).

|  |  |
| --- | --- |
| Variable Name | Type |
| Indicator ID | str |
| Indicator Name | str |
| Parent Code | str |
| Parent Name | str |
| Area Code | str |
| Area Name | str |
| Area Type | str |
| Sex | str |
| Age | str |
| Category Type | str |
| Category | str |
| Time period | str |
| Value | float |
| Lower CI 95.0 limit | float |
| Upper CI 95.0 limit | float |
| Lower CI 99.8 limit | float |
| Upper CI 99.8 limit | float |
| Count | int |
| Denominator | int |
| Value note | str |
| Recent Trend | str |
| Compared to England value or percentiles | str |
| Compared to subnational parent value or percentiles | str |