**Homework 1: Structured Data  
Alex Hyman**

**9 August 2018**

**Introduction**

Syracuse University has been collecting data on donors to the school and needs some analysis conducted on it. This report will analyze two questions:

* How is the average donation of an individual distributed broken up by zip code and the gender?
* How does homeownership or income level have a significant effect on the lifetime donation amount of an individual to the school.

Each of these questions will be accompanied with a csv file consisting of statistics for the different groups, as well as a boxplot to provide a visual representation of the distributions for each of the specified groups. The data will be represented in python by a pandas data frame, and calculations will be conducted on the indexed groups using the .loc[] method.

Required libraries for this analysis include:

* csv – for writing and reading csv files
* pandas – for converting csv files into a data frame
* matplotlib.pyplot – for creating and saving figures
* seaborn – for visually pleasing figures
* numpy – for statistic functions not provided in the base python functions

**Part 1: Average Donation by Zip Code and Gender**

The importance of looking at the average donation by zip code and by gender is that Syracuse will be able to target alumni for donations more effectively saving both time and resources. The code provided in the appendix will use the following steps to provide meaningful analysis:

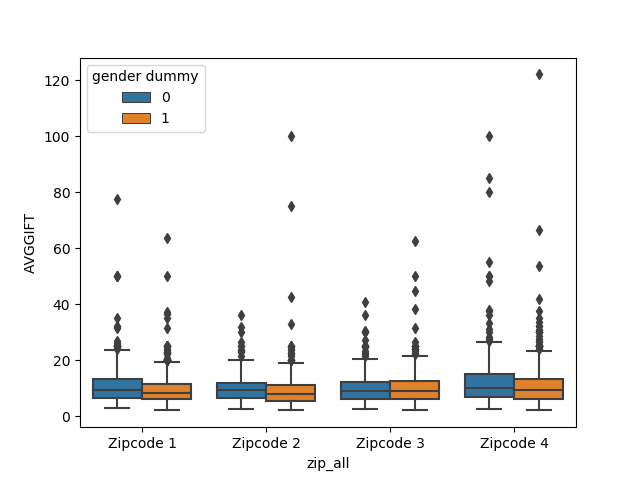
1. Read the Donors\_Data.csv file into a pandas data frame
2. Create a new column that states which zip code the donor lives in by indexing the rows that have a 1 in the column for the provided zipconvert columns
3. Create a dictionary to maintain the data that is being collected
4. Loop through each of the zip codes an genders and compute the mean AVGGIFT for those donors and save it in the dictionary using the keys zip code -> gender
5. Create a csv file called genderZipGift.csv that contains a pivot table for the mean average gift for the given zip code and gender of the donor by looping through the saved dictionary
6. Create a boxplot saved as ZipBoxplot.png that has the zip code on the x-axis and the AVGGIFT on the y-axis, using the colors to separate by gender

The csv file will look as such:



The zip code of the donor is provided by the columns and the gender is provided by the rows. Matching the column with the row will provide you the mean AVGGIFT for donors in that given zip code with the gender specified. Using this table, you can see that men in zip code 4 on average donate more money in a single instance than any other gender/zip code grouping. This table also shows that women in zip code 2 on average donate the least amount of money in a single donation instance than any other gender/zip code grouping.

The corresponding boxplot for the AVGGIFT by Zip code and Gender is provided below:



This boxplot shows that men typically donate more money to the university within a zip code, but there is not much of a difference in the AVGGIFT to the university by donors that do not behave as outlier. This boxplot also shows that Zip Code 4 is the most likely to provide a donation greater than the other zip codes as described by the propensity of donations to be outside the upper fence and by the larger upper fence in the boxplot. We can also see that the donor with the largest average donation is a woman in zip code 4.

**Part 2: Lifetime donations by Income Level and Homeowner Status**

While it is nice to look at how much an individual will donate on a single instance (AVGGIFT), it is also important to look at how much money the individual will donate over their lifetime. While we do not know the age or stage in their career the individual is in, we can see how the income level and their homeowner status effects how much money the individual has given over their lifetime. The code provided in the Appendix will provide meaningful analysis on the amount of lifetime donation for a given income and homeowner status using the following steps:

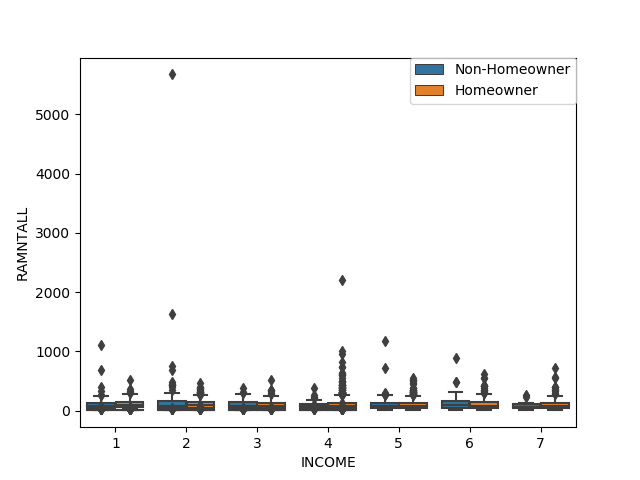
* Create a main dictionary to contain all the statistics calculated for each income level and homeowner status
* Create a list for all of the statistics, income levels, and homeownership status levels that will be used as keys in the dictionary
* Create a nested loop that has income level as the outer loop and homeownership status as the inner loop and index all the rows that have the same income level and homeownership status as stated in the loop
* Calculate the count, min, max, median, mean, standard deviation, and sum of donations for each income and home ownership status and save the value using the keys income level -> homeownership status -> statistic in the incomeHoStats dictionary
* Create an output file called IncomeHomeownerDonations.csv that will be contain two tables of the statistics for each income level for both homeowners and non-homeowners.
* Create a boxplot that has an x-axis of the income levels and a y-axis of the total donations, using color to separate by homeowner status

The csv file will be as follows:



The csv file shows that there is a huge range in the amount of lifetime donations, as shown by the massive standard deviation. The tables also show that as the income level increases, the median generally decreases. The table also shows that a huge percentage of donors are homeowners in income category 4. Homeowners in income category 4 are the category of donors that have donated the most money. On average, non-homeowners in income category 2 donate the most money, but this could be skewed as the maximum lifetime donation is significantly higher the other category’s maximum lifetime donation. Homeowners in income category 1 has the highest median of donors, so it is probably the most lucrative category of donors.

The corresponding boxplot for the Lifetime donations by income and homeowner status is provided below:



The boxplot shows that there is in fact a large influence from an outlier to the mean of the lifetime contributions of non-homeowners in income category 2 and that there is a larger upper range in the donations from homeowners in income category 4, suggesting that focusing marking and promotional efforts towards those in that category would be to our benefit.

**Appendix**

'''

File Name: HW1.property

Created By: Alex Hyman

Created on: 7 Aug 2018

Purpose: To explore a the Donors\_Data.csv file and answer some questions relating

to the data and provide some graphics

Questions:

1. How much are people in each zipcode donating to the university, and what does

the distribution look like by gender

Approach: We will create a csv file that outputs the mean of the average gift for

each gender number within each zipcode. We will also create a boxplot of the

average gift for each gender in each of the zip codes to show the spread

2. Does homeownership have an effect on the independent income variable in terms

in terms of the the response variable of lifetime donations to the school?

Approach: create a table for each of the income levels and produce a table that

includes the min, max, mean, median, sd of RAMNTALL for both categories of

homeowners and provide a boxplot

'''

import csv #Reading the csv

import pandas as pd #Pandas for data frames

import matplotlib.pyplot as plt #For plotting

import seaborn as sns #More for plotting

import numpy as np #For statistics

if \_\_name\_\_ == "\_\_main\_\_":

##Question 1

#Converting donor data into data frame

donors = pd.read\_csv("Donors\_Data.csv")

#List of all the dummy zip code columns

zip\_columns = ["zipconvert\_2", "zipconvert\_3", "zipconvert\_4", "zipconvert\_5"]

#Labels we are giving to the different zip codes

zip\_labels = ["Zipcode 1", "Zipcode 2", "Zipcode 3", "Zipcode 4"]

#for each column name and label in the zipcode columns and labels

for col, label in zip(zip\_columns, zip\_labels):

#Find all rows that have an instance of 1 in that column and in a new column

#"zip\_all", give it the corresponding value specified in zip\_labels

donors.loc[donors[col] == 1, "zip\_all"] = label

#Create a boxplot of the average gift amount by zipcode, separated by

#the gender dummy variable

genList = list(set(donors["gender dummy"]))

#Creating a dictionary to organize the means for each zip and gender

meanGift = {}

#For each of the zip codes in the zip code list

for zip in zip\_labels:

#with the zip code as the key, create an empty dictionary

meanGift[zip] = {}

#For each gender in the gender list

for gender in genList:

#find all the rows that have the same zip code as zip and the

#same gender as the gender and retrieve the AVGGIFT

allAvg = donors.loc[(donors["zip\_all"] == zip) & \

(donors["gender dummy"] == gender), "AVGGIFT"]

#In the dictionary index the zipcode and gender and place the

#average of the number in the dictionary

meanGift[zip][gender] = np.mean(allAvg)

#initializing the name of the output file

outfile = 'genderZipGift.csv'

#write a new csv file called genderZipGift

with open(outfile, "w") as outcsv:

#initializing the csv writer with the opened out file

csvwriter = csv.writer(outcsv)

#Writing the title on the first row

csvwriter.writerow(["Average gift by Gender within Zipcode"])

#Skipping two rows

csvwriter.writerow([])

csvwriter.writerow([])

#in the zip\_labels list, insert an empty string in the 0 index

#to format the headers of the zip code

zip\_labels.insert(0, "")

#write the labels for the zip code skipping the first column

csvwriter.writerow(zip\_labels)

#For each of the genders in the gender list

for gender in genList:

#creating the gender label for the table in the list

genAvg = ["gender {:d}".format(gender)]

#For each zip code

for zip in zip\_labels:

#There is an error with the empty string, so catching error

try:

#Appending the mean AVGGIFT for each gender to the genAvg list

genAvg.append(meanGift[zip][gender])

except KeyError:

#If there is an error, next in the zip\_label

continue

#After looping through the zip codes write the row in the csv file

#that has the averages for that gender dummy

csvwriter.writerow(genAvg)

#After looping through the genders, close the file

outcsv.close()

#Create the boxplot with the zip code as the x-axis and the average gift

#as the y-axis in the order of the zip\_labels (except the ""). For each

#zip code, also separate out by the gender in the zip code and use the

#donors data frame as the source

sns.boxplot(x = 'zip\_all', y = "AVGGIFT", order = zip\_labels[1:], \

hue = "gender dummy", data = donors)

#Saving the figure

plt.savefig("zipBoxplot.png")

##Question 2

#Create a dictionary for keeping track of statisitcs

incomeHoStats = {}

#list of stats we will calculate

stats = ["count", "min", "max", "median", "mean", "sd", "Total Donations"]

#List of all unique income levels

incomeLevels = list(set(donors["INCOME"]))

#categories for homeowners

homeowner = [0,1]

#Looping through incomeLevels

for level in incomeLevels:

#Create a dictionary inside of the dictionary with the income level as

#the key to that dictionary

incomeHoStats[level] = {}

#inner looping through the homewoner status creating a dictionary for

#each level of homeownership inside the dictionary created for the income

#level main dict -> income level -> homeowner status

for HOStatus in homeowner:

#Creating an empty dictionary in the homeowner status dict to

#contain the statisitcs income level -> homestatus -> statisitc

incomeHoStats[level][HOStatus] = {}

#indexing in pandas the rows that have the have the same income

#and same homeowner status as specified by the loop above

tempFrame = donors.loc[(donors["INCOME"] == level) & \

(donors["homeowner dummy"] == HOStatus)]

#Storing the count of instances in the count key

incomeHoStats[level][HOStatus]["count"] = len(tempFrame)

#Storing the minimum of instances in the min key

incomeHoStats[level][HOStatus]["min"] = min(tempFrame["RAMNTALL"])

#Storing the maximum of instances in the max key

incomeHoStats[level][HOStatus]["max"] = max(tempFrame["RAMNTALL"])

#Storing the mean of instances in the mean key

incomeHoStats[level][HOStatus]["mean"] = np.mean(tempFrame["RAMNTALL"])

#Storing the median of instances in the median key

incomeHoStats[level][HOStatus]["median"] = np.median(tempFrame["RAMNTALL"])

#Storing the sd of instances in the sd key

incomeHoStats[level][HOStatus]["sd"] = np.std(tempFrame["RAMNTALL"])

#Storing the total $ amount of donations in that class to the Total

#Donations key

incomeHoStats[level][HOStatus]["Total Donations"] = sum(tempFrame["RAMNTALL"])

#initializing the name of the second out file

outfile2 = "incomeHomeownerDonations.csv"

#writing the csv file for an output

with open(outfile2, "w") as csvout2:

#initializing the csv writer

csvwriter = csv.writer(csvout2)

#Creating a title for the file

csvwriter.writerow(["Lifetime Donation Statistics"])

#Leaving a space

csvwriter.writerow([""])

#Writing a title for the table

csvwriter.writerow(["Lifetime Donations for Non-Homeowners"])

#initializing the header for the table

header = ["Income Levels"]

#extending the header list to include the income levels

header.extend(incomeLevels)

#Writing the header

csvwriter.writerow(header)

#for each of the statistics in stats

for stat in stats:

#Start an empty list

statRow = []

#for each income level

for income in incomeLevels:

#in the dictionary using income level -> homeowner status -> statistic

#find the statisitc for the group and append it to the statRow

#list

statRow.append(incomeHoStats[income][0][stat])

#Add the name of the statvto the 0 index of the statRow list

statRow.insert(0, stat)

#Write the row in the csv file

csvwriter.writerow(statRow)

#Leaving a blank row to separate the table for homeowner table

csvwriter.writerow([""])

#writing the title in the csv file

csvwriter.writerow(["Lifetime Donations for Homeowners"])

#Writing the header for the second table

csvwriter.writerow(header)

#looping through each statisitc

for stat in stats:

#Create an empty list to place the statistic for each income level

statRow = []

#for each income level

for income in incomeLevels:

#find the statistic using income level -> homeownership -> stat

#and append it to the statRow list

statRow.append(incomeHoStats[income][1][stat])

#add the statistic name to the 0 index of the list

statRow.insert(0, stat)

#Write the row in the csv file that has the statistic name followed by

#the statistic for the omeowners in the corresponding income level

csvwriter.writerow(statRow)

#close the file

csvout2.close()

#Create a boxplot that has the distribution of total donations for each level

#of INCOME separated by homeowner status

bp = sns.boxplot(x = "INCOME", y = "RAMNTALL", hue = "homeowner dummy", data = donors)

#getting the handle and labels for the legend

handles, labels = bp.get\_legend\_handles\_labels()

#Getting rid of the duplicates and settingt the labels

l = plt.legend(handles[0:2], ["Non-Homeowner", "Homeowner"], loc=0, borderaxespad=0.)

#Saving the figure

plt.savefig("LifetimeDonations\_Income.png")