# Does Urban Development Prevent Obesity?

by Kellen Bullock Oklahoma State Univserity


### Abstract:

Obesity has been a growing epidemic in the United States of America. Pervious studies have found the causes to be complex but one aspect remains siginifcantly influencital, the enviroment. This study will be observing obesity rates in Allengheny County, PA. Using Census data, with parks and gym data collected from the Google Maps Platform Places API a least order squares regression will be constructed.


# Table of Contents



# Literature Review

**Intro:**

The United States of America has more Americans than ever are falling into the category of obese. Many studies have revealed that morbid obesity has been increasing in the past 20 to 30 years in the United States (Zang 2004; Ogden 2012; Sturm 2007). With obesity rate increasing and only predicted to further increase in the coming years (Zang 2004; Ogden 2012). With the rise of obesity, studies have shown this to be a growing problem in terms of cost of health care and finical strain on healthcare systems (Rosenberger 2005; McLaren 2007; Finkelstein 2010). Identifying environments that reduce obesity is key to implementing strategies to reduce this epidemic. The format of this review will address work performed on: what obesity is, how it is measured, how it is treated, formality of anti-obesogenic space and what geographic studies have done to attempt to measure causes.

To begin with, obesity is defined by the Center for Disease Control and Prevention as “Weight that is higher than what is considered as a healthy weight for a given height is described as overweight or obese” (CDC 2015). To further clarify, the human body is considered healthy when tissues are proportioned adequately.

For example, in the circumstance of obesity, an excess of adipose tissue (fat) and a scarcity of muscle tissue would make a person unhealthy.

Obesity is primarily measured by the Body Mass Index (BMI) (CDC 2015). Invented for screening purposes the measurement is based on height and weight squared (CDC 2015). Individuals who measure shorter and heavier fall under the category obesity as determined by the metric of 35. Multiple authors have claimed that this is not a good way of measuring obesity because of not directly observing how much adipose tissue is present (Eckel 2008; Adam 2007). The metric weight does not take into account the differences between muscle and adipose. Muscle is denser than adipose and in athletes who train frequently many of them can fall into the category of obese because of their weight. Generally, it has been used as a screening method over the past century and new measurements such as the body composition index BCI are beginning to replace it as the standard measurement. Now that obesity has been defined as individuals who have a higher weight than what is considered healthy weight.

**Treatment and more background:**

Next treatment of obesity has become frequent and wide spread the medical community has established a study known as bariatrics. Bariatric medical treatment is composed of three different broad fields that will be discussed in more detail further on in this review. Among them are dieting, surgery, physical activity. In many circumstances all are used in the treatment of obesity as shown in the literature (Ley 2007; Adam 2007; Kahn 2002; Eckel 2008; ). What dieting and surgery have in common is they are covered by insurance where as physical activity is not(Rosenberger 2005).

Dieting is often the first step in altering the proportion of adipose tissue in a individual's body(Ley 2007; Eckel 2008). Now more so than ever high caloric food that a plentiful source of fats and carbohydrates is widely available (Ley 2007; Janiszewski Ross 2007). Where as lean protein has risen in price. These factors have lead to disproportioned diets that create obesity very frequently. Strategic dieting has shown to have significant effect on reducing an individual's adipose tissue with the addition of exercise. For the purposes of this review and to better refine the focus, dieting is acknowledged as large factor in influencing obesity but largely controlled by variables that are hard to measure such as choice.

Since dieting is often not enough to give an obese individual to change, gastric bypass surgery and Colorectal surgery have been used to drastically alter an obese individual's food intake and weight (Adam 2007; Courcoulas 2015; Gloy 2013). All papers say that these methods are extremely effective at altering a person's BMI (Adam 2007; Courcoulas 2015; Gloy 2013). In many of these circumstances these operations are covered by medical insurance because of their link to treating Type II diabetes. While there are positives to these methods, what has begun to come to light are the physiological effects. In article Long-Term Mortality after Gastric Bypass Surgery, have found that suicides rates are higher in individuals that have received this treatment as oppose to individuals who opted out (Adam 2007). This is not to say these people did not receive extremely good benefits from sugery, such as reduced cardio respiratory, diabetes, and caner (Adam 2007). What I suspect is in some cases there a level of self perception trauma. The mean behind that is someone else had to change them, and they could not change themselves. Which leads to the next method that can provide the individual with the chance to shape themselves.

Physical activity has always been encouraged by physicians as a method of maintaining health(Kahn 2002; Janiszewski Ross 2007). Though encouraged by physicians, evidence for exercise alone having the desired effect of reducing body mass has mixed results (Eckel 2008, Hill Wyatt 2005). Changes in metabolic rates and how the body manages adipose tissue occur at varying degrees(cite). In combinations with dieting, physical activity has been shown to make changes (Eckel 2008; Hill Wyatt 2005). What is often not acknowledged in the literature is where these activities take place. The environment that can treat obesity is very formal. This hold true for both the surgical procedures as well as physical activity. In a controlled environment and a space made for encouraging physical activity progress toward a healthy body weight can be achieved more feasibly.

**Main body:**

Formality of the environment is key to understanding the underlying causes of obesity is the relationship with spaces where physical activity occurs. Public and private parks and gyms offer spaces for leisure time to be used for physical activity. This is not to say physical activity does not occur within the residence of an individual but communities are generally healthier when there are more formal places to exercise. Since obesity is an epidemic identifying what the anti-obese environments are may shed more understanding on where it is and is not happening (Ding, Klaus 2012). Partly what all these papers have in common is that the closer these faculties are to the population the more likely people are to use them (Ding, Klaus 2012; Humpel 2002; Brown 2009; Swimburn 1999; Pitts 2013). This point is what most studies are lacking is the combination of both private and public facilities for people to partake in physical activity. More often these spaces are treated separately because of the differences between private gyms and public parks, as well as, the formality differences between the two. Gyms generally offer more equipment and specialized help for helping individuals with their weight goals. Parks on the other hand have limited equipment that is often more suited for children than adults.

When evaluating the literature regarding parks authors are almost always observing cardio type exercises (Wolch 2011; Blanck 2012; Gies 2006). Walking, running, and bicycling are among the most common ways of engaging in physical activity in parks (Wolch 2011; Blanck 2012; Gies 2006). Since obesity does involve lifestyle choices people participating in physical activity these places offer a way to prevent obesity. Cardio exercise has easy entry for many people because of its ties to everyday life, such as walking. However, treatment of obesity often requires more aggressive and structured methods that have a higher skill level required of the participant. What has been shown is that areas with higher density of gyms encourage more physical activity and reduce obesity rate (Pitts 2013). This possible is a result of more specialized training that gyms may offer.

Gyms offer a more controlled environment that allow for mixed and high specialized methods of physical activity. Such as swimming, and weight bearing exercises. specialized activities such as swimming and weight lifting are much harder to enter because of their requirement of background skills to be an effective way of changing body composition (Eckel2008; Janiszewski 2007). This these skills are not free though. Pricing and variation of service often deters many who aspect that makes gyms often not ideal for a treatment of obesity is the variation in pricing (Evans 2013). What has been shown is that areas with higher density of gyms encourage more physical activity and reduce obesity rate (Pitts 2013).

Many studies have been done on how socio economic activity effects obesity rates. What the authors of (Abercrombie 2008), that recreation facilities in lower income and high minority neighborhoods was not found. A similar finding was in (Zhang 2004), that over time correlation has weakened between income and obesity. Both of these studies are saying that income is become less of an impact on obesity but obesity is still increasing. Also what is becoming evident is that going to a privately owned gym is become a more likely option for receiving treatment.

Due to the complexity of the problem that is obesity many geographic studies have faced numerous challenges collecting obesity data, scaling problems, and variable. These are classic geographic problems but within the ethical guidelines of Health Insurance Portability and Accountability Act (HIPPAA) keeping individuals aggregate is the most sensible thing. Differences in scaling causes issues with repeatability of methods in different locations as well as validity issues. Determining where and what constitutes as space where people can be physically active is also a overwhelming task. At the finest scale almost any space can be used for physical activity. This causes the need for aggregation which could be hiding underlying factors.

Collecting data on obesity has primarily been done and established by the CDC. What the CDC does is not reveal individuals because county level of scale they are collecting at. Issues with this are often a result of not seeing enough of the picture to know what is going on. County level data does not show how parks for facilities in local environments effect the population.

The body of work concerning how obesity is effected by space occurs a variety of scales and usually urban to suburban areas (Ding 2012; Gordon-Larsen 2006; Rosenberger 2005; Swinburn 1999; Brownson 2009). A clear problem with this is repeatability of experimentation. The underlying issue with this is data collect on obesity rates without violating privacy ethics. The finer the scale the larger this problem becomes. Ding and company have found that different scales offer different challenges for determining variables (Ding 2012). For example, when observing the neighborhood level sidewalks are usually evaluated. Where as, at the Census Tract scale they would be too fine of scale to include in the study.

Setting what variables should be evaluated has also been an issue of great debate among studies that evaluate how the environment effects obesity. The key to this problem from what both of these reviews have found is the lack of concrete definition between place and study area (Ding 2012).

**Conclusion:**

To better understand anti-obesogentic environments as revealed by the pervious studies mentioned, emphasis placed on the formal space, the scale, and the independent variables needs to be included in research. This can be done with the inclusion of private businesses done with public parks in a regressive study. The aims of this study are to present those results.

## MetaData for Obesity Data

{\rtf1\ansi\ansicpg1252\cocoartf1671\cocoasubrtf200 {\fonttbl\f0\fmodern\fcharset0 Courier;} {\colortbl;\red255\green255\blue255;\red0\green0\blue0;} {*\expandedcolortbl;;\cssrgb\c0\c0\c0;} \margl1440\margr1440\vieww10800\viewh8400\viewkind0 \deftab720 \pard\pardeftab720\partightenfactor0

\f0\fs26 \cf0 \expnd0\expndtw0\kerning0 {"@type": "dcat:Dataset", "accessLevel": "public", "contactPoint": {"@type": "vcard:Contact", "fn": "Lynda Jones", "hasEmail": "mailto:Lynda.Jones@alleghenycounty.us"}, "description": "Obesity rates for each Census Tract in Allegheny County were produced for the study \u201cDeveloping small-area predictions for smoking and obesity prevalence in the United States.\" The data is not explicitly based on population surveys or data collection conducted in Allegheny County, but rather estimated using statistical modeling techniques. In this technique, researchers applied the obesity rate of a demographically similar census tract to one in Allegheny County to compute an obesity rate.", "distribution": [{"@type": "dcat:Distribution", "downloadURL": "https://data.wprdc.org/dataset/b049da40-910a-4b9d-9438-ff584e432bb1/resource/fce248f0-8697-4d2a-bbe0-2da826776bfa/download/obesity-ac-2006-2010censustracts.csv (https://data.wprdc.org/dataset/b049da40-910a-4b9d-9438-ff584e432bb1/resource/fce248f0-8697-4d2a-bbe0-2da826776bfa/download/obesity-ac-2006-2010censustracts.csv)", "mediaType": "text/csv", "title": "Obesity Rates Allegheny County Data"}, {"@type": "dcat:Distribution", "downloadURL": "https://data.wprdc.org/dataset/b049da40-910a-4b9d-9438-ff584e432bb1/resource/9e6a2b51-90ca-47f8-b2f2-5ea275bd2f53/download/data-obesetractgamephitaugust082013.csv (https://data.wprdc.org/dataset/b049da40-910a-4b9d-9438-ff584e432bb1/resource/9e6a2b51-90ca-47f8-b2f2-5ea275bd2f53/download/data-obesetractgamephitaugust082013.csv)", "mediaType": "text/csv", "title": "Obesity Rates Statewide Data"}, {"@type": "dcat:Distribution", "downloadURL": "https://data.wprdc.org/dataset/b049da40-910a-4b9d-9438-ff584e432bb1/resource/a19a6a1a-90cb-4d39-859c-0c7eacf256e2/download/dictionary-obesity-rate.csv (https://data.wprdc.org/dataset/b049da40-910a-4b9d-9438-ff584e432bb1/resource/a19a6a1a-90cb-4d39-859c-0c7eacf256e2/download/dictionary-obesity-rate.csv)", "mediaType": "text/csv", "title": "Obesity Rates Data Dictionary"}, {"@type": "dcat:Distribution", "downloadURL": "https://data.wprdc.org/dataset/b049da40-910a-4b9d-9438-ff584e432bb1/resource/d90c9929-9b9e-49eb-b2cb-0460b7dc73e8/download/obesityrates.zip (https://data.wprdc.org/dataset/b049da40-910a-4b9d-9438-ff584e432bb1/resource/d90c9929-9b9e-49eb-b2cb-0460b7dc73e8/download/obesityrates.zip)", "mediaType": "application/zip", "title": "Obesity Rates Allegheny County Data"}, {"@type": "dcat:Distribution", "downloadURL": "https://tools.wprdc.org/guides/dash-data-guide/alleghenycountyobesityrates/ (https://tools.wprdc.org/guides/dash-data-guide/alleghenycountyobesityrates/)", "mediaType": "text/html", "title": "DASH Data Guide"}], "identifier": "b049da40-910a-4b9d-9438-ff584e432bb1", "keyword": ["ACHD DASH", "Obesity", "social determinants of health"], "license": "http://www.opendefinition.org/licenses/cc-zero (http://www.opendefinition.org/licenses/cc-zero)", "modified": "2018-04-03T18:27:14.816271", "publisher": {"@type": "org:Organization", "name": "Allegheny County / City of Pittsburgh / Western PA Regional Data Center"}, "theme": ["Health"], "title": "Allegheny County Obesity Rates"}}

# Module Imports

Modules are packages made in the Python language. Modules are packaged functions that can be called from files to prefrom calculations. These modules have been downloaded from the open source python distribution known as Anaconda. Anaconda is the leading data science for machine learning and data science for the languages R, and Python.

The following modules were used from the python distribution known as the Anaconda distribution: Pandas to tabulate data, Numpy to tabulate data, Matplotlib and pyplot to produce visuals, Geopandas to map, scipy and scit-kitlearn to perform the regression analysis.

In [79]:

```
import numpy as np
import pandas as pd
from pandas import DataFrame
import os
import matplotlib.pyplot as plt
import geopandas
from geopandas import GeoDataFrame
import fiona
from scipy import stats
from scipy.stats import kurtosis
from scipy.stats import skew

pd.set_option('display.max_rows', 5)
pd.set_option('display.max_columns', 5)
pd.set_option('display.width', 10)
```

In [80]:

```
data = pd.read_csv('Obesity_Data.csv')
data = data.rename(columns = {'2000 Tract':'Tract_Num'})
```

This data contains obesity rates at the tract scale. Looking at the Neighborhood names and compared to the tract numbers each tract as a unquine number but it can have mulpile of the same neighborhood names.

My Plan is to join this table with a tracts shapefile from the year 2000. On the field 2000 Tract which I will end up renaming to something easier.

In [81]:

```
data
```

Out[81]:

| | stname | GEOID | ... | City Neighborhood | 2006-2010 estimate of obesity |
|---|---|---|---|---|---|
| **0** | Pennsylvania | 42003010300 | ... | Bluff | 0.246936 |
| **1** | Pennsylvania | 42003020100 | ... | Central Business District | 0.668012 |
| **...** | ... | ... | ... | ... | ... |
| **414** | Pennsylvania | 42003981800 | ... | Lincoln-Lemington-Belmar | 0.574293 |
| **415** | Pennsylvania | 42003982200 | ... | North Oakland | 0.138806 |

416 rows × 8 columns

**Tract Validation in both tables**

Just confirming that this tract exist becuase in the other table it does too but loses gemoetry in the data_plot map.

In [82]:

```
data.loc[data['Tract_Num'] == 151500]
```

Out[82]:

| | stname | GEOID | ... | City Neighborhood | 2006-2010 estimate of obesity |
|---|---|---|---|---|---|
| **59** | Pennsylvania | 42003151500 | ... | Hazelwood | 0.399853 |

1 rows × 8 columns

## Duplicate Check

As seen by the test to ensure we do not have duplicate records. This code shows the dataframe without any dulicpates. If there were duplicates there what you would see is True. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.duplicated.html (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.duplicated.html)

In [83]:

```
data.duplicated('Tract_Num')
```

Out[83]:

```
0      False
1      False
       ...
414    False
415    False
Length: 416, dtype: bool
```

## Geopandas Module to visiualize data

Here we have the Allegheny shapefile for county tracts set for the year 2016. This shapefile was made by the Allegheny county gis comission. I have been having issues with using this file I think I should create a Allegheny tracts shapefile based on the 2000 cencus. http://geopandas.org (http://geopandas.org)

In [84]:

```
study_area = geopandas.read_file('All_Tracts/2000_Tracts/tl_2010_42003_tract00.shp')
#study_area = geopandas.read_file('All_Tracts/2016 Tracts/Allegheny_County_Census_Tracts_2016.shp')
#study_area = geopandas.read_file(geopandas.datasets.get_path('Allegheny_County_Census_Tracts_2016.shp'))
study_area = study_area.rename(columns={'TRACTCE00': 'Tract_Num'})
study_area.head()
```

Out[84]:

| | STATEFP00 | COUNTYFP00 | ... | INTPTLON00 | geometry |
|---|---|---|---|---|---|
| 0 | 42 | 003 | ... | -079.8210570 | POLYGON ((-79.817899 40.416443, -79.81759 40.4... |
| 1 | 42 | 003 | ... | -079.8102282 | POLYGON ((-79.814165 40.389742, -79.8142479999... |
| 2 | 42 | 003 | ... | -079.8072431 | POLYGON ((-79.81406699999999 40.383586, -79.81... |
| 3 | 42 | 003 | ... | -079.7856771 | POLYGON ((-79.795081 40.39221, -79.79525699999... |
| 4 | 42 | 003 | ... | -079.7702632 | POLYGON ((-79.766164 40.383842, -79.766178 40.... |

5 rows × 13 columns

### Quick visual of the tracts in the county

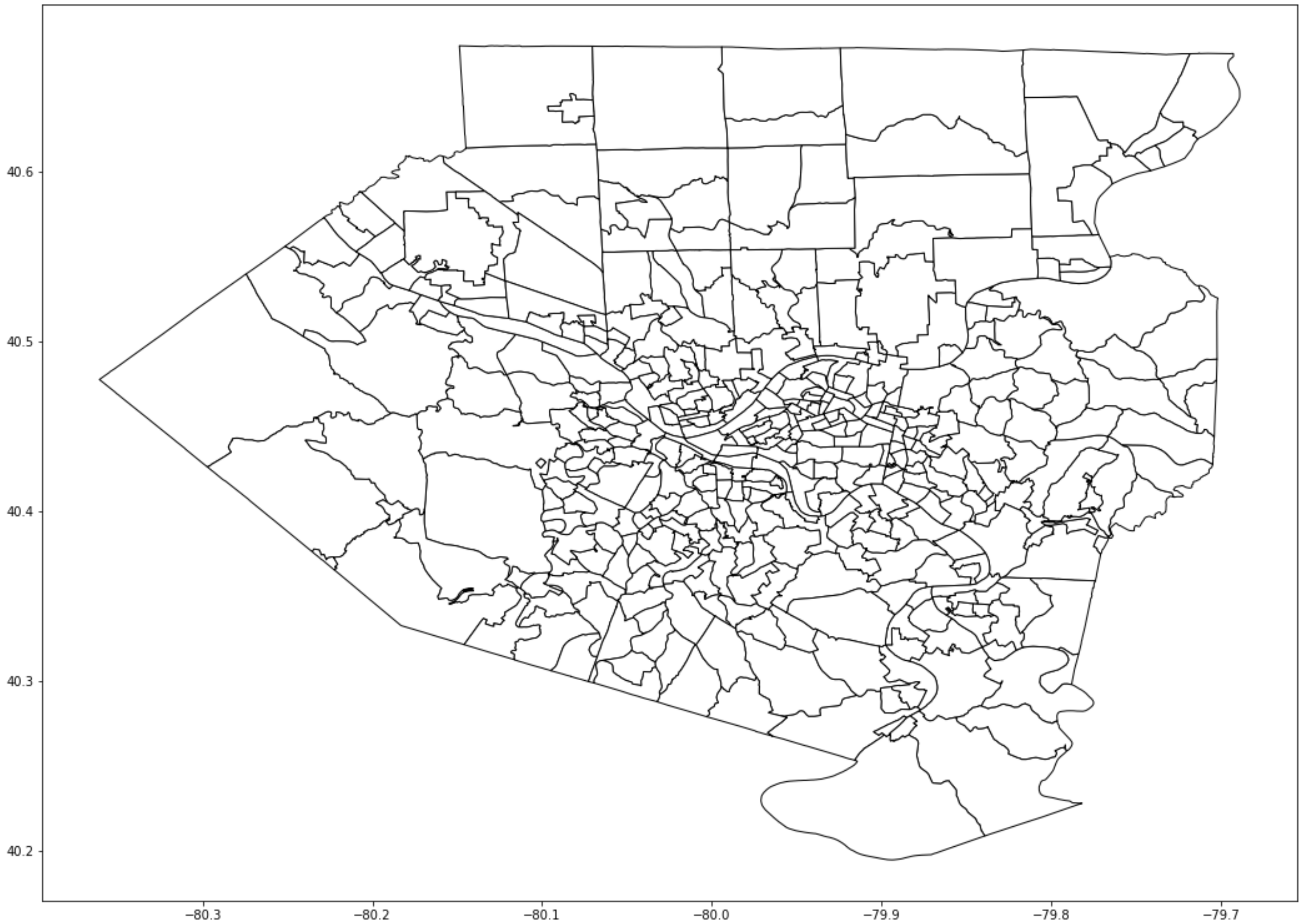
This does not show any missing tracts or incomplete polygons.

In [85]:

```
study_area.plot(color =('white'),figsize=(18,14),edgecolor=('black'))
```

Out[85]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a318f6a58>
```



# First Merge between data and study_area to make data_plot.

This code is casting the Tract_Num variable into an integer so it can be joined with the obesity data file. study_area['Tract_Num'] = study_area['Tract_Num'].astype(int) data['Tract_Num'] = data['Tract_Num'].astype(int)

Then I use the merge method from pandas.

To join files in pandas the methods merge, join, and concat(concatenation) are made avalible. The differences are listed in the documentation but the reason why I used merge here instead of join is because of the issues I was having with the duplicates in the obesity dataset. Below are the paratmeters for merge.

pd.merge(left, right, how='inner', on=None, left_on=None, right_on=None, left_index=False, right_index=False, sort=True, suffixes=('_x', '_y'), copy=True, indicator=False, validate=None)

pd.merge(left, right, on='B', how='outer', validate="one_to_many")

Converting types to matchup for the merge

df['year']=df['year'].astype(int)

I also did some clean up here to make the table nicer to look at. I dropped some columns as indicated by the drop function and renamed 006-2010 estimate of obesity to obesity for quality of life.

### Methodology

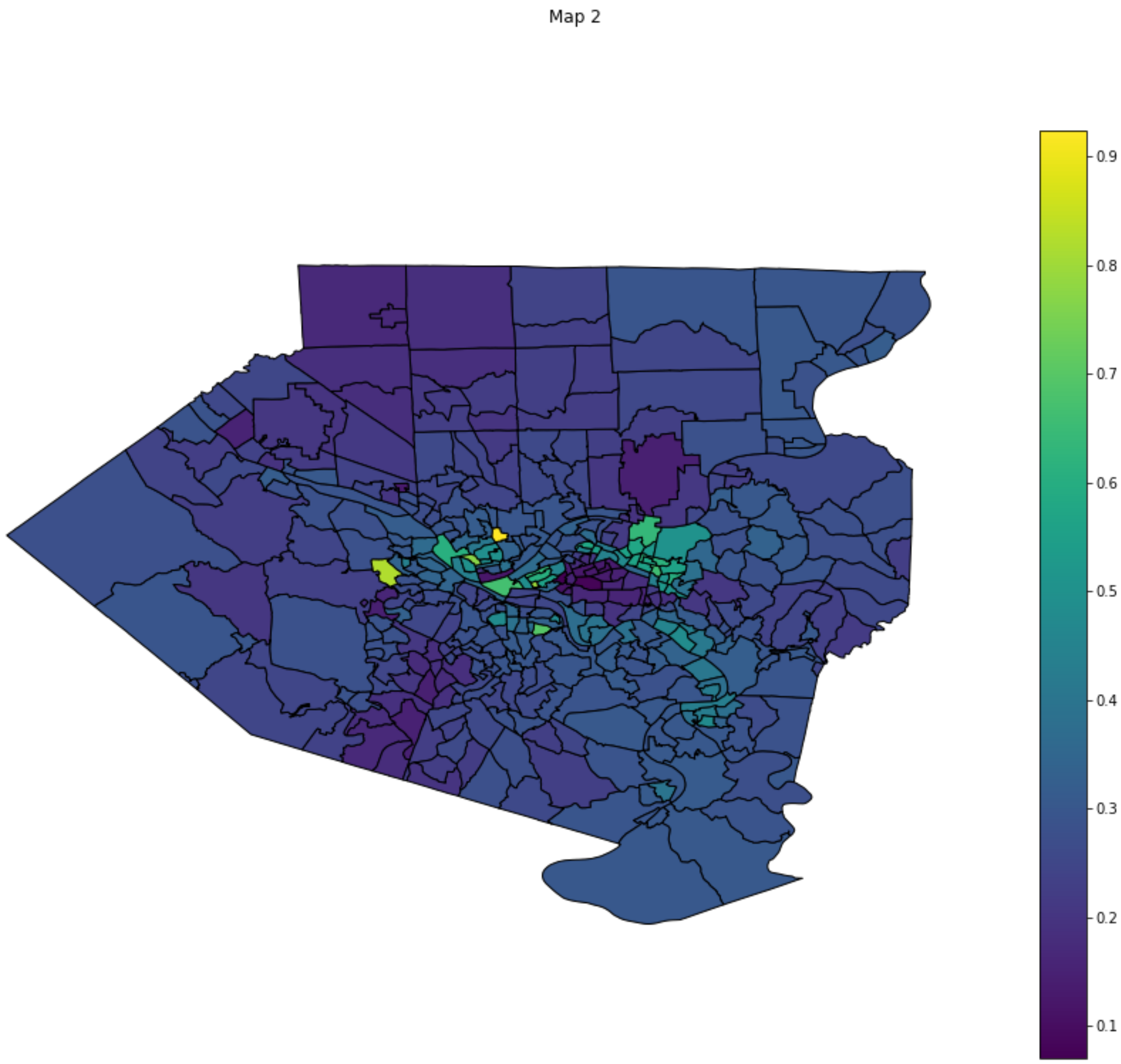
In Obesity is inversly related to the physical enviroment, the authors choose to use data at the county level and a website that stored facility data at the county level (Pitt 2013). Here I am doing exactly that but at the Census tract scale. Below is a visiualization of obesity rates in Allegheny County. Unlike Pitts study I choose to use an unclassified scheme to map the data. The advantage to doing this is there are not outliers the numbers just tell the story like it is.

In [86]:

```
study_area['Tract_Num'] = study_area['Tract_Num'].astype(int)
data['Tract_Num'] = data['Tract_Num'].astype(int)

data_plot = pd.merge(study_area, data, on='Tract_Num')
data_plot = data_plot.rename(columns = {'2006-2010 estimate of obesity': 'Obesit
y'})
#data_plot = data_plot.drop(['stname', 'GEOID_x','2010 Tract','MUNICIPALITY','Mu
nicipality','FID','STATEFP','COUNTYFP','AFFGEOID','GEOID_y','LSAD','ALAND','AWAT
ER','Shape__Are','Shape__Len'], axis=1)
#data_plot.to_csv('Joined.csv')
#data_plot = data_plot.dropna(subset=['geometry'])
#data_plot = GeoDataFrame(data_plot,geometry=data['geometry'])

f, ax = plt.subplots(1, figsize=(16,12))
ax = data_plot.plot(axes=ax, column='Obesity',edgecolor=('black'), legend=(True)
)
ax.set_axis_off()
f.suptitle('Map 2')
lims = plt.axis('equal')
plt.show()
```




**Map 2**

Above is the an unclassified scheme distribution of obesity rates throughout the tracts of Allengheny County PA. Contributions to styling and design: [http://darribas.org/gds15/content/labs/lab_03.html (http://darribas.org/gds15/content/labs/lab_03.html)](http://darribas.org/gds15/content/labs/lab_03.html)

## Attribute table for Map 2

In [87]:

```
data_plot
```

Out[87]:

| | STATEFP00 | COUNTYFP00 | ... | City Neighborhood | Obesity |
|---|---|---|---|---|---|
| **0** | 42 | 003 | ... | NaN | 0.287971 |
| **1** | 42 | 003 | ... | NaN | 0.299366 |
| **...** | ... | ... | ... | ... | ... |
| **414** | 42 | 003 | ... | NaN | 0.260487 |
| **415** | 42 | 003 | ... | NaN | 0.302539 |

416 rows × 20 columns

# Parks and Gyms Data collection

## Google Maps Platform to gather point, name, and gym type information.

The modules that are needed to do this are googlemaps to access the API. Pretty print so I can use python to extract information from the json file that will be downloaded. Time is so I can delay the code to produce the next page. Key is a method I developed to keep my API key to google unseen.

Source: [https://developers.google.com/places/web-service/intro (https://developers.google.com/places/web-service/intro)](https://developers.google.com/places/web-service/intro)

To obtain the point data for gyms and parks the Google Places API was queried for results at the 40.440174, -79.996307 latitude longitude in a 20 kilometer radius. The API produced a json file that was processed into shapefiles. This processing was performed using for loops in the standard python library. Each json file was iterated over 4 times to obtain the following information: geometry by latitude and longitude, names of both the parks and gyms, and place ID a code used by google to index all locations.

In [88]:

```
import googlemaps
import pprint
import time
from Key import GetKey

gmaps = googlemaps.Client(key = GetKey())

#places_result = gmaps.places_nearby(location = '40.440174,-79.996307', radius = 
20000, open_now = False, type = 'gym')
#time.sleep(3)
#places_result = gmaps.places_nearby(page_token = places_result['next_page_token
'])

#pprint.pprint(places_result)
```

Next idea: For loop through the dictionary and attempt to snatch up: geometry, location lat longs name place_id One I get it out into variables then maybe I can put it into a dataframe.

Okay here is me for loop going through an extracting all of the place ids. Since I can do this I can definitely harvest lat longs and gym names. (I might hold off on gym names and run a place details instead.

In [89]:

```
for place in places_result['results']:
    #making a variable for place id
    my_place_id = place['place_id']
    print(my_place_id)
```

```
---------------------------------------------------------------------------
-------
NameError                                 Traceback (most recent cal
l last)
<ipython-input-89-a32ae1a64035> in <module>
----> 1 for place in places_result['results']:
      2     #making a variable for place id
      3     my_place_id = place['place_id']
      4     print(my_place_id)

NameError: name 'places_result' is not defined
```

Below I have printed out the lat and longs for the points, but this is a lot more than what I wanted. I need to figure out how to go down one more layer in the dictionaries.

Here is an example of how to access the elements of nested dictionaries: people = {1: {'name': 'John', 'age': '27', 'sex': 'Male'}, 2: {'name': 'Marie', 'age': '22', 'sex': 'Female'}}

print(people[1]['name']) print(people[1]['age']) print(people[1]['sex'])

Incase I mess up: for place in places_result['results']:

```
#making a variable for lat
lat = place['geometry']
print(lat)
```

## Latitude

In [90]:

```
for place in places_result['results']:
    #making a variable for lat
    lat = place['geometry']['location']['lat']
    print(lat)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent cal
l last)
<ipython-input-90-9e558d466dca> in <module>
----> 1 for place in places_result['results']:
      2     #making a variable for lat
      3     lat = place['geometry']['location']['lat']
      4     print(lat)

NameError: name 'places_result' is not defined
```

## Longitude

In [91]:

```
for place in places_result['results']:
    #making a variable for lat
    long = place['geometry']['location']['lng']
    print(long)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent cal
l last)
<ipython-input-91-ca4254284c85> in <module>
----> 1 for place in places_result['results']:
      2     #making a variable for lat
      3     long = place['geometry']['location']['lng']
      4     print(long)

NameError: name 'places_result' is not defined
```

## Name?

In [92]:

```
for place in places_result['results']:
    #making a variable for lat
    name = place['name']
    print(name)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent cal
l last)
<ipython-input-92-ab2b88f1e999> in <module>
----> 1 for place in places_result['results']:
      2     #making a variable for lat
      3     name = place['name']
      4     print(name)

NameError: name 'places_result' is not defined
```

#### Creating the geometry field for geopandas

https://geopandas.readthedocs.io/en/latest/gallery/create_geopandas_from_pandas.html (https://geopandas.readthedocs.io/en/latest/gallery/create_geopandas_from_pandas.html)

I used the shapely tool set to help set the geometry for the points.

# Gym GeoDataFrame

This GeoDataFrame was used to hold the data for the gym points extracted from the Google Maps Platform. Detailed below the Gym_Points file is read into the GeoDataFrame. The Coordinates column is then formmated into the correct layout for GeoPandas to read.

In [93]:

```
from shapely.geometry import Point

gyms = pd.read_excel('Gym_Points.xlsx')
gyms['Coordinates'] = list(zip(gyms.Long, gyms.Lat))
gyms['Coordinates'] = gyms['Coordinates'].apply(Point)
gyms = GeoDataFrame(gyms, crs={'init': 'epsg:4269'}, geometry='Coordinates')
type(gyms)
gyms.head()
```

Out[93]:

| | Gym_Name | Lat | Long | Coordinates |
|---|---|---|---|---|
| 0 | Amazing Yoga | 40.451034 | -79.935016 | POINT (-79.9350163 40.4510345) |
| 1 | Town Place Fitness | 40.440750 | -80.003872 | POINT (-80.0038716 40.4407503) |
| 2 | Fitness Factory | 40.458515 | -79.925384 | POINT (-79.92538399999999 40.458515) |
| 3 | GreenTree SportsPlex | 40.420343 | -80.058117 | POINT (-80.05811679999999 40.420343) |
| 4 | Ice Castle Arena | 40.366207 | -80.026361 | POINT (-80.02636129999991 40.3662067) |

Saving layer out to shapefile:

In [13]:

```
gyms.to_file('ShapeFiles/Gyms/gyms.shp')
```

# Park Data

In [ ]:

```
gmaps = googlemaps.Client(key = GetKey())

places_result = gmaps.places_nearby(location = '40.440174,-79.996307', radius = 
20000, open_now = False, type = 'park')
time.sleep(3)
places_result = gmaps.places_nearby(page_token = places_result['next_page_token'
])

pprint.pprint(places_result)
```

## Place ID

In [ ]:

```
#appendfile = open('Parks_points.xlsx','a')
for place in places_result['results']:
    #making a variable for place id
    my_place_id = place['place_id']
    print(my_place_id)
```

## Name

In [ ]:

```
for place in places_result['results']:
    #making a variable for lat
    name = place['name']
    print(name)
```

## Lat

In [ ]:

```
for place in places_result['results']:
    #making a variable for lat
    lat = place['geometry']['location']['lat']
    print(lat)
```

## Long

In [ ]:

```
for place in places_result['results']:
    #making a variable for lat
    long = place['geometry']['location']['lng']
    print(long)
```

# Parks GeoDataFrame

In [94]:

```
parks = pd.read_excel('Park_Points.xlsx')
parks = parks.dropna()
parks['Coordinates_p'] = list(zip(parks.Long, parks.Lat))
parks['Coordinates_p'] = parks['Coordinates_p'].apply(Point)
parks = GeoDataFrame(parks, crs={'init': 'epsg:4269'}, geometry='Coordinates_p')
parks.head()
```

Out[94]:

| | Name | Lat | Long | Coordinates_p |
|---|---|---|---|---|
| **0** | Phipps Conservatory and Botanical Gardens | 40.439197 | -79.9474 | POINT (-79.9473787 40.4391973) |
| **1** | Gateway Center Park | 40.440537 | -80.0055 | POINT (-80.0054548 40.4405366999999) |
| **2** | Pittsburgh Parks Conservancy | 40.429858 | -79.9726 | POINT (-79.97261519999989 40.429858) |
| **3** | Schenley Park | 40.434849 | -79.9425 | POINT (-79.9424885999999 40.4348492) |
| **4** | Point State Park | 40.441572 | -80.0079 | POINT (-80.00785379999991 40.4415719) |

Saving layer out to shapefile:

In [403]:

```
parks.to_file(filename='ShapeFiles/Parks/parks.shp')
```

# Map with combined layers

### Methodology

This step was done to address facility deserts found in a study done "Neighbourhood deprivation and the cost of accessing gyms and fitness centres: National study in Wales"(Evans 2013). While their study did do a nearest neighbor analysis I do not think that would be worth wild here because of the small size of this data and the clustering in urban areas.

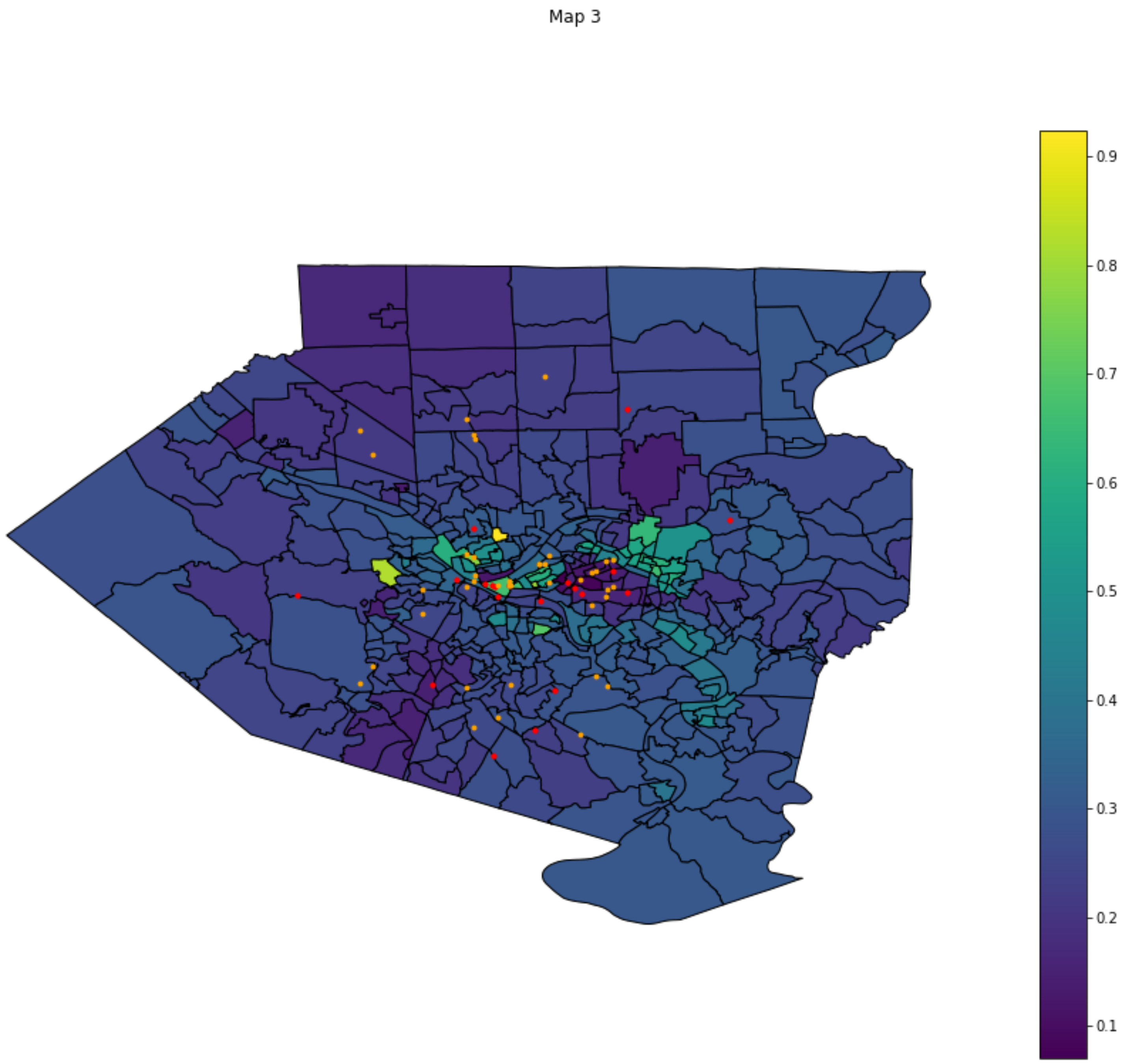
Below is **map 3**. This displays

Orange points are gyms. Red points are parks.

In [13]:

```
f, ax = plt.subplots(1, figsize=(16,12))
ax = data_plot.plot(axes=ax, column='Obesity',edgecolor=('black'), legend=(True)
)
parks.plot(ax=ax, marker='o', color='red', markersize=8)
gyms.plot(ax=ax, marker='o', color='Orange', markersize=8)
ax.set_axis_off()
f.suptitle('Map 3')
lims = plt.axis('equal')
plt.show()
```

```
/anaconda3/lib/python3.6/site-packages/geopandas/plotting.py:385: Fu
tureWarning: 'axes' is deprecated, please use 'ax' instead (for cons
istency with pandas)
  "(for consistency with pandas)", FutureWarning)
```




## Checking coordinate systems with crs

In [14]:

```
print(data_plot.crs)
```

{'init': 'epsg:4269'}

In [15]:

```
print(parks.crs)
```

{'init': 'epsg:4269'}

In [16]:

```
print(gyms.crs)
```

{'init': 'epsg:4269'}

# Spatial Joins

**This is the documentation for how to do a spaital join**

geopandas.sjoin(left_df, right_df, how='inner', op='intersects', lsuffix='left', rsuffix='right') Because this function is reseting the dataframe to behave like a panadas dataframe I needed to reset the geometry to make it a geodataframe again.

# Join instructions and workflow:

Spaital join gyms to data_plot

spatial join parks to data_plot

group by( like what you did down there) apply to

merge data_plots to demographic data

In [95]:

```
joined_park = geopandas.sjoin(parks, data_plot, how='right', op ='intersects')
joined_park
```

Out[95]:

| | index_left | Name | ... | City Neighborhood | Obesity |
|---|---|---|---|---|---|
| **index_right** | | | | | |
| **88** | 0.0 | Phipps Conservatory and Botanical Gardens | ... | Squirrel Hill South | 0.164996 |
| **88** | 3.0 | Schenley Park | ... | Squirrel Hill South | 0.164996 |
| **...** | ... | ... | ... | ... | ... |
| **414** | NaN | NaN | ... | NaN | 0.260487 |
| **415** | NaN | NaN | ... | NaN | 0.302539 |

440 rows × 24 columns

In [96]:

```
joined_gyms = geopandas.sjoin(gyms, data_plot, how='right', op = 'intersects')
joined_gyms
```

Out[96]:

| | index_left | Gym_Name | ... | City Neighborhood | Obesity |
|---|---|---|---|---|---|
| **index_right** | | | | | |
| **256** | 0.0 | Amazing Yoga | ... | Shadyside | 0.128451 |
| **256** | 16.0 | X Shadyside | ... | Shadyside | 0.128451 |
| **...** | ... | ... | ... | ... | ... |
| **414** | NaN | NaN | ... | NaN | 0.260487 |
| **415** | NaN | NaN | ... | NaN | 0.302539 |

427 rows × 24 columns

# Census Data Merge

The Census data contains demographic information such as age, race, and physical environmental varaibles. The Census data will be merged to data_plot. To reiterate the data_plot GeoDataFrame is the shapefile that contains the information on Allegheny county tracts.

- The demo variable will be used as the storage container for the Census Data.
- The column GEOID10 will be converted to GEOID to enable the merge to occur.
- Demo will be merged to data_plot using the right parameter to preserve the index of data_plots.
- The final GeoDataFrame will be created and the index column will be renamed to Key.

In [97]:

```
demo = geopandas.read_file('All_Tracts/2010_Tracts/Tract_2010Census_DP1.shp')
```

In [98]:

```
demo = demo.rename(columns={'GEOID10':'GEOID'})
demo['GEOID'] = demo['GEOID'].astype(int)
#join_2['GEOID'] = join_2['GEOID'].astype(int)
```

In [99]:

```
final = demo.merge(data_plot, how='right', on = 'GEOID')
#final = final.reset_index()
#final = final.drop(columns={'index_right','index_left'})
#final = GeoDataFrame(final, crs={'init':'epsg:4269'}, geometry='geometry_x')
final.index.name = 'Key'
final
```

Out[99]:

| | GEOID | NAMELSAD10 | ... | City Neighborhood | Obesity |
|---|---|---|---|---|---|
| **Key** | | | | | |
| **0** | 42003560500 | Census Tract 5605 | ... | NaN | 0.224347 |
| **1** | 42003560400 | Census Tract 5604 | ... | NaN | 0.437485 |
| **...** | ... | ... | ... | ... | ... |
| **414** | 42003151500 | NaN | ... | Hazelwood | 0.399853 |
| **415** | 42003453001 | NaN | ... | NaN | 0.205551 |

416 rows × 214 columns

In [22]:

```
final.to_excel('Final_DataFrame.xlsx')
```

In [23]:

```
final = GeoDataFrame(final, crs={'init':'epsg:4269'}, geometry='geometry_y')
```

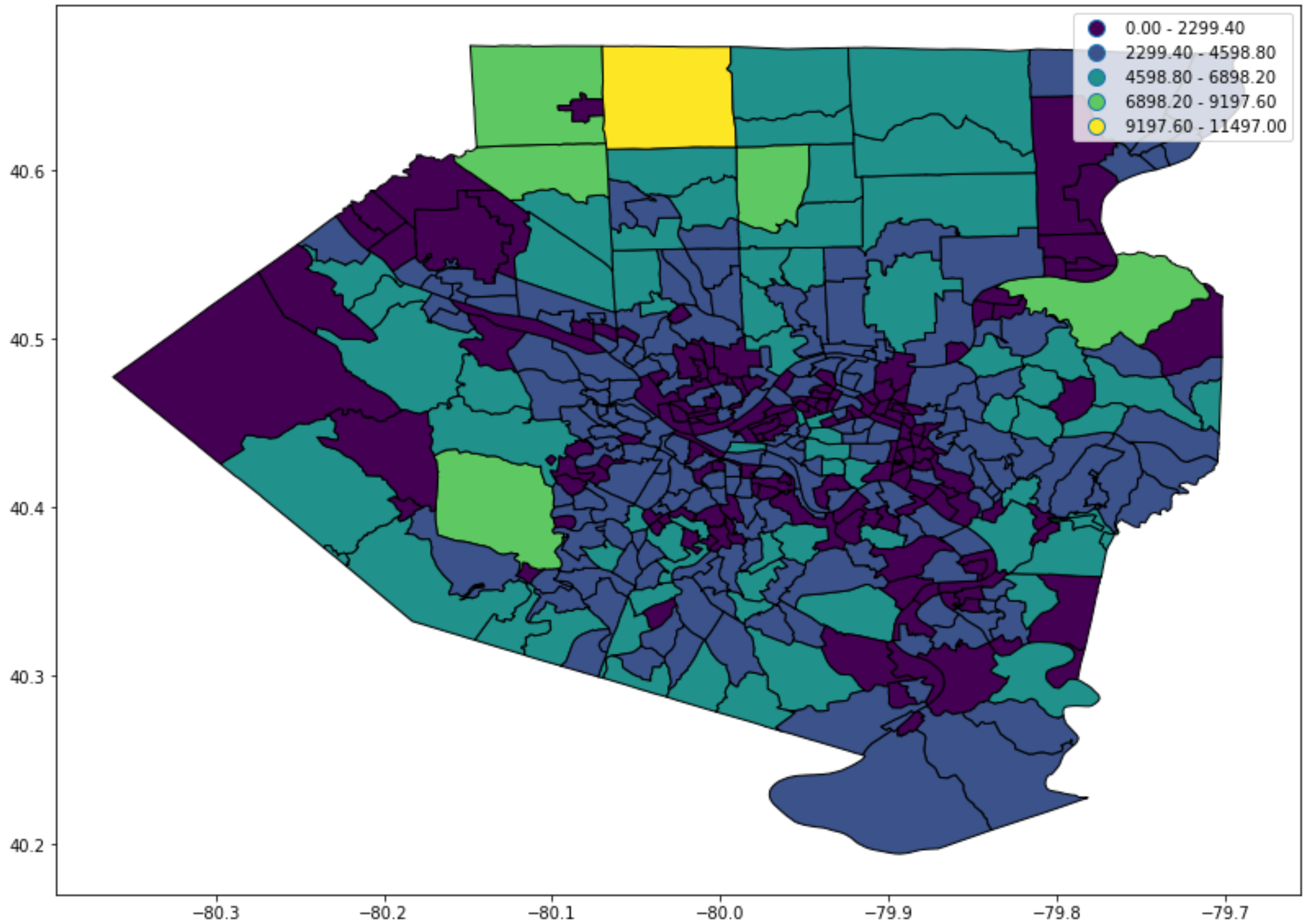
In [24]:

```
final.plot(column=('DP0010001'), edgecolor='black', scheme='equal_interval', fig
size=(16,10),legend=True)
```

```
/anaconda3/lib/python3.6/site-packages/pysal/__init__.py:65: Visible
DeprecationWarning: PySAL's API will be changed on 2018-12-31. The l
ast release made with this API is version 1.14.4. A preview of the n
ext API version is provided in the `pysal` 2.0 prelease candidate. T
he API changes and a guide on how to change imports is provided at h
ttps://pysal.org/about
  ), VisibleDeprecationWarning)
/anaconda3/lib/python3.6/site-packages/pysal/esda/mapclassify.py:267
: RuntimeWarning: invalid value encountered in greater
  binIds += (x > l) * (x <= r) * k
/anaconda3/lib/python3.6/site-packages/pysal/esda/mapclassify.py:267
: RuntimeWarning: invalid value encountered in less_equal
  binIds += (x > l) * (x <= r) * k
/anaconda3/lib/python3.6/site-packages/numpy/lib/function_base.py:32
50: RuntimeWarning: Invalid value encountered in median
  r = func(a, **kwargs)
```

Out[24]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a2c8a4390>
```



# Transform of Gyms and Parks from cataogrical to numerical

Since parks and gyms have been spaially joined into dataset I can now run a function to convert the count of gyms and parks into a measurment of desnity. What that would look like is gyms/persons and parks/person.

Why am I using population? I am making an assumption that people that live closer to the facility are the ones that best represent the relationship in the tract to the facitliy.

Steps:

1. Select the column (either parks or gyms)
2. Add a new column called either gym_num or park_num and count the number of gyms per tract
3. Perform a (park per tract/ population) * 100 function for a new column called Park-Pop_Density
4. Perform a (gym per tract / population) * 100 calculation on a new column called Gym-Pop_Density

In [100]:

```
#joined_park.groupby('GEOID')['Name'].count()
temp_parks = DataFrame({'park_num': joined_park.groupby('GEOID')['Name'].count()
}).reset_index()
temp_parks
```

Out[100]:

| | GEOID | park_num |
|---|---|---|
| **0** | 42003010300 | 0 |
| **1** | 42003020100 | 8 |
| **...** | ... | ... |
| **388** | 42003981800 | 0 |
| **389** | 42003982200 | 0 |

390 rows × 2 columns

In [101]:

```
temp_parks['GEOID'] = temp_parks["GEOID"].astype(int)
final = final.merge(temp_parks, on = 'GEOID', how = 'left')
final
```

Out[101]:

| | GEOID | NAMELSAD10 | ... | Obesity | park_num |
|---|---|---|---|---|---|
| **0** | 42003560500 | Census Tract 5605 | ... | 0.224347 | 0 |
| **1** | 42003560400 | Census Tract 5604 | ... | 0.437485 | 0 |
| **...** | ... | ... | ... | ... | ... |
| **414** | 42003151500 | NaN | ... | 0.399853 | 0 |
| **415** | 42003453001 | NaN | ... | 0.205551 | 0 |

416 rows × 215 columns

In [102]:

```
temp_gyms = DataFrame({'gym_num' : joined_gyms.groupby('GEOID')['Gym_Name'].coun
t()}).reset_index()
temp_gyms
```

Out[102]:

| | GEOID | gym_num |
|---|---|---|
| **0** | 42003010300 | 0 |
| **1** | 42003020100 | 4 |
| **...** | ... | ... |
| **388** | 42003981800 | 0 |
| **389** | 42003982200 | 0 |

390 rows × 2 columns

In [103]:

```
temp_gyms['GEOID'] = temp_gyms["GEOID"].astype(int)
final = final.merge(temp_gyms, on = 'GEOID', how = 'left')
final
```

Out[103]:

| | GEOID | NAMELSAD10 | ... | park_num | gym_num |
|---|---|---|---|---|---|
| **0** | 42003560500 | Census Tract 5605 | ... | 0 | 0 |
| **1** | 42003560400 | Census Tract 5604 | ... | 0 | 0 |
| **...** | ... | ... | ... | ... | ... |
| **414** | 42003151500 | NaN | ... | 0 | 0 |
| **415** | 42003453001 | NaN | ... | 0 | 0 |

416 rows × 216 columns

## Variable Creation:

In [104]:

```
final['Park-Pop_Density'] = ''
final['Park-Pop_Density'] = (final['park_num']/final['DP0010001'])*100
```

In [105]:

```
final['Gym-Pop_Density'] = ''
final['Gym-Pop_Density'] = (final['gym_num']/final['DP0010001'])*100
#final.to_excel('Final_polish.xlsx')
```

In [106]:

```
final.dropna()
```

Out[106]:

| | GEOID | NAMELSAD10 | ... | Park-Pop_Density | Gym-Pop_Density |
|---|---|---|---|---|---|
| **17** | 42003562000 | Census Tract 5620 | ... | 0.0 | 0.0 |
| **20** | 42003561600 | Census Tract 5616 | ... | 0.0 | 0.0 |
| **...** | ... | ... | ... | ... | ... |
| **410** | 42003561900 | Census Tract 5619 | ... | 0.0 | 0.0 |
| **411** | 42003563200 | Census Tract 5632 | ... | 0.0 | 0.0 |

29 rows × 218 columns

In [107]:

```
final = final.reset_index()
final = final.rename(columns={'index':'Key'})
```

# Final GeoDataFrame Polishing

Using the drop function apart of the Pandas module. Appròminately 200 columns will be drops to clean the final GeoDataFrame.

In [108]:

```
clean_data = pd.read_excel('Final_polish.xlsx')
```

In [109]:

```
clean_data = clean_data.reset_index()
clean_data = clean_data.rename(columns={'index':'Key'})
```

In [110]:

```
clean_data
```

Out[110]:

| | Key | GEOID | ... | Gym-Pop_Density | AWATER00 |
|---|---|---|---|---|---|
| **0** | 0 | 42003560500 | ... | 0.0 | 0 |
| **1** | 1 | 42003560400 | ... | 0.0 | 0 |
| **...** | ... | ... | ... | ... | ... |
| **414** | 414 | 42003151500 | ... | NaN | 0 |
| **415** | 415 | 42003453001 | ... | NaN | 0 |

416 rows × 15 columns

In [111]:

```
clean_data = clean_data.drop(columns=['gym_num_y','geometry_x'])
clean_data = clean_data.rename(columns={'gym_num_x':'gym_num','geometry_y':'geom
etry'})
clean_data.to_excel('clean_data.xlsx')
```

# Exploratory Data Analysis

Dependent Variable: Obesity

Independent Variables: Gyms, Parks, Municipalitiy and Water_Area

Purpose: to ensure that the data will meet the assumptions for least ordered squares regression.

1. Data is normally distributed.
2. There is a linear relationship.
3. Residuals are homosecatic.
4. Data is not catagorical
5. Data is not spatially autocorrelated
6. Multicollinearity is not present
7. Identification of outliers

Plan of attack:

1. Histograms, boxplots, and Wilcoxcon form scistats
2. Scatter plots will address this
3. Running a test regression and plotting the residuals
4. Data dimensions will explain why there are not catagorical
5. Moran's I will be used to check
6. Person's r, and if I can figure out if there is any VIF output for scipy or statmat or sci-kitlearn
7. This will be solved with the Histograms, and boxplots

***EDA method***

I made this method to put all the tools for the EDA in one place I have more information above it to help explain what is going on.

In [112]:

```
# the x parameter is the input column from a pandas dataframe ie. df['colmun']
# This method requires pandas, numpy, scipy stats, scipy.stats kurtosis, scipy.s
tats skew, matplotlib .
def EDA(x,df,column_name_string):
    #tables and print lines
    print('Kurtosis: ',kurtosis(x))
    print('Skewness: ',skew(x))
    one = stats.wilcoxon(x)
    two = x.describe()

    fig = plt.figure(1)
    #graphs
    gridsize = (2,2)
    plt.subplot2grid(gridsize,(0,0))
    #scatter plot
    plt.scatter(final['Key'], x)

    plt.subplot2grid(gridsize,(0,1))
    #histogram
    plt.hist([column_name_string], bins=5)

    plt.subplot2grid(gridsize,(1,0))
    #boxplot
    df.boxplot([column_name_string],grid=False)

    plt.subplot2grid(gridsize,(1,1))
    #Probability Plot
    stats.probplot(x, plot=plt)
    return one, two
```

**Obesity**

Data Dimensions:

1. Indvidiual vs Spaitally aggregated
2. Sample vs population
3. Implicitly spatial vs Explicitly spatial
4. Discrete vs continuous
5. Quantitative vs qualitative
6. Nominal vs ordinal vs interval vs ratio

Dimensions classifications and explainations:

1. Obesity is a measrument of **individuals** whom fall under the catagory of a BMI index rating of 30 or higher. In this circumstance obesity has been aggrgated to the tract scale, making the individuals undistinguishable from the rest of the tract. Therefore this data is of a **spatially aggregated** nature.
2. Since obesity rates are collected by the CDC, this data is based on a **sample** of the population because they measure at the county level. As referenced in the metadata, this is a staticial desmination of population of each cencus tract with obesity rates measured from the county level.
3. This data is **Explicitly spatial**, because of the rates of obesity being directly tied to each tract
4. Obesity is a **continous** form of measrument because it can be measured with even finer percision thn what is presented in this study.
5. This is a **Quantitative** measurment, because of the numerical nature of the data.
6. This data is **Ratio** level. There are no negatives and any tract with a rate of zero obesity has meaning. Zero is not arbitrary.

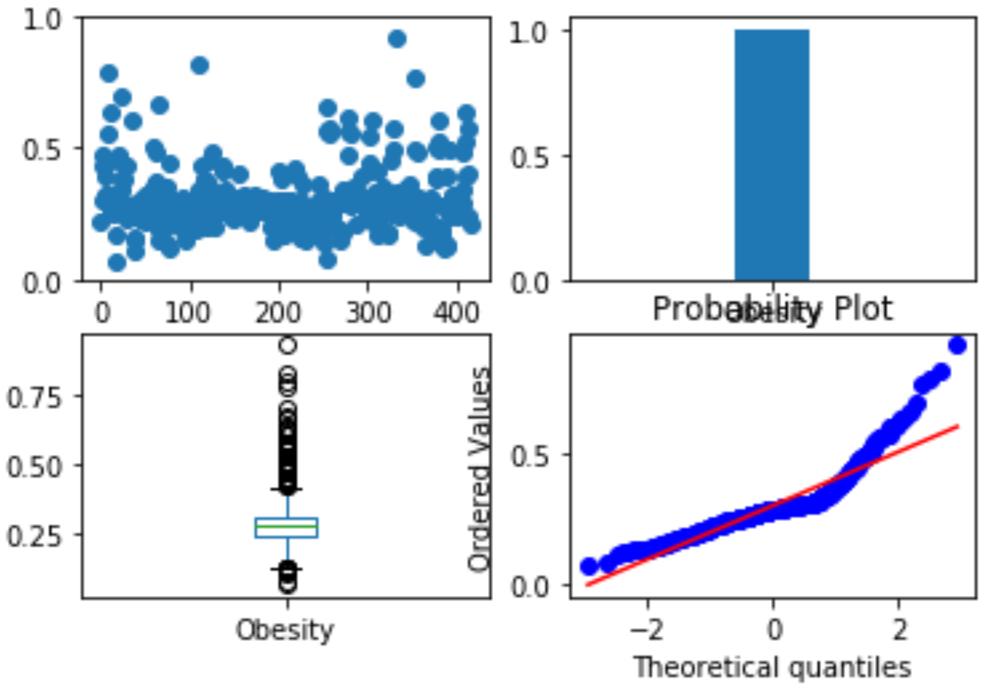
In [42]:

```
EDA(clean_data['Obesity'],clean_data,'Obesity')
```

```
Kurtosis:  5.1868767219770575
Skewness:  1.8353889055046517
```

Out[42]:

```
(WilcoxonResult(statistic=0.0, pvalue=6.636685802753196e-70),
 count    416.000000
 mean       0.299506
 std        0.113258
 min        0.070353
 25%        0.240699
 50%        0.282818
 75%        0.313238
 max        0.922932
 Name: Obesity, dtype: float64)
```




**Park_num**

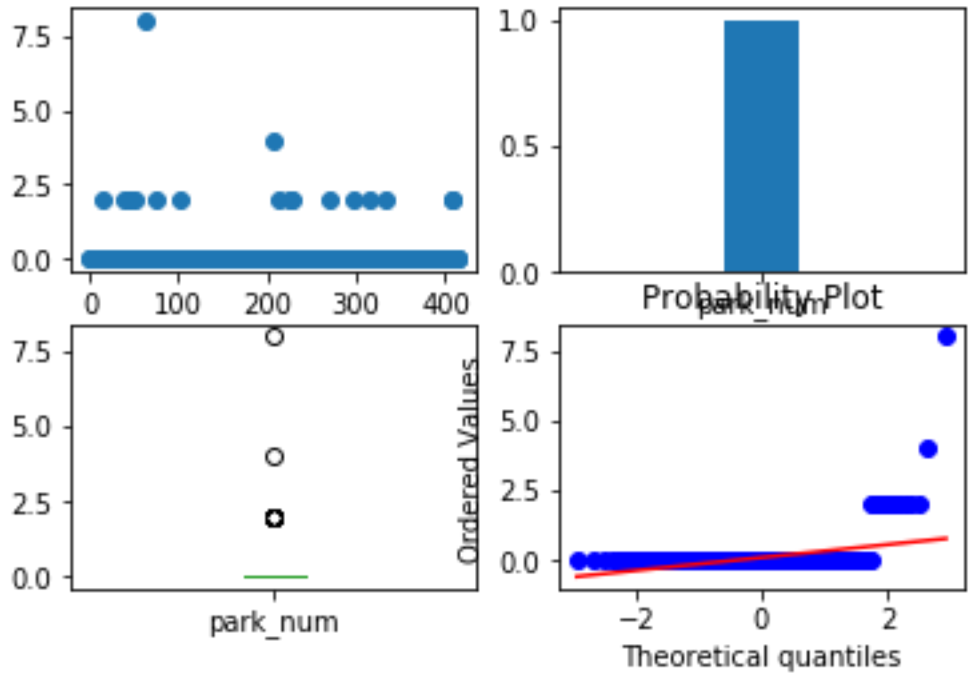
In [43]:

```
EDA(clean_data['park_num'],clean_data,'park_num')
```

```
Kurtosis:  94.49247702342261
Skewness:  8.438569311499537
```

Out[43]:

```
(WilcoxonResult(statistic=0.0, pvalue=8.017534109495179e-05),
 count    416.000000
 mean       0.100962
 std        0.571954
 min        0.000000
 25%        0.000000
 50%        0.000000
 75%        0.000000
 max        8.000000
 Name: park_num, dtype: float64)
```




**Gym_num**

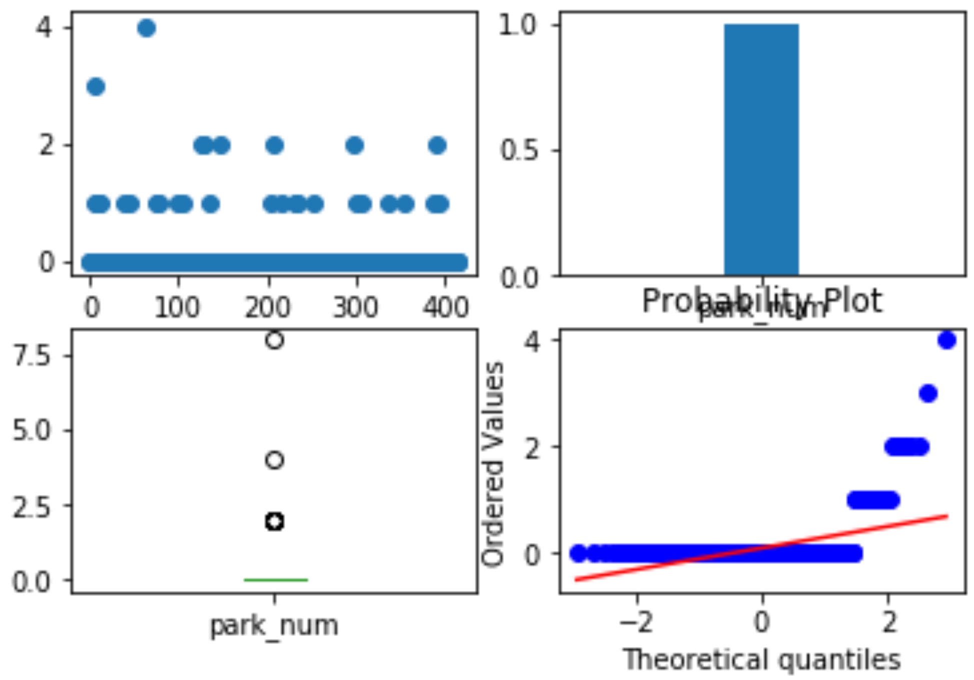
In [59]:

```
EDA(clean_data['gym_num'],clean_data,'park_num')
```

```
Kurtosis:  34.66511087074095
Skewness:  5.3276616900477745
```

Out[59]:

```
(WilcoxonResult(statistic=0.0, pvalue=7.986438914528027e-07),
 count    416.000000
 mean       0.096154
 std        0.399258
 min        0.000000
 25%        0.000000
 50%        0.000000
 75%        0.000000
 max        4.000000
 Name: gym_num, dtype: float64)
```




Same as before I know there are things that will need to be added to this matplotlib.pyplot.boxplot

**Water Area**

Data Dimensions:

1. Indvidiual vs Spaitally aggregated
2. Sample vs population
3. Implicitly spatial vs Explicitly spatial
4. Discrete vs continuous
5. Quantitative vs qualitative
6. Nominal vs ordinal vs interval vs ratio

1. Water area is **Spatially Aggregated**, because there are no polygons that show the boundry of the water.
2. Since this measurement was collected by the Cencus it will be treated as **population** data.
3. **Explicily spatial** is how this data would be described because of in certain tracts there is bodies of water and in some tracts there are not. Location does matter.
4. Water Area is a **continuous** measuresment because the level of percision can be increased than what is presented.
5. This varaible is a **quantitative** measurment because of the numerical nature of the data
6. **Ratio** is what type of data this variable fall under, because there are no negative values and having a zero amount of water area has significant meaning.

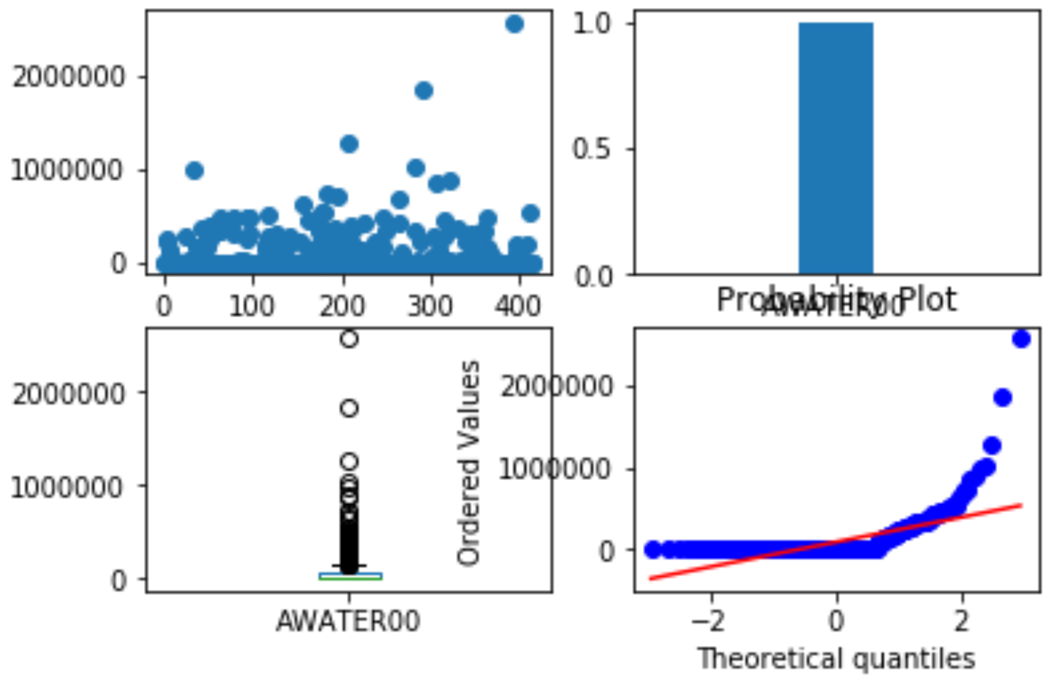
In [44]:

```
EDA(clean_data['AWATER00'],clean_data,'AWATER00')
```

```
Kurtosis:  43.09164065856403
Skewness:  5.388116807366086
```

Out[44]:

```
(WilcoxonResult(statistic=0.0, pvalue=1.3894515759554441e-22),
 count    4.160000e+02
 mean     8.994591e+04
 std      2.284063e+05
 min      0.000000e+00
 25%      0.000000e+00
 50%      0.000000e+00
 75%      6.474225e+04
 max      2.562012e+06
 Name: AWATER00, dtype: float64)
```




# Pearson's R Matrix

In [113]:

```
corr = clean_data.corr()
corr.style.background_gradient(cmap='coolwarm')
```

Out[113]:

| | Key | GEOID | AWATER10 | DP0010001 | Tract_Num | ALAND00 |
|---|---|---|---|---|---|---|
| **Key** | 1 | -0.215157 | 0.108059 | -0.0572447 | -0.24294 | 0.0671438 |
| **GEOID** | -0.215157 | 1 | 0.173785 | 0.097974 | 0.688841 | 0.136606 |
| **AWATER10** | 0.108059 | 0.173785 | 1 | -0.0121916 | 0.110891 | 0.155647 |
| **DP0010001** | -0.0572447 | 0.097974 | -0.0121916 | 1 | 0.256372 | 0.415731 |
| **Tract_Num** | -0.24294 | 0.688841 | 0.110891 | 0.256372 | 1 | 0.232849 |
| **ALAND00** | 0.0671438 | 0.136606 | 0.155647 | 0.415731 | 0.232849 | 1 |
| **Obesity** | 0.0184568 | -0.0950488 | -0.0146974 | -0.378094 | -0.220418 | -0.166292 |
| **park_num** | -0.0464983 | -0.0998771 | 0.0432174 | 0.0672327 | -0.147105 | 0.00340068 |
| **gym_num** | -0.0813185 | -0.156852 | -0.0226501 | 0.0606826 | -0.172603 | 0.0205497 |
| **Park-Pop_Density** | -0.055269 | 0.162372 | 0.0617646 | -0.0966529 | -0.0610673 | -0.0239519 |
| **Gym-Pop_Density** | -0.0747728 | 0.220886 | 0.0911616 | -0.135876 | -0.0777234 | -0.0306929 |
| **AWATER00** | 0.0214804 | 0.095167 | 0.838042 | -0.0278315 | 0.0791038 | 0.196182 |

# Moran's I spatial auto-correlation

Source

@online{darribas_gds15, author = {Dani Arribas-Bel}, title = {Geographic Data Science'15}, year = 2016, url = {http://darribas.org/gds15} (http://darribas.org/gds15}), urldate = {2016-02-19} doi = {10.5281/zenodo.46313} }

In [51]:

```
import pysal as ps
```

In [53]:

```
w_queen = ps.queen_from_shapefile(final, idVariable='Obesity')
w_queen
```

```
---------------------------------------------------------------------
-------
TypeError                                 Traceback (most recent cal
l last)
<ipython-input-53-415f03d7235c> in <module>
----> 1 w_queen = ps.queen_from_shapefile(final, idVariable='Obesity
')
      2 w_queen

/anaconda3/lib/python3.6/site-packages/pysal/weights/user.py in quee
n_from_shapefile(shapefile, idVariable, sparse)
     65     """
     66 
---> 67     w = Queen.from_shapefile(shapefile, idVariable=idVariabl
e)
     68     if sparse:
     69         w = pysal.weights.WSP(w.sparse, id_order=w.id_order)

/anaconda3/lib/python3.6/site-packages/pysal/weights/Contiguity.py
in from_shapefile(cls, filepath, idVariable, full, **kwargs)
    246         sparse = kwargs.pop('sparse', False)
    247         if idVariable is not None:
--> 248             ids = get_ids(filepath, idVariable)
    249             id_order = ids
    250         else:

/anaconda3/lib/python3.6/site-packages/pysal/weights/util.py in get_
ids(shapefile, idVariable)
   1017 
   1018     try:
-> 1019         dbname = os.path.splitext(shapefile)[0] + '.dbf'
   1020         db = pysal.open(dbname)
   1021         var = db.by_col[idVariable]

/anaconda3/lib/python3.6/posixpath.py in splitext(p)
    120 
    121 def splitext(p):
--> 122     p = os.fspath(p)
    123     if isinstance(p, bytes):
    124         sep = b'/'

TypeError: expected str, bytes or os.PathLike object, not GeoDataFra
me
```

# Multivariate Regression with Sci-KitLearn

Using Sci-Kit learn's module for Multivariate Regression Output table made with Statsmodel

- The indpenedent variables are stored in the X variable.
- Y is the variable in which the dependent variable is stored in.
- Using the preprocessing function in the Sci-KitLearn Module the training data is split from the test data
- Linear Regression from Sci-KitLearn is imported and then saved to regressor.
- Regessor is then called to fit the training data for the independent and dependent variables.
- Then the regession is run.

#### Methodology

In Rosenberger 2005, their regression involved nesting hierarchy. Where I will differentiate is with using a least ordered squares and analyze the predictive nature of this model. One of the reasons why I can avoid the nested hierarchy regression used previously is I am not using county level data.

In [114]:

```
final = final.fillna(0)
```

In [115]:

```
final
```

Out[115]:

| | Key | GEOID | ... | Park-Pop_Density | Gym-Pop_Density |
|---|---|---|---|---|---|
| **0** | 0 | 42003560500 | ... | 0.0 | 0.0 |
| **1** | 1 | 42003560400 | ... | 0.0 | 0.0 |
| **...** | ... | ... | ... | ... | ... |
| **414** | 414 | 42003151500 | ... | 0.0 | 0.0 |
| **415** | 415 | 42003453001 | ... | 0.0 | 0.0 |

416 rows × 219 columns

In [116]:

```
X = final.iloc[:,215:216].values
y = final.iloc[:, 214].values
```

In [117]:

```
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, rando
m_state = 0)
```

In [118]:

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Out[118]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normaliz
e=False)
```

In [119]:

```
# Predicting the Test set results
y_pred = regressor.predict(X_test)
```

source for score function: https://towardsdatascience.com/simple-and-multiple-linear-regression-in-python-c928425168f9 (https://towardsdatascience.com/simple-and-multiple-linear-regression-in-python-c928425168f9)

In [120]:

```
regressor.score(X_test,y_test)
```

Out[120]:

```
0.00017138492145007955
```

In [121]:

```
import statsmodels.api as sm
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)
```

Out[121]:

```
0.00017138492145007955
```

In [122]:

```
import statsmodels
X = final.iloc[:,214:21].values
obesity = final.iloc[:, 213].values

X2 = sm.add_constant(X)
regress2 = sm.OLS(y, X2)
regression = regress2.fit()
print(regression.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                      y   R-squared:
0.000
Model:                            OLS   Adj. R-squared:
0.000
Method:                 Least Squares   F-statistic:
nan
Date:                Fri, 12 Apr 2019   Prob (F-statistic):
nan
Time:                        15:27:16   Log-Likelihood:
316.30
No. Observations:                 416   AIC:
-630.6
Df Residuals:                     415   BIC:
-626.6
Df Model:                           0
Covariance Type:            nonrobust
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-
const          0.2995      0.006     53.936      0.000       0.289
0.310
==============================================================================
=
Omnibus:                      172.360   Durbin-Watson:
1.162
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
699.891
Skew:                           1.835   Prob(JB):
1.05e-152
Kurtosis:                       8.187   Cond. No.
1.00
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.
```

```
/anaconda3/lib/python3.6/site-packages/statsmodels/regression/linear
_model.py:1554: RuntimeWarning: invalid value encountered in double_
scalars
  return self.ess/self.df_model
```

## Disscussion

Based on the R-squared value of 0.006 this model is a good predictor of obesity rates per tract based on the number of gyms and parks within each tract. The F-statistic is below 1, indicating that this model is not significant.

# Conclusion

This model need more data to be evaluated effectively.

**Works Cited**


Humpel, Nancy, Neville Owen, and Eva Leslie. "Environmental factors associated with adults' participation in physical activity: a review." American journal of preventive medicine 22, no. 3 (2002): 188-199.

Kahn, Emily B., Leigh T. Ramsey, Ross C. Brownson, Gregory W. Heath, Elizabeth H. Howze, Kenneth E. Powell, Elaine J. Stone, Mummy W. Rajab, and Phaedra Corso. "The effectiveness of interventions to increase physical activity: a systematic review." American journal of preventive medicine 22, no. 4 (2002): 73-107.

Ding, Ding, and Klaus Gebel. "Built environment, physical activity, and obesity: what have we learned from reviewing the literature?." Health & place 18, no. 1 (2012): 100-105.

Janiszewski, Peter M., and Robert Ross. "Physical activity in the treatment of obesity: beyond body weight reduction." Applied Physiology, Nutrition, and Metabolism 32, no. 3 (2007): 512-522.

Eckel, Robert H. "Nonsurgical management of obesity in adults." New England Journal of Medicine 358, no. 18 (2008): 1941-1950.

Hill, James O., and Holly R. Wyatt. "Role of physical activity in preventing and treating obesity." Journal of Applied Physiology 99, no. 2 (2005): 765-770.

Courcoulas, Anita P., Steven H. Belle, Rebecca H. Neiberg, Sheila K. Pierson, Jessie K. Eagleton, Melissa A. Kalarchian, James P. DeLany, Wei Lang, and John M. Jakicic. "Three-year outcomes of bariatric surgery vs lifestyle intervention for type 2 diabetes mellitus treatment: a randomized clinical trial." JAMA surgery 150, no. 10 (2015): 931-940.

Gloy, Viktoria L., Matthias Briel, Deepak L. Bhatt, Sangeeta R. Kashyap, Philip R. Schauer, Geltrude Mingrone, Heiner C. Bucher, and Alain J. Nordmann. "Bariatric surgery versus non-surgical treatment for obesity: a systematic review and meta-analysis of randomised controlled trials." Bmj 347 (2013): f5934.

Gordon-Larsen, Penny, Melissa C. Nelson, Phil Page, and Barry M. Popkin. "Inequality in the built environment underlies key health disparities in physical activity and obesity." Pediatrics 117, no. 2 (2006): 417-424.

Rosenberger, Randall S., Yoav Sneh, Tim T. Phipps, and Rachel Gurvitch. "A spatial analysis of linkages between health care expenditures, physical inactivity, obesity and recreation supply." Journal of Leisure Research 37, no. 2 (2005): 216-235.

Brownson, Ross C., Christine M. Hoehner, Kristen Day, Ann Forsyth, and James F. Sallis. "Measuring the built environment for physical activity: state of the science." American journal of preventive medicine 36, no. 4 (2009): S99-S123.

McLaren, Lindsay. "Socioeconomic status and obesity." Epidemiologic reviews 29, no. 1 (2007): 29-48. Swinburn, Boyd, Garry Egger, and Fezeela Raza. "Dissecting obesogenic environments: the development and application of a framework for identifying and prioritizing environmental interventions for obesity." Preventive medicine 29, no. 6 (1999): 563-570.

Finkelstein, Eric A., and Kiersten L. Strombotne. "The economics of obesity." The American journal of clinical nutrition 91, no. 5 (2010): 1520S-1524S.

Pitts, Stephanie B. Jilcott, Michael B. Edwards, Justin B. Moore, Kindal A. Shores, Katrina Drowatzky DuBose, and David McGranahan. "Obesity is inversely associated with natural amenities and recreation facilities per capita." Journal of physical activity and health 10, no. 7 (2013): 1032-1038.

Wolch, Jennifer, Michael Jerrett, Kim Reynolds, Rob McConnell, Roger Chang, Nicholas Dahmann, Kirby Brady, Frank Gilliland, Jason G. Su, and Kiros Berhane. "Childhood obesity and proximity to urban parks and recreational resources: a longitudinal cohort study." Health & place 17, no. 1 (2011): 207-214.

Abercrombie, Lauren C., James F. Sallis, Terry L. Conway, Lawrence D. Frank, Brian E. Saelens, and James E. Chapman. "Income and racial disparities in access to public parks and private recreation facilities." American journal of preventive medicine 34, no. 1 (2008): 9-15.

Sturm, Roland. "Increases in morbid obesity in the USA: 2000–2005." Public health 121, no. 7 (2007): 492-496. Zhang, Qi, and Youfa Wang. "Trends in the association between obesity and socioeconomic status in US adults: 1971 to 2000." Obesity research 12, no. 10 (2004): 1622-1632.

Blanck, Heidi M., Diana Allen, Zarnaaz Bashir, Nina Gordon, Alyson Goodman, Dee Merriam, and Candace Rutt. "Let's go to the park today: The role of parks in obesity prevention and improving the public's health." Childhood Obesity (Formerly Obesity and Weight Management) 8, no. 5 (2012): 423-428.

Gies, Erica. "The health benefits of parks." The Trust for Public Land (2006): 1-24.

Ogden, Cynthia L., Margaret D. Carroll, Brian K. Kit, and Katherine M. Flegal. "Prevalence of childhood and adult obesity in the United States, 2011-2012." Jama 311, no. 8 (2014): 806-814.

Center for disease control and prevention. 2015. Body mass index. Atlanta. https://www.cdc.gov/healthyweight/assessing/bmi/index.html (https://www.cdc.gov/healthyweight/assessing/bmi/index.html) (last accessed 21 March 2019).

Ley, Sylvia H., Osama Hamdy, Viswanathan Mohan, and Frank B. Hu. "Prevention and management of type 2 diabetes: dietary components and nutritional strategies." The Lancet383, no. 9933 (2014): 1999-2007.

Evans, Tim, Steven Cummins, and Tim Brown. "Neighbourhood deprivation and the cost of accessing gyms and fitness centres: National study in Wales." Health & place24 (2013): 16-19.