Green Space and Impacts on Health

Kenny Siegler, Harrison Riebow, Cody Allison

Goals

We seek to analyze what impact the prevalence of green space has on health metrics.

What is Green Space?

Land planning for plants life, water space, and other natural elements for use in social, recreational, and public spaces

Our analysis includes green space figures
From the UN calculated in Square Green Meters
Per Million people

https://www.forestparkforever.org/playscape

What Health Indicators are in scope?

Our health data comes from a 2022 release of information from the PLACES CDC initiative.

PLACES has measures for over 29 health indicators in all 50 states.

Due to data availability we are focusing on the following:

-Percent of Population w/ High Blood Pressure Among Adults 18+

-Percent of Population w/ High Cholesterol Among Adults 18+

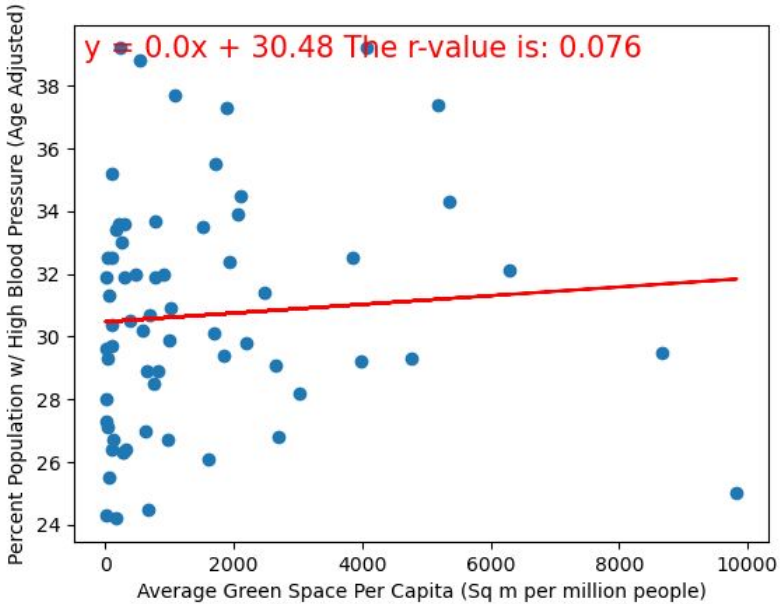
Health Data Measurements

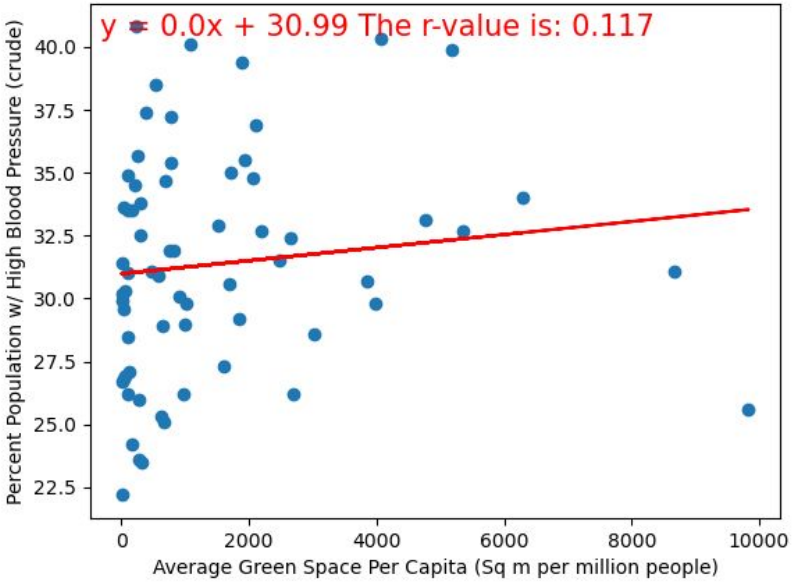
CDC includes two data types for each Data Measurement

- Crude Prevalence
  - All Data with no weighting/scaling
- Age Adjusted Prevalence
  - All Data with weighting/scaling to account for age being a common factor in these health measurements

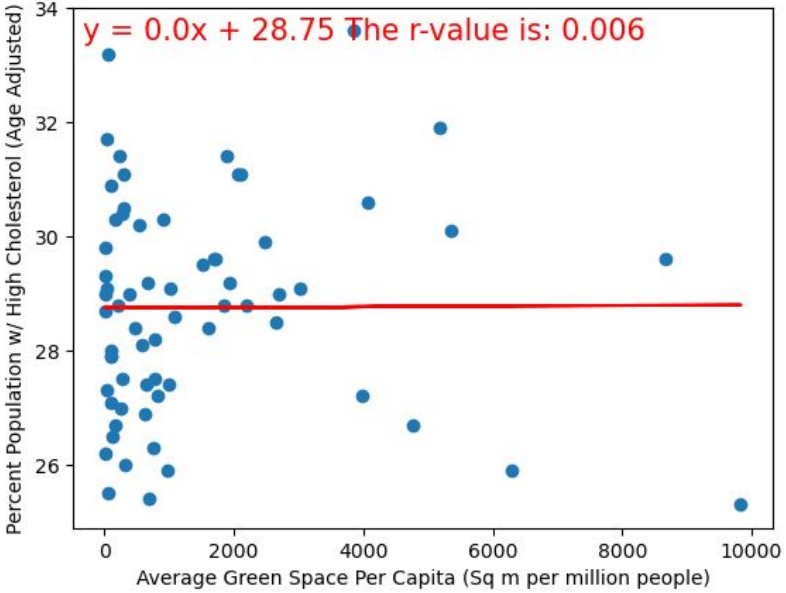
Live Demo

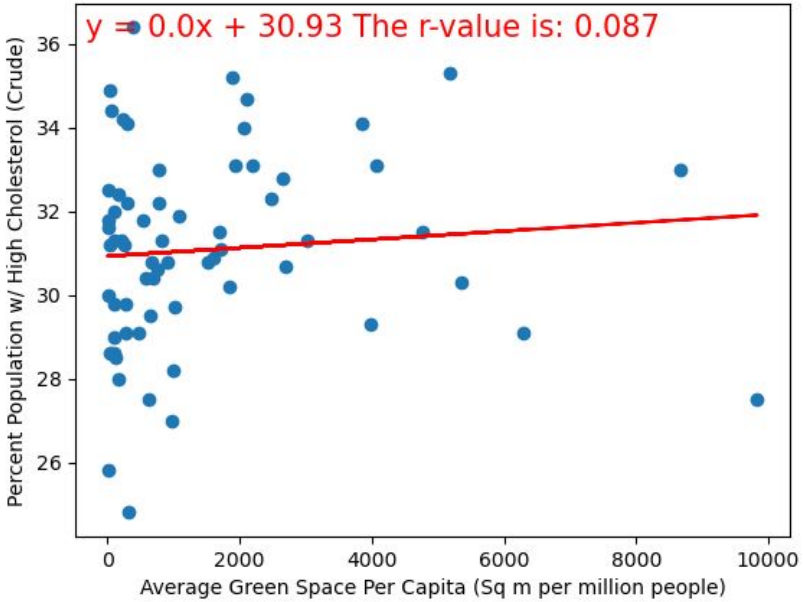
Our Findings





Our Findings





How Did We Get Here?

Data Source

Clean Up

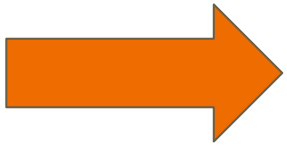
Data Storage

Flask API

PyMongo

How Did We Get Here? (cont.)

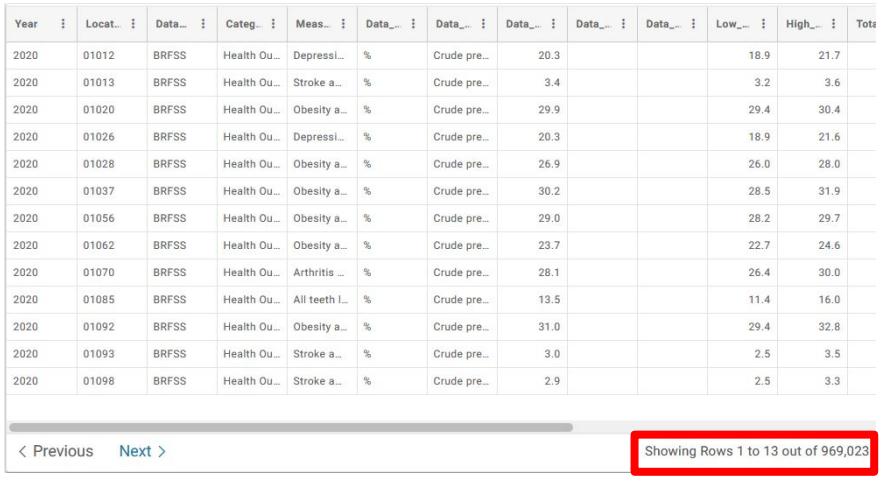
HTML
Javascript
CSS



Frontend
Visualization

Data Source & Clean Up

CDC PLACES data:



Year	Locat...	Data...	Categ...	Meas...	Data_...	Data_...	Data_...	Data_...	Data_...	Low_...	High_...	Tota
2020	01012	BRFSS	Health Ou...	Depressi...	%	Crude pre...	20.3			18.9	21.7	
2020	01013	BRFSS	Health Ou...	Stroke a...	%	Crude pre...	3.4			3.2	3.6	
2020	01020	BRFSS	Health Ou...	Obesity a...	%	Crude pre...	29.9			29.4	30.4	
2020	01026	BRFSS	Health Ou...	Depressi...	%	Crude pre...	20.3			18.9	21.6	
2020	01028	BRFSS	Health Ou...	Obesity a...	%	Crude pre...	26.9			26.0	28.0	
2020	01037	BRFSS	Health Ou...	Obesity a...	%	Crude pre...	30.2			28.5	31.9	
2020	01056	BRFSS	Health Ou...	Obesity a...	%	Crude pre...	29.0			28.2	29.7	
2020	01062	BRFSS	Health Ou...	Obesity a...	%	Crude pre...	23.7			22.7	24.6	
2020	01070	BRFSS	Health Ou...	Arthritis ...	%	Crude pre...	28.1			26.4	30.0	
2020	01085	BRFSS	Health Ou...	All teeth l...	%	Crude pre...	13.5			11.4	16.0	
2020	01092	BRFSS	Health Ou...	Obesity a...	%	Crude pre...	31.0			29.4	32.8	
2020	01093	BRFSS	Health Ou...	Stroke a...	%	Crude pre...	3.0			2.5	3.5	
2020	01098	BRFSS	Health Ou...	Stroke a...	%	Crude pre...	2.9			2.5	3.3	

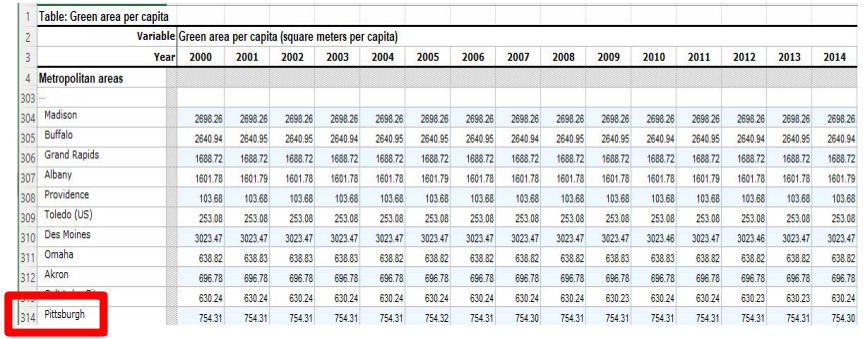
< Previous Next >

Showing Rows 1 to 13 out of 969,023

29 Health indicators x 2 Years of data x 2 types of data measurement

Up to 116 rows per county

UN Green Space Data:



1	Table: Green area per capita															
2	Variable	Green area per capita (square meters per capita)														
3	Year	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
4	Metropolitan areas															
303	...															
304	Madison	2698.26	2698.26	2698.26	2698.26	2698.26	2698.26	2698.26	2698.26	2698.26	2698.26	2698.26	2698.26	2698.26	2698.26	2698.26
305	Buffalo	2640.94	2640.95	2640.95	2640.94	2640.95	2640.95	2640.95	2640.94	2640.95	2640.95	2640.94	2640.95	2640.94	2640.95	2640.94
306	Grand Rapids	1688.72	1688.72	1688.72	1688.72	1688.72	1688.72	1688.72	1688.72	1688.72	1688.72	1688.72	1688.72	1688.72	1688.72	1688.72
307	Albany	1601.78	1601.79	1601.78	1601.78	1601.78	1601.79	1601.78	1601.78	1601.78	1601.78	1601.78	1601.79	1601.78	1601.78	1601.79
308	Providence	103.68	103.68	103.68	103.68	103.68	103.68	103.68	103.68	103.68	103.68	103.68	103.68	103.68	103.68	103.68
309	Toledo (US)	253.08	253.08	253.08	253.08	253.08	253.08	253.08	253.08	253.08	253.08	253.08	253.08	253.08	253.08	253.08
310	Des Moines	3023.47	3023.47	3023.47	3023.47	3023.47	3023.47	3023.47	3023.47	3023.47	3023.47	3023.46	3023.47	3023.46	3023.47	3023.47
311	Omaha	638.82	638.83	638.83	638.83	638.82	638.82	638.82	638.82	638.82	638.83	638.83	638.82	638.82	638.82	638.82
312	Akron	696.78	696.78	696.78	696.78	696.78	696.78	696.78	696.78	696.78	696.78	696.78	696.78	696.78	696.78	696.78
313	[illegible]	630.24	630.24	630.24	630.24	630.24	630.24	630.24	630.24	630.24	630.23	630.24	630.24	630.23	630.23	630.24
314	Pittsburgh	754.31	754.31	754.31	754.31	754.31	754.32	754.31	754.30	754.31	754.31	754.31	754.31	754.31	754.31	754.30

70 US Cities - 4 for data integrity

Final list of 66 US Cities

Data Source & Clean Up

CDC PLACES Data:

-Intended to look at “behavioral health”

-2019 data only. Insufficient data to tell a compelling story once COVID enters the picture

-2020 start of many new health measures collected by CDC PLACES

-Blood Pressure and Cholesterol remaining data points

-Clean up Coordinates

-White Spaces

```
## Create an empty list
lats = []
lons = []

## Iterate through the rows
for i in new_df['Geolocation']:
    ## Set each item to a string
    s = str(i)
    # get lon-lat from "POINT (lon lat)"
    loc = s[7:-1]
    ## Turn the returned string into a list
    l = loc.split(' ')
    ## Reverse the list

    try:
        lats.append(l[1])

    except:
        lats.append('')

    try:
        lons.append(l[0])

    except:
        lats.append('')
```

UN Green Space Data:

-14 years of Green Space data

-Latest figure in 2014

-Data Largely static (65 out of 66 cities had less than 1% change over 14 years)

-Manually added county for each city to insure clean and accurate merge

-White Space clean up

Data Source & Clean Up

```
In [13]: final_df = pd.merge(final_df, gs_df, how='inner', left_on=['LocationName', 'StateAbbr'], right_on=['county', 'state'])
         final_df
```

Out[13]:

	Year	StateAbbr	LocationName	Data_Value_Type	Data_Value	TotalPopulation	DataValueTypeID	Short_Question_Text	Lat	Lon	city
0	2019	WI	Milwaukee	Age-adjusted prevalence	31.4	945016	AgeAdjPrv	High Blood Pressure	42.912643	-87.862312	Milwaukee
1	2019	WI	Milwaukee	Crude prevalence	31.5	945016	CrdPrv	High Blood Pressure	42.912643	-87.862312	Milwaukee
2	2019	WI	Milwaukee	Age-adjusted prevalence	29.9	945016	AgeAdjPrv	High Cholesterol	42.912643	-87.862312	Milwaukee
3	2019	WI	Milwaukee	Crude prevalence	32.3	945016	CrdPrv	High Cholesterol	42.912643	-87.862312	Milwaukee
4	2019	WI	Dane	Crude prevalence	26.2	552536	CrdPrv	High Blood Pressure	43.067468	-89.417852	Madison
...	...	...	...	...	...	...	...	...	...	...	...
259	2019	AZ	Maricopa	Age-adjusted prevalence	29.3	4579081	AgeAdjPrv	High Blood Pressure	33.345176	-112.498930	Phoenix
260	2019	AL	Jefferson	Crude prevalence	40.1	655342	CrdPrv	High Blood Pressure	33.553444	-86.896536	Birmingham
261	2019	AL	Jefferson	Age-adjusted prevalence	28.6	655342	AgeAdjPrv	High Cholesterol	33.553444	-86.896536	Birmingham
262	2019	AL	Jefferson	Crude prevalence	31.9	655342	CrdPrv	High Cholesterol	33.553444	-86.896536	Birmingham
263	2019	AL	Jefferson	Age-adjusted prevalence	37.7	655342	AgeAdjPrv	High Blood Pressure	33.553444	-86.896536	Birmingham

264 rows × 14 columns

Data Storage

```
data_list = []
for row in group:
    empty_dict = {

    }
    county = row[0][0]
    empty_dict['county'] = county

    state = row[0][1]
    empty_dict['state'] = state

    data = row[1]
    parameters = [x for x in data['Short_Question_Text']]
    data_types = [x for x in data['DataValueTypeID']]

    data_values = [x for x in data['Data_Value']]

    population = data['TotalPopulation'].head(1)
    empty_dict['population'] = int(population.values[0])

    city = data['city'].head(1).values[0]
    empty_dict['city'] = city

    gpc = data['average_greenspace'].head(1).values[0]
    empty_dict['greenspace_per_capita'] = round(gpc,2)


    lat = data['Lat'].head(1)

    lon = data['Lon'].head(1)
    empty_dict['location'] = [float(lat.values[0]),float(lon.values[0])]

    ##print(county, state, parameters, data_types, measurements, population, lat, lon)
    measurements = {}
    for i in range(4):
        m = parameters[i].replace(' ','')
        dt = data_types[i]
        m=m+dt
        measurements[m] = data_values[i]
        empty_dict['measurements'] = measurements

    data_list.append(empty_dict)
```

```
Out[54]: [{'county': 'Albany',
           'state': 'NY',
           'population': 303654,
           'city': 'Albany',
           'greenspace_per_capita': 1601.78,
           'location': [42.5882401, -73.9740095],
           'measurements': {'HighBloodPressureAgeAdjPrv': 26.1,
            'HighBloodPressureCrdPrv': 27.3,
            'HighCholesterolAgeAdjPrv': 28.4,
            'HighCholesterolCrdPrv': 30.9}},
          {'county': 'Allegheny',
           'state': 'PA',
           'population': 1211358,
           'city': 'Pittsburgh',
           'greenspace_per_capita': 754.31,
           'location': [40.4697574, -79.9804515],
           'measurements': {'HighCholesterolCrdPrv': 30.6,
            'HighCholesterolAgeAdjPrv': 26.3,
            'HighBloodPressureAgeAdjPrv': 28.5,
```

```
import json

with open("../Resources/data.json", "w") as outfile:
    json.dump(data_list, outfile)
```

```
mongoimport --type json -d met -c healthandgreenspacedata --drop --jsonArray data.json
```

Flask API & PyMongo

```python
from pydoc import doc
from flask import Flask, jsonify
import json
from pymongo import MongoClient
from flask_cors import CORS


#mongodb
mongo = MongoClient(port=27017)
db = mongo.met
healthandgreenspacedata = db['healthandgreenspacedata']

#################################################
# Flask Setup
#################################################
app = Flask(__name__)
CORS(app)

@app.route("/api/health-green-data")
def health_green_data():
    cursor = healthandgreenspacedata.find({})
    documents ={}
    pymongo_cursor = db.healthandgreenspacedata.find({}, {'_id': False})
    all_data = list(pymongo_cursor)
    print(all_data)
    documents['data'] = all_data
    documents = jsonify(documents)
    return(documents)

if __name__ == "__main__":
    app.run(debug=True)
```


```
{
    "data": [
        {
            "city": "Albany",
            "county": "Albany",
            "greenspace_per_capita": 1601.78,
            "location": [
                42.5882401,
                -73.9740095
            ],
            "measurements": {
                "HighBloodPressureAgeAdjPrv": 26.1,
                "HighBloodPressureCrdPrv": 27.3,
                "HighCholesterolAgeAdjPrv": 28.4,
                "HighCholesterolCrdPrv": 30.9
            },
            "population": 303654,
            "state": "NY"
        },
        {
            "city": "Pittsburgh",
            "county": "Allegheny",
            "greenspace_per_capita": 754.31,
            "location": [
                40.4697574,
                -79.9804515
            ],
            "measurements": {
                "HighBloodPressureAgeAdjPrv": 28.5,
                "HighBloodPressureCrdPrv": 31.9,
                "HighCholesterolAgeAdjPrv": 26.3,
                "HighCholesterolCrdPrv": 30.6
            },
            "population": 1211358,
            "state": "PA"
        },
```

Javascript & Leaflet

-Data Loaded in from the Flask API

-Randomized Index populates initial Dataset

- Create Metadata
- Create Bar Chart
- Move Map to County Location

-On Menu Selection, New Data is populated

-Scatterplot also created w/ all data

JQuery

-Included JQuery Specific CSS, JS, HTML

Web Scrape Demo


```
from splinter import Browser
from bs4 import BeautifulSoup as soup
```

```
# Set up Splinter
browser = Browser('chrome')
```

```
# Visit the color picker to Scrape site
url = 'https://hihayk.github.io/scale/#4/6/50/30/-50/84/100/0/FFB100/255/180/0/white'
browser.visit(url)
```

```
# Parse the HTML
html = browser.html
html_soup = soup(html, 'html.parser')
```

```
# Scrape the heading
color
colorValues = html_soup.find_all('div', class_="ColorBlockCode")
print(colors)

for color in colors:
    print(color.text)
```

```
[<div class="ColorBlockCode sc-dnqmqq fwgsuC">#3C8000</div>, <div class="ColorBlockCode sc-dnqmqq fwgsuC">#6D9F00</div>, <div c
lass="ColorBlockCode sc-dnqmqq fwgsuC">#AABF00</div>, <div class="ColorBlockCode sc-dnqmqq fwgsuC">#DFC900</div>, <div class="C
olorBlockCode sc-dnqmqq fwgsuC">#FFB100</div>, <div class="ColorBlockCode sc-dnqmqq fwgsuC">#FF7B0D</div>, <div class="ColorBlo
ckCode sc-dnqmqq fwgsuC">#FF391A</div>, <div class="ColorBlockCode sc-dnqmqq fwgsuC">#FF2626</div>, <div class="ColorBlockCode
sc-dnqmqq fwgsuC">#FF3340</div>, <div class="ColorBlockCode sc-dnqmqq fwgsuC">#FF4096</div>, <div class="ColorBlockCode sc-dnqm
qq fwgsuC">#FF4DEF</div>]
#3C8000
#6D9F00
#AABF00
#DFC900
#FFB100
#FF7B0D
#FF391A
#FF2626
#FF3340
#FF4096
#FF4DEF
```

Questions?