General Analytics-Insights

March 24, 2020

[1]:
```
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:100% !important; height:100%}</style>"))
```

```
<IPython.core.display.HTML object>
```

[2]:
```
import json
import csv
import pandas as pd
import numpy as np

#libraries for plotting
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
```

1 Data preprocessing

Source1 : https://www.ers.usda.gov/data-products/food-access-research-atlas/download-the-data/

Filtering the data Set and considering the neighboring counties with the New hanover Counties to check status.

[3]:
```
data1 = "food_desert_clean.csv"
data2 = "TRACT_ZIP_122019.csv"


cleanmerge_df = pd.read_csv(data1)

counties = ["New Hanover", "Brunswick","Pender"]
cleanmerge_df.County.isin(counties)

cleanmerge_df = cleanmerge_df[cleanmerge_df.County.isin(counties)]
cleanmerge_df

lowaccess_income_df = cleanmerge_df[cleanmerge_df.State.isin(["North␣
 ↪Carolina"])]
```

```
print(lowaccess_income_df.isna().sum())
lowaccess_income_df['County'].value_counts()

lowaccess_income_df = lowaccess_income_df[(lowaccess_income_df.CensusTract !=␣
 ↪37129990100) & (lowaccess_income_df.Urban !=0)]


zipcodes_for_HN_df = pd.read_csv(data2)
```

```
CensusTract      0
State            0
County           0
Urban            0
POP2010          0
                ..
TractAIAN        0
TractOMultir     0
TractHispanic    0
TractHUNV        0
TractSNAP        0
Length: 147, dtype: int64
```

As the tract numbers were hard to understand, I decided to map the Censustract with the respective zip code for that area.

Data Source2 : https://www.huduser.gov/portal/datasets/usps_crosswalk.html#codebook

[4]:
```
#Merge the two datasets on "CensusTract" inorder to map zipcodes.

combined_tracts_df = pd.merge(lowaccess_income_df, zipcodes_for_HN_df,
                              how='outer', left_on='CensusTract',␣
 ↪right_on='CensusTract')
combined_tracts_df.head()
```

[4]:

	CensusTract	State	County	Urban	POP2010	OHU2010	\
0	37019020103	North Carolina	Brunswick	1.0	7052.0	2977.0	
1	37019020104	North Carolina	Brunswick	1.0	5305.0	2135.0	
2	37019020201	North Carolina	Brunswick	1.0	3300.0	1355.0	
3	37019020204	North Carolina	Brunswick	1.0	5391.0	2204.0	
4	37019020306	North Carolina	Brunswick	1.0	2352.0	995.0	

	GroupQuartersFlag	NUMGQTRS	PCTGQTRS	LILATracts_1And10	...	TractWhite	\
0	0.0	0.0	0.000000	0.0	...	5874.0	
1	0.0	0.0	0.000000	1.0	...	3425.0	
2	0.0	0.0	0.000000	0.0	...	2496.0	
3	0.0	14.0	0.002597	1.0	...	4913.0	
4	0.0	192.0	0.081633	0.0	...	2040.0	

	TractBlack	TractAsian	TractNHOPI	TractAIAN	TractOMultir	TractHispanic	\

```
0       691.0        97.0         2.0        33.0         355.0          433.0
1      1363.0        27.0         0.0        68.0         422.0          595.0
2       631.0        42.0         1.0        24.0         106.0          120.0
3       282.0        32.0         2.0        30.0         132.0          102.0
4       214.0         3.0         0.0         8.0          87.0           93.0

   TractHUNV  TractSNAP  Zip Code
0      144.0      490.0       NaN
1      141.0      503.0       NaN
2       33.0      163.0       NaN
3      106.0      272.0       NaN
4      199.0      250.0       NaN

[5 rows x 148 columns]
```

[5]:
```
combined_tracts_df.to_csv('DataMerge1.csv')
```

[6]:
```
data3 ="DataMerge1.csv"
cleanmerge_df =pd.read_csv(data3)
cleanmerge_df.head()
cleanmerge_df.shape
```

[6]:
```
(60, 149)
```

Compare New hanover County with the neighboring counties

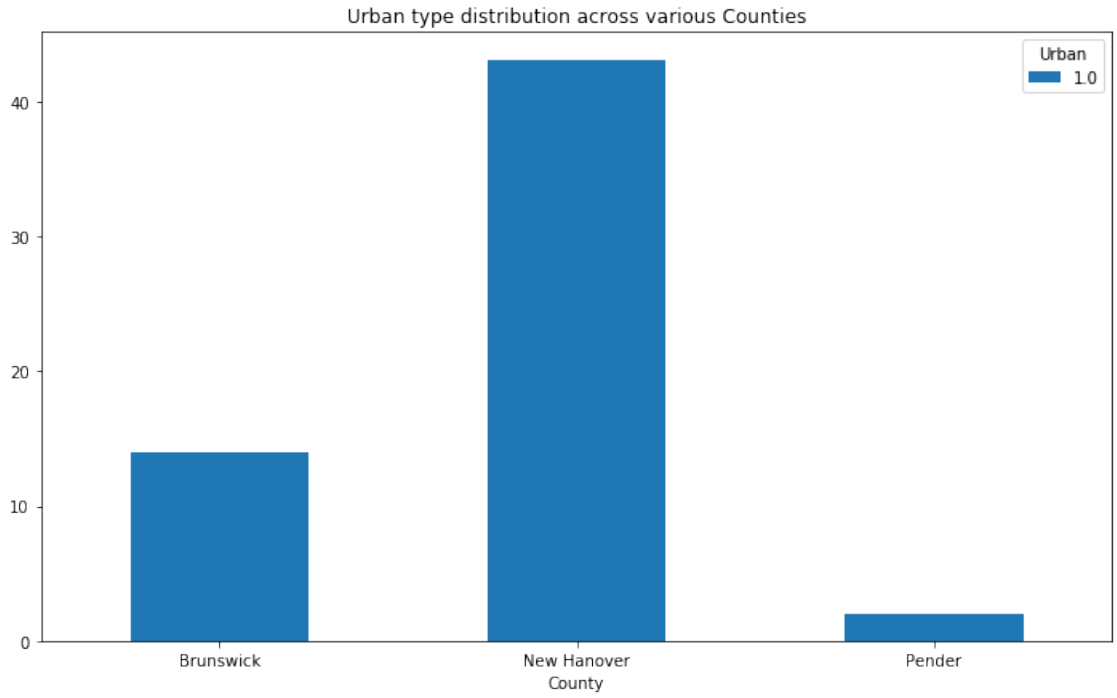
[7]:
```
#identify number of tracts in these counties.
df_county = cleanmerge_df.County
countytract_distribution = df_county.value_counts(dropna=False)
fig = go.FigureWidget(data=go.Bar(x=countytract_distribution.index,␣
 ↪y=countytract_distribution.values))
fig.update_layout(title='Tract Distribution across various counties',␣
 ↪xaxis_title='Neighbourhood Counties', yaxis_title='Count', width=800,␣
 ↪height=600)
fig.show()
```

[8]:
```
#the affect of food desert is different in rural and urban based counties.
#Identifying the number of Urban tracts in these counties

print(cleanmerge_df['Urban'].value_counts())


df_urban = cleanmerge_df.groupby(by=['County','Urban']).count()['State']
print(df_urban.head())
fig, ax = plt.subplots(1, 1, figsize=(12, 7))
df_urban.unstack().plot(kind='bar', ax=ax, title='Urban type distribution␣
 ↪across various Counties')
plt.xticks(rotation=0)
```

```
1.0    59
```

```
Name: Urban, dtype: int64
County       Urban
Brunswick    1.0      14
New Hanover  1.0      43
Pender       1.0       2
Name: State, dtype: int64
```

[8]:
```
(array([0, 1, 2]), <a list of 3 Text xticklabel objects>)
```



[9]:
```
#Analyse the Number of Housing Units

df_householdunits = cleanmerge_df.groupby(by=['County'])['OHU2010'].sum()
print(df_householdunits.head())

fig = go.Figure(data=[go.Pie(labels=df_householdunits.index,␣
 ↪values=df_householdunits.values)])
fig.update_layout(title='Total Number of housing units in the␣
 ↪county',width=1000, height=500)
fig.show()
```

```
County
Brunswick      20722.0
New Hanover    83033.0
Pender          4761.0
Name: OHU2010, dtype: float64
```

#Observation:
As the number of house units are greater, so is the number of SNAP benefits in the county

[10]:
```
#Analyse the Number of SNAP benefits in a county
df_householdunits = cleanmerge_df.groupby(by=['County'])['TractSNAP'].sum()
df_householdunits

fig = go.Figure(data=[go.Pie(labels=df_householdunits.index,␣
 ↪values=df_householdunits.values)])
fig.update_layout(title='Total percentage of housing units receiving SNAP␣
 ↪benefits in a county',width=1000, height=500)
fig.show()
```

New Hanover DataFrame

[11]:
```
dataframe_Newhanover = cleanmerge_df[cleanmerge_df.County.isin(["New Hanover"])]
dataframe_Newhanover.head()
```

[11]:
```
    Unnamed: 0  CensusTract           State       County  Urban  POP2010  \
14          14  37129010100  North Carolina  New Hanover    1.0   2512.0
15          15  37129010200  North Carolina  New Hanover    1.0   3378.0
16          16  37129010300  North Carolina  New Hanover    1.0   5365.0
17          17  37129010400  North Carolina  New Hanover    1.0   2909.0
18          18  37129010501  North Carolina  New Hanover    1.0   3820.0

    OHU2010  GroupQuartersFlag  NUMGQTRS  PCTGQTRS  ...  TractWhite  \
14   1020.0                0.0      96.0  0.038217  ...       737.0
15   1457.0                0.0      18.0  0.005329  ...      1475.0
16   2235.0                0.0      53.0  0.009879  ...      2198.0
17   1321.0                0.0      14.0  0.004813  ...      2457.0
18   1974.0                0.0       4.0  0.001047  ...      2050.0

    TractBlack  TractAsian  TractNHOPI  TractAIAN  TractOMultir  \
14      1680.0         4.0         3.0       11.0          77.0
15      1755.0        16.0         2.0       17.0         113.0
16      2891.0        30.0         5.0       36.0         205.0
17       310.0        33.0         0.0       13.0          96.0
18      1123.0        52.0         1.0       26.0         568.0

    TractHispanic  TractHUNV  TractSNAP  Zip Code
14           93.0      187.0      249.0   28401.0
15          136.0      113.0      309.0   28401.0
16          193.0      251.0      695.0   28403.0
17           96.0       58.0       70.0   28403.0
18          810.0       96.0      559.0   28403.0

[5 rows x 149 columns]
```

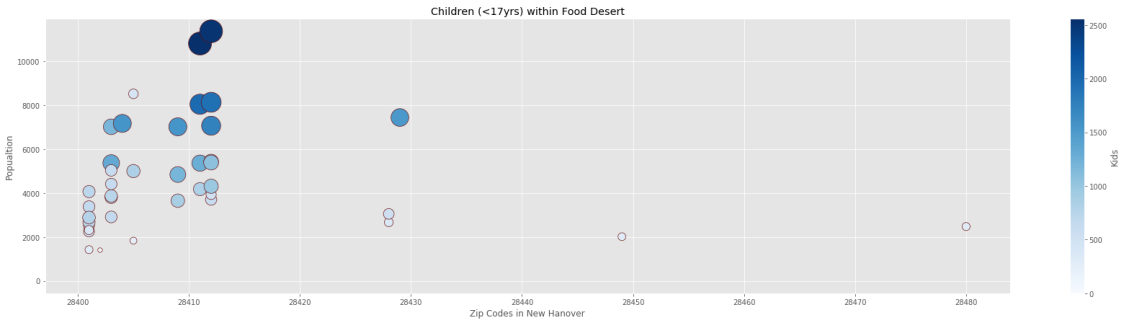
[ ]:
```

```

I focused my analysis on the New Hanover county and wanted to identify the most affected

regions based on ZipCode in this region which needs our attention

[12]:
```
plt.style.use("ggplot")
fig,ax=plt.subplots(figsize=(30, 7))
plt1=ax.scatter(dataframe_Newhanover["Zip Code"],
                dataframe_Newhanover["POP2010"],
s=dataframe_Newhanover["TractKids"]/(1e2)*40,
c=dataframe_Newhanover["TractKids"],
cmap=plt.cm.Blues,
vmin=0,
vmax=dataframe_Newhanover["TractKids"].max(),
edgecolor="#6b0c08",
linewidth=.65)
cbar=fig.colorbar(plt1)
cbar.set_label("Kids")
ax.set_xlabel("Zip Codes in New Hanover")
ax.set_ylabel("Popualtion")
ax.set_title("Children (<17yrs) within Food Desert")
#plt.xlim(1300,7500)
#plt.ylim(0,55)
```

[12]:
```
Text(0.5, 1.0, 'Children (<17yrs) within Food Desert')
```



[ ]:

[13]:
```
plt.savefig("NewData3.jpg")
plt.show()
```

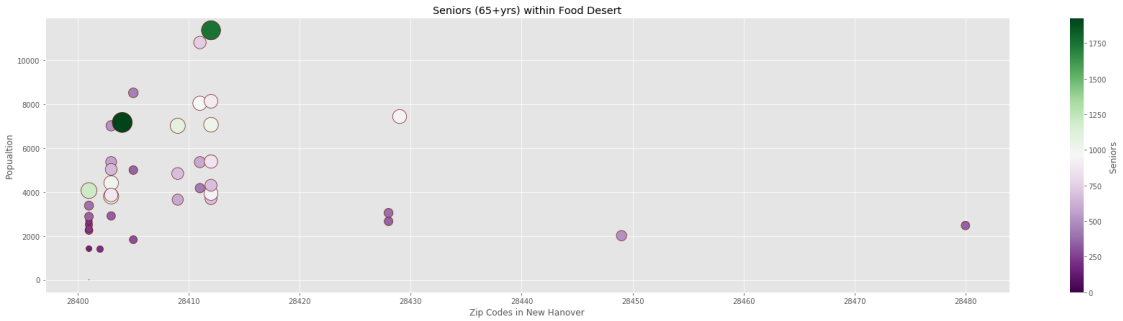
```
<Figure size 432x288 with 0 Axes>
```

[ ]:
```
#Regions where situation needs to be addressed
```

[14]:
```
plt.style.use("ggplot")
fig,ax=plt.subplots(figsize=(30, 7))
plt1=ax.scatter(dataframe_Newhanover["Zip Code"],
                dataframe_Newhanover["POP2010"],
s=dataframe_Newhanover["TractSeniors"]/(1e2)*40,
```

```
c=dataframe_Newhanover["TractSeniors"],
cmap=plt.cm.PRGn,
vmin=0,
vmax=dataframe_Newhanover["TractSeniors"].max(),
edgecolor="#6b0c08",
linewidth=.65)
cbar=fig.colorbar(plt1)
cbar.set_label("Seniors ")
ax.set_xlabel("Zip Codes in New Hanover")
ax.set_ylabel("Popualtion")
ax.set_title("Seniors (65+yrs) within Food Desert")
```

[14]: `Text(0.5, 1.0, 'Seniors (65+yrs) within Food Desert')`

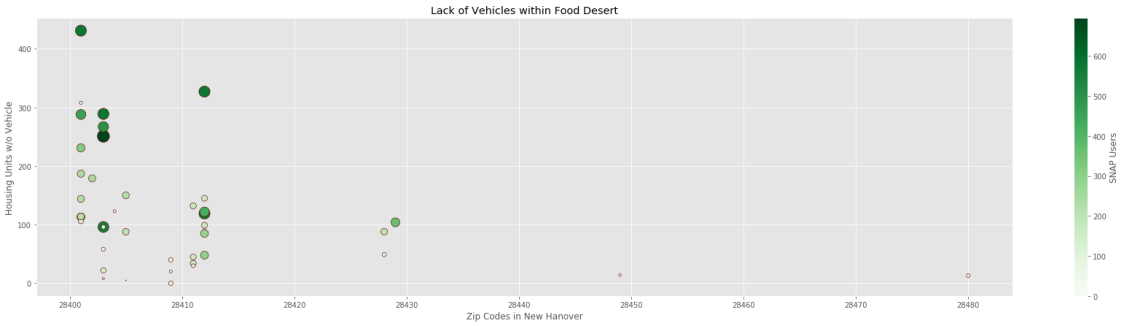


[16]:
```
plt.savefig("NewData4.jpg")
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

[15]:
```
plt.style.use("ggplot")
fig,ax=plt.subplots(figsize=(30, 7))
plt1=ax.scatter(dataframe_Newhanover["Zip Code"],
                dataframe_Newhanover["TractHUNV"],
s=dataframe_Newhanover["TractSNAP"]/(1e2)*40,
c=dataframe_Newhanover["TractSNAP"],
cmap=plt.cm.Greens,
vmin=0,
vmax=dataframe_Newhanover["TractSNAP"].max(),
edgecolor="#6b0c08",
linewidth=.65)
cbar=fig.colorbar(plt1)
cbar.set_label("SNAP Users")
ax.set_xlabel("Zip Codes in New Hanover")
ax.set_ylabel("Housing Units w/o Vehicle")
ax.set_title("Lack of Vehicles within Food Desert")
```

[15]:
```
Text(0.5, 1.0, 'Lack of Vehicles within Food Desert')
```



[18]:
```
plt.savefig("NewData5.jpg")
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

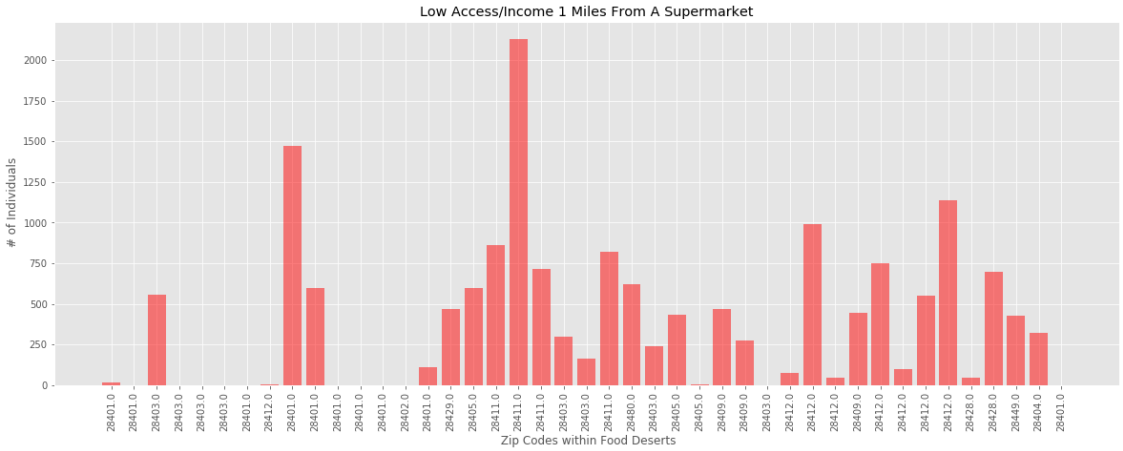
2 Number of housing units with no vehicles is highest in the areas with zip code 28400 - 28410. These areas are highly prone to food desert.

Notice: this figure also signifies the SNAP users. Now even if a person benefits from SNAP, if don't owns a vehicle he won't be able to visit grocery stores to buy healthy food.

[19]:
```
# Set x axis and tick locations
x_axis = np.arange(len(dataframe_Newhanover))
tick_locations = [value+0.4 for value in x_axis]
```

[20]:
```
# Create a list indicating where to write x labels and set figure size to␣
 ↪adjust for space
plt.figure(figsize=(20,7))
plt.bar(x_axis, dataframe_Newhanover["LALOWI1_10"], color='r', alpha=0.5,␣
 ↪align="edge")
plt.xticks(tick_locations, dataframe_Newhanover["Zip Code"],␣
 ↪rotation="vertical")
plt.xlabel(" Zip Codes within Food Deserts")
plt.ylabel(" # of Individuals")
plt.title("Low Access/Income 1 Miles From A Supermarket")
```

[20]:
```
Text(0.5, 1.0, 'Low Access/Income 1 Miles From A Supermarket')
```



Again Zipcode 28411 needs our attention

[ ]:

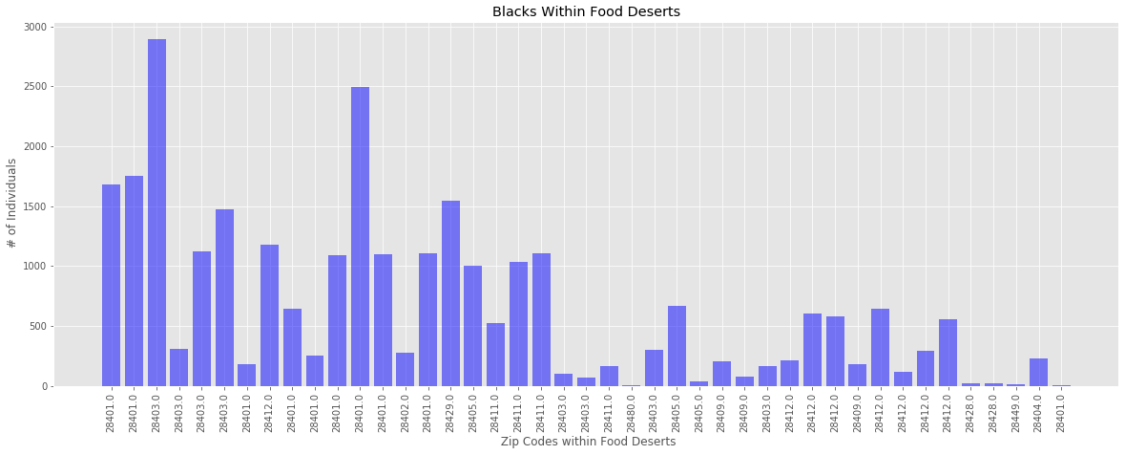
[21]:
```
# Save our graph and show the grap
plt.tight_layout()
plt.savefig("NewData8")
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

3 Now I decided to do some analysis on distribution of race and ethnicity around various Tracts. I don't have any religious/race biases but from a data science problem point of view, we need to consider all the factors as during the supply chain management process, it is very important to understand what type of food the person living in that tract prefer to eat as it will help us as a community to build a robust product distribution address system.

[22]:
```
# Create a list indicating where to write x labels and set figure size to␣
 ↪adjust for space
plt.figure(figsize=(20,7))
plt.bar(x_axis, dataframe_Newhanover["TractBlack"], color='b', alpha=0.5,␣
 ↪align="edge")
plt.xticks(tick_locations, dataframe_Newhanover["Zip Code"],␣
 ↪rotation="vertical")
plt.xlabel(" Zip Codes within Food Deserts")
plt.ylabel(" # of Individuals")
plt.title("Blacks Within Food Deserts")
```

[22]: Text(0.5, 1.0, 'Blacks Within Food Deserts')

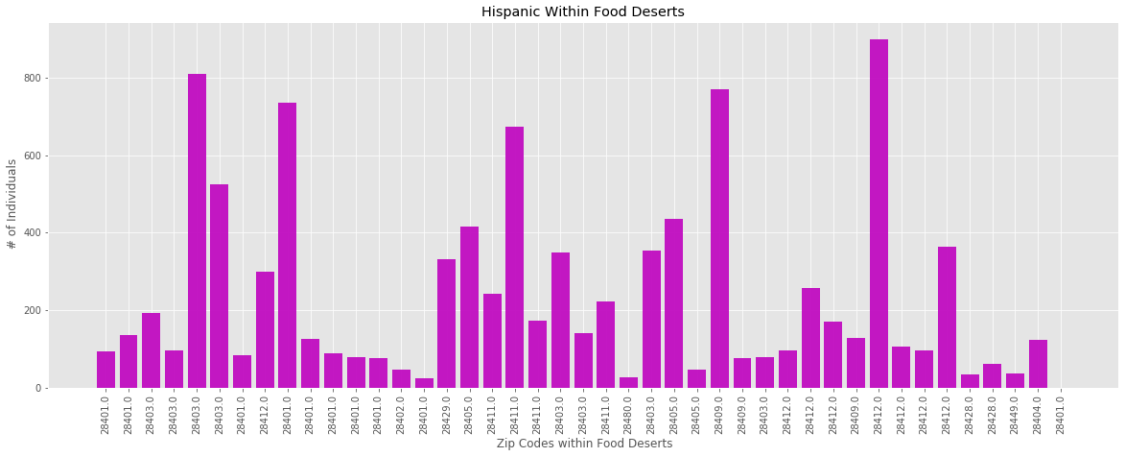


[23]:
```
# Save our graph and show the grap
plt.tight_layout()
plt.savefig("DataBlack7")
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

[24]:
```
# Create a list indicating where to write x labels and set figure size to␣
 ↪adjust for space
plt.figure(figsize=(20,7))
plt.bar(x_axis, dataframe_Newhanover["TractHispanic"], color='m', alpha=0.9,␣
 ↪align="edge")
plt.xticks(tick_locations, dataframe_Newhanover["Zip Code"],␣
 ↪rotation="vertical")
plt.xlabel(" Zip Codes within Food Deserts")
plt.ylabel(" # of Individuals")
plt.title("Hispanic Within Food Deserts")
```

[24]: Text(0.5, 1.0, 'Hispanic Within Food Deserts')

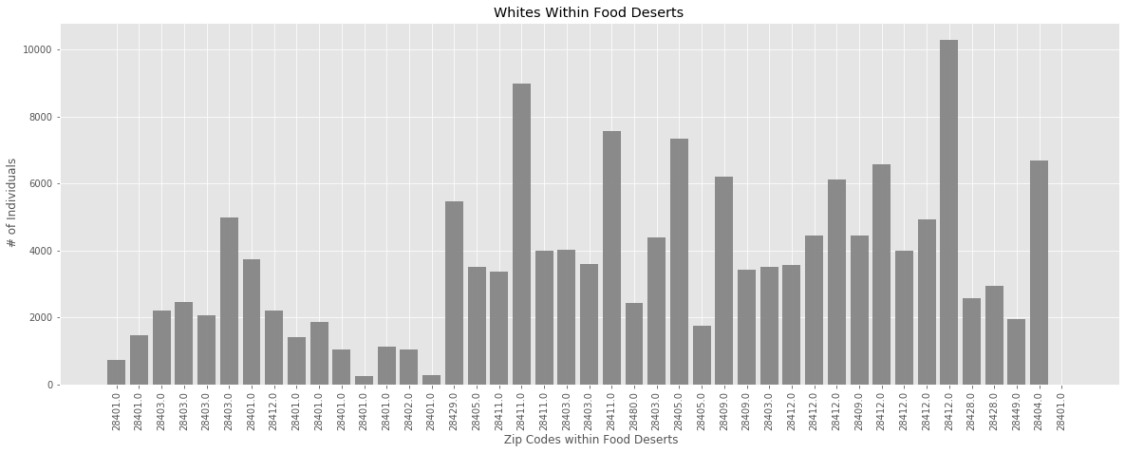


[25]:
```
# Save our graph and show the grap
plt.tight_layout()
plt.savefig("DataHispanic8")
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

[26]:
```
# Create a list indicating where to write x labels and set figure size to␣
 ↪adjust for space
plt.figure(figsize=(20,7))
plt.bar(x_axis, dataframe_Newhanover["TractWhite"], color='grey', alpha=0.9,␣
 ↪align="edge")
plt.xticks(tick_locations, dataframe_Newhanover["Zip Code"],␣
 ↪rotation="vertical")
plt.xlabel(" Zip Codes within Food Deserts")
plt.ylabel(" # of Individuals")
plt.title("Whites Within Food Deserts")
```

[26]:
```
Text(0.5, 1.0, 'Whites Within Food Deserts')
```

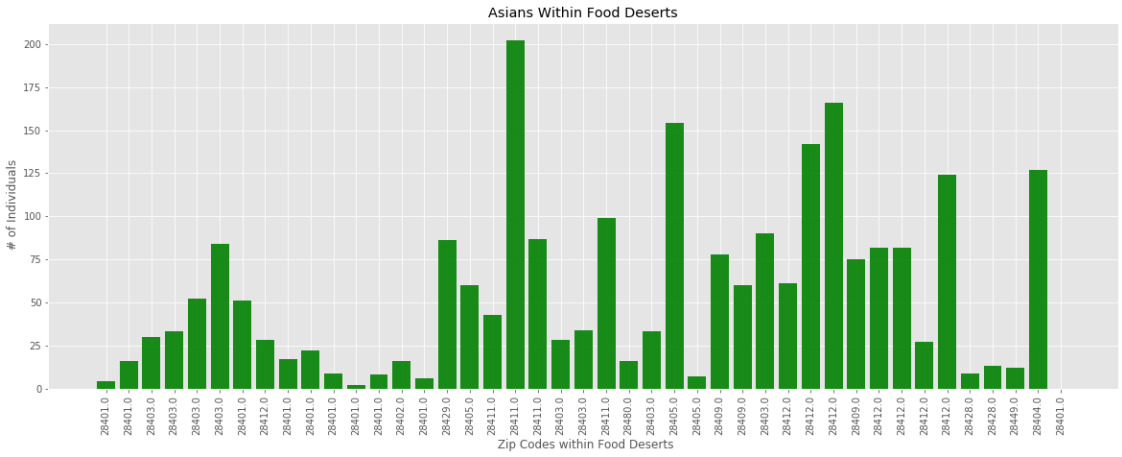


[27]:
```
# Save our graph and show the grap
plt.tight_layout()
plt.savefig("DataWhites9")
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

[37]:
```
# Create a list indicating where to write x labels and set figure size to␣
 ↪adjust for space
plt.figure(figsize=(20,7))
plt.bar(x_axis, dataframe_Newhanover["TractAsian"], color='Green', alpha=0.9,␣
 ↪align="edge")
plt.xticks(tick_locations, dataframe_Newhanover["Zip Code"],␣
 ↪rotation="vertical")
plt.xlabel(" Zip Codes within Food Deserts")
plt.ylabel(" # of Individuals")
plt.title("Asians Within Food Deserts")
```

[37]:
```
Text(0.5, 1.0, 'Asians Within Food Deserts')
```

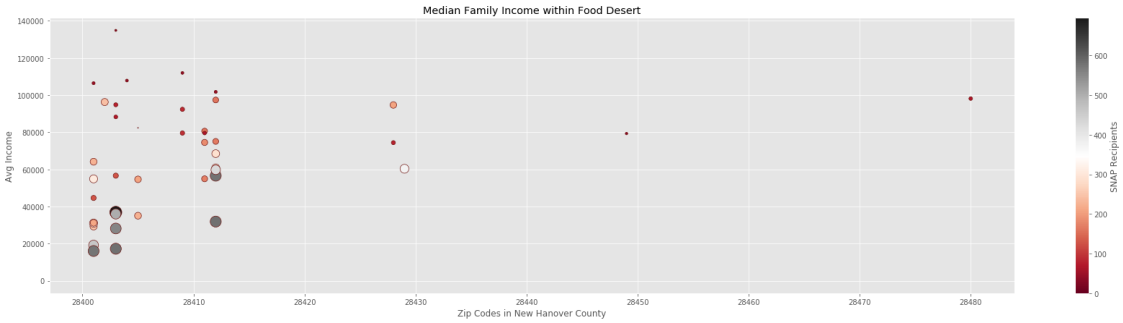


[29]:
```
# Save our graph and show the grap
plt.tight_layout()
plt.savefig("DataMulti10")
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

[30]:
```
plt.style.use("ggplot")
fig,ax=plt.subplots(figsize=(30, 7))
plt1=ax.scatter(dataframe_Newhanover["Zip Code"],
                dataframe_Newhanover["MedianFamilyIncome"],
s=dataframe_Newhanover["TractSNAP"]/(1e2)*40,
c=dataframe_Newhanover["TractSNAP"],
cmap=plt.cm.RdGy,
vmin=0,
vmax=dataframe_Newhanover["TractSNAP"].max(),
edgecolor="#6b0c08",
linewidth=.65)
cbar=fig.colorbar(plt1)
cbar.set_label("SNAP Recipients")
ax.set_xlabel("Zip Codes in New Hanover County")
ax.set_ylabel("Avg Income")
ax.set_title("Median Family Income within Food Desert")
```

[30]:
```
Text(0.5, 1.0, 'Median Family Income within Food Desert')
```



[31]:
```
plt.savefig("DataIncome11.jpg")
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

[32]:
```
dataframe_Newhanover['MedianFamilyIncome'].describe()
```

[32]:
```
count        43.000000
mean      65042.465116
std       30918.356051
min           0.000000
25%       36405.500000
50%       64100.000000
75%       90292.000000
max      134877.000000
Name: MedianFamilyIncome, dtype: float64
```

[33]:
```
import plotly.express as px
df = px.data.tips()
fig = px.box(dataframe_Newhanover, y="MedianFamilyIncome")
fig.show()
```

#remove the comments below to compare the box plot distribution across different race types.

[39]:
```
'''
import plotly.graph_objects as go
import numpy as np
np.random.seed(1)

fig = go.Figure()
fig.add_trace(go.Box(y=dataframe_Newhanover['TractWhite'],name = "White"))
fig.add_trace(go.Box(y=dataframe_Newhanover['TractAsian'],name = "Asian"))
fig.add_trace(go.Box(y=dataframe_Newhanover['TractBlack'],name = "Black"))
fig.add_trace(go.Box(y=dataframe_Newhanover['TractHispanic'],name =
 ↪"Hispanic"))



fig.show()
'''
```

[39]:
```
'\'\nimport plotly.graph_objects as go\nimport numpy as
np\nnp.random.seed(1)\n\nfig =
go.Figure()\nfig.add_trace(go.Box(y=dataframe_Newhanover[\'TractWhite\'],name =
"White"))\nfig.add_trace(go.Box(y=dataframe_Newhanover[\'TractAsian\'],name =
"Asian"))\nfig.add_trace(go.Box(y=dataframe_Newhanover[\'TractBlack\'],name =
"Black"))\nfig.add_trace(go.Box(y=dataframe_Newhanover[\'TractHispanic\'],name =
"Hispanic"))\n\n\n\nfig.show()\n'
```