Lab1_Group2

May 24, 2020

1 Lab One: Vizualization and Data Preprocessing

Group 2 Members: Reagan Meagher, Jaclyn Coate, Megan Riley, and Matthew Chinchilla

Kaggle Link To Data: https://www.kaggle.com/muonneutrino/us-census-demographic-data

1.1 Business Understanding

The data we used for this lab is a 2015 United States census data set for the all 50 states at the census tract level, accessed via Kaggle. This data set contains demographic data about each census tract such as ethnicity, job function, commute types, poverty, income, and unemployment. We also have the 2017 version of this data set.

This data set was orignally collected by the United States Census Bureau to gain insights into the demographics of areas of the U.S. on a granular level.

For us, the purpose of this data set is to determine which characteristics of a geographical area determine the level of poverty in that respective area. Poverty in this data set is given as a percentage of the population, but in order for this to be a classification we turned that percentage into a factor with four levels: Low Poverty, Average Poverty, High Poverty, and Extreme Poverty. We will discuss this new dependent variable in detail in the new features section.

We will have successfully mined useful knowledge from this data set if we can determine which charactersitcs of a geographical area from the attributes in our data set are the most highly correlated with certain poverty classes. This will help lead us to building a classification model that can accruately predict which poverty class that geographical area is.

Our classification model will be built with the data at the census tract level and we will test our model at various geographical levels including census tract, zip code, city, and county.

The poverty level of an area is very important as it has social, cultural, political, and economic implications. Therefore, we need to know for sure if an area is at a certain poverty level. We also need to know that if we change certain characteristics of a geographical area if that would likely change the poverty level or not. That leads us to thinking that the accuracy of our model, that being the rate of correct poverty class predictions, is highly important if we are to label our model as an effective model to predict poverty class of a geographical area.

Therefore, we will focus on the accuracy metric as a determintent of a successfull classification model or not. An 85% accuracy level would suffice as an effective classifcation model. We will also do an 80/20 cross-validation to assess the performance of our model. In addition to accuracy and cross validation we will use our model on the 2017 data set to see if it can accurately predict the Poverty classes of each census tract with updated variable values.

We are not concerend with sensitivity and specificity as both would mean a wrong classification of the geographical area. Depending on the use of the model in certain industries and sectors a wrong poverty class classification could infinite impacts such as employement prospects, government funding, school budgets, and community services. For example, if this model were to be used by a company looking to open an office in a certain area a wrong poverty class classification of that area could cause them to not open, leading to lost employment prospects in that area.

1.2 Data Meaning Type

The data we are using is from a Kaggle dataset containing U.S. Census Demographic Data. The author describes the data as being from the DP03 and DP05 tables of the 2015 and 2017 American Community Survey 5-year estimates. We have both the 2015 and 2017 datasets at the census tract level of detail. The census bureau defines a census tract as an area roughly equivelent to a neighborhood containing around 5000 residents. We created several new variables for this data set such as PovertyClass, IncomeClass, PopulationClass, and population percentage breakdowns of men and women. These variables were created to provide us with a categorical dependent variable, additional predictors for our model and to aid in our analysis of the data. The complete data set includes 43 variables and 74001 rows with the majority of the variables being percentages classified as float64 variables.

The table below contains data definitions, most definitions regarding race or ethnicity were sourced from the original census brief overview. Other data definitions were obtained from the Census bureau glossary. Below the data definitions table is addition information related to the variables.

Attribute	Description
CensusTract	The Census Tract Id as assigned by the US Census Bureau
State	The State the data comes from
County	The county the data comes from
TotalPop	The total Population in a tract or county
Men	The total population of men in a given tract or county
Women	The total population of women in a given tract or county
PrcMen	Percentage of men in a tract
PrcWomen	Percentage of Women in a tract
Hispanic	Percentage of population in a given tract or county that Identifies as Hispanic. A hispanic is defined as a person of Cuban, Mexican, Puerto Rican, South or Central American, or other Spanish culture or origin regardless of race.
White	Percentage of population in a given tract or county that identifies as white. “White” refers to a person having origins in any of the original peoples of Europe, the Middle East, or North Africa. It includes people who indicated their race(s) as “White” or reported entries such as Irish, German, Italian, Lebanese,Arab, Moroccan, or Caucasian.
Black	Percentage of population in a given tract or county that identifies as black.“Black or African American” refers to a person having origins in any of the Black racial groups of Africa. It includes people who indicated their race(s) as “Black, African Am., or Negro” or reported entries such as African American, Kenyan, Nigerian, or Haitian.

Attribute	Description
Native	Percentage of population in a given tract or county that identifies as native. "American Indian or Alaska Native" refers to a person having origins in any of the original peoples of North and South America (including Central America) and who maintains tribal affiliation or community attachment. This category includes people who indicated their race(s) as "American Indian or Alaska Native" or reported their enrolled or principal tribe, such as Navajo, Blackfeet, Inupiat, Yup'ik, or Central American Indian groups or South American Indian groups.
Asian	Percentage of population in a given tract or county that identifies as Asian. "Asian" refers to a person having origins in any of the original peoples of the Far East, Southeast Asia, or the Indian subcontinent, including, for example, Cambodia, China, India, Japan, Korea, Malaysia, Pakistan, the Philippine Islands, Thailand, and Vietnam. It includes people who indicated their race(s) as "Asian" or reported entries such as "Asian Indian," "Chinese," "Filipino," "Korean," "Japanese," "Vietnamese," and "Other Asian" or provided other detailed Asian responses.
Pacific	Percentage of population in a given tract or county that identifies as Pacific. "Native Hawaiian or Other Pacific Islander" refers to a person having origins in any of the original peoples of Hawaii, Guam, Samoa, or other Pacific Islands. It includes people who indicated their race(s) as "Pacific Islander" or reported entries such as "Native Hawaiian," "Guamanian or Chamorro," "Samoan," and "Other Pacific Islander" or provided other detailed Pacific Islander responses.
Other	Percentage of population in a given tract or county that does not fit into the classes of Hispanic, White, Black, Native, Asian, or Pacific. This variable was created to account for other races not explicitly called out in the original data.
Citizen	Number of total population in a tract or county that are United States citizens.
Income	Estimated household income in a given tract or county.
IncomeErr	Income standard error.
IncomePerCap	Per capita income is the mean income computed for every man, woman, and child in a particular group. It is derived by dividing the total income of a particular group by the total population.
IncomePerCapErr	Income Per Capita standard error.
Poverty	Percentage of total population considered in poverty. The Census Bureau uses a set of money income thresholds that vary by family size and composition to determine who is in poverty. If the total income for a family or unrelated individual falls below the relevant poverty threshold, then the family (and every individual in it) or unrelated individual is considered in poverty.
ChildPoverty	Percentage of total population considered children and in poverty.
Professional	Percentage of total population who's occupation is considered professional. The Professional, Scientific, and Technical Services sector comprises establishments that specialize in performing professional, scientific, and technical activities for others. These activities require a high degree of expertise and training. Activities performed include: legal advice and representation; accounting, bookkeeping, and payroll services; architectural, engineering, and specialized design services; computer services; consulting services; research services; advertising services; photographic services; translation and interpretation services; veterinary services; and other professional, scientific, and technical services.

Attribute	Description
Service	Percentage of total population who's occupation is considered part of the service or hospitality industry. The Accommodation and Food Services sector is comprised of establishments providing customers with lodging and/or preparing meals, snacks, and beverages for immediate consumption. The sector includes both accommodation and food services establishments because the two activities are often combined at the same establishment.
Office	Percentage of total population who's occupation is considered administrative or office oriented.
Construction	Percentage of total population who's occupation is considered to be part of the construction sector. The Construction sector comprises establishments primarily engaged in the construction of buildings or engineering projects (e.g., highways and utility systems). Construction work done may include new work, additions, alterations, or maintenance and repairs.
Production	Percentage of total population who's occupation is considered to be part of the production sector. The "production workers" number includes workers (up through the line-supervisor level) engaged in fabricating, processing, assembling, inspecting, receiving, storing, handling, packing, warehousing, shipping (but not delivering), maintenance, repair, janitorial and guard services, product development, auxiliary production for plant's own use (e.g., power plant), recordkeeping, and other services closely associated with these production operations at the establishment covered by the report. Employees above the working-supervisor level are excluded from this item.
Drive	Percentage of total population who's primary mode of transportation is driving.
Carpool	Percentage of total population who's primary mode of transportation is carpool.
Transit	Percentage of total population who's primary mode of transportation is the public transit system.
Walk	Percentage of total population who's primary mode of transportation is walking.
OtherTransp	Percentage of total population who's primary mode of transportation is other than driving, carpooling, public transit, or walking.
WorkAtHome	Percentage of total population who work at or from their place of residence.
MeanCommute	The average commute distance in miles for persons that are employed but not working at or from their place of residence.
Employed	Total number of population who are considered employed. Employed includes all civilians 16 years old and over who were either (1) "at work" – those who did any work at all during the reference week as paid employees, worked in their own business or profession, worked on their own farm, or worked 15 hours or more as unpaid workers on a family farm or in a family business; or (2) were "with a job but not at work" – those who did not work during the reference week but had jobs or businesses from which they were temporarily absent due to illness, bad weather, industrial dispute, vacation, or other personal reasons. Excluded from the employed are people whose only activity consisted of work around the house or unpaid volunteer work for religious, charitable, and similar organizations; also excluded are people on active duty in the United States Armed Forces.
PrivateWork	Percentage of total population that work for a private company or structures not owned by any federal, state, or local government.
PublicWork	Percentage of total population that work for a federal, state, or local government.
SelfEmployed	Percentage of total population that are self employed.
FamilyWork	Percentage of total population that work with or for immediate or extended family.

Attribute	Description
Unemployment	Percentage of total population that is considered unemployed. All civilians 16 years old and over are classified as unemployed if they (1) were neither "at work" nor "with a job but not at work" during the reference week, and (2) were actively looking for work during the last 4 weeks, and (3) were available to accept a job.
IncomeClass	Classification of poverty based on income per capita. 'Lower Class' = 0-19189, 'Lower-Middle Class' = 19190-25367, 'Upper-Middle Class' = 25369-33889, 'Upper Class' = 33890-1000000
PopulationClass	A classifier indicating the size of the population. 'Small Population' = 0-2944, 'Medium Population' = 2945-4098, 'large Population' = 4099-5467, 'Extra Large Population' = 5486 - 100000
PovertyClassifier	A classifier indicating the class of poverty in a given tract. The classifier was calculated based on the percent of poverty indicated for each tract 'Low Poverty' = 0 to 6.9%, 'Average Poverty' = 7 to 12.4%,'High Poverty' = 12.5 to 21.9%, 'Extreme Poverty' = 22 to 100%

[97]:
```
data2015.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74001 entries, 0 to 74000
Data columns (total 43 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   CensusTract      74001 non-null  int64
 1   State            74001 non-null  object
 2   County           74001 non-null  object
 3   TotalPop         74001 non-null  int64
 4   Men              74001 non-null  int64
 5   Women            74001 non-null  int64
 6   PercMen          73311 non-null  float64
 7   PercWomen        73311 non-null  float64
 8   Hispanic         73311 non-null  float64
 9   White            73311 non-null  float64
 10  Black            73311 non-null  float64
 11  Native           73311 non-null  float64
 12  Asian            73311 non-null  float64
 13  Pacific          73311 non-null  float64
 14  Other            73311 non-null  float64
 15  Citizen          74001 non-null  int64
 16  Income           72901 non-null  float64
 17  IncomeErr        72901 non-null  float64
 18  IncomePerCap     73261 non-null  float64
 19  IncomePerCapErr  73261 non-null  float64
 20  Poverty          73166 non-null  float64
 21  ChildPoverty     72883 non-null  float64
 22  Professional     73194 non-null  float64
```

```
 23  Service          73194 non-null  float64
 24  Office           73194 non-null  float64
 25  Construction     73194 non-null  float64
 26  Production       73194 non-null  float64
 27  Drive            73204 non-null  float64
 28  Carpool          73204 non-null  float64
 29  Transit          73204 non-null  float64
 30  Walk             73204 non-null  float64
 31  OtherTransp      73204 non-null  float64
 32  WorkAtHome       73204 non-null  float64
 33  MeanCommute      73052 non-null  float64
 34  Employed         74001 non-null  int64
 35  PrivateWork      73194 non-null  float64
 36  PublicWork       73194 non-null  float64
 37  SelfEmployed     73194 non-null  float64
 38  FamilyWork       73194 non-null  float64
 39  Unemployment     73199 non-null  float64
 40  IncomeClass      73261 non-null  category
 41  PopulationClass  73311 non-null  category
 42  PovertyClass     73003 non-null  category
dtypes: category(3), float64(32), int64(6), object(2)
memory usage: 22.8+ MB
```

1.2.1 Data Load & New Feature Creation

Loading Our Initial Data Sets To set up the rest of our analysis sections we first need to load our 2015 and 2017 census data sets. The subsequent analysis sections will all use the 2015 data set. To make sure we have the correct data set, we will use the .head() python method.

[2]:
```
#Load and Explore Census 2015 and 2017 Data
import pandas as pd
import numpy as np

census2015 = "https://raw.githubusercontent.com/megnn/SMUMSDS_ML1/master/
 ↪acs2015_census_tract_data.csv"
census2017 = "https://raw.githubusercontent.com/megnn/SMUMSDS_ML1/master/
 ↪acs2017_census_tract_data.csv"


data2015 = pd.read_csv(census2015)
data2017 = pd.read_csv(census2017)
data2015.head()
```

[2]:

	CensusTract	State	County	TotalPop	Men	Women	Hispanic	White	\
0	1001020100	Alabama	Autauga	1948	940	1008	0.9	87.4	
1	1001020200	Alabama	Autauga	2156	1059	1097	0.8	40.4	
2	1001020300	Alabama	Autauga	2968	1364	1604	0.0	74.5	
3	1001020400	Alabama	Autauga	4423	2172	2251	10.5	82.8	

```
4   1001020500  Alabama  Autauga     10763  4922   5841       0.7   68.5

   Black  Native  …  Walk  OtherTransp  WorkAtHome  MeanCommute  Employed  \
0    7.7     0.3  …   0.5          2.3         2.1         25.0       943
1   53.3     0.0  …   0.0          0.7         0.0         23.4       753
2   18.6     0.5  …   0.0          0.0         2.5         19.6      1373
3    3.7     1.6  …   0.0          2.6         1.6         25.3      1782
4   24.8     0.0  …   0.0          0.6         0.9         24.8      5037

   PrivateWork  PublicWork  SelfEmployed  FamilyWork  Unemployment
0         77.1        18.3           4.6         0.0           5.4
1         77.0        16.9           6.1         0.0          13.3
2         64.1        23.6          12.3         0.0           6.2
3         75.7        21.2           3.1         0.0          10.8
4         67.1        27.6           5.3         0.0           4.2

[5 rows x 37 columns]
```

Feature Creation: Poverty Class Since we are dealing with a classification model we need to create a new cateogrical variable from our continuous Poverty variable (continuous percentage from 0-100%). We will discuss this new variable in more detail in the New Features section. The code for this variable can be seen below.

[3]:
```
#Create new categorical dependent variable: PovertyClass
PovertyClass = pd.cut(data2015.Poverty,bins=[0,7,12.5,22,100],labels=
['Low Poverty','Average Poverty','High Poverty','Extreme Poverty'])
data2015.insert(37,'PovertyClass',PovertyClass)

PovertyClass = pd.cut(data2017.Poverty,bins=[0,7,12.5,22,100],labels=
['Low Poverty','Average Poverty','High Poverty','Extreme Poverty'])
data2017.insert(37,'PovertyClass',PovertyClass)
data2015.head()
```

[3]:
```
   CensusTract    State   County  TotalPop   Men  Women  Hispanic  White  \
0   1001020100  Alabama  Autauga      1948   940   1008       0.9   87.4
1   1001020200  Alabama  Autauga      2156  1059   1097       0.8   40.4
2   1001020300  Alabama  Autauga      2968  1364   1604       0.0   74.5
3   1001020400  Alabama  Autauga      4423  2172   2251      10.5   82.8
4   1001020500  Alabama  Autauga     10763  4922   5841       0.7   68.5

   Black  Native  …  OtherTransp  WorkAtHome  MeanCommute  Employed  \
0    7.7     0.3  …          2.3         2.1         25.0       943
1   53.3     0.0  …          0.7         0.0         23.4       753
2   18.6     0.5  …          0.0         2.5         19.6      1373
3    3.7     1.6  …          2.6         1.6         25.3      1782
4   24.8     0.0  …          0.6         0.9         24.8      5037
```

```
   PrivateWork  PublicWork  SelfEmployed  FamilyWork  Unemployment  \
0         77.1        18.3           4.6         0.0           5.4
1         77.0        16.9           6.1         0.0          13.3
2         64.1        23.6          12.3         0.0           6.2
3         75.7        21.2           3.1         0.0          10.8
4         67.1        27.6           5.3         0.0           4.2

      PovertyClass
0  Average Poverty
1  Extreme Poverty
2     High Poverty
3      Low Poverty
4  Average Poverty

[5 rows x 38 columns]
```

Feature Creation: Other Ethnicity Percentage The percentages for the ethnicity variables do add up to 100%. Therefore, we created a continuous percentage variable that rounds out the additional ethnicity percentages to equal 100% for back calcuation in next section. This variable is called "Other" and can be seen between the "Pacific" and "Citizen" variables. The code for this variable is below. This new varaible will be discussed further in our New Features section.

[4]:
```
#Create new numerical independent variable: Other to account for reat of␣
 ↪percentage of races
Other = ''
data2015.insert(12,'Other',Other)
data2017.insert(12,'Other',Other)

data2015['Other'] = 100 - (data2015['Hispanic'] + data2015['White'] +␣
 ↪data2015['Black'] + data2015['Native'] + data2015['Asian'] +␣
 ↪data2015['Pacific'])
data2015.head()
```

[4]:
```
   CensusTract    State   County  TotalPop   Men  Women  Hispanic  White  \
0   1001020100  Alabama  Autauga      1948   940   1008       0.9   87.4
1   1001020200  Alabama  Autauga      2156  1059   1097       0.8   40.4
2   1001020300  Alabama  Autauga      2968  1364   1604       0.0   74.5
3   1001020400  Alabama  Autauga      4423  2172   2251      10.5   82.8
4   1001020500  Alabama  Autauga     10763  4922   5841       0.7   68.5

   Black  Native  …  OtherTransp  WorkAtHome  MeanCommute  Employed  \
0    7.7     0.3  …          2.3         2.1         25.0       943
1   53.3     0.0  …          0.7         0.0         23.4       753
2   18.6     0.5  …          0.0         2.5         19.6      1373
3    3.7     1.6  …          2.6         1.6         25.3      1782
4   24.8     0.0  …          0.6         0.9         24.8      5037
```

```
   PrivateWork  PublicWork  SelfEmployed  FamilyWork  Unemployment  \
0         77.1        18.3           4.6         0.0           5.4
1         77.0        16.9           6.1         0.0          13.3
2         64.1        23.6          12.3         0.0           6.2
3         75.7        21.2           3.1         0.0          10.8
4         67.1        27.6           5.3         0.0           4.2

      PovertyClass
0  Average Poverty
1  Extreme Poverty
2     High Poverty
3      Low Poverty
4  Average Poverty

[5 rows x 39 columns]
```

Feature Creation: Aggregate Numbers Many of our variables are percentages of the total population for the respective cateogries. This can be an issue when we want to roll up the view to different levels of detail such as the county level for visualizations and analysis. Therefore, we created a new data frame called data2015agg as a reference for visualization and analysis that contains the aggregate population numbers instead of the percentages that were given in the original data set. This new aggregated data set can be seen below using the python .head() method. This new data set will be discussed further in our New Features section.

[5]:
```
#Code to create aggregated data frames at the county level
data2015agg = data2015.copy()
data2015agg['Hispanic'] = (data2015['Hispanic']/100) * data2015['TotalPop']
data2015agg['White'] = (data2015['White']/100) * data2015['TotalPop']
data2015agg['Black'] = (data2015['Black']/100) * data2015['TotalPop']
data2015agg['Native'] = (data2015['Native']/100) * data2015['TotalPop']
data2015agg['Asian'] = (data2015['Asian']/100) * data2015['TotalPop']
data2015agg['Pacific'] = (data2015['Pacific']/100) * data2015['TotalPop']
data2015agg['Other'] = (data2015['Other']/100) * data2015['TotalPop']
data2015agg['Poverty'] = (data2015['Poverty']/100) * data2015['TotalPop']
data2015agg['ChildPoverty'] = (data2015['ChildPoverty']/100) *␣
 ↪data2015['TotalPop']
data2015agg['Professional'] = (data2015['Professional']/100) *␣
 ↪data2015['TotalPop']
data2015agg['Service'] = (data2015['Service']/100) * data2015['TotalPop']
data2015agg['Office'] = (data2015['Office']/100) * data2015['TotalPop']
data2015agg['Construction'] = (data2015['Construction']/100) *␣
 ↪data2015['TotalPop']
data2015agg['Production'] = (data2015['Production']/100) * data2015['TotalPop']
data2015agg['Drive'] = (data2015['Drive']/100) * data2015['TotalPop']
data2015agg['Carpool'] = (data2015['Carpool']/100) * data2015['TotalPop']
data2015agg['Transit'] = (data2015['Transit']/100) * data2015['TotalPop']
data2015agg['Walk'] = (data2015['Walk']/100) * data2015agg['TotalPop']
```

```
data2015agg['OtherTransp'] = (data2015['OtherTransp']/100) *␣
 ↪data2015['TotalPop']
data2015agg['WorkAtHome'] = (data2015['WorkAtHome']/100) * data2015['TotalPop']
data2015agg['PrivateWork'] = (data2015['PrivateWork']/100) *␣
 ↪data2015['TotalPop']
data2015agg['PublicWork'] = (data2015['PublicWork']/100) * data2015['TotalPop']
data2015agg['SelfEmployed'] = (data2015['SelfEmployed']/100) *␣
 ↪data2015['TotalPop']
data2015agg['FamilyWork'] = (data2015['FamilyWork']/100) * data2015['TotalPop']
data2015agg['Unemployment'] = (data2015['Unemployment']/100) *␣
 ↪data2015['TotalPop']

df = pd.DataFrame(data2015agg)
df.to_csv('2015CensusAgg.csv')
data2015agg.head()
```

[5]:

	CensusTract	State	County	TotalPop	Men	Women	Hispanic	White	\
0	1001020100	Alabama	Autauga	1948	940	1008	17.532	1702.552	
1	1001020200	Alabama	Autauga	2156	1059	1097	17.248	871.024	
2	1001020300	Alabama	Autauga	2968	1364	1604	0.000	2211.160	
3	1001020400	Alabama	Autauga	4423	2172	2251	464.415	3662.244	
4	1001020500	Alabama	Autauga	10763	4922	5841	75.341	7372.655	

	Black	Native	…	OtherTransp	WorkAtHome	MeanCommute	Employed	\
0	149.996	5.844	…	44.804	40.908	25.0	943	
1	1149.148	0.000	…	15.092	0.000	23.4	753	
2	552.048	14.840	…	0.000	74.200	19.6	1373	
3	163.651	70.768	…	114.998	70.768	25.3	1782	
4	2669.224	0.000	…	64.578	96.867	24.8	5037	

	PrivateWork	PublicWork	SelfEmployed	FamilyWork	Unemployment	\
0	1501.908	356.484	89.608	0.0	105.192	
1	1660.120	364.364	131.516	0.0	286.748	
2	1902.488	700.448	365.064	0.0	184.016	
3	3348.211	937.676	137.113	0.0	477.684	
4	7221.973	2970.588	570.439	0.0	452.046	

	PovertyClass
0	Average Poverty
1	Extreme Poverty
2	High Poverty
3	Low Poverty
4	Average Poverty

[5 rows x 39 columns]

Feature Creation: Population Size Class The population size is of the census tract is given by a total population count. It may be useful to bin the Total Population (TotalPop) variable to seperate out small, medium, large, and extra large geographical areas by population. The code for this new varaible can be seen below. This new variable will be discussed further in our New Features section.

[6]:
```
#New feature creation Population Class
PopulationClass = pd.cut(data2015.
 ↪TotalPop,bins=[0,2945,4099,5468,100000],labels=
['Small Population','Medium Population','Large Population','Extra Large␣
 ↪Population'])
data2015.insert(38,'PopulationClass',PopulationClass)

PopulationClass = pd.cut(data2017.
 ↪TotalPop,bins=[0,2945,4099,5468,100000],labels=
['Small Population','Medium Population','Large Population','Extra Large␣
 ↪Population'])
data2017.insert(38,'PopulationClass',PopulationClass)
data2015.head()
```

[6]:

	CensusTract	State	County	TotalPop	Men	Women	Hispanic	White	\
0	1001020100	Alabama	Autauga	1948	940	1008	0.9	87.4	
1	1001020200	Alabama	Autauga	2156	1059	1097	0.8	40.4	
2	1001020300	Alabama	Autauga	2968	1364	1604	0.0	74.5	
3	1001020400	Alabama	Autauga	4423	2172	2251	10.5	82.8	
4	1001020500	Alabama	Autauga	10763	4922	5841	0.7	68.5	

	Black	Native	…	WorkAtHome	MeanCommute	Employed	PrivateWork	\
0	7.7	0.3	…	2.1	25.0	943	77.1	
1	53.3	0.0	…	0.0	23.4	753	77.0	
2	18.6	0.5	…	2.5	19.6	1373	64.1	
3	3.7	1.6	…	1.6	25.3	1782	75.7	
4	24.8	0.0	…	0.9	24.8	5037	67.1	

	PublicWork	SelfEmployed	FamilyWork	Unemployment	PopulationClass	\
0	18.3	4.6	0.0	5.4	Small Population	
1	16.9	6.1	0.0	13.3	Small Population	
2	23.6	12.3	0.0	6.2	Medium Population	
3	21.2	3.1	0.0	10.8	Large Population	
4	27.6	5.3	0.0	4.2	Extra Large Population	

	PovertyClass
0	Average Poverty
1	Extreme Poverty
2	High Poverty
3	Low Poverty
4	Average Poverty

```
[5 rows x 40 columns]
```

Feature Creation: Income Class The income level of each census tract is given by income per capita and median income of the population. It may be useful to bin the Income Per Capita varialbe to seperate out lower, middle, and upper class geographical areas. The code for this new variable can be seen below. This new variable will be discussed further in our New Features section.

[7]:
```
#New feature creation income class
IncomeClass = pd.cut(data2015.
 ↪IncomePerCap,bins=[0,19190,25368,33890,1000000],labels=
['Lower Class','Lower-Middle Class','Upper-Middle Class','Upper Class'])
data2015.insert(38,'IncomeClass',IncomeClass)

IncomeClass = pd.cut(data2017.
 ↪IncomePerCap,bins=[0,19190,25368,33890,1000000],labels=
['Lower Class','Lower-Middle Class','Upper-Middle Class','Upper Class'])
data2017.insert(38,'IncomeClass',IncomeClass)
data2015.head()
```

[7]:

	CensusTract	State	County	TotalPop	Men	Women	Hispanic	White	\
0	1001020100	Alabama	Autauga	1948	940	1008	0.9	87.4	
1	1001020200	Alabama	Autauga	2156	1059	1097	0.8	40.4	
2	1001020300	Alabama	Autauga	2968	1364	1604	0.0	74.5	
3	1001020400	Alabama	Autauga	4423	2172	2251	10.5	82.8	
4	1001020500	Alabama	Autauga	10763	4922	5841	0.7	68.5	

	Black	Native	…	MeanCommute	Employed	PrivateWork	PublicWork	\
0	7.7	0.3	…	25.0	943	77.1	18.3	
1	53.3	0.0	…	23.4	753	77.0	16.9	
2	18.6	0.5	…	19.6	1373	64.1	23.6	
3	3.7	1.6	…	25.3	1782	75.7	21.2	
4	24.8	0.0	…	24.8	5037	67.1	27.6	

	SelfEmployed	FamilyWork	Unemployment	IncomeClass	\
0	4.6	0.0	5.4	Upper-Middle Class	
1	6.1	0.0	13.3	Lower Class	
2	12.3	0.0	6.2	Lower-Middle Class	
3	3.1	0.0	10.8	Lower-Middle Class	
4	5.3	0.0	4.2	Upper-Middle Class	

	PopulationClass	PovertyClass
0	Small Population	Average Poverty
1	Small Population	Extreme Poverty
2	Medium Population	High Poverty
3	Large Population	Low Poverty
4	Extra Large Population	Average Poverty

```
[5 rows x 41 columns]
```

Feature Creation: Men/Women Percentage Breakdown The Men and Women variables in our data set are each given as total population numbers, respectively. However, many of our variables are given as percentages of our population. For standardization we added two new variables to reflect the percentage of men and women in each census tract. The code for these variables can be seen below. These variables will be discussed futher in our New Features section.

[8]:
```
#New feature creation: Percent of Genders
PercMen = ''
data2015.insert(6,'PercMen',PercMen)
data2017.insert(6,'PercMen',PercMen)

data2015['PercMen'] = (data2015['Men']/data2015['TotalPop'])
data2015.head()

PercWomen = ''
data2015.insert(7,'PercWomen',PercWomen)
data2017.insert(7,'PercWomen',PercWomen)

data2015['PercWomen'] = (data2015['Women']/data2015['TotalPop'])

df = pd.DataFrame(data2015)
df.to_csv('2015Census.csv')
data2015.head()
```

[8]:

	CensusTract	State	County	TotalPop	Men	Women	PercMen	PercWomen	\
0	1001020100	Alabama	Autauga	1948	940	1008	0.482546	0.517454	
1	1001020200	Alabama	Autauga	2156	1059	1097	0.491187	0.508813	
2	1001020300	Alabama	Autauga	2968	1364	1604	0.459569	0.540431	
3	1001020400	Alabama	Autauga	4423	2172	2251	0.491069	0.508931	
4	1001020500	Alabama	Autauga	10763	4922	5841	0.457307	0.542693	

	Hispanic	White	…	MeanCommute	Employed	PrivateWork	PublicWork	\
0	0.9	87.4	…	25.0	943	77.1	18.3	
1	0.8	40.4	…	23.4	753	77.0	16.9	
2	0.0	74.5	…	19.6	1373	64.1	23.6	
3	10.5	82.8	…	25.3	1782	75.7	21.2	
4	0.7	68.5	…	24.8	5037	67.1	27.6	

	SelfEmployed	FamilyWork	Unemployment	IncomeClass	\
0	4.6	0.0	5.4	Upper-Middle Class	
1	6.1	0.0	13.3	Lower Class	
2	12.3	0.0	6.2	Lower-Middle Class	
3	3.1	0.0	10.8	Lower-Middle Class	
4	5.3	0.0	4.2	Upper-Middle Class	

```
          PopulationClass     PovertyClass
0        Small Population  Average Poverty
1        Small Population  Extreme Poverty
2       Medium Population     High Poverty
3        Large Population      Low Poverty
4  Extra Large Population  Average Poverty

[5 rows x 43 columns]
```

1.3 Data Quality

1.3.1 NA Evaluation Analysis

From the below output we can see there were betwen about six hundred to about twelve hundred NA values found in most of the columns. Overall there were about 1500 instances with some values of NAs, as shown in the breakdown by column a lot of these had multiple NA values and spread across multiple attributes, this resulted in several columns with several hundred values missing.

Rather than imputation, given some instances had many NAs, we chose to remove them from the dataset. As seen in the code chunkcs below this only removed less than two percent of the original data leaving us with a varied and robust sample size to continue exploration and eventual model building. Overall no particular geographical area was removed at a disproportionate rate.

[98]:
```
#Checking for NAs
data2015agg.isnull().sum()
```

[98]:
```
CensusTract           0
State                 0
County                0
TotalPop              0
Men                   0
Women                 0
Hispanic            690
White               690
Black               690
Native              690
Asian               690
Pacific             690
Other               690
Citizen               0
Income             1100
IncomeErr          1100
IncomePerCap        740
IncomePerCapErr     740
Poverty             835
ChildPoverty       1118
Professional        807
```

```
Service             807
Office              807
Construction        807
Production          807
Drive               797
Carpool             797
Transit             797
Walk                797
OtherTransp         797
WorkAtHome          797
MeanCommute         949
Employed              0
PrivateWork         807
PublicWork          807
SelfEmployed        807
FamilyWork          807
Unemployment        802
PovertyClass        998
dtype: int64
```

[99]:
```
#NA remove and check
data2015original = data2015agg.copy()
data2015agg = data2015agg.dropna()

data2015agg.isnull().sum()
```

[99]:
```
CensusTract         0
State               0
County              0
TotalPop            0
Men                 0
Women               0
Hispanic            0
White               0
Black               0
Native              0
Asian               0
Pacific             0
Other               0
Citizen             0
Income              0
IncomeErr           0
IncomePerCap        0
IncomePerCapErr     0
Poverty             0
ChildPoverty        0
Professional        0
```

```
Service            0
Office             0
Construction       0
Production         0
Drive              0
Carpool            0
Transit            0
Walk               0
OtherTransp        0
WorkAtHome         0
MeanCommute        0
Employed           0
PrivateWork        0
PublicWork         0
SelfEmployed       0
FamilyWork         0
Unemployment       0
PovertyClass       0
dtype: int64
```

[100]:
```
#Data that has been Removed comparison as percentage of states
removed = data2015original[data2015original.isnull().any(axis=1)]
removed.groupby('State').count() /data2015agg.groupby('State').count()
```

[100]:

State	CensusTract	County	TotalPop	Men	Women	\
Alabama	0.007679	0.007679	0.007679	0.007679	0.007679	
Alaska	0.018293	0.018293	0.018293	0.018293	0.018293	
Arizona	0.031778	0.031778	0.031778	0.031778	0.031778	
Arkansas	0.004392	0.004392	0.004392	0.004392	0.004392	
California	0.016271	0.016271	0.016271	0.016271	0.016271	
Colorado	0.022095	0.022095	0.022095	0.022095	0.022095	
Connecticut	0.020833	0.020833	0.020833	0.020833	0.020833	
Delaware	0.023474	0.023474	0.023474	0.023474	0.023474	
District of Columbia	0.022857	0.022857	0.022857	0.022857	0.022857	
Florida	0.033098	0.033098	0.033098	0.033098	0.033098	
Georgia	0.010262	0.010262	0.010262	0.010262	0.010262	
Hawaii	0.147059	0.147059	0.147059	0.147059	0.147059	
Idaho	0.003367	0.003367	0.003367	0.003367	0.003367	
Illinois	0.005473	0.005473	0.005473	0.005473	0.005473	
Indiana	0.008005	0.008005	0.008005	0.008005	0.008005	
Iowa	0.004872	0.004872	0.004872	0.004872	0.004872	
Kansas	0.014493	0.014493	0.014493	0.014493	0.014493	
Kentucky	0.011797	0.011797	0.011797	0.011797	0.011797	
Louisiana	0.024086	0.024086	0.024086	0.024086	0.024086	
Maine	0.019943	0.019943	0.019943	0.019943	0.019943	
Maryland	0.023290	0.023290	0.023290	0.023290	0.023290	

Massachusetts	0.017906	0.017906	0.017906	0.017906	0.017906
Michigan	0.031536	0.031536	0.031536	0.031536	0.031536
Minnesota	0.005259	0.005259	0.005259	0.005259	0.005259
Mississippi	0.015291	0.015291	0.015291	0.015291	0.015291
Missouri	0.006503	0.006503	0.006503	0.006503	0.006503
Montana	0.011194	0.011194	0.011194	0.011194	0.011194
Nebraska	0.009488	0.009488	0.009488	0.009488	0.009488
Nevada	0.023845	0.023845	0.023845	0.023845	0.023845
New Hampshire	0.013746	0.013746	0.013746	0.013746	0.013746
New Jersey	0.015152	0.015152	0.015152	0.015152	0.015152
New Mexico	0.004024	0.004024	0.004024	0.004024	0.004024
New York	0.029732	0.029732	0.029732	0.029732	0.029732
North Carolina	0.016674	0.016674	0.016674	0.016674	0.016674
North Dakota	NaN	NaN	NaN	NaN	NaN
Ohio	0.008541	0.008541	0.008541	0.008541	0.008541
Oklahoma	0.008679	0.008679	0.008679	0.008679	0.008679
Oregon	0.010909	0.010909	0.010909	0.010909	0.010909
Pennsylvania	0.012905	0.012905	0.012905	0.012905	0.012905
Puerto Rico	0.081236	0.081236	0.081236	0.081236	0.081236
Rhode Island	0.016667	0.016667	0.016667	0.016667	0.016667
South Carolina	0.023191	0.023191	0.023191	0.023191	0.023191
South Dakota	0.004525	0.004525	0.004525	0.004525	0.004525
Tennessee	0.017675	0.017675	0.017675	0.017675	0.017675
Texas	0.013670	0.013670	0.013670	0.013670	0.013670
Utah	0.012048	0.012048	0.012048	0.012048	0.012048
Vermont	0.005464	0.005464	0.005464	0.005464	0.005464
Virginia	0.028032	0.028032	0.028032	0.028032	0.028032
Washington	0.013204	0.013204	0.013204	0.013204	0.013204
West Virginia	NaN	NaN	NaN	NaN	NaN
Wisconsin	0.016595	0.016595	0.016595	0.016595	0.016595
Wyoming	0.007634	0.007634	0.007634	0.007634	0.007634

	Hispanic	White	Black	Native	Asian	…	\
State						…	
Alabama	0.003413	0.003413	0.003413	0.003413	0.003413	…	
Alaska	0.018293	0.018293	0.018293	0.018293	0.018293	…	
Arizona	0.027721	0.027721	0.027721	0.027721	0.027721	…	
Arkansas	0.002928	0.002928	0.002928	0.002928	0.002928	…	
California	0.010595	0.010595	0.010595	0.010595	0.010595	…	
Colorado	0.016367	0.016367	0.016367	0.016367	0.016367	…	
Connecticut	0.014706	0.014706	0.014706	0.014706	0.014706	…	
Delaware	0.004695	0.004695	0.004695	0.004695	0.004695	…	
District of Columbia	0.022857	0.022857	0.022857	0.022857	0.022857	…	
Florida	0.015332	0.015332	0.015332	0.015332	0.015332	…	
Georgia	0.004105	0.004105	0.004105	0.004105	0.004105	…	
Hawaii	0.032680	0.032680	0.032680	0.032680	0.032680	…	
Idaho	0.003367	0.003367	0.003367	0.003367	0.003367	…	

Illinois	0.002898	0.002898	0.002898	0.002898	0.002898	…
Indiana	0.004003	0.004003	0.004003	0.004003	0.004003	…
Iowa	0.002436	0.002436	0.002436	0.002436	0.002436	…
Kansas	0.001318	0.001318	0.001318	0.001318	0.001318	…
Kentucky	0.006352	0.006352	0.006352	0.006352	0.006352	…
Louisiana	0.006244	0.006244	0.006244	0.006244	0.006244	…
Maine	0.000000	0.000000	0.000000	0.000000	0.000000	…
Maryland	0.010917	0.010917	0.010917	0.010917	0.010917	…
Massachusetts	0.008264	0.008264	0.008264	0.008264	0.008264	…
Michigan	0.009168	0.009168	0.009168	0.009168	0.009168	…
Minnesota	0.002254	0.002254	0.002254	0.002254	0.002254	…
Mississippi	0.006116	0.006116	0.006116	0.006116	0.006116	…
Missouri	0.002890	0.002890	0.002890	0.002890	0.002890	…
Montana	0.007463	0.007463	0.007463	0.007463	0.007463	…
Nebraska	0.009488	0.009488	0.009488	0.009488	0.009488	…
Nevada	0.011923	0.011923	0.011923	0.011923	0.011923	…
New Hampshire	0.003436	0.003436	0.003436	0.003436	0.003436	…
New Jersey	0.011111	0.011111	0.011111	0.011111	0.011111	…
New Mexico	0.002012	0.002012	0.002012	0.002012	0.002012	…
New York	0.016960	0.016960	0.016960	0.016960	0.016960	…
North Carolina	0.005558	0.005558	0.005558	0.005558	0.005558	…
North Dakota	NaN	NaN	NaN	NaN	NaN	…
Ohio	0.004783	0.004783	0.004783	0.004783	0.004783	…
Oklahoma	0.007715	0.007715	0.007715	0.007715	0.007715	…
Oregon	0.000000	0.000000	0.000000	0.000000	0.000000	…
Pennsylvania	0.007869	0.007869	0.007869	0.007869	0.007869	…
Puerto Rico	0.014874	0.014874	0.014874	0.014874	0.014874	…
Rhode Island	0.000000	0.000000	0.000000	0.000000	0.000000	…
South Carolina	0.012987	0.012987	0.012987	0.012987	0.012987	…
South Dakota	0.004525	0.004525	0.004525	0.004525	0.004525	…
Tennessee	0.008158	0.008158	0.008158	0.008158	0.008158	…
Texas	0.006353	0.006353	0.006353	0.006353	0.006353	…
Utah	0.006885	0.006885	0.006885	0.006885	0.006885	…
Vermont	0.000000	0.000000	0.000000	0.000000	0.000000	…
Virginia	0.012938	0.012938	0.012938	0.012938	0.012938	…
Washington	0.004170	0.004170	0.004170	0.004170	0.004170	…
West Virginia	NaN	NaN	NaN	NaN	NaN	…
Wisconsin	0.004329	0.004329	0.004329	0.004329	0.004329	…
Wyoming	0.000000	0.000000	0.000000	0.000000	0.000000	…

State	OtherTransp	WorkAtHome	MeanCommute	Employed	\
Alabama	0.002560	0.002560	0.002560	0.007679	
Alaska	0.012195	0.012195	0.012195	0.018293	
Arizona	0.024341	0.024341	0.020960	0.031778	
Arkansas	0.002928	0.002928	0.000000	0.004392	
California	0.007947	0.007947	0.005424	0.016271	

Colorado	0.012275	0.012275	0.011457	0.022095
Connecticut	0.013480	0.013480	0.012255	0.020833
Delaware	0.004695	0.004695	0.004695	0.023474
District of Columbia	0.022857	0.022857	0.011429	0.022857
Florida	0.014115	0.014115	0.011438	0.033098
Georgia	0.002565	0.002565	0.001539	0.010262
Hawaii	0.029412	0.029412	0.022876	0.147059
Idaho	0.000000	0.000000	0.000000	0.003367
Illinois	0.002576	0.002576	0.001932	0.005473
Indiana	0.004003	0.004003	0.002668	0.008005
Iowa	0.002436	0.002436	0.002436	0.004872
Kansas	0.000000	0.000000	0.000000	0.014493
Kentucky	0.005445	0.005445	0.003630	0.011797
Louisiana	0.005352	0.005352	0.001784	0.024086
Maine	0.000000	0.000000	0.000000	0.019943
Maryland	0.008734	0.008734	0.007278	0.023290
Massachusetts	0.006887	0.006887	0.005510	0.017906
Michigan	0.007334	0.007334	0.003300	0.031536
Minnesota	0.000751	0.000751	0.000751	0.005259
Mississippi	0.004587	0.004587	0.000000	0.015291
Missouri	0.002890	0.002890	0.001445	0.006503
Montana	0.003731	0.003731	0.003731	0.011194
Nebraska	0.003795	0.003795	0.003795	0.009488
Nevada	0.010432	0.010432	0.010432	0.023845
New Hampshire	0.003436	0.003436	0.003436	0.013746
New Jersey	0.008081	0.008081	0.006566	0.015152
New Mexico	0.002012	0.002012	0.002012	0.004024
New York	0.014866	0.014866	0.008794	0.029732
North Carolina	0.005095	0.005095	0.003705	0.016674
North Dakota	NaN	NaN	NaN	NaN
Ohio	0.004100	0.004100	0.003416	0.008541
Oklahoma	0.007715	0.007715	0.002893	0.008679
Oregon	0.000000	0.000000	0.000000	0.010909
Pennsylvania	0.005980	0.005980	0.003462	0.012905
Puerto Rico	0.013730	0.013730	0.009153	0.081236
Rhode Island	0.000000	0.000000	0.000000	0.016667
South Carolina	0.010204	0.010204	0.009276	0.023191
South Dakota	0.004525	0.004525	0.004525	0.004525
Tennessee	0.006118	0.006118	0.002719	0.017675
Texas	0.005583	0.005583	0.003851	0.013670
Utah	0.005164	0.005164	0.001721	0.012048
Vermont	0.000000	0.000000	0.000000	0.005464
Virginia	0.011321	0.011321	0.009704	0.028032
Washington	0.004170	0.004170	0.003475	0.013204
West Virginia	NaN	NaN	NaN	NaN
Wisconsin	0.003608	0.003608	0.002886	0.016595
Wyoming	0.000000	0.000000	0.000000	0.007634

State	PrivateWork	PublicWork	SelfEmployed	FamilyWork	\
Alabama	0.002560	0.002560	0.002560	0.002560	
Alaska	0.012195	0.012195	0.012195	0.012195	
Arizona	0.024341	0.024341	0.024341	0.024341	
Arkansas	0.002928	0.002928	0.002928	0.002928	
California	0.007694	0.007694	0.007694	0.007694	
Colorado	0.011457	0.011457	0.011457	0.011457	
Connecticut	0.013480	0.013480	0.013480	0.013480	
Delaware	0.004695	0.004695	0.004695	0.004695	
District of Columbia	0.022857	0.022857	0.022857	0.022857	
Florida	0.013872	0.013872	0.013872	0.013872	
Georgia	0.002565	0.002565	0.002565	0.002565	
Hawaii	0.029412	0.029412	0.029412	0.029412	
Idaho	0.000000	0.000000	0.000000	0.000000	
Illinois	0.002576	0.002576	0.002576	0.002576	
Indiana	0.004003	0.004003	0.004003	0.004003	
Iowa	0.002436	0.002436	0.002436	0.002436	
Kansas	0.000000	0.000000	0.000000	0.000000	
Kentucky	0.005445	0.005445	0.005445	0.005445	
Louisiana	0.005352	0.005352	0.005352	0.005352	
Maine	0.000000	0.000000	0.000000	0.000000	
Maryland	0.008734	0.008734	0.008734	0.008734	
Massachusetts	0.006887	0.006887	0.006887	0.006887	
Michigan	0.007334	0.007334	0.007334	0.007334	
Minnesota	0.000751	0.000751	0.000751	0.000751	
Mississippi	0.004587	0.004587	0.004587	0.004587	
Missouri	0.002890	0.002890	0.002890	0.002890	
Montana	0.003731	0.003731	0.003731	0.003731	
Nebraska	0.003795	0.003795	0.003795	0.003795	
Nevada	0.010432	0.010432	0.010432	0.010432	
New Hampshire	0.003436	0.003436	0.003436	0.003436	
New Jersey	0.008081	0.008081	0.008081	0.008081	
New Mexico	0.002012	0.002012	0.002012	0.002012	
New York	0.014657	0.014657	0.014657	0.014657	
North Carolina	0.003705	0.003705	0.003705	0.003705	
North Dakota	NaN	NaN	NaN	NaN	
Ohio	0.004100	0.004100	0.004100	0.004100	
Oklahoma	0.007715	0.007715	0.007715	0.007715	
Oregon	0.000000	0.000000	0.000000	0.000000	
Pennsylvania	0.005980	0.005980	0.005980	0.005980	
Puerto Rico	0.013730	0.013730	0.013730	0.013730	
Rhode Island	0.000000	0.000000	0.000000	0.000000	
South Carolina	0.010204	0.010204	0.010204	0.010204	
South Dakota	0.004525	0.004525	0.004525	0.004525	
Tennessee	0.005438	0.005438	0.005438	0.005438	

Texas	0.005583	0.005583	0.005583	0.005583
Utah	0.005164	0.005164	0.005164	0.005164
Vermont	0.000000	0.000000	0.000000	0.000000
Virginia	0.010782	0.010782	0.010782	0.010782
Washington	0.004170	0.004170	0.004170	0.004170
West Virginia	NaN	NaN	NaN	NaN
Wisconsin	0.003608	0.003608	0.003608	0.003608
Wyoming	0.000000	0.000000	0.000000	0.000000

State	Unemployment	PovertyClass
Alabama	0.002560	0.001706
Alaska	0.012195	0.018293
Arizona	0.024341	0.022989
Arkansas	0.002928	0.001464
California	0.007947	0.004667
Colorado	0.011457	0.005728
Connecticut	0.013480	0.009804
Delaware	0.004695	0.004695
District of Columbia	0.022857	0.017143
Florida	0.014115	0.012412
Georgia	0.002565	0.002052
Hawaii	0.029412	0.009804
Idaho	0.000000	0.000000
Illinois	0.002898	0.002576
Indiana	0.004003	0.002668
Iowa	0.002436	0.002436
Kansas	0.000000	0.000000
Kentucky	0.005445	0.002722
Louisiana	0.005352	0.003568
Maine	0.000000	0.000000
Maryland	0.008734	0.005822
Massachusetts	0.006887	0.003444
Michigan	0.007334	0.004034
Minnesota	0.000751	0.000000
Mississippi	0.004587	0.004587
Missouri	0.002890	0.001445
Montana	0.003731	0.003731
Nebraska	0.003795	0.000000
Nevada	0.010432	0.010432
New Hampshire	0.003436	0.000000
New Jersey	0.008081	0.005556
New Mexico	0.002012	0.000000
New York	0.014866	0.006700
North Carolina	0.003705	0.000926
North Dakota	NaN	NaN
Ohio	0.004100	0.002392

```
Oklahoma                0.007715      0.005786
Oregon                  0.000000      0.000000
Pennsylvania            0.005980      0.003148
Puerto Rico             0.013730      0.012586
Rhode Island            0.000000      0.000000
South Carolina          0.010204      0.006494
South Dakota            0.004525      0.000000
Tennessee               0.005438      0.002039
Texas                   0.005583      0.002888
Utah                    0.005164      0.000000
Vermont                 0.000000      0.000000
Virginia                0.010782      0.005391
Washington              0.004170      0.000695
West Virginia                NaN           NaN
Wisconsin               0.003608      0.003608
Wyoming                 0.000000      0.000000

[52 rows x 38 columns]
```

[101]:
```
perc = 72671/74001
print("Remaining Data: ", perc)
```

```
Remaining Data:  0.982027269901758
```

1.3.2 Duplicate Data Analysis

This data was overall quite clean and without duplication, each census tract occurs once and represents a specific geographical location in the country. There may be census tract with some shared occurrences, but that is expected.

[102]:
```
#Duplicates
print("Number of Duplicate Census Tracts Recorded:", data2015agg.
 ↪duplicated(subset = "CensusTract").sum())
```

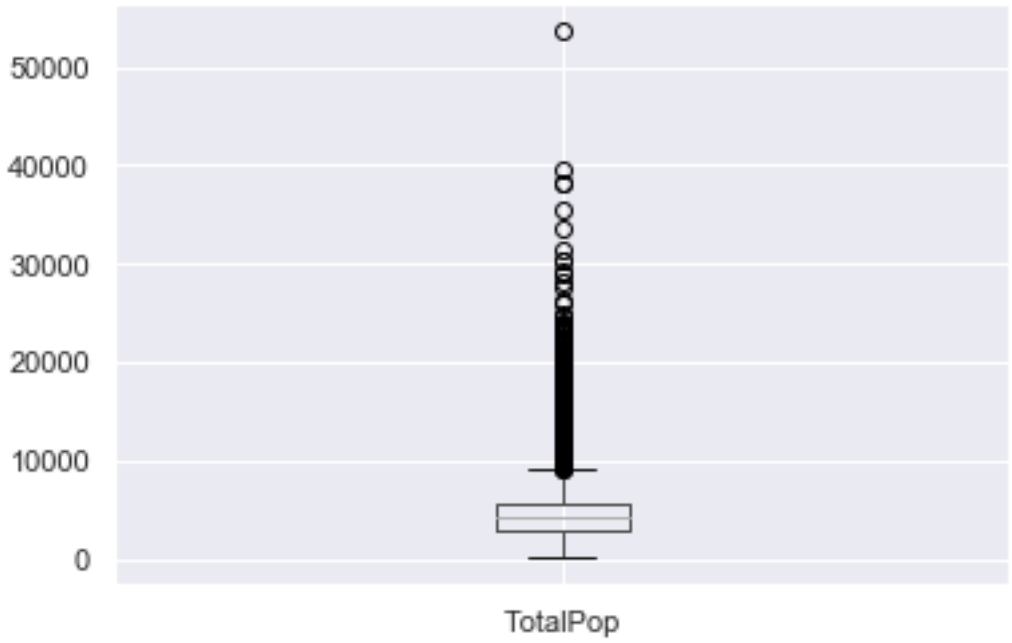
```
Number of Duplicate Census Tracts Recorded: 0
```

1.3.3 Outliers Analysis

Overall any columns with population counts demonstrate major skew, as shown in the boxplots outlined in the visualizations section. Several larger census tracts are well above the population of other tracts and may require examination as we build models but for now can remain in the data without issue.

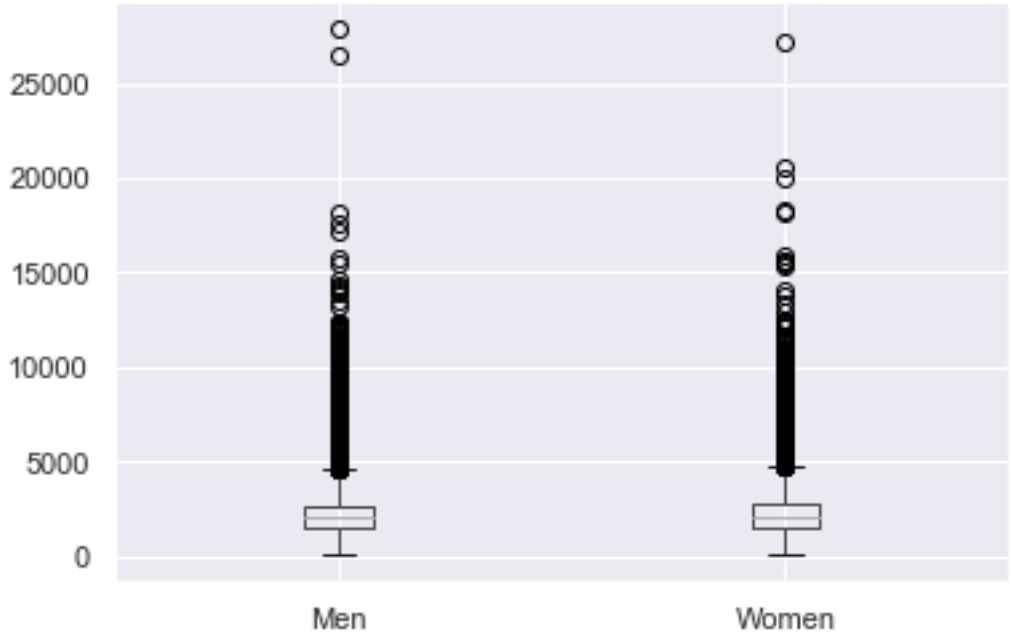
[103]:
```
#Outliers
#Total Population
data2015agg.boxplot(column = "TotalPop")
```

[103]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x155cf7f50>
```



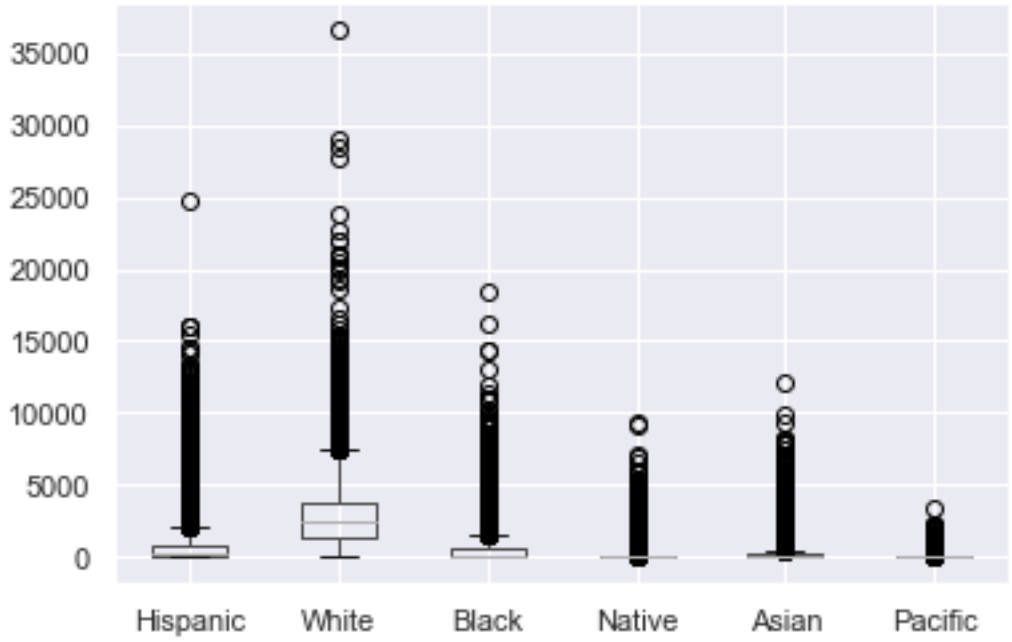
[104]:
```
#Men and Women are both total counts and will skew like total population
#Men and Women Columns
data2015agg.boxplot(column = ["Men","Women"])
```

[104]: <matplotlib.axes._subplots.AxesSubplot at 0x14965dd10>



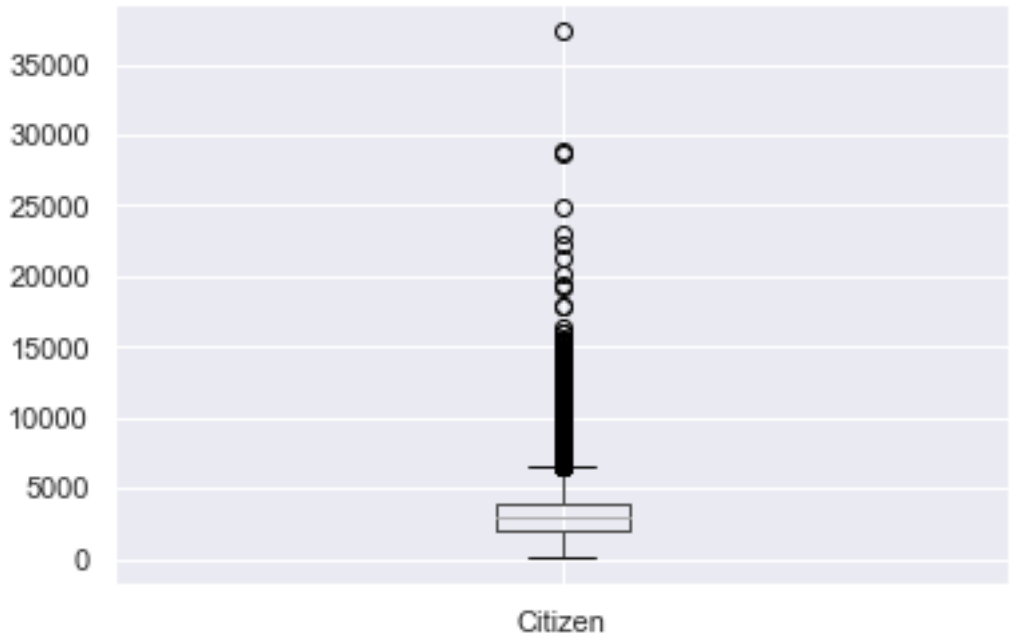
[105]:
```
#Racial Column Comparsions
data2015agg.boxplot(column =␣
↪["Hispanic","White","Black","Native","Asian","Pacific"])
```

[105]: <matplotlib.axes._subplots.AxesSubplot at 0x14acc0990>



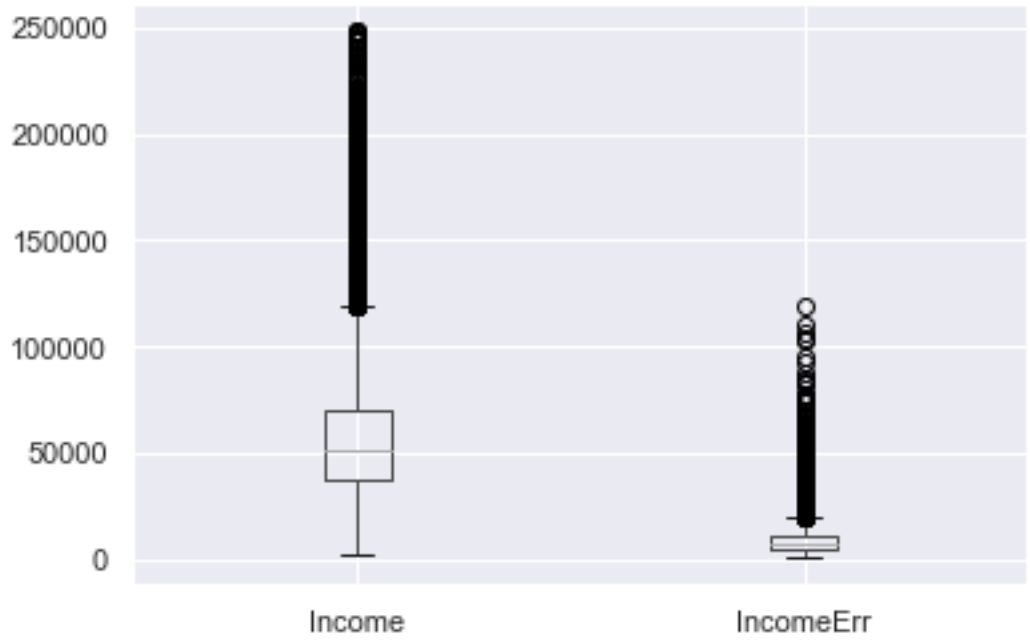
[106]:
```
#Citizen Column
#Perct Citizen of Total Pop may be a better function of this.
data2015agg.boxplot("Citizen")
```

[106]: <matplotlib.axes._subplots.AxesSubplot at 0x1950fe710>



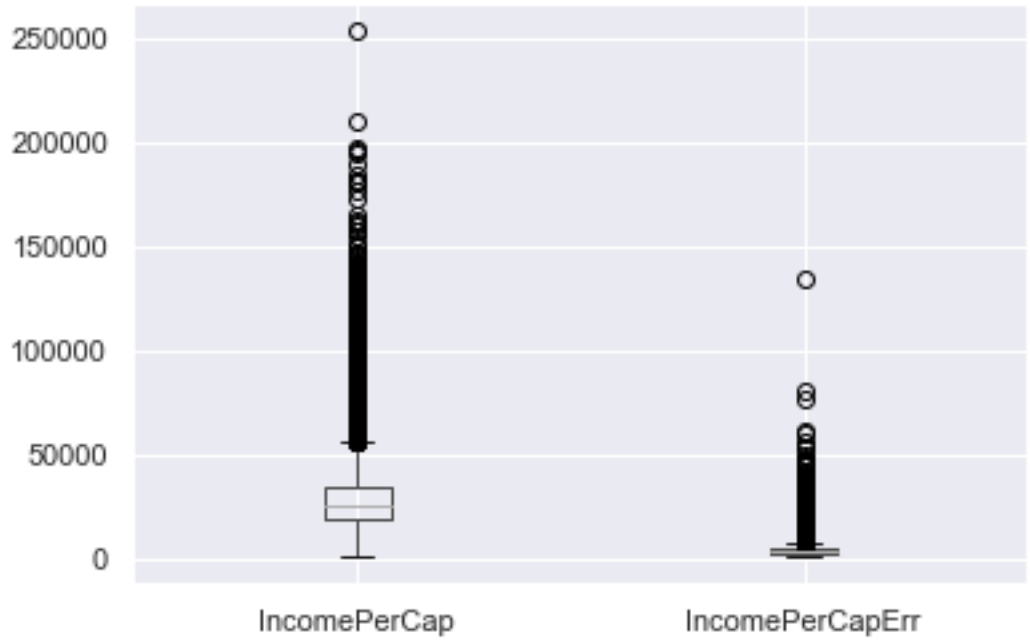
[107]:
```
#Income Column
#Calculated  column combining the two may be helpful , maybe max (income +
 ↪incomeerr) and min (income - err)
#Income reflects household income
data2015agg.boxplot(["Income", "IncomeErr"])
```

[107]: <matplotlib.axes._subplots.AxesSubplot at 0x19a0d8e50>



[108]:
```
#Incomes per cap
#Reflects per person incomes
data2015agg.boxplot(["IncomePerCap", "IncomePerCapErr"])
```

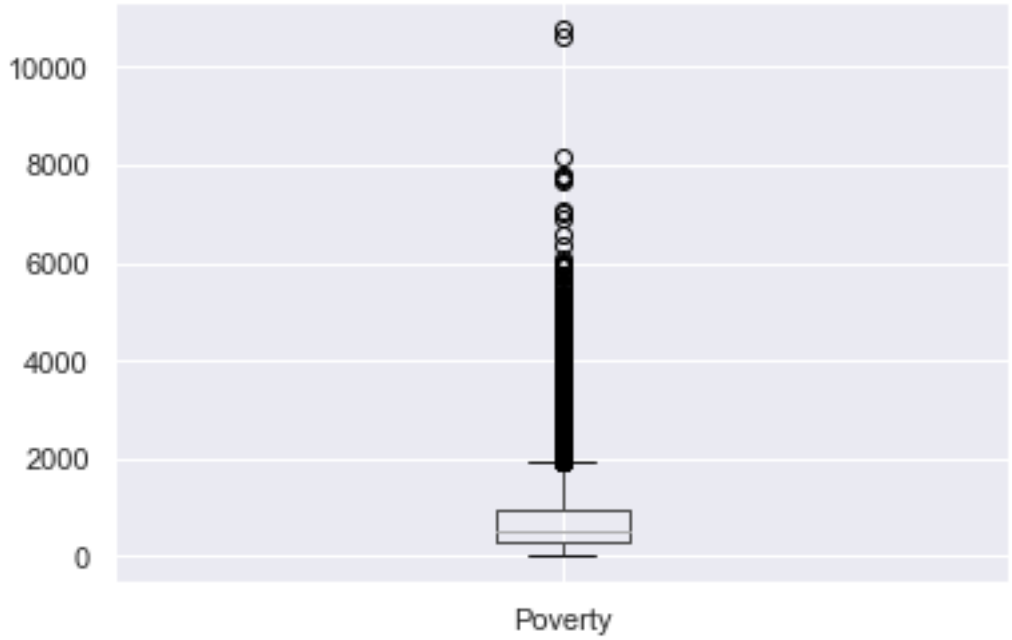
[108]: <matplotlib.axes._subplots.AxesSubplot at 0x14acdfa10>



Poverty Analysis First below we can see the spreads of Poverty and Child Poverty. Both are centered in the lower levels of poverty, with a few outliers that have larger populations of impoverished people. These outliers are a result of a combination of a high percentile of poverty and of a potentially large population in the tract explored. To manage the outliers present in the data, we binned the poverty variable as a class variable.

[109]:
```
#Poverty
data2015agg.boxplot("Poverty")
```

[109]: <matplotlib.axes._subplots.AxesSubplot at 0x15b29b250>



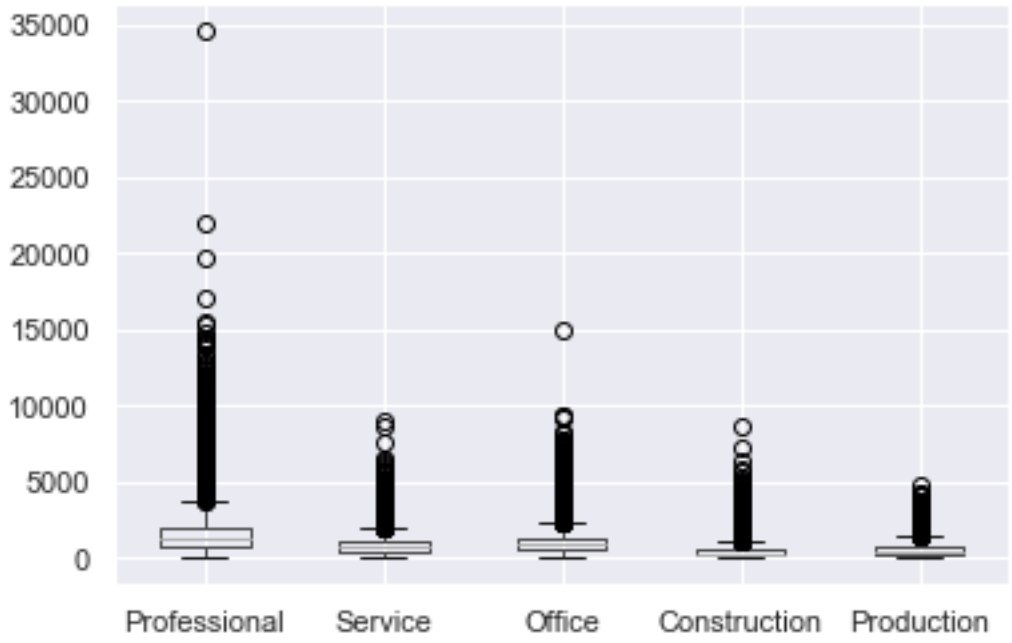
[110]:
```
#Child Poverty
data2015agg.boxplot("ChildPoverty")
```

[110]: <matplotlib.axes._subplots.AxesSubplot at 0x150b57910>

Job Function Variable Analysis Next, we can see the types of jobs grouped below and their spreads. There are a few instances of tracts having high proportions of one job or another and these stand out as outliers

[111]:
```
#Types of Jobs Columns
data2015agg.
 ↪boxplot(["Professional","Service","Office","Construction","Production"])
```

[111]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x159b7c050>
```

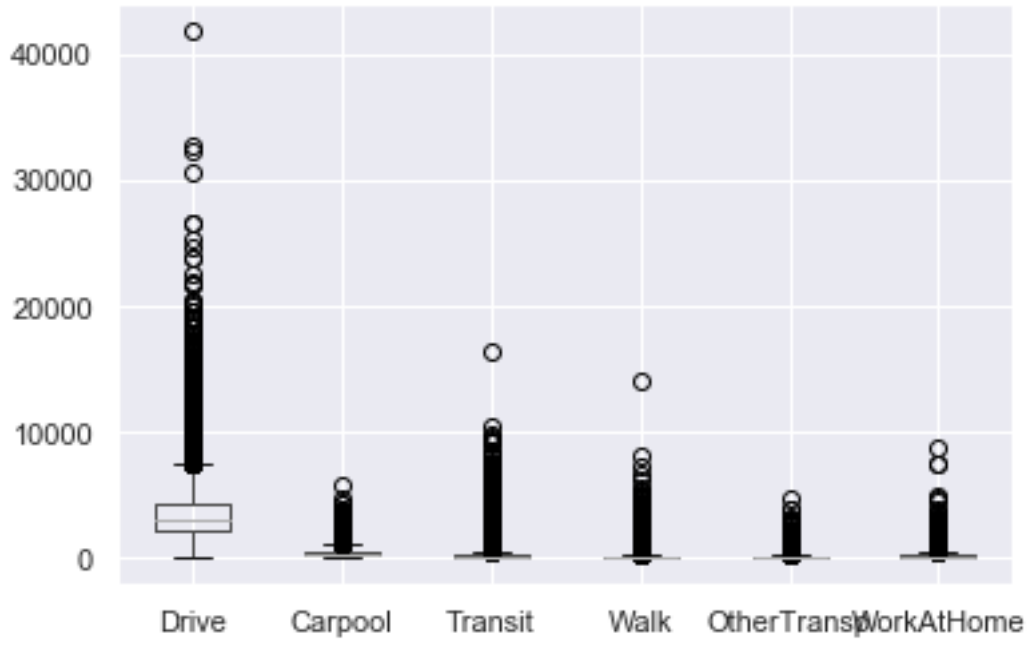


Transportation Variable Analysis Here we gave see the types of transportation to work proportions grouped below and their spreads. There are a few instances of tracts having exceedingly high proportions, but those are usually down to particularly a few tracts with little leverage on the data as a whole.

Related to this, below we have mean commute time per tract, there is a decenty wide spread, and only an outlier or two that don't stand out as far from the pack overall.

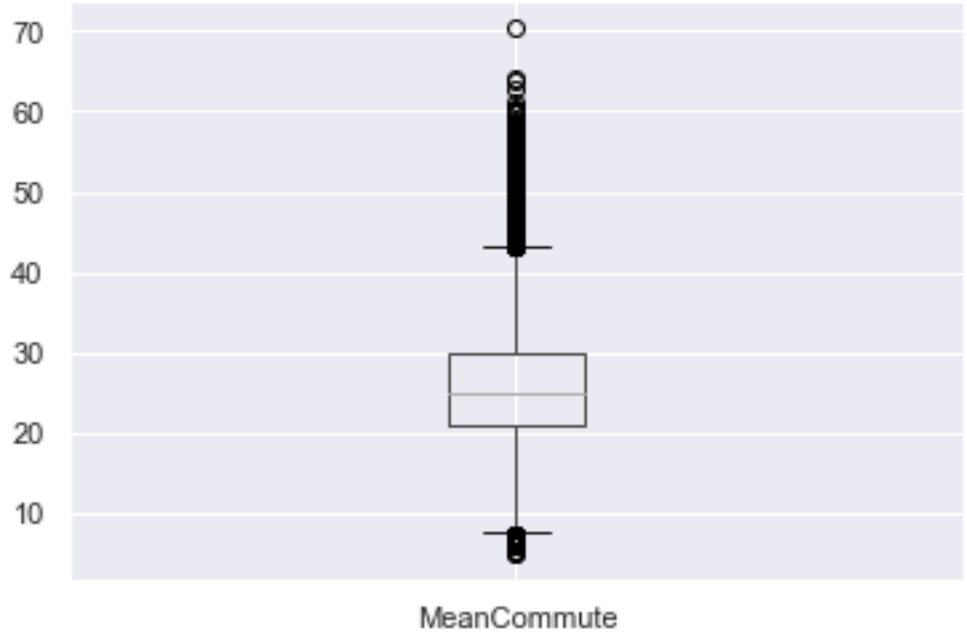
[112]:
```
#Types of Transportations

data2015agg.
 ↪boxplot(["Drive","Carpool","Transit","Walk","OtherTransp","WorkAtHome"])
```

[112]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x13ca91e10>
```



[113]:
```
#Mean Commute Time
data2015agg.boxplot("MeanCommute")
```

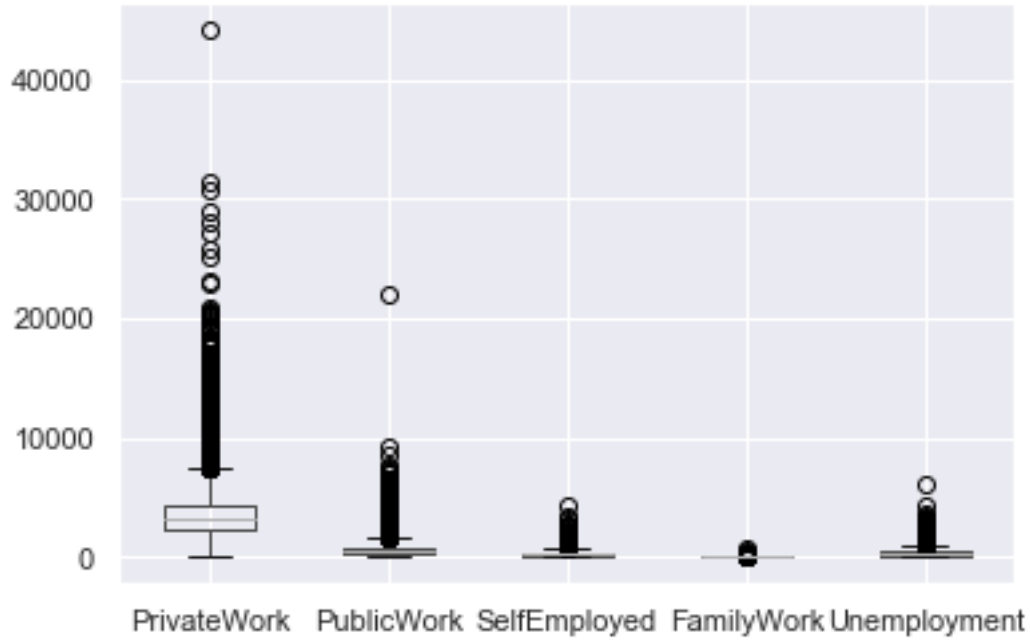
[113]: <matplotlib.axes._subplots.AxesSubplot at 0x196eaab90>



Employmnet Variable Anlaysis Finally we can see employment and job proportion as distributions in the tracts. There are few outliers we can see are likely connected to total population.

[114]:
```
data2015agg.
↪boxplot(["PrivateWork","PublicWork","SelfEmployed","FamilyWork","Unemployment"])
```

[114]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x198fabd50>
```



[115]:
```
data2015agg.boxplot(column=["Employed"])
```

[115]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x14cf78710>
```

Plotted Outlier Analysis With variables plotted, we can explore a few outlier portions.

There are a few dozen tracts with high incomes in locations that are not surprising.

There are also several tracts with larger populations than most the spread, likely in census tracts that have grown since they were last assigned.

[116]:
```
#Check on Some Outliers

data2015agg[data2015agg["Income"] > 230000]

#High incomes show up in expected places, DC, LA, Silicon Valley, Connecticut,␣
↪NYC etc.
```

[116]:

	CensusTract	State	County	TotalPop	\
5563	6037262400	California	Los Angeles	3030	
6842	6037700600	California	Los Angeles	4998	
7634	6059062643	California	Orange	7120	
10297	6081605700	California	San Mateo	5392	
10368	6081611400	California	San Mateo	4419	
10378	6081612700	California	San Mateo	2113	
10384	6081613400	California	San Mateo	3244	
10696	6085507002	California	Santa Clara	2956	
12868	9001010201	Connecticut	Fairfield	3344	
12913	9001035300	Connecticut	Fairfield	4030	
12937	9001045101	Connecticut	Fairfield	4275	

13929	11001000901	District of Columbia	District of Columbia	7790
17886	12109020708	Florida	St. Johns	3241
30484	24031700608	Maryland	Montgomery	5858
30672	24031706008	Maryland	Montgomery	5070
30673	24031706009	Maryland	Montgomery	5359
45713	36061013000	New York	New York	3277
45724	36061014200	New York	New York	4467
48009	36119009800	New York	Westchester	4610
62889	48113013300	Texas	Dallas	2072
63180	48113019700	Texas	Dallas	1740
64278	48201411200	Texas	Harris	1823
64353	48201430600	Texas	Harris	3842
68119	51059470100	Virginia	Fairfax	2913
68136	51059480100	Virginia	Fairfax	4354
68137	51059480201	Virginia	Fairfax	4500

	Men	Women	Hispanic	White	Black	Native	…	OtherTransp	\
5563	1478	1552	187.860	2469.450	21.210	0.000	…	30.300	
6842	2373	2625	414.834	4293.282	39.984	4.998	…	54.978	
7634	3699	3421	263.440	4777.520	0.000	28.480	…	142.400	
10297	2823	2569	258.816	3450.880	75.488	0.000	…	97.056	
10368	2202	2217	234.207	3486.591	13.257	0.000	…	61.866	
10378	1002	1111	84.520	1857.327	0.000	0.000	…	101.424	
10384	1611	1633	243.300	2799.572	0.000	0.000	…	81.100	
10696	1618	1338	156.668	2246.560	70.944	0.000	…	65.032	
12868	1650	1694	227.392	2782.208	143.792	0.000	…	56.848	
12913	1964	2066	245.830	3554.460	4.030	0.000	…	88.660	
12937	2147	2128	47.025	3894.525	25.650	4.275	…	17.100	
13929	3273	4517	934.800	5671.120	599.830	0.000	…	467.400	
17886	1603	1638	142.604	2865.044	100.471	25.928	…	19.446	
30484	2901	2957	492.072	3180.894	87.870	0.000	…	76.154	
30672	2458	2612	167.310	3822.780	15.210	0.000	…	182.520	
30673	2724	2635	310.822	4115.712	150.052	0.000	…	150.052	
45713	1407	1870	157.296	2932.915	13.108	0.000	…	501.381	
45724	1862	2605	218.883	3935.427	8.934	35.736	…	844.263	
48009	2124	2486	271.990	3747.930	23.050	0.000	…	119.860	
62889	1025	1047	97.384	1732.192	47.656	0.000	…	109.816	
63180	873	867	97.440	1564.260	0.000	5.220	…	0.000	
64278	888	935	92.973	1587.833	5.469	0.000	…	0.000	
64353	1901	1941	265.098	2935.288	15.368	0.000	…	80.682	
68119	1471	1442	262.170	2097.360	72.825	0.000	…	145.650	
68136	2074	2280	139.328	3544.156	108.850	0.000	…	26.124	
68137	2181	2319	85.500	3303.000	162.000	0.000	…	0.000	

	WorkAtHome	MeanCommute	Employed	PrivateWork	PublicWork	\
5563	439.350	30.9	1212	2466.420	124.230	
6842	1169.532	23.2	2252	3958.416	229.908	

7634	1160.560	29.7	3537	5817.040	512.640
10297	501.456	28.3	2398	4362.128	361.264
10368	649.593	25.1	1662	3473.334	123.732
10378	194.396	23.3	857	1760.129	131.006
10384	489.844	28.6	1615	2780.108	90.832
10696	227.612	29.4	1332	2373.668	141.888
12868	451.440	39.4	1208	2962.784	130.416
12913	386.880	38.5	1293	3554.460	193.440
12937	384.750	38.0	1930	3620.925	410.400
13929	1191.870	27.3	2613	5632.170	1223.030
17886	139.363	24.2	1439	2952.551	168.532
30484	732.250	37.1	2870	4662.968	896.274
30672	496.860	31.0	2453	3772.080	679.380
30673	610.926	33.8	2576	3997.814	755.619
45713	403.071	21.2	1399	2379.102	232.667
45724	665.583	24.1	2122	3743.346	58.071
48009	732.990	45.3	1846	3508.210	244.330
62889	236.208	21.4	1057	1715.616	101.528
63180	137.460	14.8	729	1270.200	78.300
64278	196.884	17.4	916	1418.294	158.601
64353	311.202	22.1	1472	3250.332	215.152
68119	364.125	29.2	1239	1858.494	646.686
68136	539.896	32.1	1825	3226.314	500.710
68137	445.500	30.5	1954	3505.500	643.500

	SelfEmployed	FamilyWork	Unemployment	PovertyClass
5563	424.200	15.150	221.190	Low Poverty
6842	789.684	24.990	219.912	Low Poverty
7634	719.120	71.200	206.480	Low Poverty
10297	668.608	0.000	188.720	Low Poverty
10368	773.325	48.609	318.168	Low Poverty
10378	221.865	0.000	14.791	Low Poverty
10384	360.084	12.976	68.124	Low Poverty
10696	440.444	0.000	153.712	Average Poverty
12868	210.672	40.128	147.136	Low Poverty
12913	282.100	0.000	261.950	Low Poverty
12937	239.400	0.000	68.400	Low Poverty
13929	810.160	116.850	747.840	Average Poverty
17886	119.917	0.000	165.291	Low Poverty
30484	287.042	11.716	158.166	Low Poverty
30672	618.540	0.000	218.010	Low Poverty
30673	568.054	42.872	85.744	Low Poverty
45713	563.644	101.587	137.634	Average Poverty
45724	665.583	0.000	84.873	Low Poverty
48009	834.410	23.050	281.210	Low Poverty
62889	244.496	10.360	97.384	Low Poverty
63180	351.480	40.020	81.780	Low Poverty

```
64278       231.521      14.584         0.000      Low Poverty
64353       376.516       0.000        42.262      Low Poverty
68119       396.168      11.652       110.694      Low Poverty
68136       626.976       0.000       222.054      Low Poverty
68137       351.000       0.000       162.000      Low Poverty

[26 rows x 39 columns]
```

[117]:
```
data2015[data2015["TotalPop"]>30000]
#Couple really high density Tracts
```

[117]:
```
       CensusTract         State      County  TotalPop    Men  Women  \
9724    6073018700    California   San Diego     39454  27962  11492
16886  12095016730       Florida      Orange     30258  14659  15599
18070  12115002712       Florida    Sarasota     35527  17250  18277
18180  12119911200       Florida      Sumter     38169  17638  20531
50374  38017040500  North Dakota        Cass     30256  14321  15935
62174  48039660602         Texas    Brazoria     33655  15473  18182
63652  48157672900         Texas   Fort Bend     38137  18139  19998
63656  48157673101         Texas   Fort Bend     53812  26562  27250
65439  48339692001         Texas  Montgomery     31493  15813  15680

        PercMen  PercWomen  Hispanic  White  …  MeanCommute  Employed  \
9724   0.708724   0.291276      23.0   60.5  …         12.3      3548
16886  0.484467   0.515533      35.1   46.8  …         30.5     15521
18070  0.485546   0.514454       8.5   78.1  …         29.5     13962
18180  0.462103   0.537897       1.4   96.0  …         23.2      5407
50374  0.473328   0.526672       1.5   93.7  …         17.8     18538
62174  0.459753   0.540247      22.8   27.1  …         35.1     18425
63652  0.475627   0.524373      23.9   22.2  …         38.5     18577
63656  0.493607   0.506393      18.5   53.9  …         36.5     24075
65439  0.502112   0.497888      17.0   71.8  …         34.5     15828

       PrivateWork  PublicWork  SelfEmployed  FamilyWork  Unemployment  \
9724          41.4        55.7           2.9         0.0          15.5
16886         83.2        13.9           2.8         0.0           6.1
18070         81.5        13.8           4.4         0.2           7.1
18180         80.4        10.8           8.9         0.0           8.0
50374         85.1        11.8           3.1         0.0           2.3
62174         80.4        16.0           3.6         0.0           2.3
63652         82.4        14.7           2.4         0.5           6.7
63656         82.2        12.3           5.2         0.3           3.0
65439         89.0         8.8           2.2         0.0           3.8

                IncomeClass         PopulationClass     PovertyClass
9724           Lower Class  Extra Large Population  Average Poverty
16886  Upper-Middle Class  Extra Large Population      Low Poverty
```

```
18070  Lower-Middle Class  Extra Large Population     High Poverty
18180         Upper Class  Extra Large Population      Low Poverty
50374         Upper Class  Extra Large Population      Low Poverty
62174         Upper Class  Extra Large Population      Low Poverty
63652  Upper-Middle Class  Extra Large Population      Low Poverty
63656         Upper Class  Extra Large Population      Low Poverty
65439         Upper Class  Extra Large Population      Low Poverty

[9 rows x 43 columns]
```

1.4 Simple Statistics

1.4.1 data2015

This is an analysis on the raw dataset imported into our notebook. The original data set looks very simple however, upon closer investigation you see that a large portion of the variables are percentages (ethnicities, job functions, work types, and transit functions). Therefore, we saw very low numbers compared to the few variables that are not percentages (TotalPop, Men, Women, income variables, and citizen). Due to this we will call out a few interesting facts within these different variable clusters that are related.

Within the TotalPop, Men, and Women variables we see data split pretty evenly. The quartiles are similar and min and maxs are only 50 or so persons apart.

For the race variable cluster we noticed that the Native and Pacific ethnicities have the lowest means. They don't even reach the total of the Other variable that was manually calculated. These ethnic groups are very small and therefore likely under-represented in the community. However, if they have a strong impact with Poverty later on it would be important to note.

When reviewing the poverty variable cluster we noticed that the ChildPoverty 50% and 75% quartiles are slightly higher than its counterpart Poverty. This was a little disconcerting due to the fact that we thought this would be a component of Poverty as a whole, and therefore smaller than Poverty.

In the job function variable cluster we observed that the Professional variable holds the highest percentages in the 25, 50, 75 quartiles. While all job groups rounded out their minimum and maximums at 0 and 100 respectively. We also noted that the Construction and Production variables were very even in their quartile ranges.

Upon reviewing the transportation mode variable cluster we see that there is an overwhelming majority who fall into the Drive or Carpool category. We expect this to reflect as a linear relationship in our scatterplots later on.

Lastly, we reviewed the employment variable cluster. Here we see the large majority claim private sector employment. This is overwhelmingly in the lead with over triple the mean of the closest other variable: PublicWork. The FamilyWork variable is by far the farthest behind, even behind Unemployment which was an interesting find. Family work has the lowest mean, quartiles at 0, and the max does not even reach 100 (like every other variable).

[118]:
```
#Glance at top records of data2015 dataset
#data2015.head()
```

[119]:
```
#Glance at simple statistics of data2015 dataset
data2015.describe()
```

[119]:

	CensusTract	TotalPop	Men	Women	PercMen
count	7.400100e+04	74001.000000	74001.000000	74001.000000	73311.000000
mean	2.839113e+10	4325.591465	2127.648816	2197.942649	0.491956
std	1.647593e+10	2129.306903	1072.332031	1095.730931	0.047507
min	1.001020e+09	0.000000	0.000000	0.000000	0.000000
25%	1.303901e+10	2891.000000	1409.000000	1461.000000	0.468835
50%	2.804700e+10	4063.000000	1986.000000	2066.000000	0.490535
75%	4.200341e+10	5442.000000	2674.000000	2774.000000	0.511406
max	7.215375e+10	53812.000000	27962.000000	27250.000000	1.000000

	PercWomen	Hispanic	White	Black	Native
count	73311.000000	73311.000000	73311.000000	73311.000000	73311.000000
mean	0.508044	16.862810	62.032106	13.272581	0.727726
std	0.047507	22.940695	30.684152	21.762483	4.488340
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.488594	2.400000	39.400000	0.700000	0.000000
50%	0.509465	7.000000	71.400000	3.700000	0.000000
75%	0.531165	20.400000	88.300000	14.400000	0.400000
max	1.000000	100.000000	100.000000	100.000000	100.000000

	…	Walk	OtherTransp	WorkAtHome	MeanCommute
count	…	73204.000000	73204.000000	73204.000000	73052.000000
mean	…	3.123340	1.891606	4.368093	25.667357
std	…	5.881237	2.596198	3.904990	6.964881
min	…	0.000000	0.000000	0.000000	1.200000
25%	…	0.400000	0.400000	1.800000	20.800000
50%	…	1.400000	1.100000	3.500000	25.000000
75%	…	3.500000	2.500000	5.900000	29.800000
max	…	100.000000	100.000000	100.000000	80.000000

	Employed	PrivateWork	PublicWork	SelfEmployed	FamilyWork
count	74001.000000	73194.000000	73194.000000	73194.000000	73194.000000
mean	1983.907366	78.975238	14.621566	6.233814	0.169772
std	1073.429808	8.345758	7.535786	4.042990	0.458227
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1249.000000	74.600000	9.600000	3.500000	0.000000
50%	1846.000000	80.100000	13.400000	5.500000	0.000000
75%	2553.000000	84.600000	18.200000	8.100000	0.000000
max	24075.000000	100.000000	100.000000	100.000000	26.500000

	Unemployment
count	73199.000000
mean	9.028663
std	5.955441

```
min        0.000000
25%        5.100000
50%        7.700000
75%       11.400000
max      100.000000

[8 rows x 38 columns]
```

1.4.2 data2015agg

This data set contains manipulated data variables that are aggregate values and have replaced all the original data variables that were percentage. This allowed us to visualize the data in a different way and compare all the metrics more easily to each other.

As we have done a thorough analysis of the data above, this is merely back transformed aggregates of the same data. This means the data will reflect similar quartiles ranges with just different totals. That being said, we have included a few additional noteable facts below.

We observed that while we can have extreme poverty there is never a zero income. There are also many income variables and it will be important for us to review these in more detail to make sure we are not receiving duplicate information from different variables. The poverty variable has been transitioned into a poverty class variable and will need to be dropped in order not to affect modeling negatively.

Lastly, we again see that ChildPoverty is higher than overall poverty confirming our strange observation that somehow ChildPoverty is not just a component of Poverty as a whole.

[120]:
```
#Glance at top records of data2015agg dataset
#data2015agg.head()
```

[121]:
```
#Glance at simple statistics of data2015agg dataset
data2015agg.describe()
```

[121]:
```
        CensusTract      TotalPop           Men         Women      Hispanic  \
count  7.267100e+04  72671.000000  72671.000000  72671.000000  72671.000000
mean   2.837860e+10   4384.921757   2154.407191   2230.514566    792.328427
std    1.644041e+10   2087.444919   1049.758408   1072.290816   1258.682851
min    1.001020e+09     41.000000     16.000000     25.000000      0.000000
25%    1.305100e+10   2945.000000   1435.000000   1490.000000     87.209500
50%    2.804700e+10   4099.000000   2003.000000   2086.000000    282.555000
75%    4.200341e+10   5468.000000   2686.000000   2788.000000    885.344500
max    7.215375e+10  53812.000000  27962.000000  27250.000000  24793.824000

              White         Black        Native         Asian       Pacific  \
count  72671.000000  72671.000000  72671.000000  72671.000000  72671.000000
mean    2702.709017    530.525038     28.479432    219.790890      6.845628
std     1861.599602    900.982402    178.493189    477.838773     50.082200
min        0.000000      0.000000      0.000000      0.000000      0.000000
25%     1319.085000     28.676000      0.000000      6.102000      0.000000
```

```
50%     2524.325000    152.315000      0.000000     54.260000      0.000000
75%     3801.517500    609.666000     15.394500    214.729000      0.000000
max    36642.240000  18414.819000   9413.376000  12179.565000   3441.480000

       …          Walk   OtherTransp    WorkAtHome   MeanCommute  \
count  …  72671.000000  72671.000000  72671.000000  72671.000000
mean   …    119.156109     79.314357    188.461188     25.690735
std    …    246.301025    116.641495    194.226328      6.932892
min    …      0.000000      0.000000      0.000000      4.900000
25%    …     16.172000     14.193000     66.574500     20.900000
50%    …     56.736000     45.570000    139.510000     25.000000
75%    …    132.707000    102.848500    252.371000     29.800000
max    …  14085.078000   4719.780000   8841.654000     70.500000

           Employed   PrivateWork    PublicWork  SelfEmployed    FamilyWork  \
count  72671.000000  72671.000000  72671.000000  72671.000000  72671.000000
mean    2016.064372   3471.691812    639.623452    266.472936      7.149024
std     1053.626941   1701.272728    472.617252    197.326796     19.652160
min       20.000000     15.369000      0.000000      0.000000      0.000000
25%     1280.000000   2298.474000    333.185500    128.727000      0.000000
50%     1867.000000   3234.168000    530.738000    223.392000      0.000000
75%     2568.000000   4352.250000    815.123500    354.382000      0.000000
max    24075.000000  44233.464000  21975.878000   4284.776000    852.240000

       Unemployment
count  72671.000000
mean     379.581399
std      279.257776
min        0.000000
25%      187.962000
50%      313.044000
75%      493.776500
max     6115.370000

[8 rows x 36 columns]
```

1.5 Visualize Attributes

Using an aggregated data set allows us to view our percentile variables as actual population numbers among our census tracts, but these changes mean the distributions of these variables follow the trend of our total populaiton trend. For that reason we visualized the violin plot of total population of all our census tracts.

Most of the tracts fall in a fairly normal distribution but there is a large skew of larger populated census tracts. Whenever we visualize our data, we can remember it will be influenced by this trend of total population when using actual population counts.

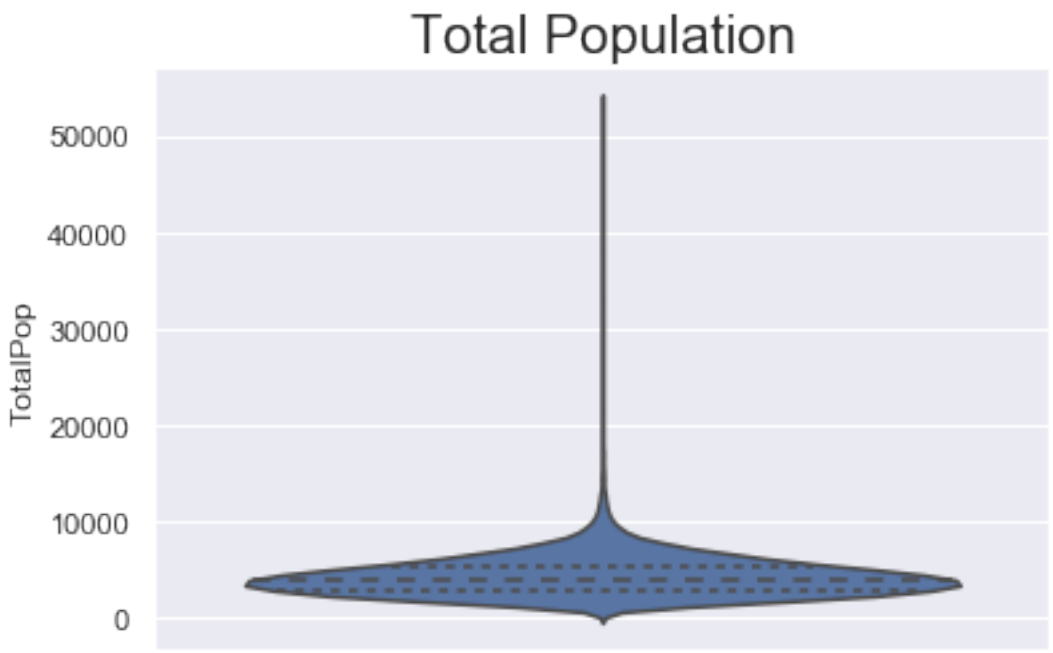
[122]:
```
import matplotlib.pyplot as plt
import warnings
warnings.simplefilter('ignore', DeprecationWarning)
%matplotlib inline
```

[123]:
```
#Import seaborn for better scatterplot visualization
import seaborn as sns
```

1.5.1 Total Population Analysis

[124]:
```
#Violin Plot for Total Population
sns.violinplot(y = "TotalPop", data = data2015agg, inner = "quart").
 ↪set_title('Total Population', fontsize = 20)
```

[124]:
```
Text(0.5, 1.0, 'Total Population')
```



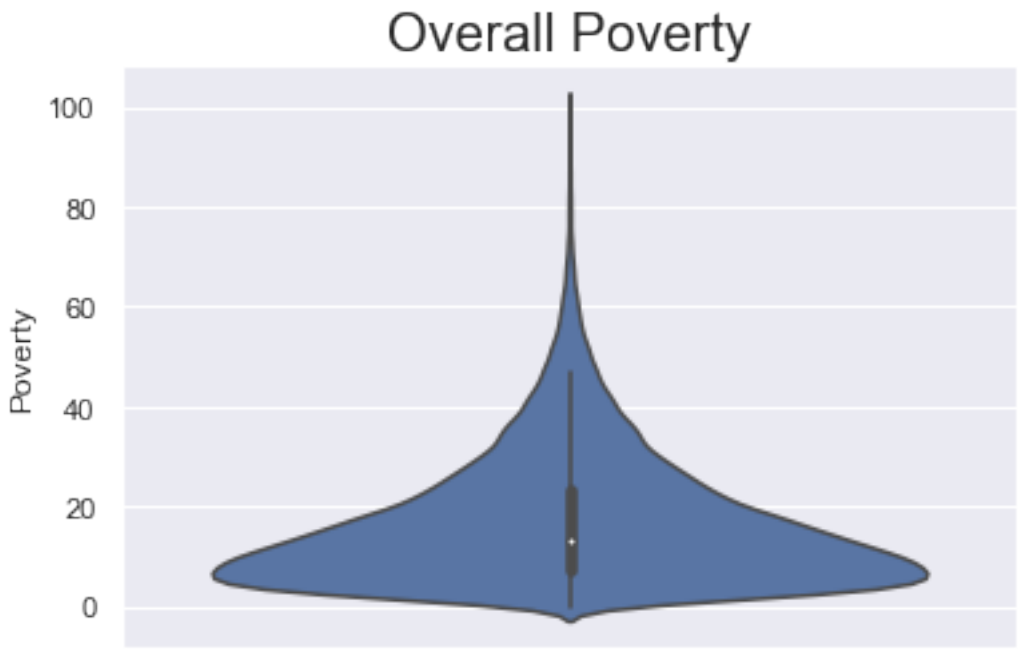
1.5.2 Child Poverty and Poverty Analysis

Using the original data set we can see the distribution of both Poverty and Child Poverty in our data. A binned version of Poverty, poverty class further defined in new features is our predicted variable. It can help us to see here how the original variable was distributed, and how we made our choices when binning the variable.

Child Poverty may also come to be a useful variable as we build our model. We can see it has a very different distribution from poverty alone, and indicates the factors behind child poverty while potentially similar to poverty as a whole work in a different method.

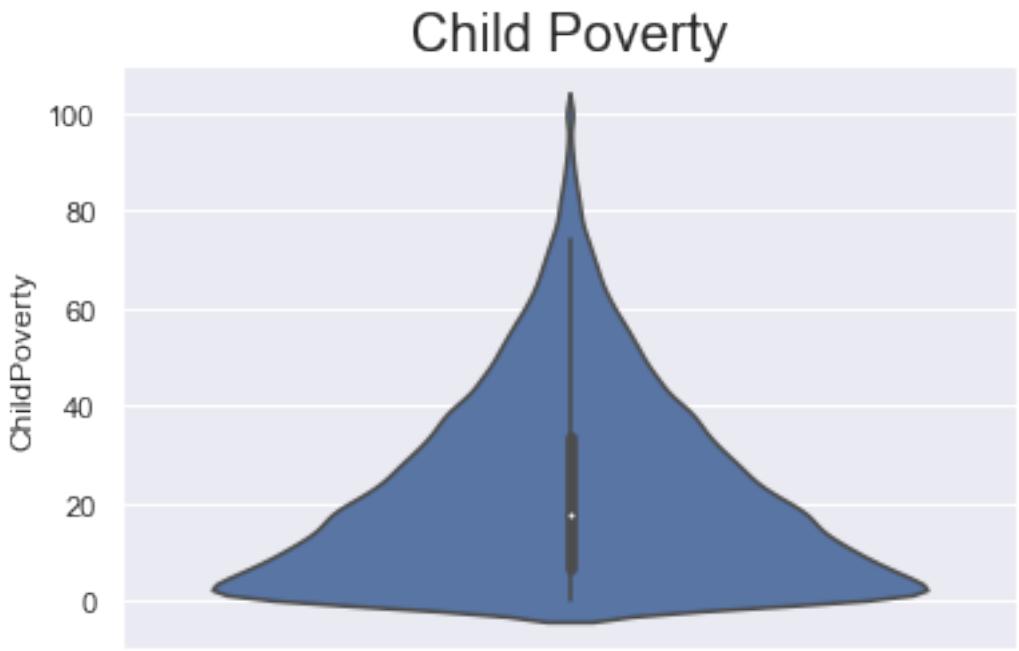
[125]:
```
sns.violinplot(y = "Poverty", data = data2015, split = True).set_title('Overall␣
↪Poverty', fontsize = 20)
```

[125]:
```
Text(0.5, 1.0, 'Overall Poverty')
```



[126]:
```
sns.violinplot(y = "ChildPoverty", data = data2015, split = True).
↪set_title('Child Poverty', fontsize = 20)
```

[126]:
```
Text(0.5, 1.0, 'Child Poverty')
```



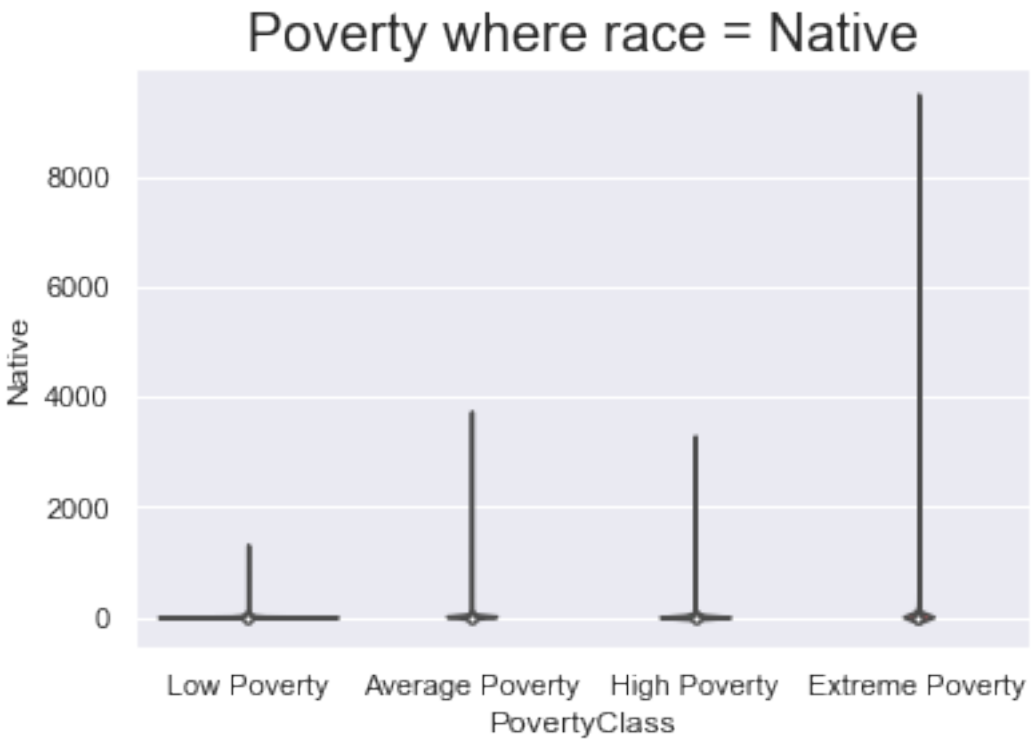
1.5.3 Ethnicities Analysis

As we later disucss, a few variables out of the clusters we explore, show some promise in understanding important features for modeling or describing poverty. For example, among racial lines, large native populations demonstrated different distributions when those census tracts were in more extreme states of poverty.

Hispanic was another racial group that showed differing distributions in different states of poverty. The portion of Hispanic population also seems to interact with the number of citizens when predictive of poverty status in a tract.

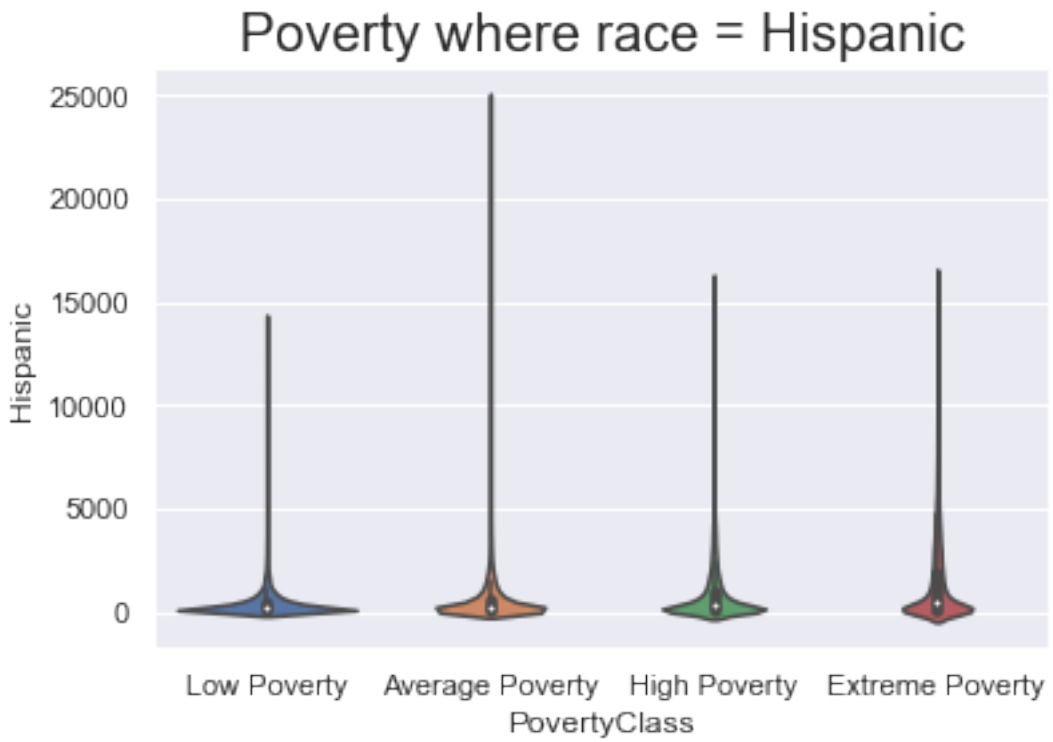
[127]:
```
sns.violinplot(x = "PovertyClass", y = "Native", data = data2015agg,  split =␣
↪True).set_title('Poverty where race = Native', fontsize = 20)
```

[127]:
```
Text(0.5, 1.0, 'Poverty where race = Native')
```



[128]:
```
sns.violinplot(x = "PovertyClass", y = "Hispanic", data = data2015agg,  split =␣
↪True).set_title('Poverty where race = Hispanic', fontsize = 20)
```

[128]: Text(0.5, 1.0, 'Poverty where race = Hispanic')

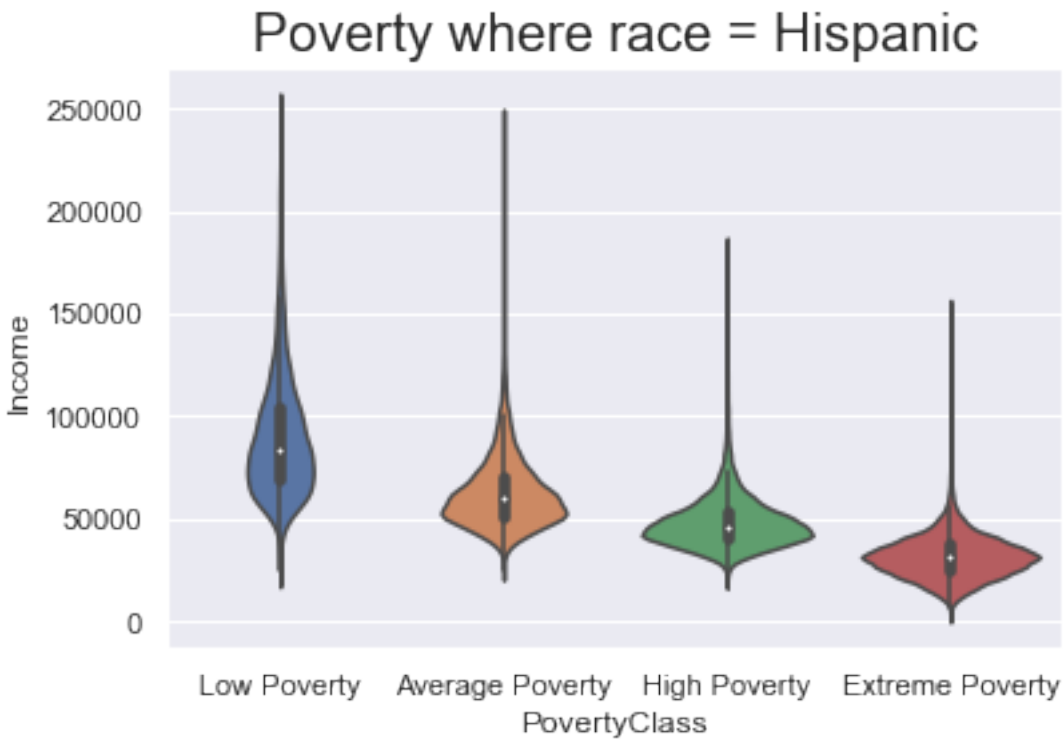


1.5.4 Income Analysis

Finally, income was an interesting income for poverty prediction, while it, as expected, lowers at higher poverty levels, the distribution across these different types of poverty changes as well. Given the multiple measures of income we are provided with, this indicates we could use some useful combinations to predict poverty better.

[129]:
```
sns.violinplot(x = "PovertyClass", y = "Income", data = data2015agg,  split =␣
↪True).set_title('Poverty where race = Hispanic', fontsize = 20)
```

[129]:
```
Text(0.5, 1.0, 'Poverty where race = Hispanic')
```

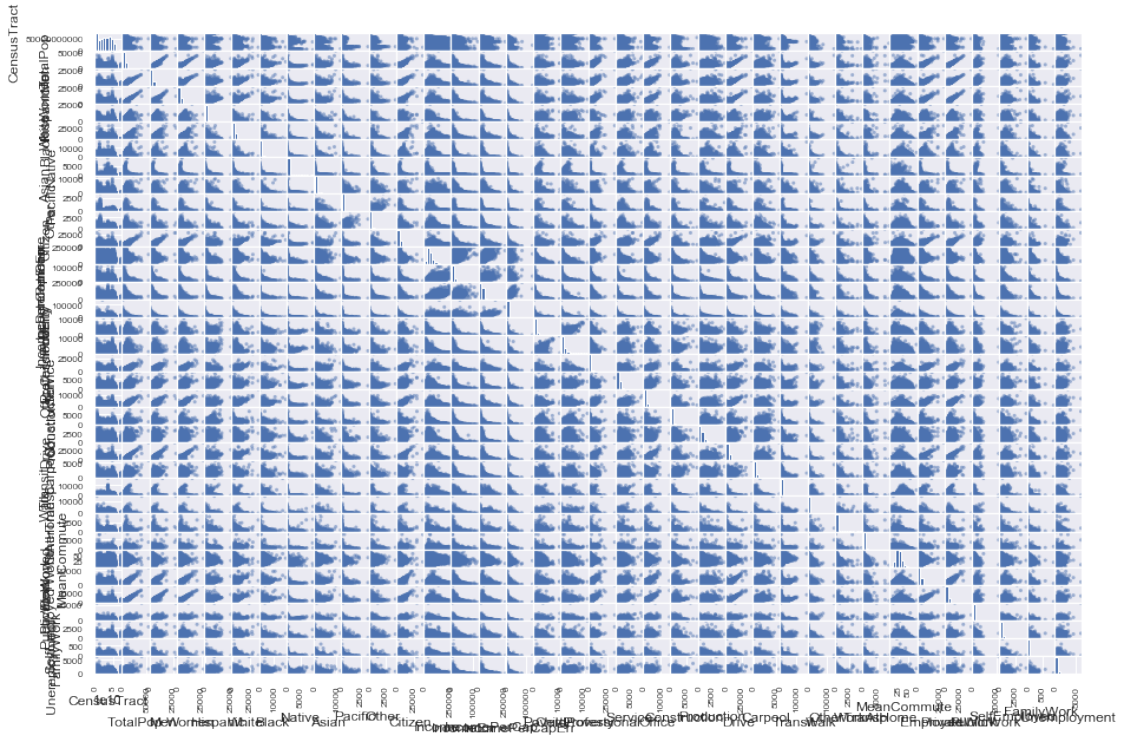


1.6 Explore Joint Attributes

When exploring joint attributes our goal is to surface important relationships between our independent variables. We started by obtaining a birdseye view of our continuous variables. We leveraged a simple scatterplot matrix with all continuous independent variables.

Upon first look (see scatterplot below) we can see that our vast number of numeric variables for evaluation make it impossible to gleen any useful information from our scatterplot. We can also see that it is difficult to see any trend with the color and style of the graph. Moving forward we will leverage cleaner visualziations to be able to see our variable trends more clearly.

[130]:
```
#Continuous Variable Scatterplot
from pandas.plotting import scatter_matrix

ax = scatter_matrix(data2015agg,figsize=(15, 10))
```



1.6.1 Variable Clustering

As we continue to investigate the data we decided to break out our continuous variables into related groups. In order to properly evaluate these attributes against each other we needed to identify variables into these clusters and then use a pairs scatter plot for visual display of any possible linear trends. In order to supplement these scatterplots we also created linearity correlation matrices.

A data example: we have a group of continuous variables that are all different ethnicities. This was broken into our race cluster. We observed that job functions was another group of related variables in the data. This was broken into our job function cluster, and so on.

NOTE: Our variable cluster comparisons are made from our aggregated data set (not the percentile data set).

1. Race Cluster
2. Job Function Cluster
3. Transportation Mode Cluster
4. Employment Cluster
5. Income Cluster
6. Additional Cluster

Race Variable Cluster Analysis For our race variable cluster we created the below scatterplot. This was in order to see if we could identify any linearity between the registered ethnic groups. We didn't see any immediate relationships between them, so we also included the Citizen variable in

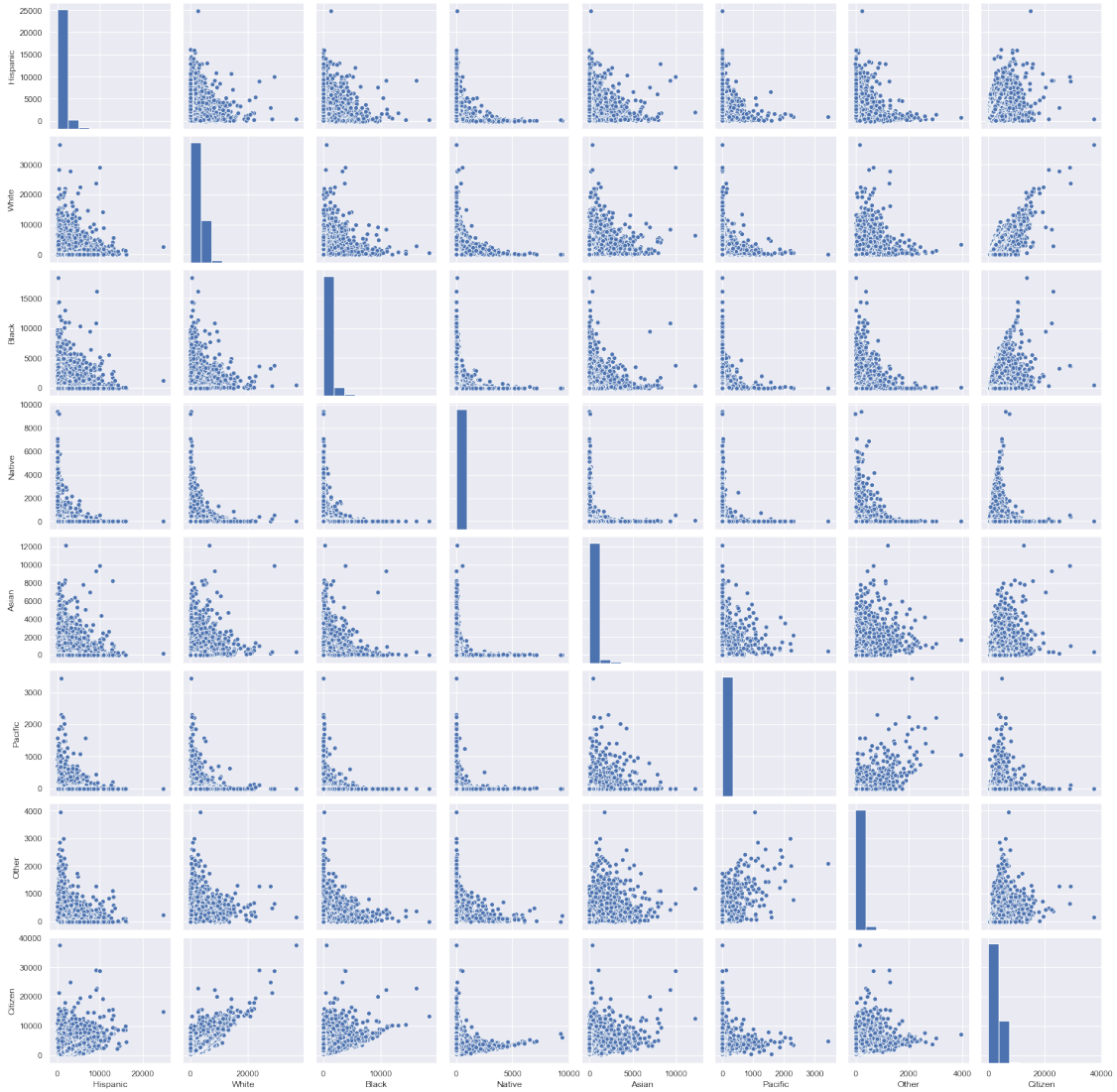
the data. Once this additional variable was added into the race cluster it showed some relationships between the ethnicities and citizenship.

Specifically we think we see a linear correlation between multiple races and citizenship, but most strongly with white and black race groups. We decided to further investigate this with a correlation plot.

[131]:
```
#Race Cluster dataframe creation
racecluster = data2015agg.copy()
```

[132]:
```
#racecluster df column selection
racecluster =␣
 ↪racecluster[['Hispanic','White','Black','Native','Asian','Pacific','Other','Citizen','Pover
```

[133]:
```
#Race Cluster scatter plot
racescatterSea = sns.pairplot(racecluster)
```

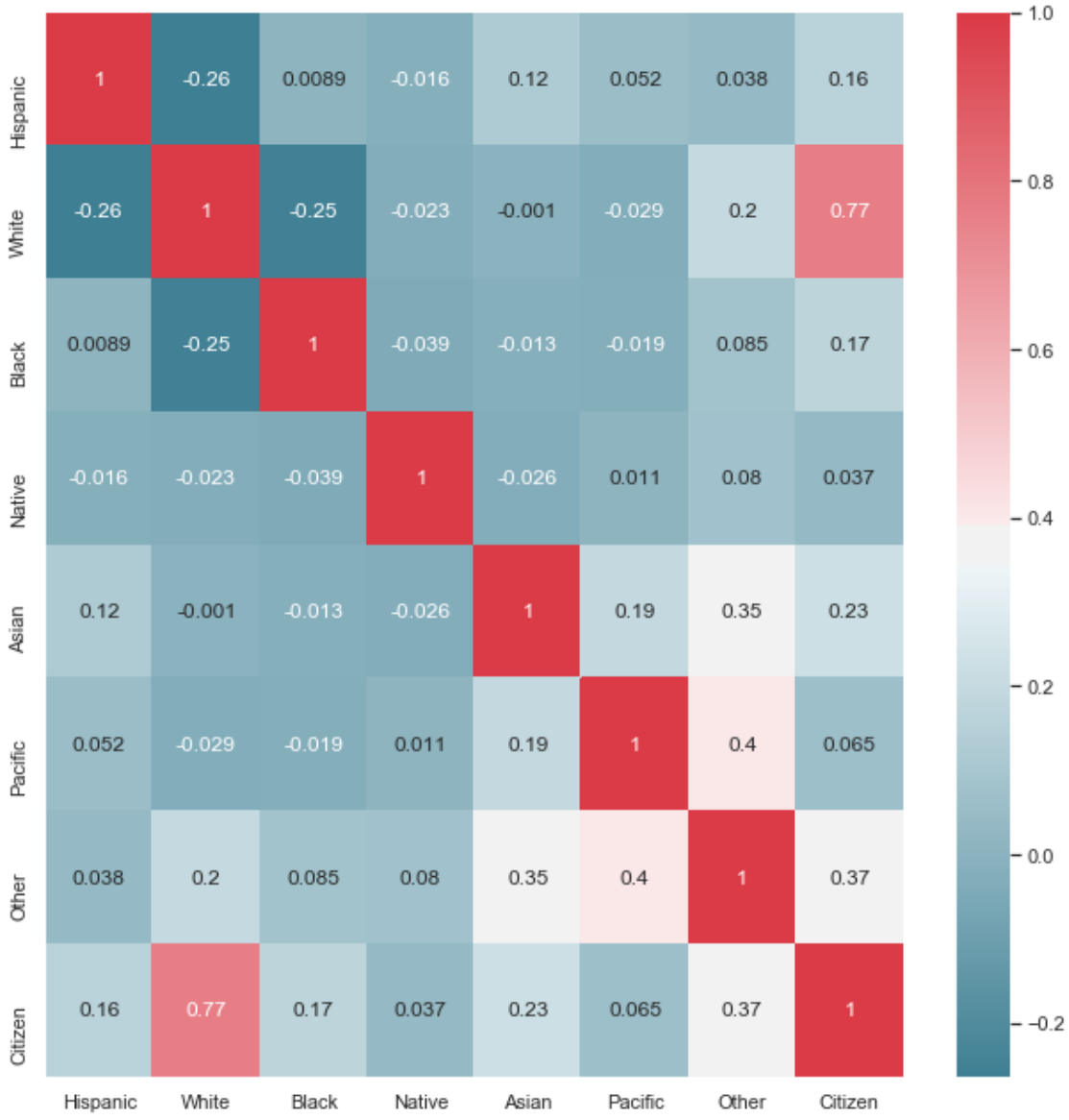


[134]:
```
#this will allow plot to be embedded into the notebook
import matplotlib.pyplot as plt
```

[135]:
```
#color mapping setup
cmap = sns.diverging_palette(220, 10, as_cmap=True) # one of the many color␣
 ↪mappings
```

For our race variable cluster correlation matrix we see that there are the strongest correlations are between Citizenship and White, Other, Hispanic, and Black groups. However, all of the groups do have some correlation with citizenship. We also noted a high correlation between Pacific and Other ethnic groups.

[136]:
```
#Race Cluster correlation plot (multicollinearity check)
sns.set(style="darkgrid") # one of the many styles to plot using

f, ax = plt.subplots(figsize=(9, 9))

sns.heatmap(racecluster.corr(), cmap=cmap, annot=True)

f.tight_layout()
```

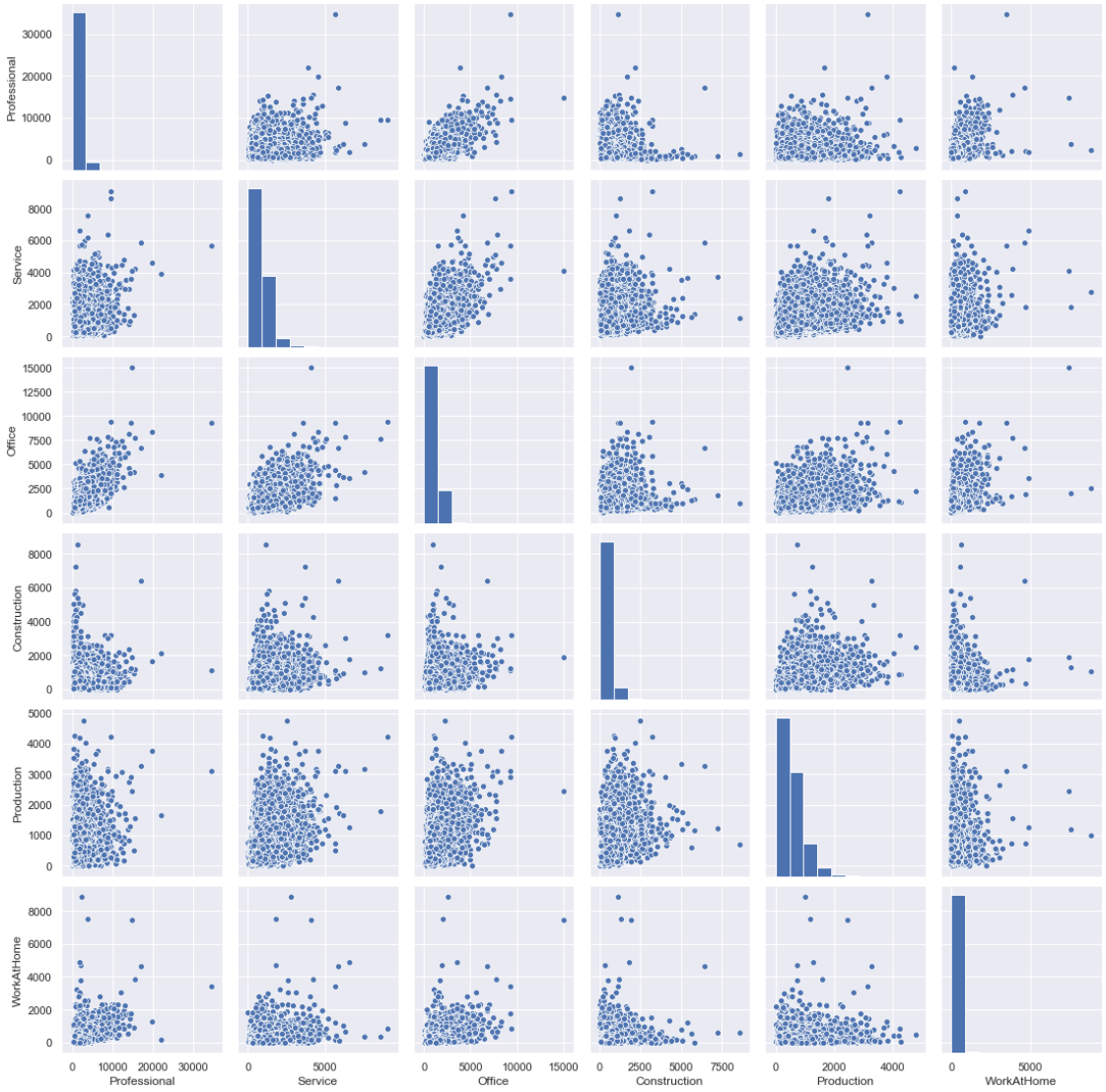


Job Cluster Variable Analysis Our job variable scatterplot cluster we created is below. This was in order to see if we could identify any linearity between the collected job functions. Upon review we see a few relationships that could easily be classified as having a positive linear relationship. Specifically we observed what appeared to be linear relationships between Professional and Office, Professional and Service, as well as WorkAtHome and numerous other job types. We investigated this further with a correlation plot. We have continued on with a correlation matrix to solifying our findings.

[137]:
```
#Job Cluster dataframe creation
jobcluster = data2015agg.copy()
```

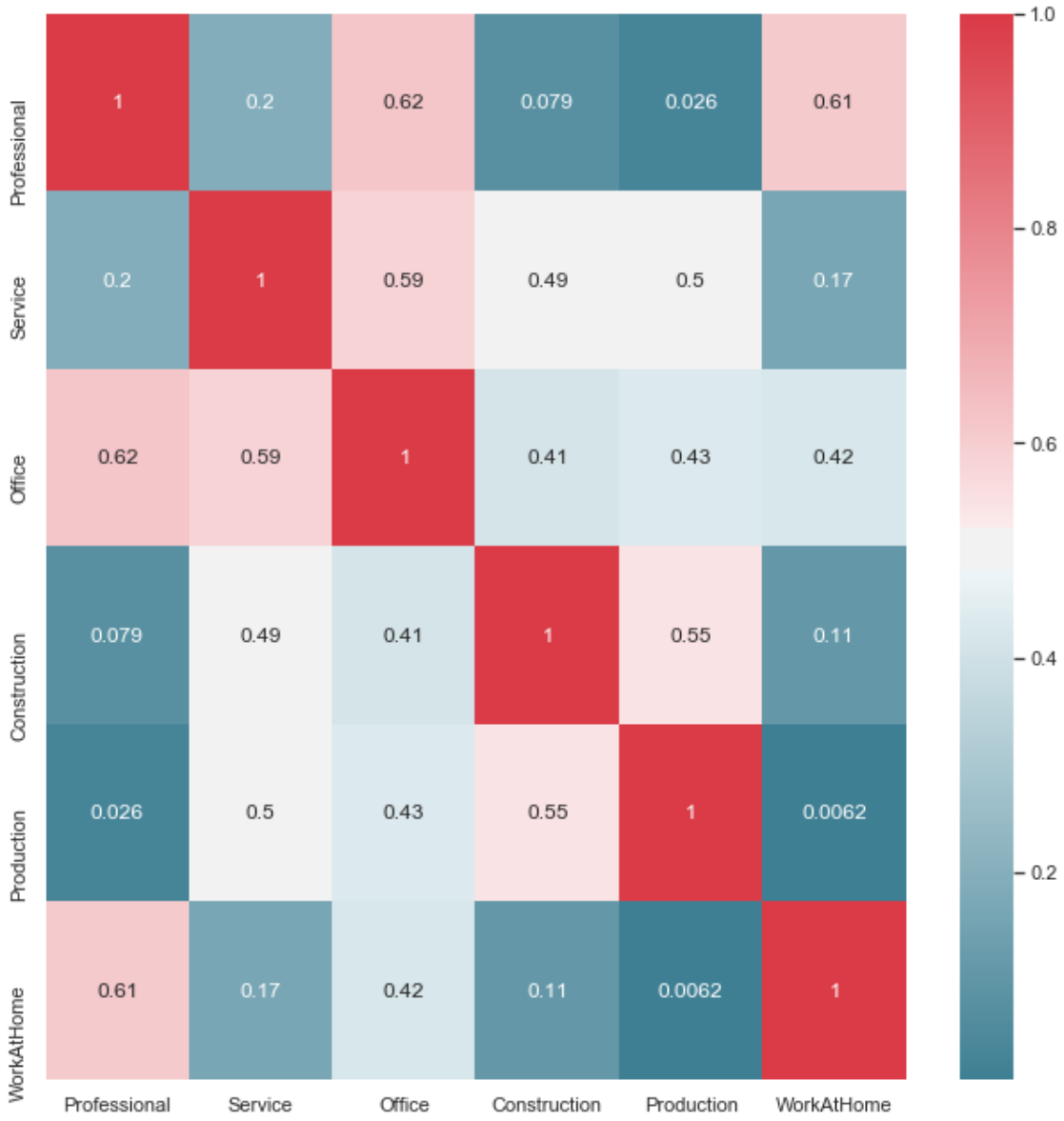
[138]:
```
#Job Cluster df column selection
jobcluster =␣
 ↪jobcluster[['Professional','Service','Office','Construction','Production','WorkAtHome','Pov
```

[139]:
```
#Job Cluster scatter plot
jobscatterSea = sns.pairplot(jobcluster)
```



For our job variable cluster correlation matrix we see that there are the strongest correlations are between Professional and multiple variables such as: Office and WorkAtHome. We also have surfaced a high correlation between Porduction and Construction. Lastly, we see a strong correlation between Office and Service, which was unexpected.

[140]:
```
#Job Cluster correlation plot (multicollinearity check)
sns.set(style="darkgrid") # one of the many styles to plot using

f, ax = plt.subplots(figsize=(9, 9))

sns.heatmap(jobcluster.corr(), cmap=cmap, annot=True)

f.tight_layout()
```



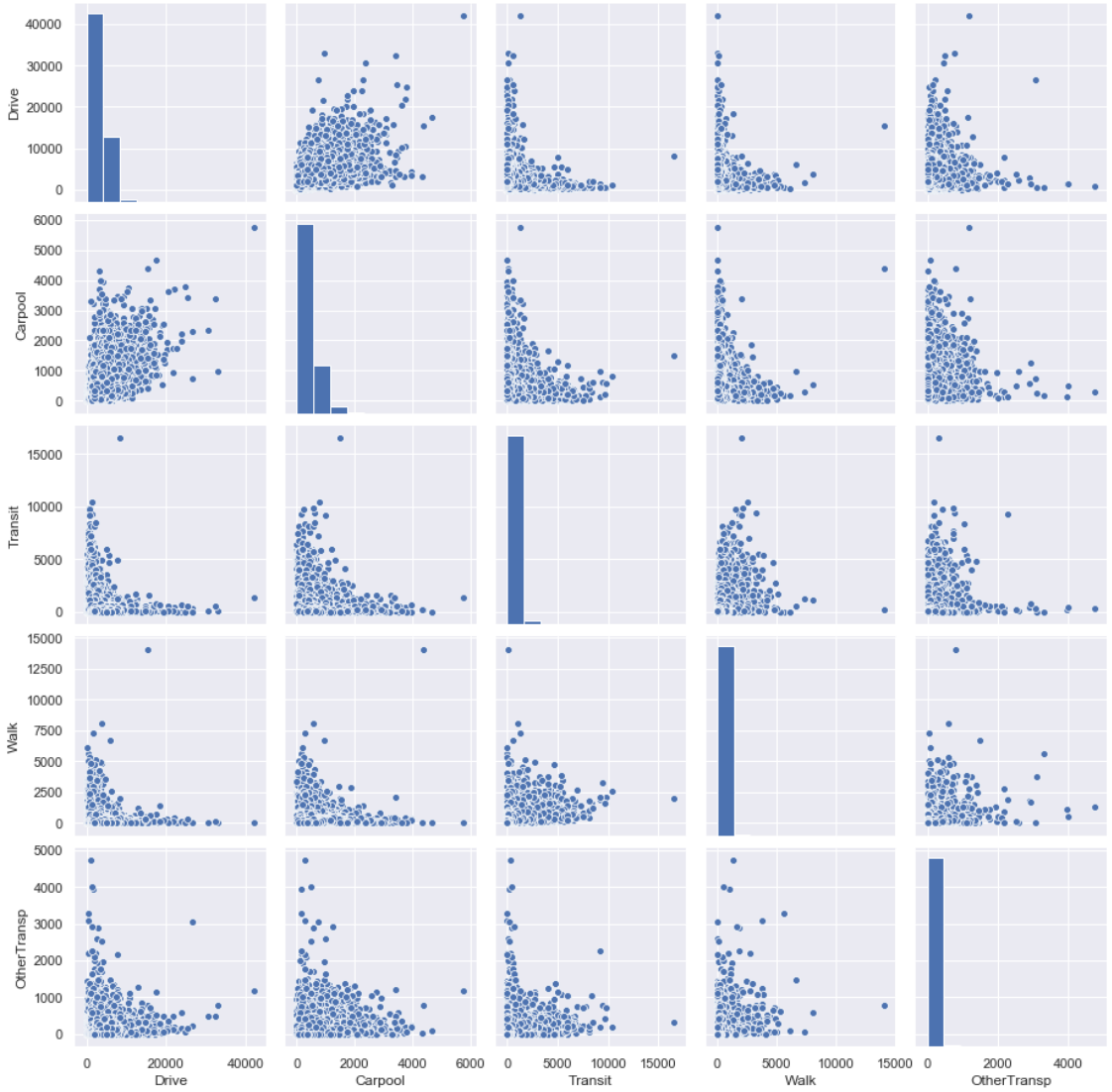
Transportation Cluster Variable Analysis Our transportation variable scatterplot cluster we created is below. This was in order to see if we could identify any linearity between the collected

types of commute options by individuals. Upon review, we see a very few relationships that seem linearly correlated. However, one that can be called out is between Carpool and Drive. All to the other comparisons could potentially have some correlation but none that are easily identifiable through the scatterplot matrix. We confirm our assumptions with a correlation plot.

[141]:
```
#Transportation Cluster dataframe creation
transcluster = data2015agg.copy()
```

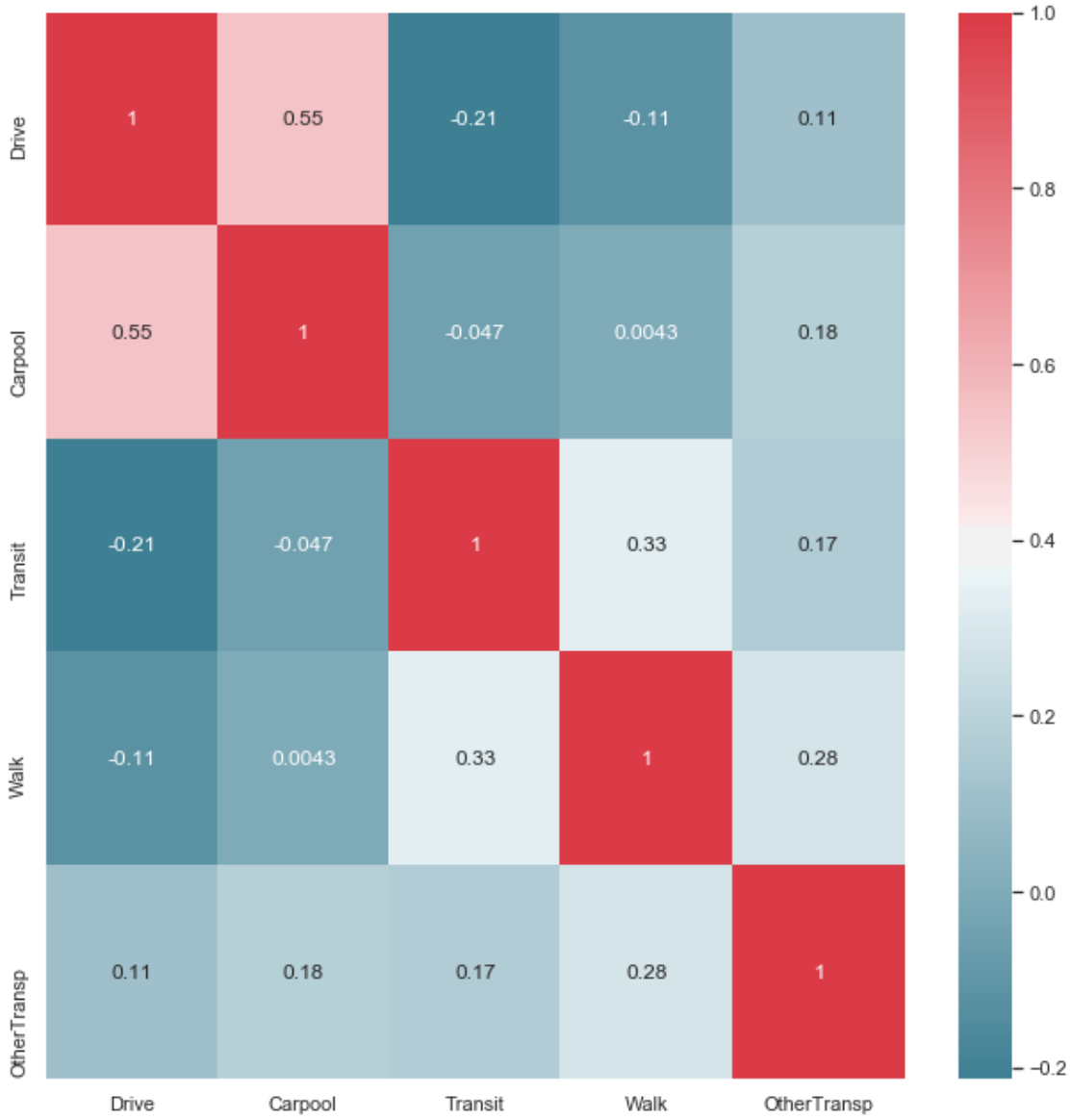
[142]:
```
#Transportation Cluster df column selection
transcluster =␣
↪transcluster[['Drive','Carpool','Transit','Walk','OtherTransp','PovertyClass']]
```

[143]:
```
#Transportation Cluster scatter plot
transscatterSea = sns.pairplot(transcluster)
```



For our transportation variable cluster correlation matrix we see that there very few strong correlations. However, like in our scatterplot we do see one notable correlation between CarPool and Drive while the other correlation measurements are quite low.

[144]:
```
#Transportation Cluster correlation plot (multicollinearity check)
sns.set(style="darkgrid") # one of the many styles to plot using

f, ax = plt.subplots(figsize=(9, 9))

sns.heatmap(transcluster.corr(), cmap=cmap, annot=True)

f.tight_layout()
```



Employment Variable Cluster Analysis For our employment variable cluster we created the below scatterplot. In this group of data we included two 'blanket' variables that would encompass many of the variables listed; these were Unemployed and Employed. When using a blanket variable such as this we can expect to see certain high correlations, and we confirm this with the scatterplots for Employed and PrivateWork as well as Employed and PublicWork. These two pairs have a visually strong correlation. What is interesting is that not more of the employment variables are strongly correlated with Employed. One would assume if you are employed in any of these listed fields you would also consider yourself Employed overall but the data doesn't seem to support this in its entirety. We decided to further investigate this with a correlation plot.

[145]:
```
#Employement Cluster dataframe creation
empcluster = data2015agg.copy()
```

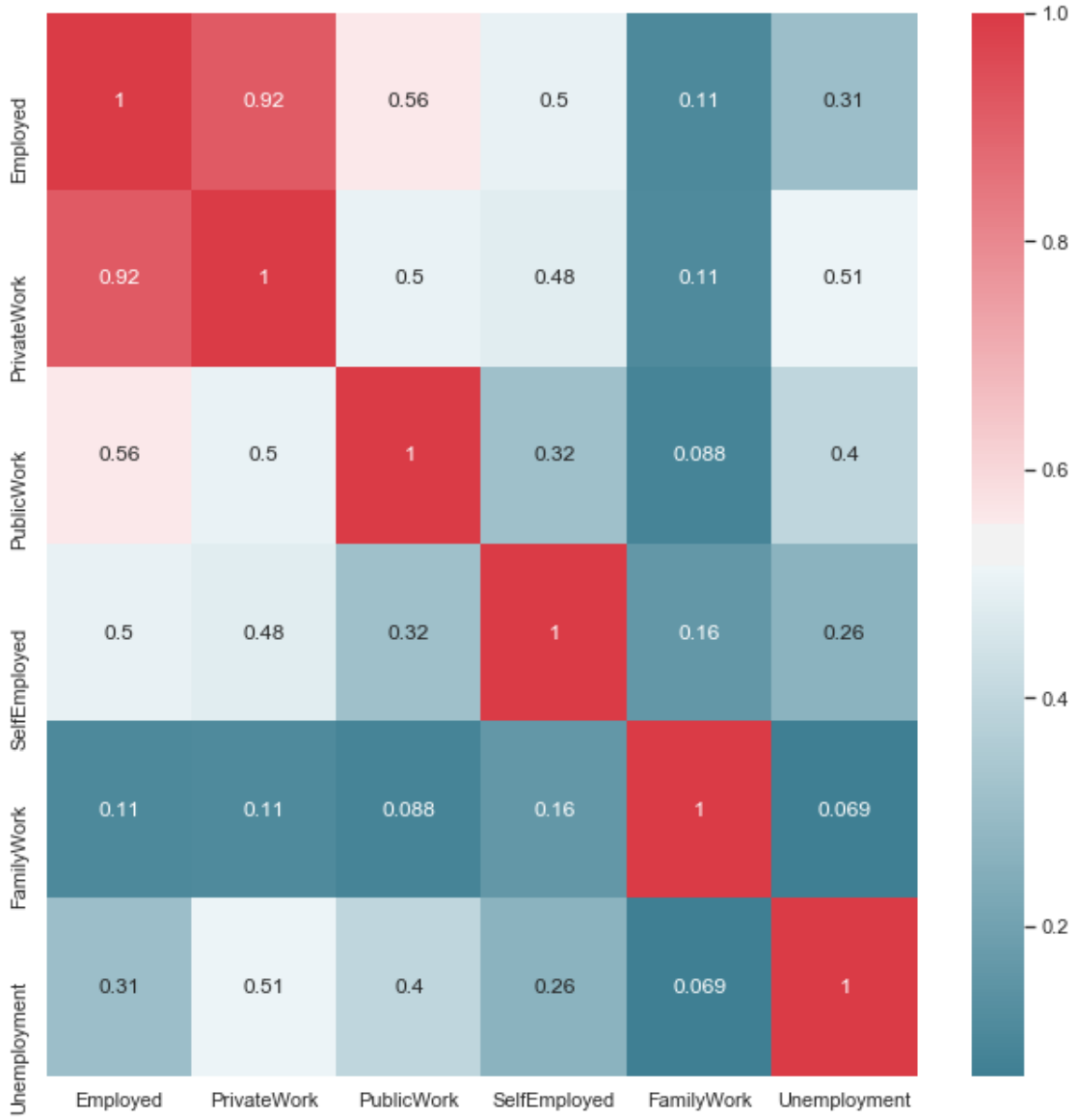
[146]:
```
#Employement Cluster df column selection
empcluster =␣
 ↪empcluster[['Employed','PrivateWork','PublicWork','SelfEmployed','FamilyWork','Unemployment
```

[147]:
```
#Employed Cluster scatter plot
empscatterSea = sns.pairplot(empcluster)
```

For our employment variable cluster correlation matrix we see that there are a couple strong correlations and a few moderately strong correlations. We can confirm our assumptions from above and see that Employed has strong correlations with PrivateWork and PublicWork. In addition, we see a moderately strong correlation between Employed and SelfEmployed, as well as Unemployed. In surfacing this contradictory information with a correlation between Unemployed and Employed there may be a collection error here we should be cautious of when building our model for classification.

[148]:
```
#Employment Cluster correlation plot (multicollinearity check)
sns.set(style="darkgrid") # one of the many styles to plot using

f, ax = plt.subplots(figsize=(9, 9))

sns.heatmap(empcluster.corr(), cmap=cmap, annot=True)
```

```
f.tight_layout()
```

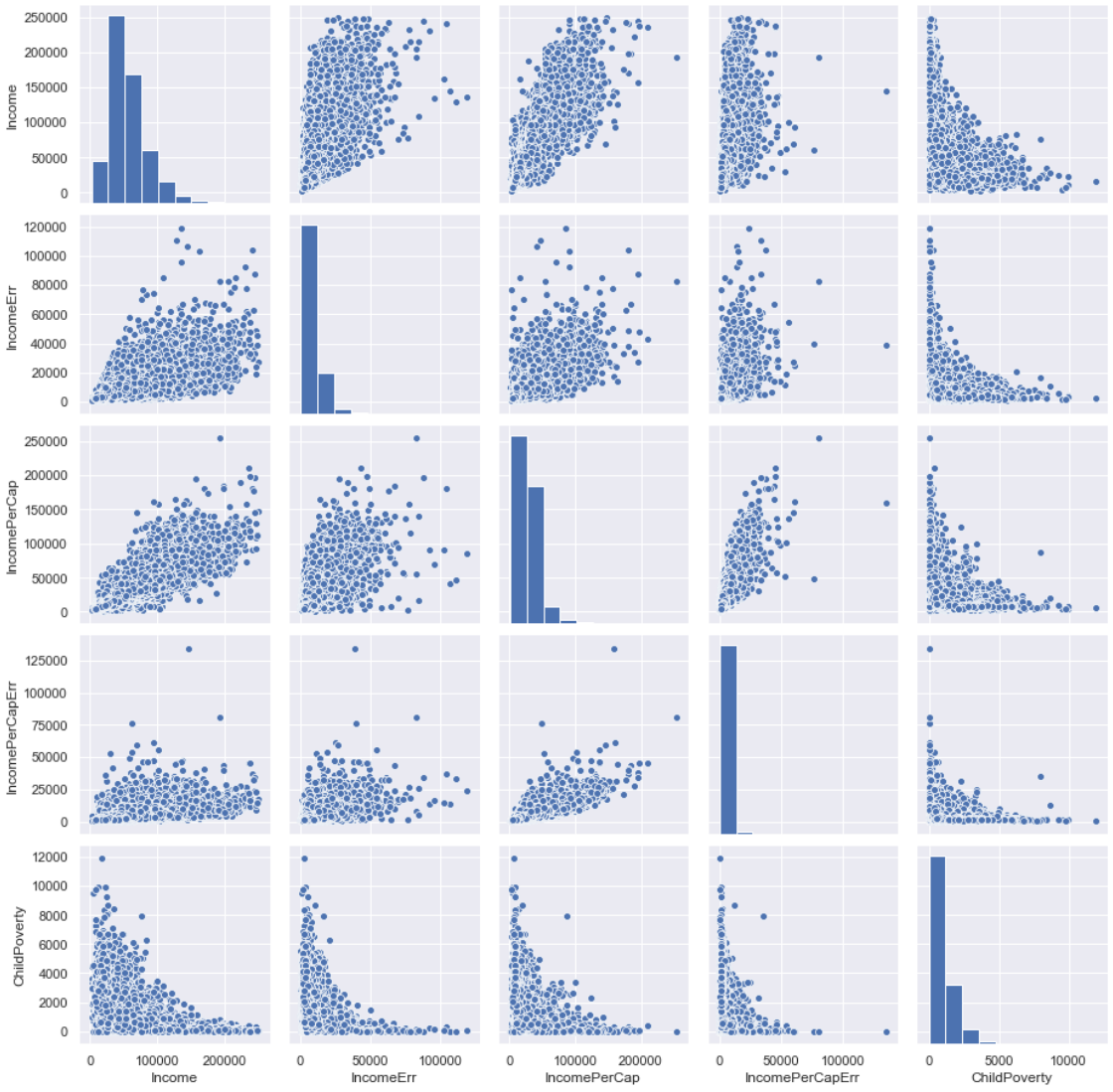


Income Variable Cluster Analysis For our income variable cluster we created the below scatterplot. Since many of the variables are income based we expect to see a high correlation between them. As we review the scatterplots below we see strong linear correlation between Income, IncomeErr, IncomePerCap, and IncomePerCapErr confirming our thoughts that these variables would be highly correlated.

[149]:
```
#Income Cluster dataframe creation
inccluster = data2015agg.copy()
```

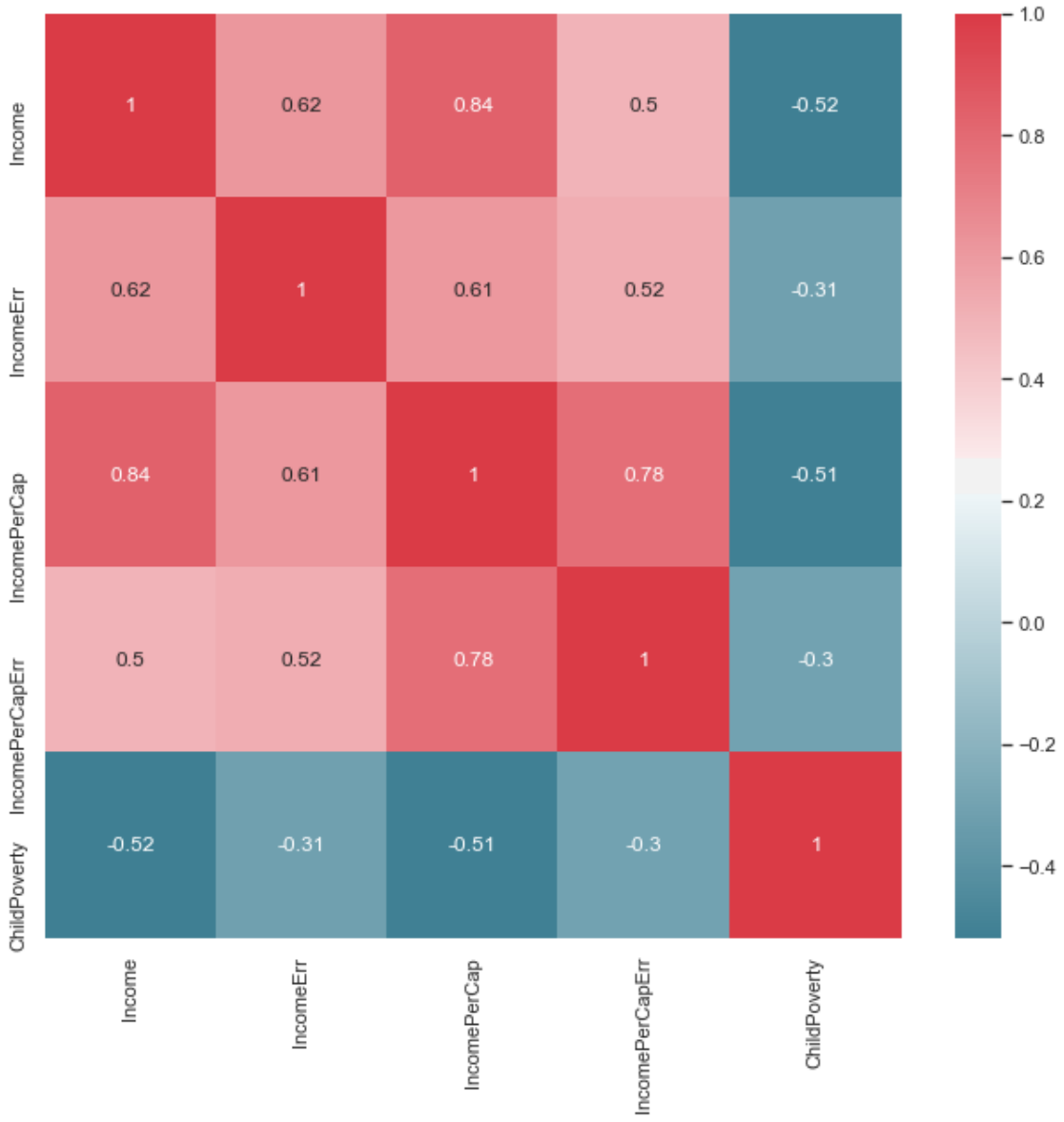
[150]:
```
#Income Cluster df column selection
inccluster = inccluster[['Income','IncomeErr','IncomePerCap',␣
↪'IncomePerCapErr', 'ChildPoverty','PovertyClass']]
```

[151]:
```
#Income Cluster scatter plot
incscatterSea = sns.pairplot(inccluster)
```



To further support our findings above, we see that this correlation matrix of our Income variable cluster is full of highly correlated variables. This is important to note because when reviewing this later it may be necessary to remove variables from our model that could be depicting the same information. Since we will be classifying poverty levels it's likely we would like to use IncomePerCap and IncomePerCapErr in order to help support our geographical models and checks.

[152]:
```
#Income Cluster correlation plot (multicollinearity check)
sns.set(style="darkgrid") # one of the many styles to plot using

f, ax = plt.subplots(figsize=(9, 9))

sns.heatmap(inccluster.corr(), cmap=cmap, annot=True)

f.tight_layout()
```



Additional Variable Cluster Analysis The additional variable cluster we created the below scatterplot. In this variable group we include a lot of obviously related variables like Total Popula-

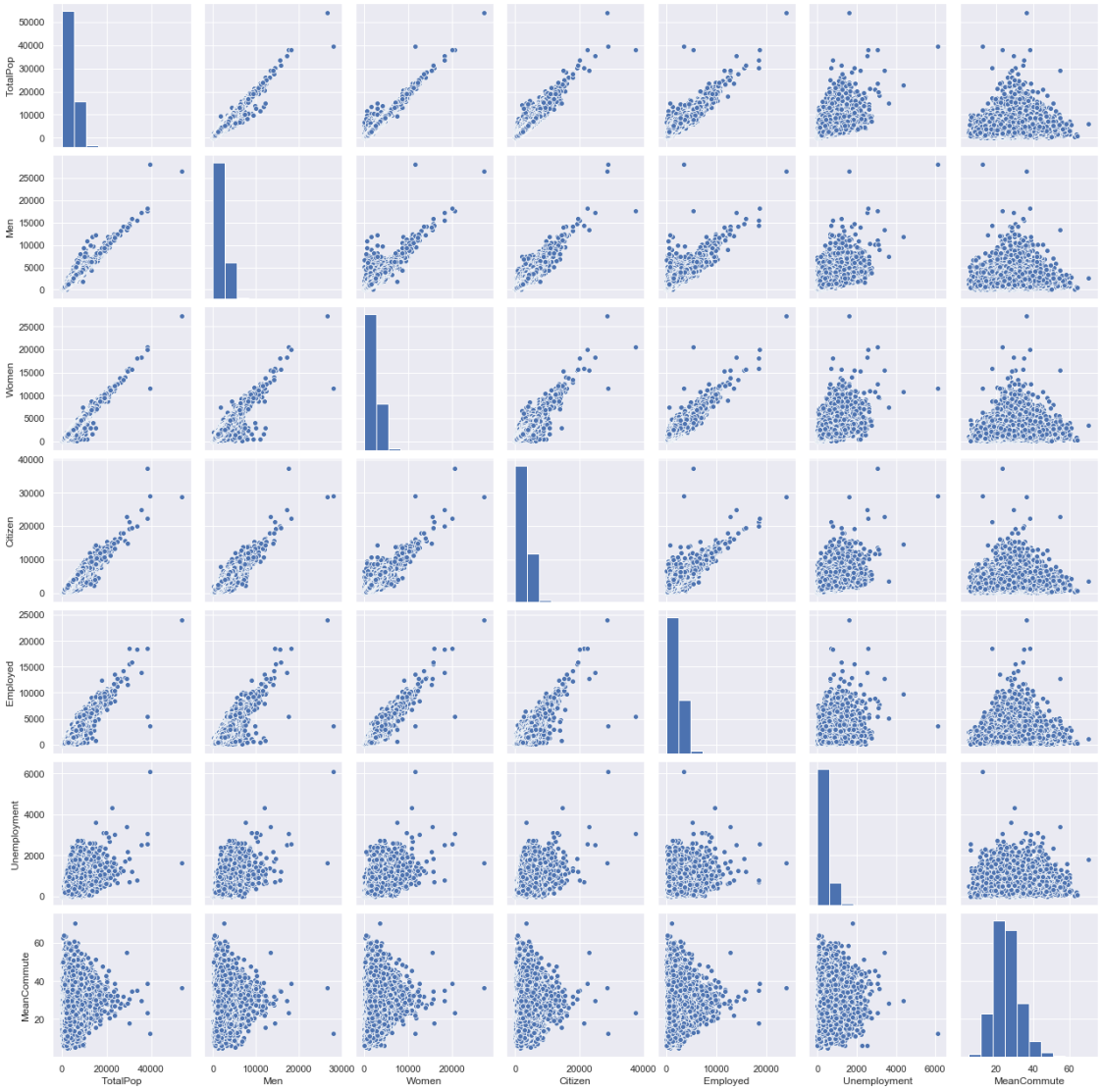
tion versus Men or Women. The Men and Women variables make up the total population therefore we expect to see high correlation between those variables.

Note: Total Population is expected to be correlated with any metric since all metrics were measured from the total population. The only question would be how correlated. We confirm our assumption with the scatterplot matrix below.

[153]:
```
#Additional Cluster dataframe creation
addcluster = data2015agg.copy()
```

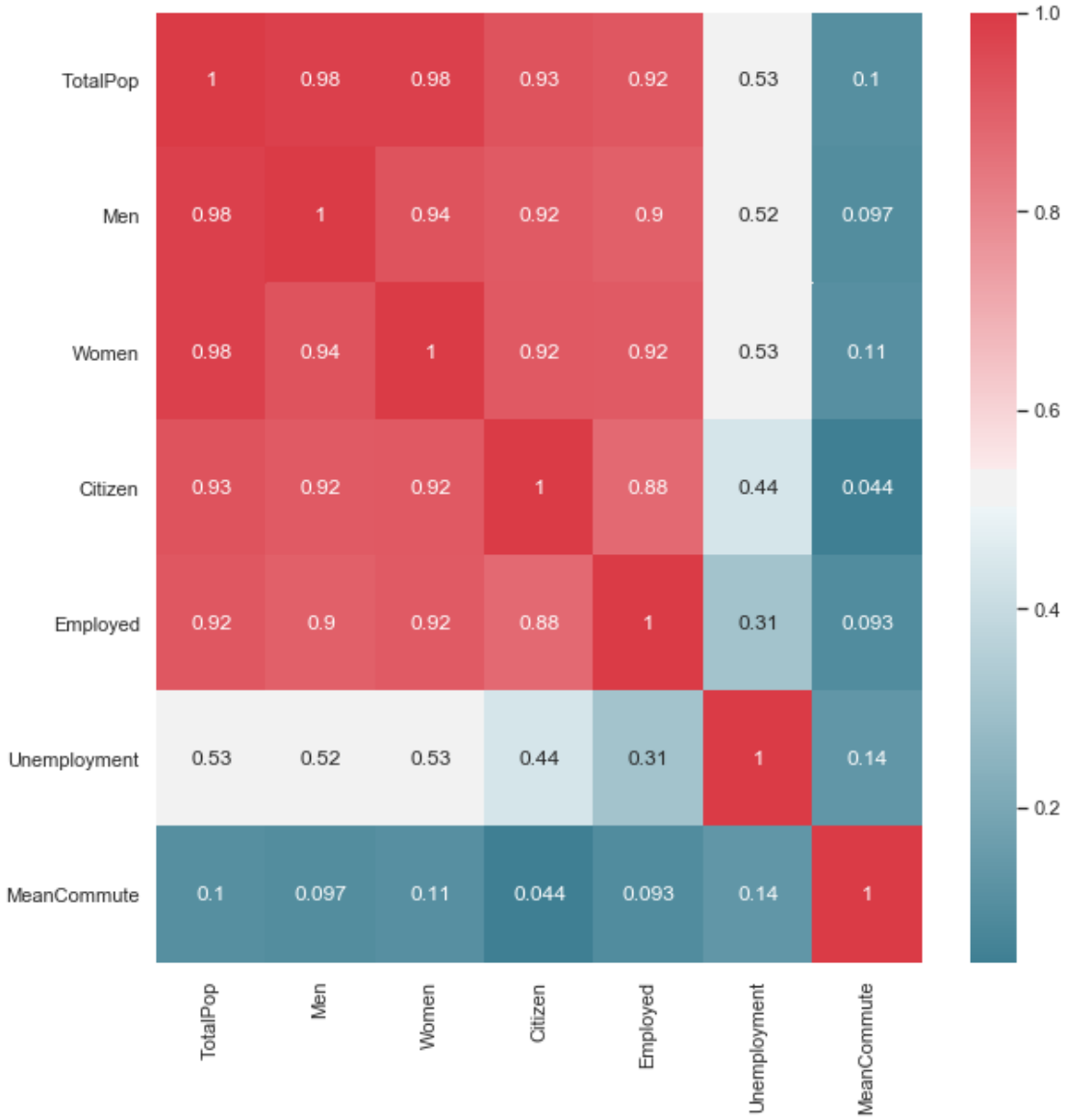
[154]:
```
#Additional Cluster df column selection
addcluster = addcluster[['TotalPop','Men','Women', 'Citizen',␣
↪'Employed','Unemployment', 'MeanCommute','PovertyClass']]
```

[155]:
```
#Additional Cluster scatter plot
addscatterSea = sns.pairplot(addcluster)
```



For our additional variable cluster correlation matrix we are simply confirming the observations we made with our scatterplot matrix. As we can see below our correlation matrix does just that. There is heavy correlation between Total Population and the other variables, as expected. This is notable now as it may affect how we treat these variables farther down the line during our classification model creation.

[156]:
```
#Additional Cluster correlation plot (multicollinearity check)
sns.set(style="darkgrid") # one of the many styles to plot using

f, ax = plt.subplots(figsize=(9, 9))

sns.heatmap(addcluster.corr(), cmap=cmap, annot=True)

f.tight_layout()
```



1.7 Explore Attributes and Class

As we continue to explore our data, we have to be sure to use our variable clusters in condition with the dependent classification variable: Poverty Class. Below we have broken out each scatterplot matrix and colored by Poverty Class to observe trends or notable insight about our dataset. 1. Race Cluster 2. Job Function Cluster 3. Transportation Mode Cluster 4. Employment Cluster 5. Income Cluster 6. Additional Cluster

Included in this section are some additiona new Features. We have evaluated by Class (Poverty-Class). 1. Income Class 2. Population Size Class 3. Men/Women Percentage Breakdown

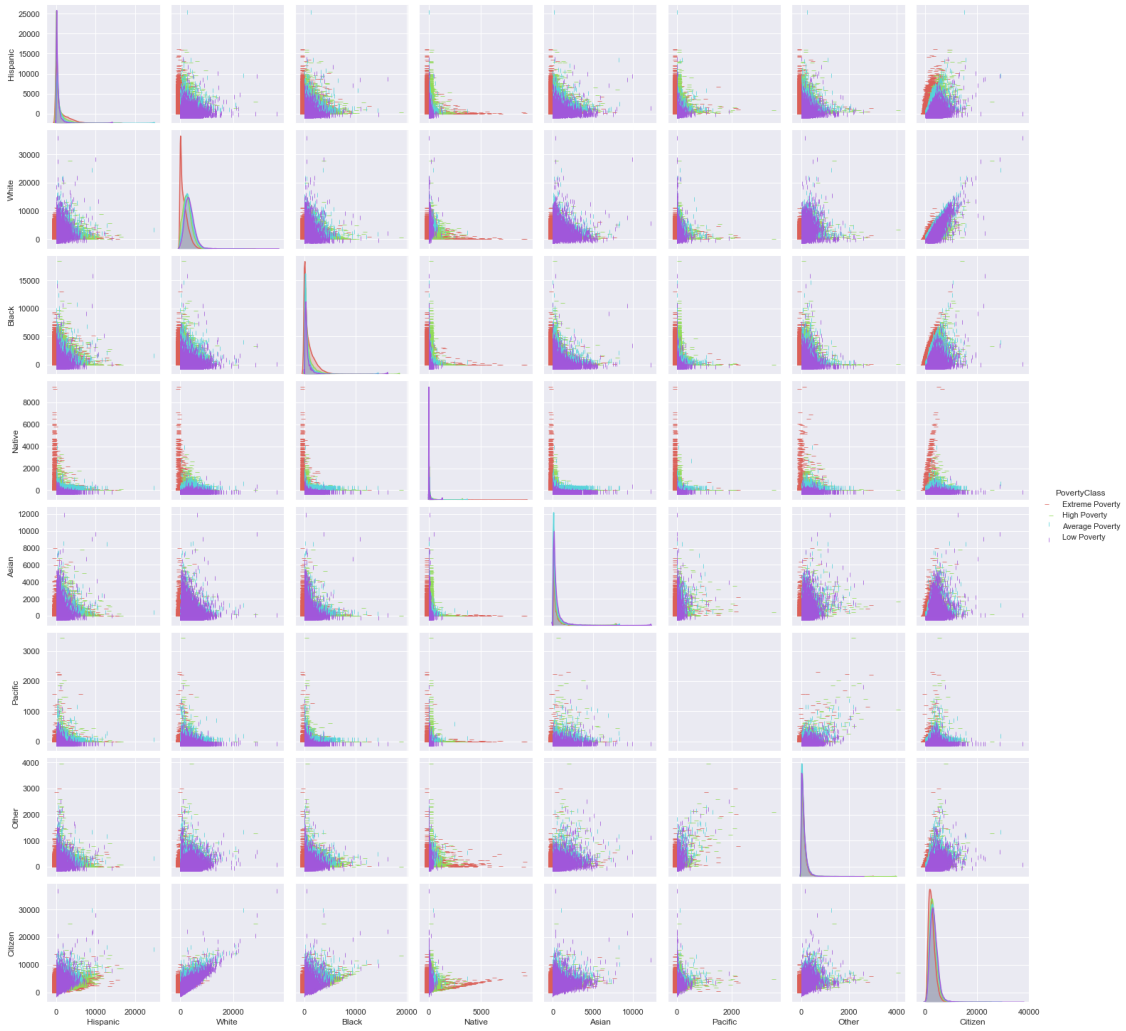
1.7.1 Race Variable Cluster with Poverty Class Analysis

Below is our race variable cluster scatterplot anlaysis hued by Poverty Class. When including PovertyClass we can see white a bit of separation between the Native group and Citizenship. This would indicate these variables used together would be essential in determining PovertyClass in our model. Note: we can see Native graphed against any ethnic group separated by Poverty Class resulted in higher separation than other pais in this plot.

We can also see that there is some separation of Poverty Class when comparing the Hispanic group against Citizenship. This is also important to note for our future model.

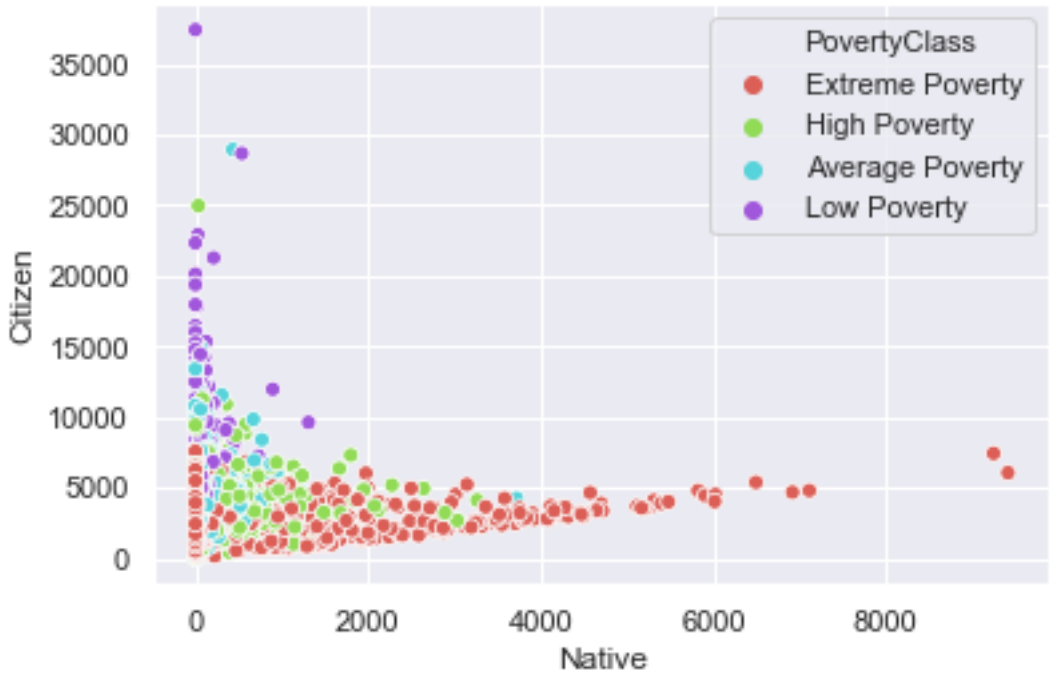
[157]:
```python
#Race Cluster scatterplot colored by Poverty Class
racescatterPC = sns.pairplot(racecluster, hue="PovertyClass",␣
 ↪hue_order=["Extreme Poverty","High Poverty","Average Poverty", "Low␣
 ↪Poverty"], palette = "hls", markers=[0, 1, 2, 3])
```

```
/Users/matt/opt/anaconda3/envs/ML1/lib/python3.7/site-
packages/seaborn/distributions.py:369: UserWarning: Default bandwidth for data
is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
/Users/matt/opt/anaconda3/envs/ML1/lib/python3.7/site-
packages/seaborn/distributions.py:369: UserWarning: Default bandwidth for data
is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
/Users/matt/opt/anaconda3/envs/ML1/lib/python3.7/site-
packages/seaborn/distributions.py:369: UserWarning: Default bandwidth for data
is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
/Users/matt/opt/anaconda3/envs/ML1/lib/python3.7/site-
packages/seaborn/distributions.py:369: UserWarning: Default bandwidth for data
is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
```



A closer look at Production versus Professional by Class

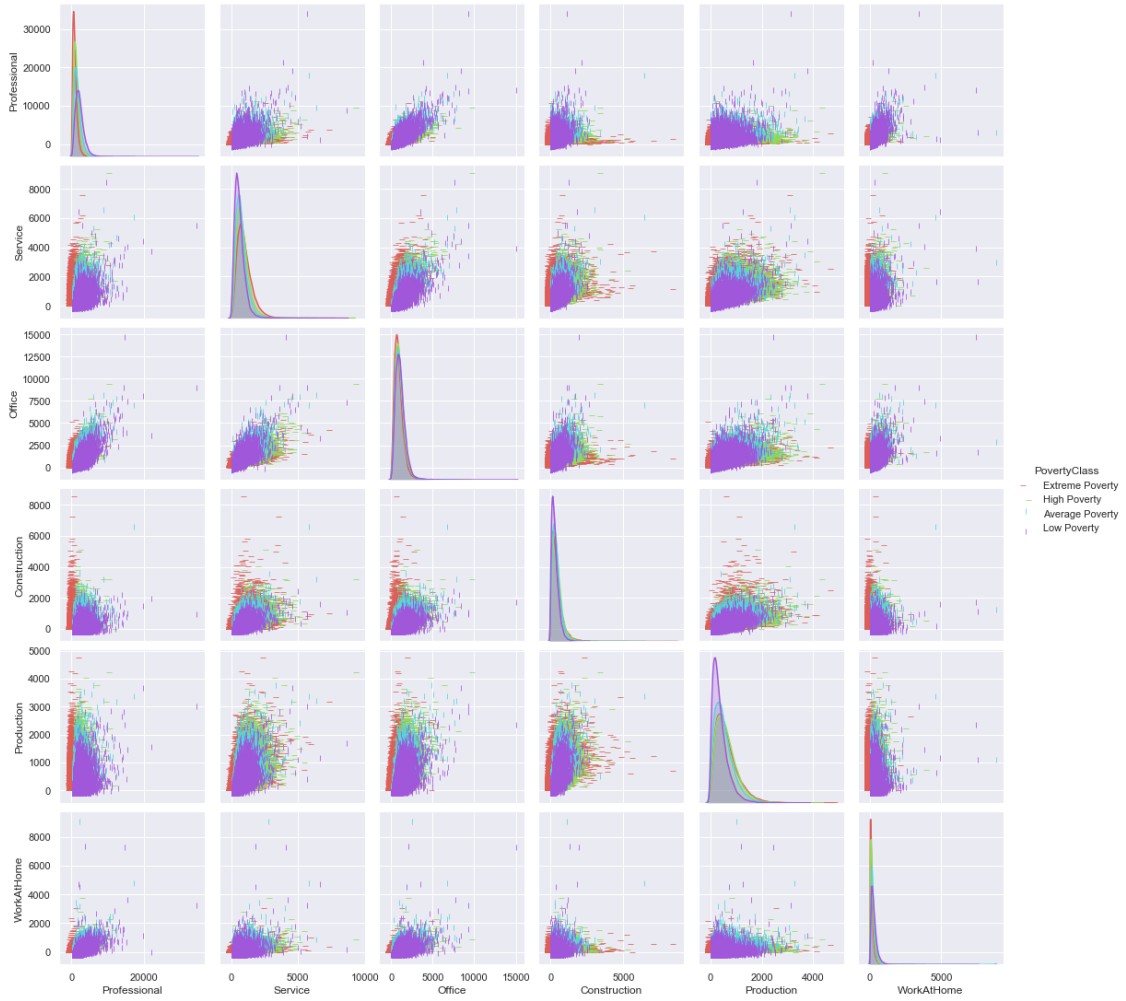
[158]:
```
#Native v Citizen by Class
ax = sns.scatterplot(x="Native", y="Citizen",
                     hue="PovertyClass", hue_order=["Extreme Poverty","High␣
↪Poverty","Average Poverty", "Low Poverty"], palette = "hls",data=racecluster)
```



1.7.2 Job Variable Cluster with Poverty Class Analysis

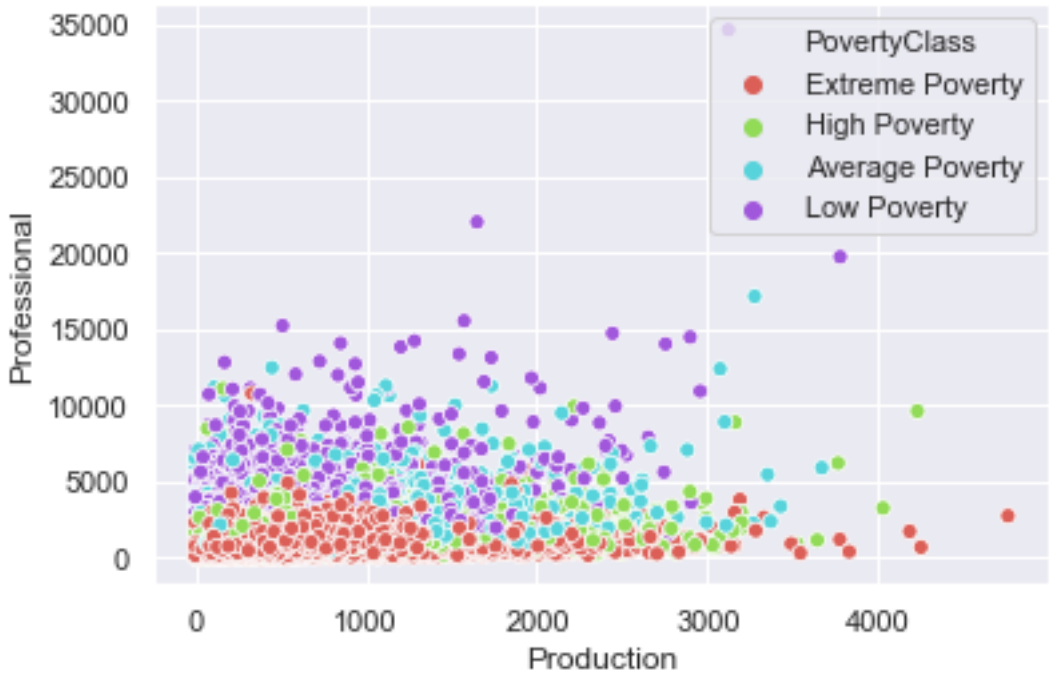
When reviewing the job variable cluster with the Poverty Class (dependent variable) as an additional component we can see that there is a lot of overlapping between the different poverty classes. An interesting thing we did surface was a higher volume of extreme poverty in the Construction and Production job variables when comparing them to the professional job category.

[159]:
```
#Job Cluster scatterplot colored by Poverty Class
jobscatterPC = sns.pairplot(jobcluster, hue="PovertyClass", hue_order=["Extreme␣
↪Poverty","High Poverty","Average Poverty", "Low Poverty"], palette = "hls",␣
↪markers=[0, 1, 2, 3])
```



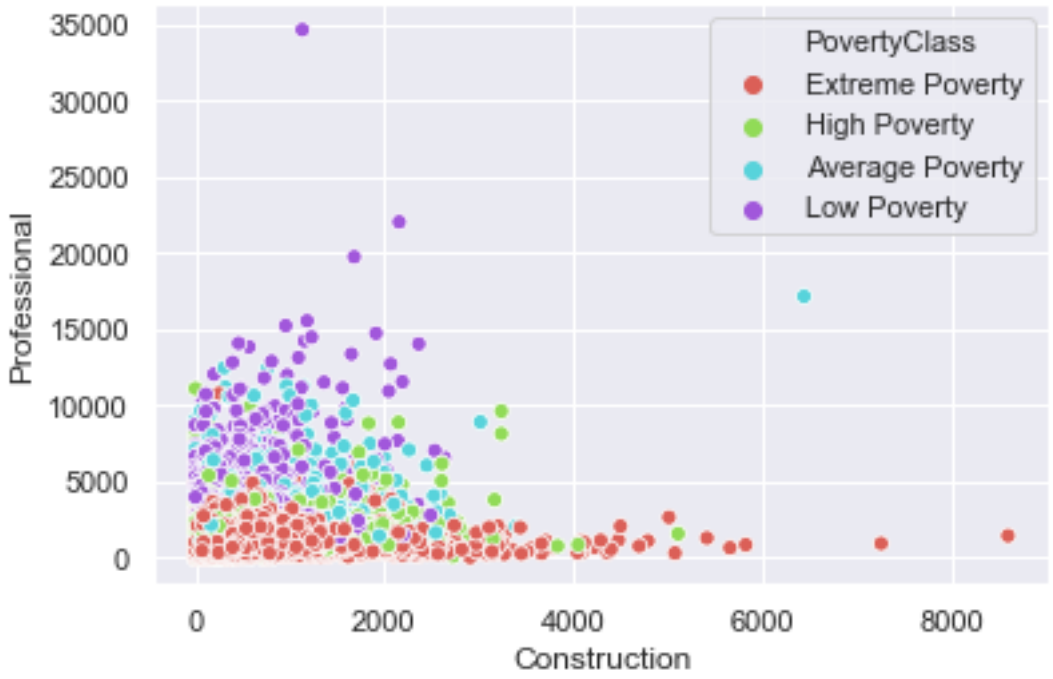
A closer look at Production versus Professional by Class

[160]:
```
#Production v Professional
ax = sns.scatterplot(x="Production", y="Professional",
                     hue="PovertyClass", hue_order=["Extreme Poverty","High␣
↪Poverty","Average Poverty", "Low Poverty"], palette = "hls",data=jobcluster)
```



A closer look at Construction versus Professional by Class

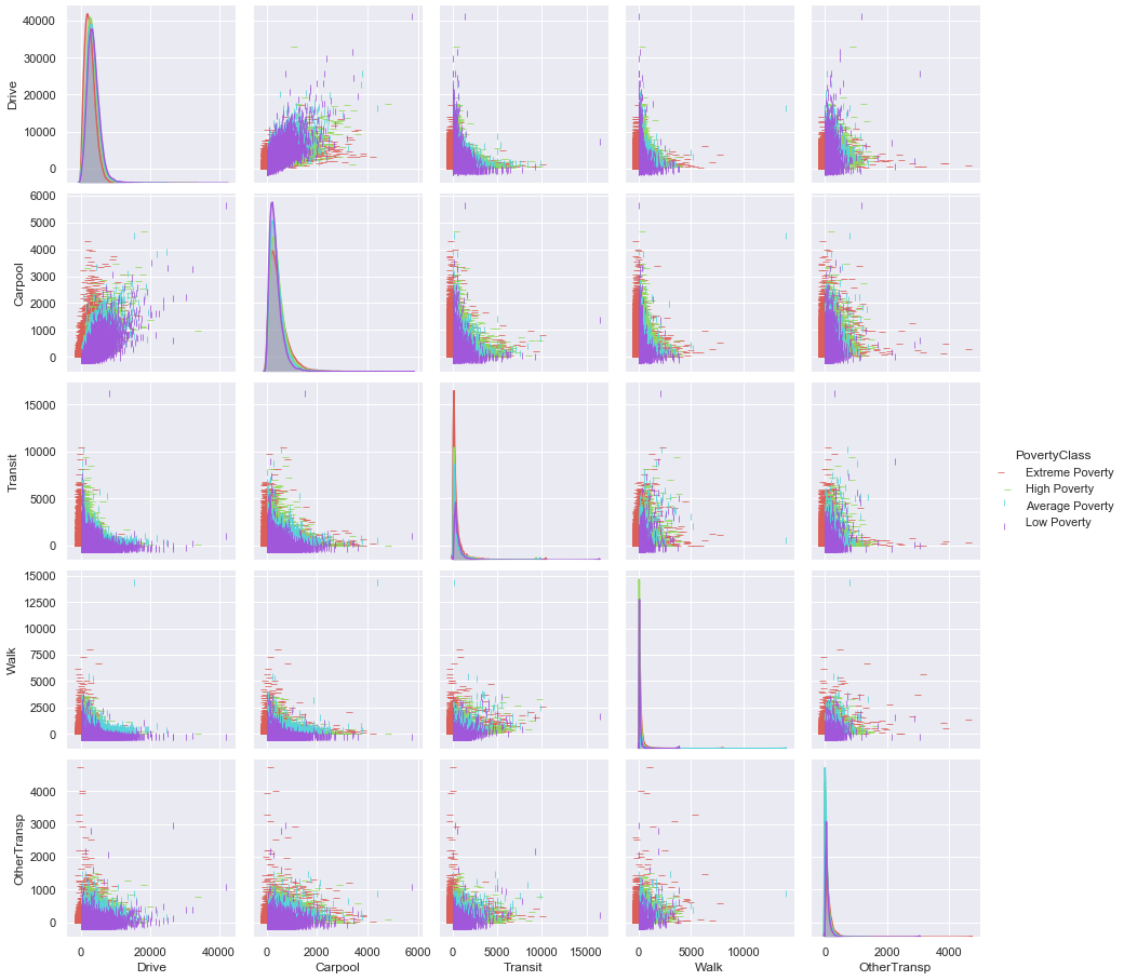
[161]:
```
#Construction v Professional
ax = sns.scatterplot(x="Construction", y="Professional",
                     hue="PovertyClass", hue_order=["Extreme Poverty","High␣
↪Poverty","Average Poverty", "Low Poverty"], palette = "hls",data=jobcluster)
```



1.7.3 Transportation Variable Cluster with Poverty Class Analysis

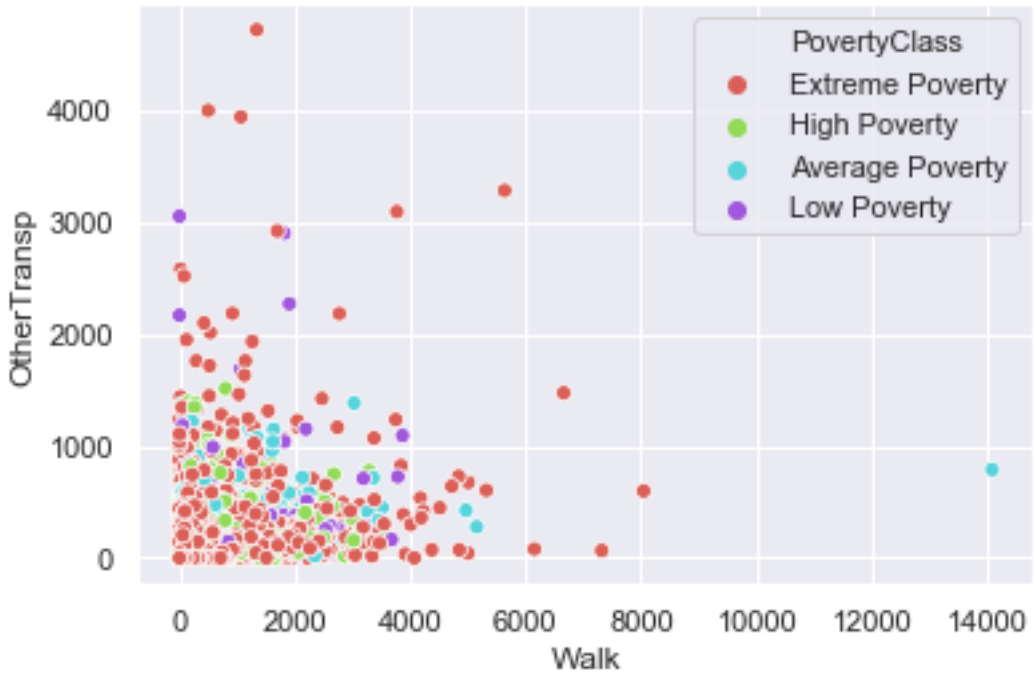
In our scatterplot that includes poverty class levels highlighted we see again there is a lot of overlapping with many of the variables. However we do see a spike in extreme poverty when looking at the transit transportation method. As well as a high poverty level spike with the walking transportation variable.

[162]:
```
#Trasportation Cluster scatterplot colored by Poverty Class
transscatterPC = sns.pairplot(transcluster, hue="PovertyClass",␣
↪hue_order=["Extreme Poverty","High Poverty","Average Poverty", "Low␣
↪Poverty"], palette = "hls", markers=[0, 1, 2, 3])
```



A closer look at Walk versus OtherTransp by Class

[163]:
```
#Walk v OtherTransp
ax = sns.scatterplot(x="Walk", y="OtherTransp",
                     hue="PovertyClass", hue_order=["Extreme Poverty","High␣
↪Poverty","Average Poverty", "Low Poverty"], palette =␣
↪"hls",data=transcluster)
```

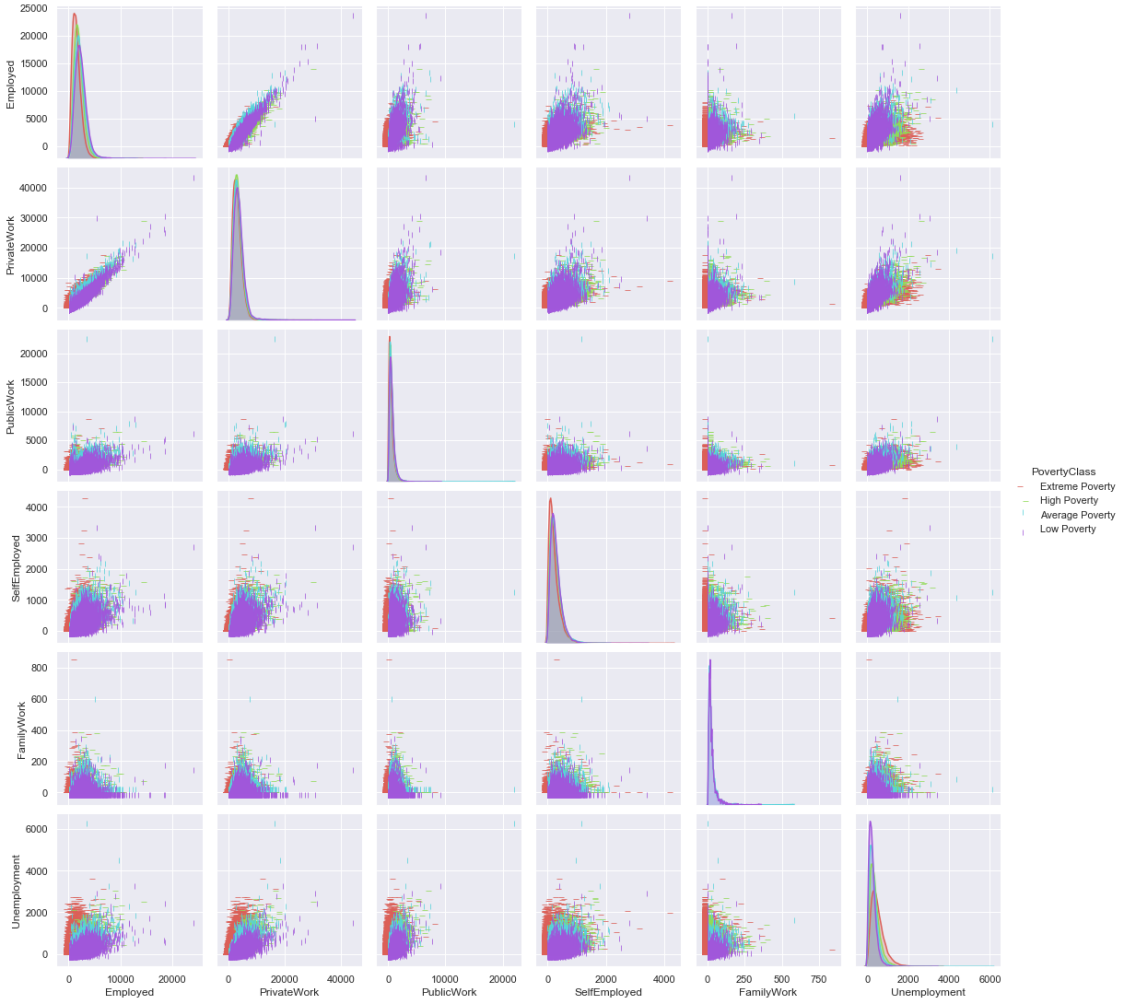


1.7.4 Employment Variable Cluster with Poverty Class Analysis

When reviewing the employment variable cluster there is a lot of overlap. However, we do see some seprataion of Poverty Class when evaluating the scatterplot for Unemployed v Employed. This is important to note moving forward.

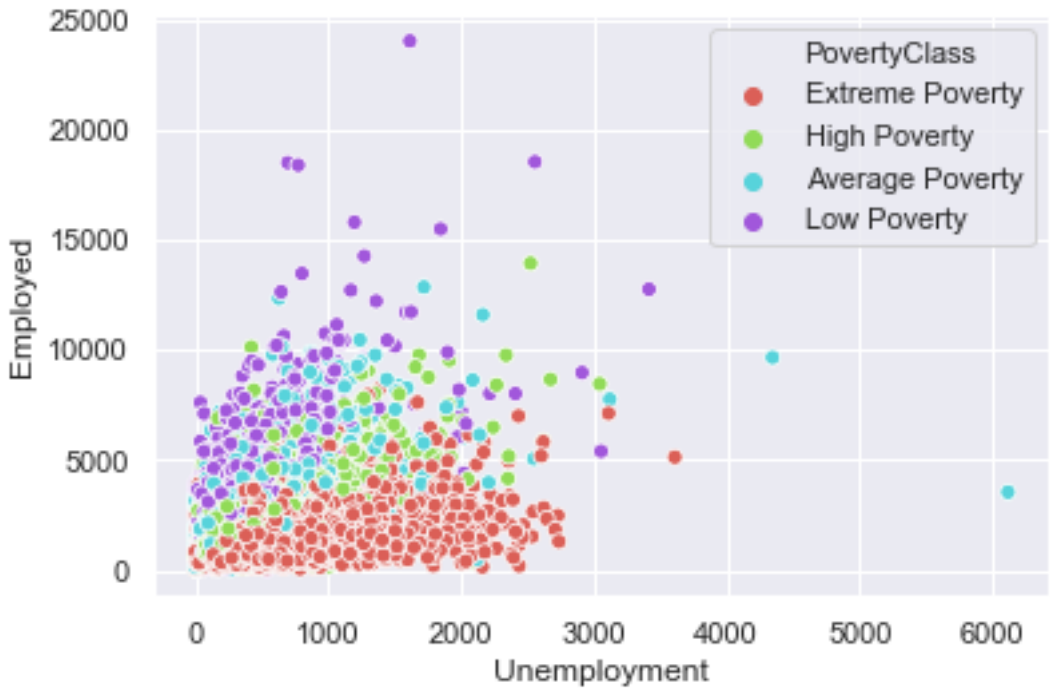
[164]:
```
#Employment Cluster scatterplot colored by Poverty Class
empscatterPC = sns.pairplot(empcluster, hue="PovertyClass", hue_order=["Extreme␣
↪Poverty","High Poverty","Average Poverty", "Low Poverty"], palette = "hls",␣
↪markers=[0, 1, 2, 3])
```

```
/Users/matt/opt/anaconda3/envs/ML1/lib/python3.7/site-
packages/seaborn/distributions.py:369: UserWarning: Default bandwidth for data
is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
/Users/matt/opt/anaconda3/envs/ML1/lib/python3.7/site-
packages/seaborn/distributions.py:369: UserWarning: Default bandwidth for data
is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
```



A closer look at Employed versus Unemployment by Class

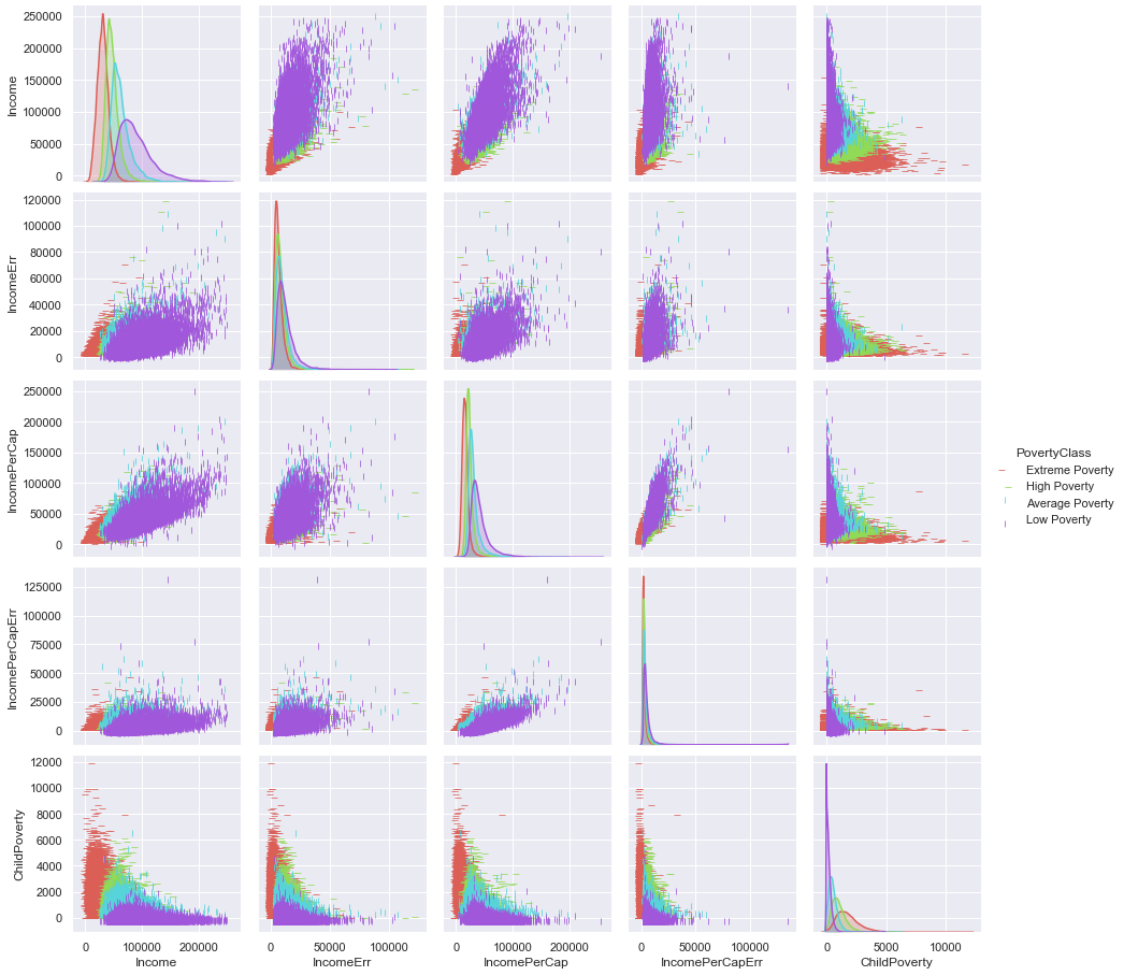
[165]:
```
#Employed v Unemployed
ax = sns.scatterplot(x="Unemployment", y="Employed",
                     hue="PovertyClass", hue_order=["Extreme Poverty","High␣
↪Poverty","Average Poverty", "Low Poverty"], palette = "hls",data=empcluster)
```



1.7.5 Income Variable Cluster with Poverty Class Analysis

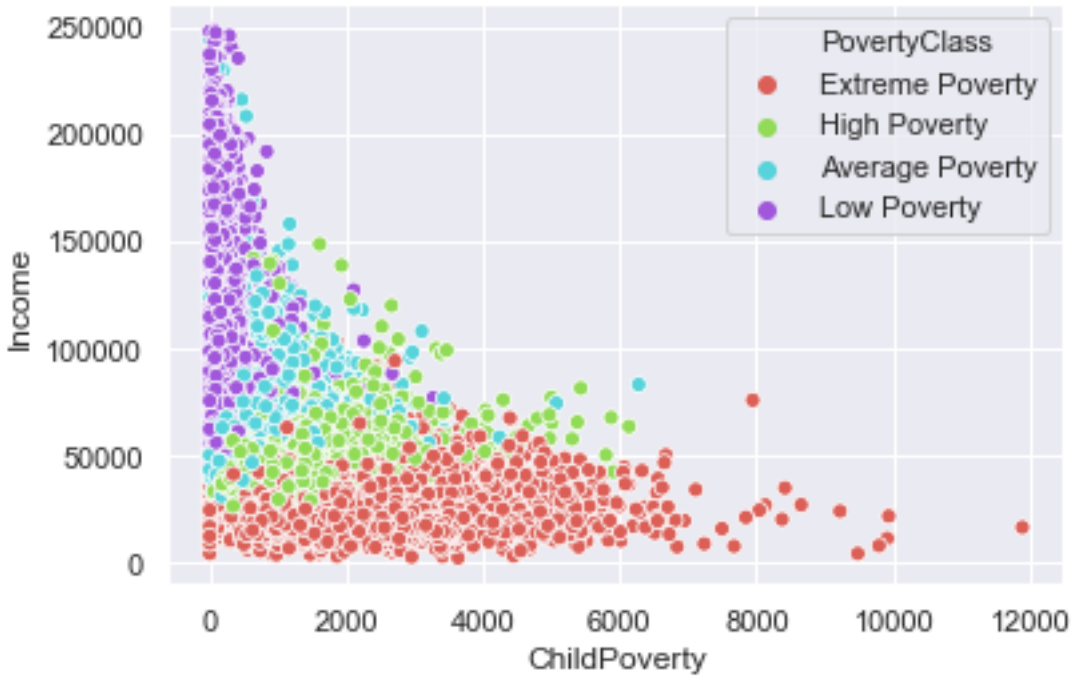
When reviewing the income variable cluster with the poverty class levels highlighted the most notable graph is the ChildPoverty versus Income scatterplot and the strongest separation we've seen among all the scatterplots so far. We can clearly see that extreme and high poverty are more common and in high numbers when the income is lower. Indication this would be an important variable in our classification model.

[166]:
```
#Income Cluster scatterplot colored by Poverty Class
incscatterPC = sns.pairplot(inccluster, hue="PovertyClass", hue_order=["Extreme␣
↪Poverty","High Poverty","Average Poverty", "Low Poverty"], palette = "hls",␣
↪markers=[0, 1, 2, 3])
```



A closer look at ChildPoverty versus Income by Class

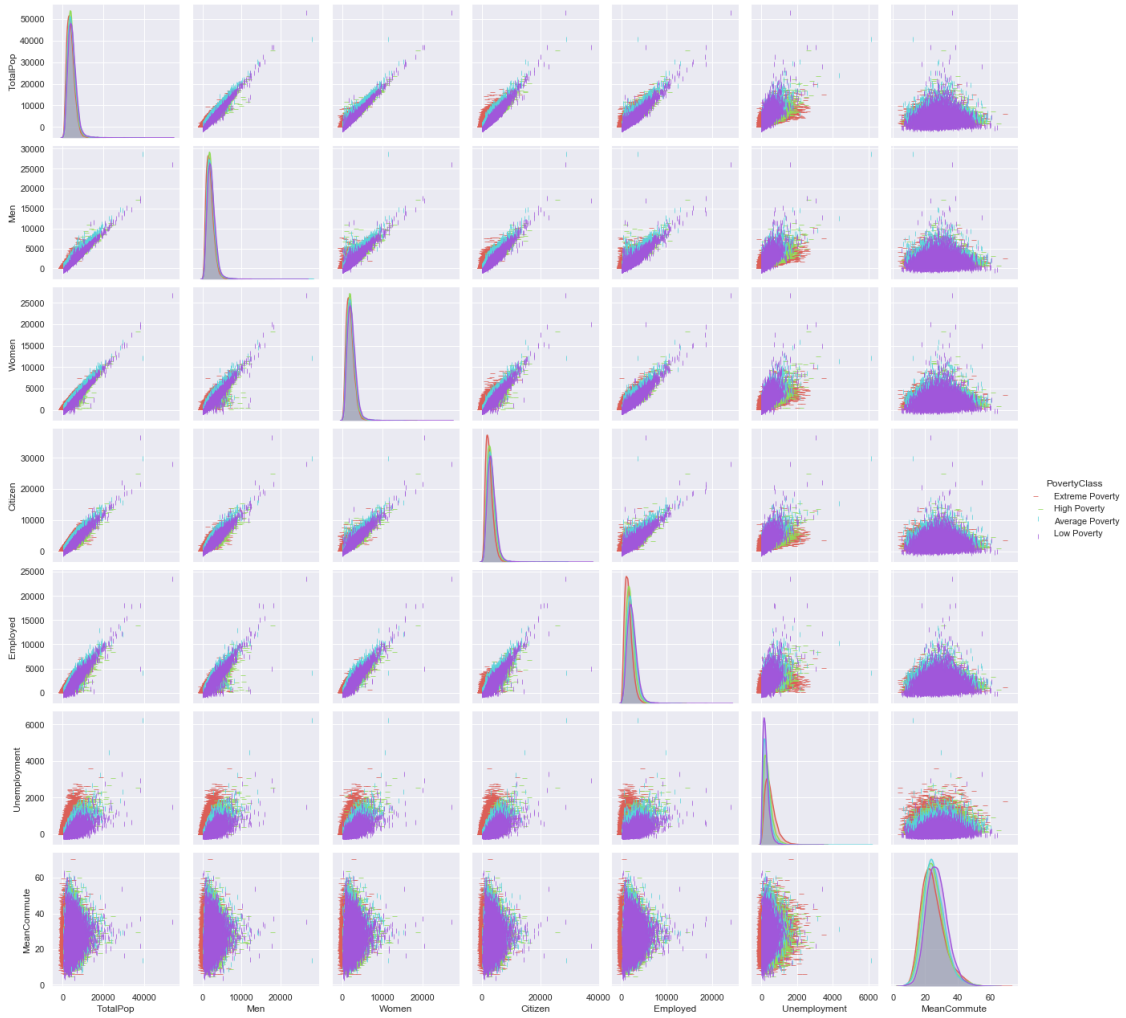
[167]:
```
#ChildPoverty v Income by Class
ax = sns.scatterplot(x="ChildPoverty", y="Income",
                     hue="PovertyClass", hue_order=["Extreme Poverty","High␣
↪Poverty","Average Poverty", "Low Poverty"], palette = "hls",data=inccluster)
```



1.7.6 Additional Variable Cluster with Poverty Class Analysis

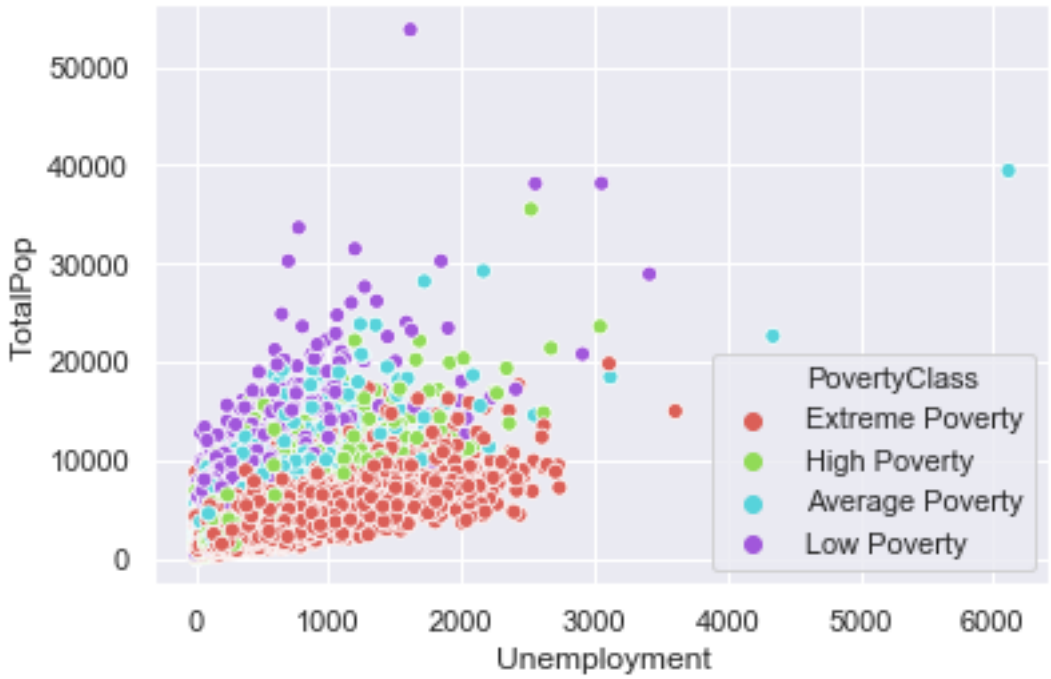
Lastly, we reviewed the additional variable cluster with the poverty class highlighted. Again we see a lot of overlap, however it would be important to note that the Unemployment variable seems to see the most separation between the poverty levels. This would indicate this is an important variable in determining poverty level classification during our model building.

[168]:
```
#Additional Cluster scatterplot colored by Poverty Class
addscatterPC = sns.pairplot(addcluster, hue="PovertyClass", hue_order=["Extreme␣
↪Poverty","High Poverty","Average Poverty", "Low Poverty"], palette = "hls",␣
↪markers=[0, 1, 2, 3])
```



A closer look at Unemployment versus TotalPop by Class

[169]:
```
#Unemployment v TotalPop by Class
ax = sns.scatterplot(x="Unemployment", y="TotalPop",
                     hue="PovertyClass", hue_order=["Extreme Poverty","High␣
↪Poverty","Average Poverty", "Low Poverty"], palette = "hls",data=addcluster)
```

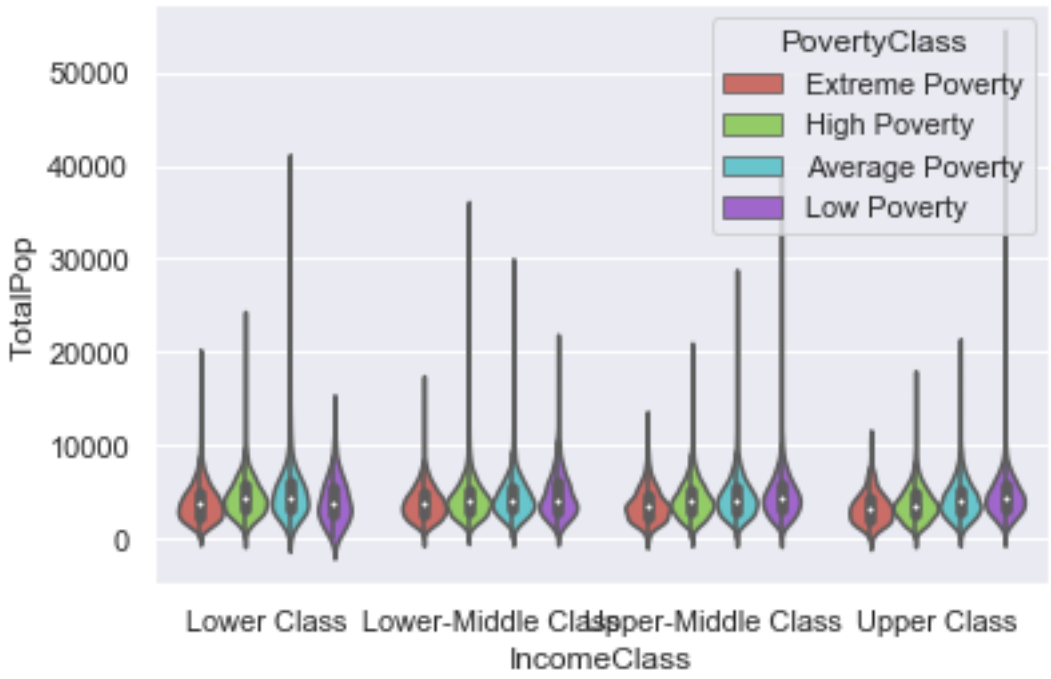


1.7.7 Income Class Bin by Poverty Class Analysis

When we take a loot at the Income as a class instead of as a numerical value we see some clear weight towards Extreme Poverty in the Lower Class income level. We also see a large range of Low Poverty when we look at the Upper Class income level. These trends are expected and are a little easier to read than the scatter plots. We may want to move forward with income classed instead of numerical. However, we would then lose the strength of separation we found in the Child Poverty and Income by Class evaluation earlier.

[170]:
```
#TotalPop by Income Class by Poverty Class
sns.violinplot(x="IncomeClass", y="TotalPop",␣
↪hue="PovertyClass",hue_order=["Extreme Poverty","High Poverty","Average␣
↪Poverty", "Low Poverty"], data=data2015,palette='hls')
```

[170]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x14503bdd0>
```



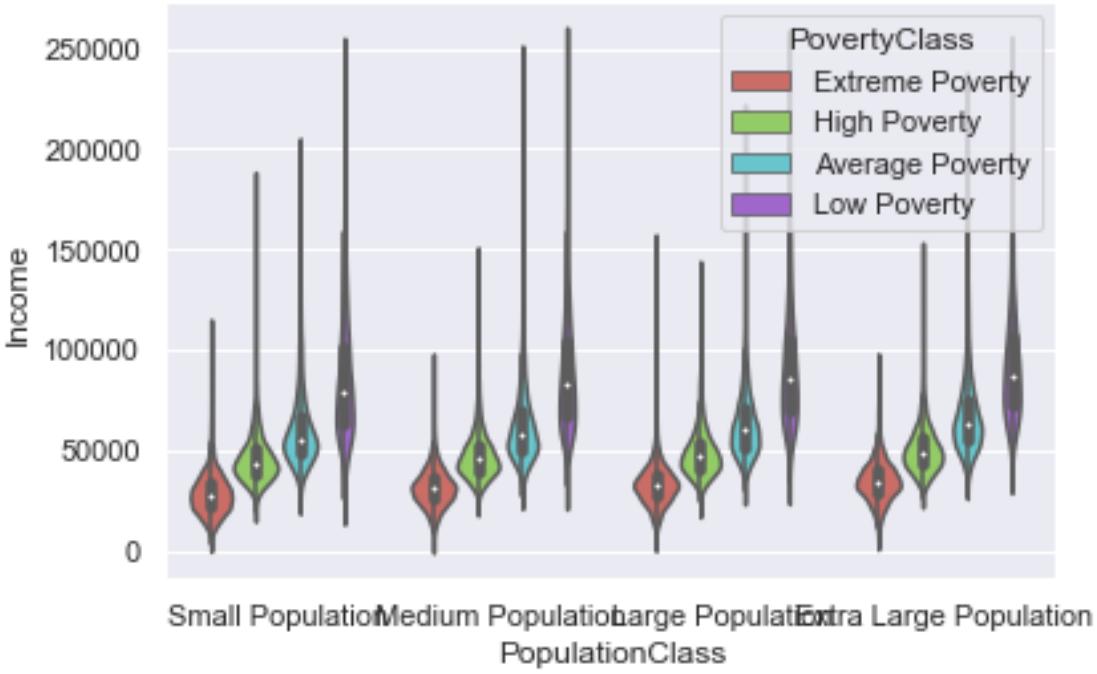
1.7.8 Population Class Bin by Poverty Class Analysis

This chart came our very interesting. Here we can clearly see that no matter the population size the volume of Extreme Poverty is always the highest and then decends in volme as it goes through the Poverty Classes to the weathiest group. This brings a saddening light to the weathiest individuals with the lowest amount of poverty seem to the be lowest in population. The Population Class might be more information for our modeling than as a numerical value.

Extreme Poverty > High Poverty > Average Poverty > Low Poverty

[171]:
```
#Income by Population Class by Poverty Class
sns.violinplot(x="PopulationClass", y="Income",␣
↪hue="PovertyClass",hue_order=["Extreme Poverty","High Poverty","Average␣
↪Poverty", "Low Poverty"], data=data2015,palette='hls')
```

[171]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x19a1a9490>
```



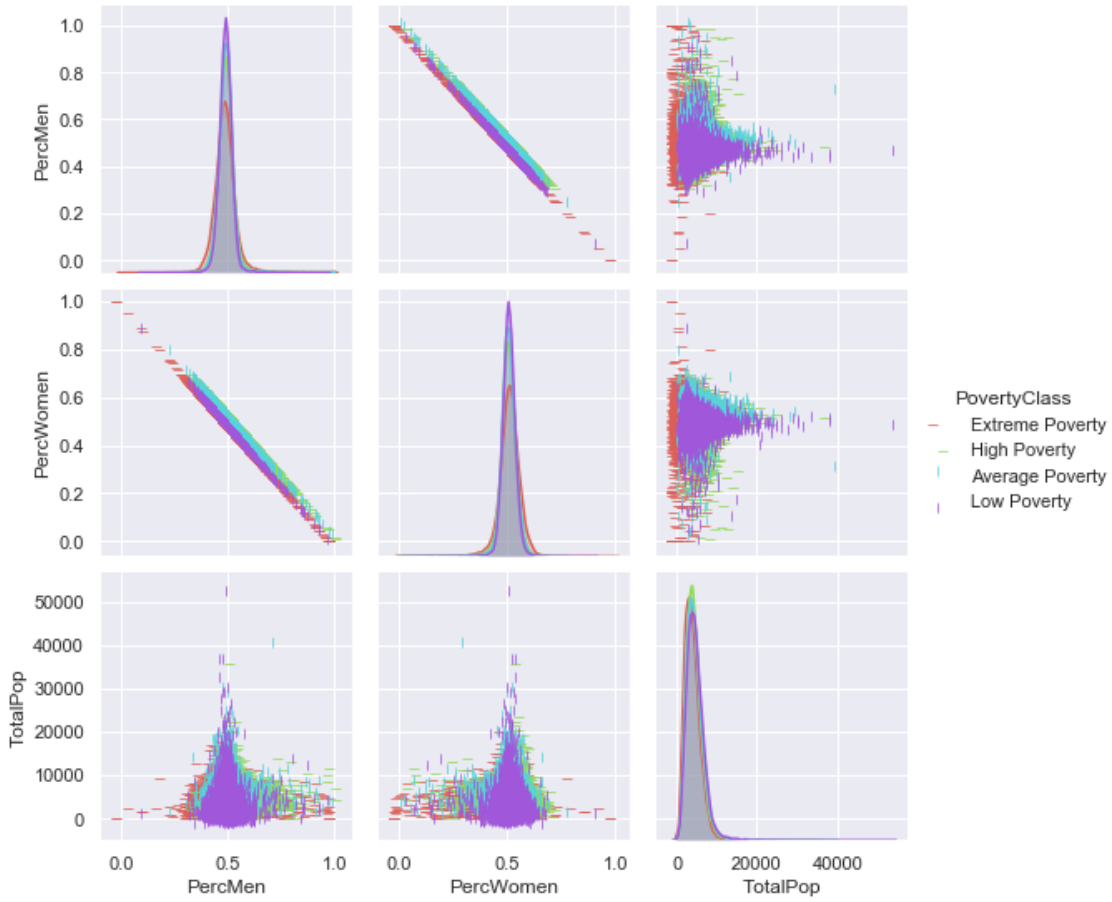
1.7.9 Gender Percentage by Poverty Class Analysis

When evaluating the percentage of genders by class we do not see a lot of separation. We will look closer as the individual scatterplots to support this conclusoin.

[172]:
```
genderperc = data2015.copy()
```

[173]:
```
genderpercclusterˍ
↪genderperc[['PercMen','PercWomen','TotalPop','PovertyClass']]
```

[174]:
```
#Additional Cluster scatterplot colored by Poverty Class
genderperc
genderpercscatterPC = sns.pairplot(genderpercclusterˍ, hue="PovertyClass",ˍ
↪hue_order=["Extreme Poverty","High Poverty","Average Poverty", "Lowˍ
↪Poverty"], palette = "hls", markers=[0, 1, 2, 3])
```

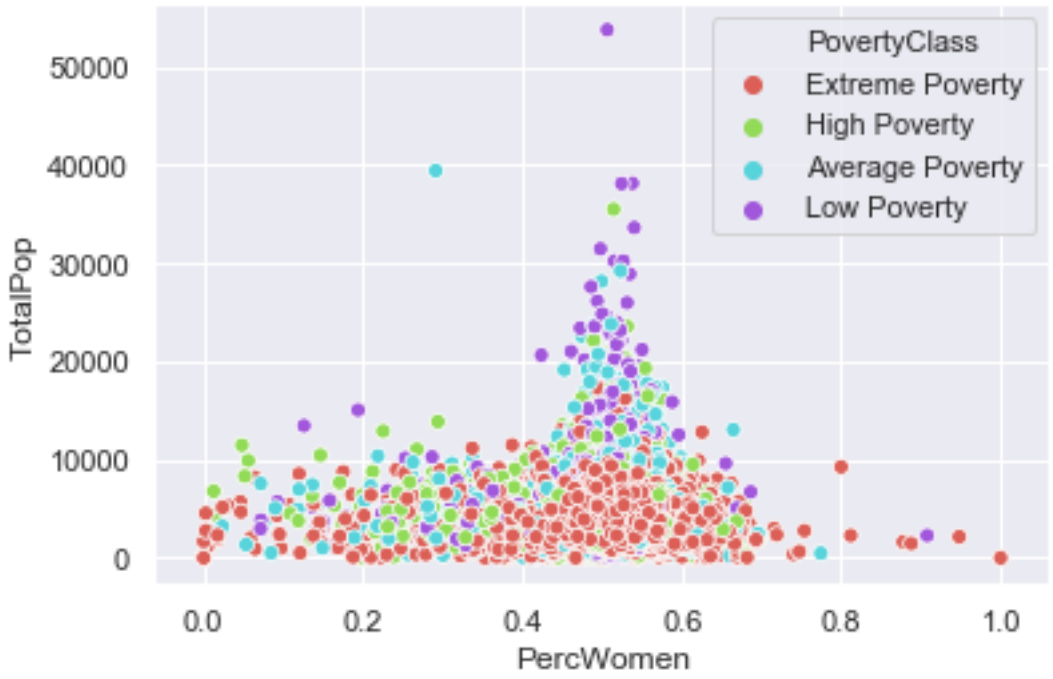


A closer look at TotalPop versus PercMen by Class

[175]:
```
#PercMen v TotalPop by Class
ax = sns.scatterplot(x="PercMen", y="TotalPop",
                     hue="PovertyClass", hue_order=["Extreme Poverty","High␣
↪Poverty","Average Poverty", "Low Poverty"], palette =␣
↪"hls",data=genderperccluster)
```

A closer look at TotalPop versus PercWomen by Class

[176]:
```
#PercWomen v TotalPop by Class
ax = sns.scatterplot(x="PercWomen", y="TotalPop",
                     hue="PovertyClass", hue_order=["Extreme Poverty","High␣
↪Poverty","Average Poverty", "Low Poverty"], palette =␣
↪"hls",data=genderpercclusterr)
```



1.8 New Features

1.8.1 Poverty Class

As dsicussed in our Business Understanding section, the goal of our data set is to predict which Poverty Class (Low Poverty, Average Poverty, High Poverty, or Extreme Poverty) a geographical area is. This is a catagorical variable, but the Poverty Percentage variable in our data set is a continious variable showing poverty percentage of a census tract from 0 - 100%. To create the categorical Poverty Class variable from the Poverty Percentage we binned this variable by the quartiles of the continuous (Quartile 1: 0-7%: Low Poverty, Quartile 2: 7-12.5%: Average Poverty, Quartile 3: 12.5-22%: High Poverty, Quartile 4: 22%+: Extreme Poverty). This changed what would have initially been a continuous variable prediction to a categorical variable prediction and therefore framing our lab as a classification problem. The code for creating this new variable can be seen below:

1.8.2 Other Ethnicity Variable

As was previously discussed the ethnicity percentage variables do not add up to 100%. Therefore we created a new variable, "Other", that fills the gap. This calculation takes 100% - the Hispanic, White, Black, Native, Asian, and Pacific percentages. This variable gives us another to add to the Race Variable Cluster. Below is the code for the variable.

1.8.3 Aggregated Variables

As we previously discussed, we determined that it would work to our advantage if we converted percentage variables back to their whole numbers. We felt this regarding the a visual perspective

and the ability to aggregate geographically will be beneficial moving forward. Below is the code we used in order to do this and create a new data set data2015agg.

1.8.4 Income Class

As previously discussed, in order to try and obtain deeper insight into the Class variable compared to our Income variable, we decided to bin our income and create levels. This is a categorical variable labeling each census tract as lower class, lower-middle class, upper-middle class, or upper class. The distiction between the four levels of the factor is made based on the quartiles for the Income Per Capita variable from our simple statistics table. Below is the code for the feature.

1.8.5 Population Size Class

Again, in an attempt to gain deeper insight into our data, we determined that binning our Population variable could provide better insight. This is a categorical variable labeling each census tract as a small, medium, large, or extra large population area. The distinction between the four levels of the factor is made based on the quartiles from our simple statistics table. The code for this feature creation is below.

1.8.6 Men/Women Percentage Breakdown

As previously discussed, since we were given men and women totals instead of percentages. We transformed this to percentages to see if any additional insight could be obtained from exmaining this data differently. When we actually do model building this allows us to consider sex/gender as a predictor since many of our other varaibles are percentages, therefore sex/gender is now standardized to the other variable clusters. The code for this feature creation can be found below.

1.9 Exceptional Work

1.9.1 New Features

The new features that we created that we described in our New Features were both thought of and created in order to have an in depth exploratory data analysis of our dataset. We would like to put forth this work to be considered with Exceptional Work.

1.9.2 Python Visualizations

We took extra effort to dissect and depict our variables in a fashion that is easily digestible for our reader. We would like to put forth this work to be considered with Exceptional Work.

1.9.3 Tableau Visualization

Python's vizualization capabilties fall short of what we can do in third party tools such as Tableau. Therefore, we have leveraged our data set with Tableau to build unique visualizations.

Since the data set we are working with is census data, it is inherently geographic in nature. Therefore, some of the most powerful visualizations we can create will have geopgraphical components.

To build these visualizations in Tableau we combined our data set with spatial files that map out census tracts provided by census.gov.

This effort culminated in an interactive Tableau Public Dashboard that can be seen at: https://public.tableau.com/profile/reagan.t.meagher#!/vizhome/PredictingPovertyClass-NewYorkCity/NewYork

We would like these Tableau visualizations in the linked Tableau Dashboard to be considered for Exceptional Work.

[183]:
```
from IPython.display import Image
```

This Tableau visualization shows the Greater New York City by Income Class:

[184]:
```
Image(url= "https://raw.githubusercontent.com/megnn/SMUMSDS_ML1/master/
↪Income_Class.png")
```

[184]:
```
<IPython.core.display.Image object>
```

This Tableau visualization shows the Greater New York City by Population Class:

[185]:
```
Image(url= "https://raw.githubusercontent.com/megnn/SMUMSDS_ML1/master/
↪Population_Class.png")
```

[185]:
```
<IPython.core.display.Image object>
```

This Tableau visualization shows the Greater New York City by Poverty Class:

[186]:
```
Image(url= "https://raw.githubusercontent.com/megnn/SMUMSDS_ML1/master/
↪Poverty_Class.png")
```

[186]:
```
<IPython.core.display.Image object>
```