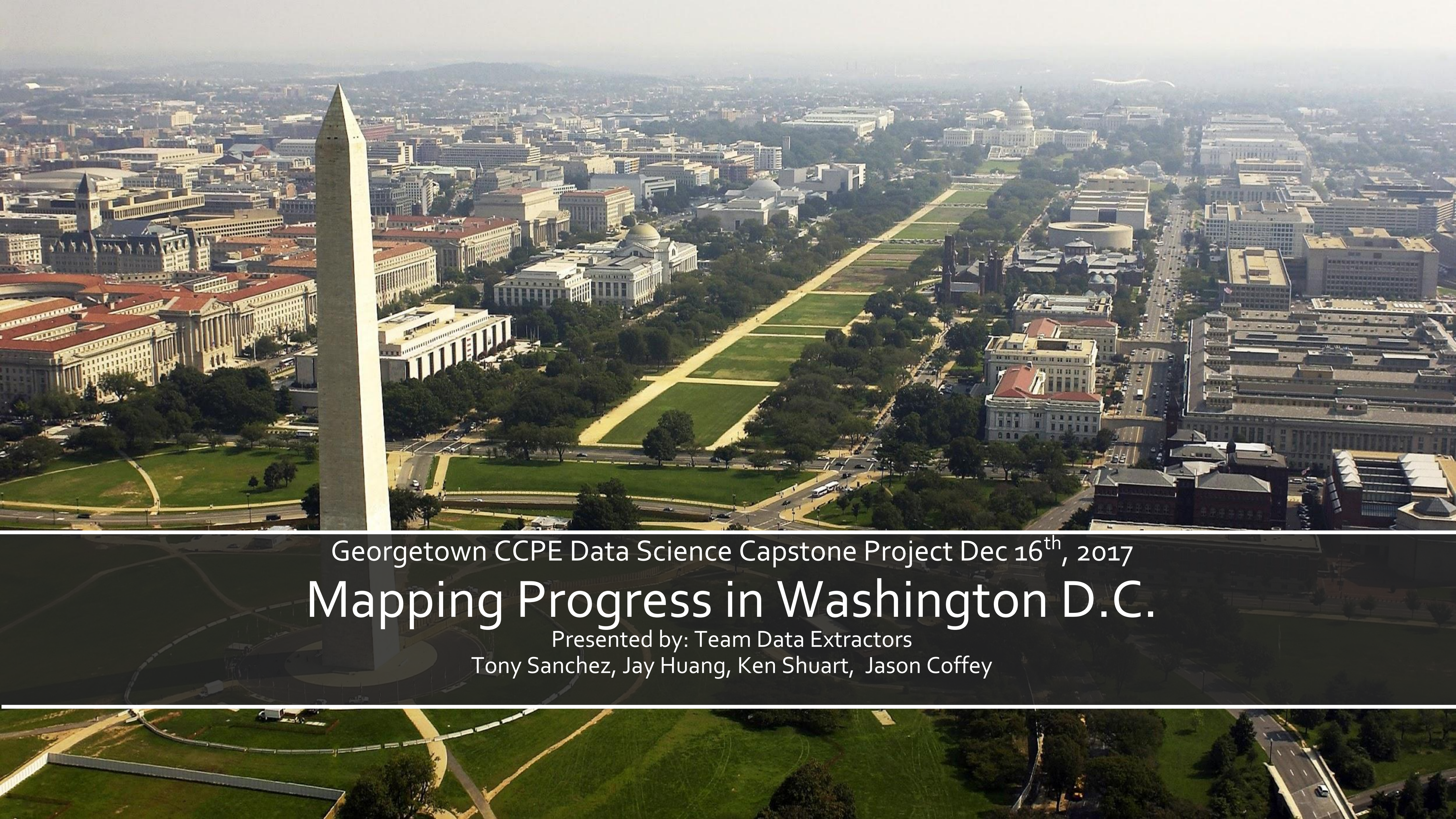


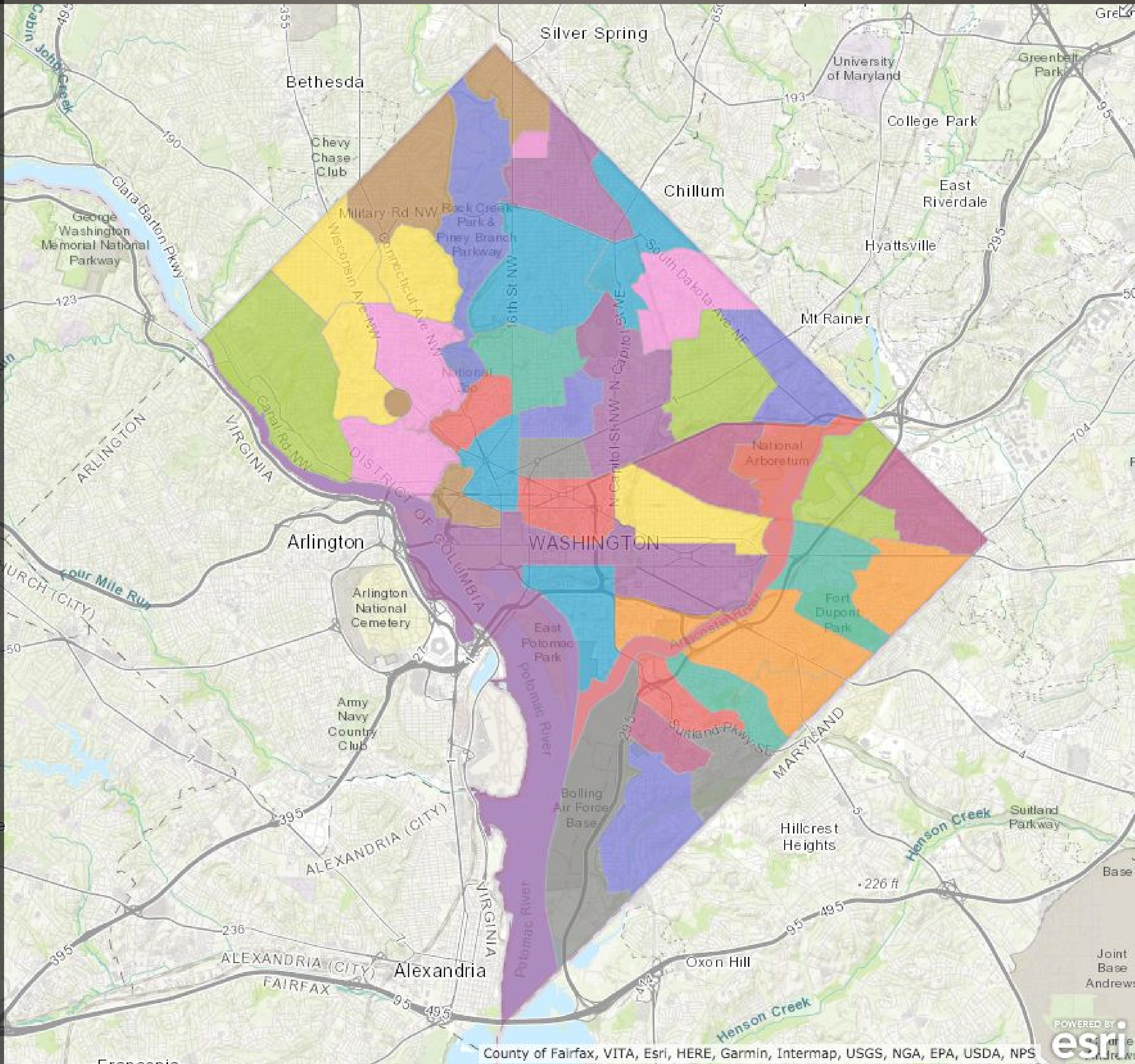
Georgetown CCPE Data Science Capstone Project Dec 16th, 2017

Mapping Progress in Washington D.C.

Presented by: Team Data Extractors
Tony Sanchez, Jay Huang, Ken Shuart, Jason Coffey

Overview

- Project Description
- Implementation of the Data Science Pipeline
- Key Insights
- Lessons Learned

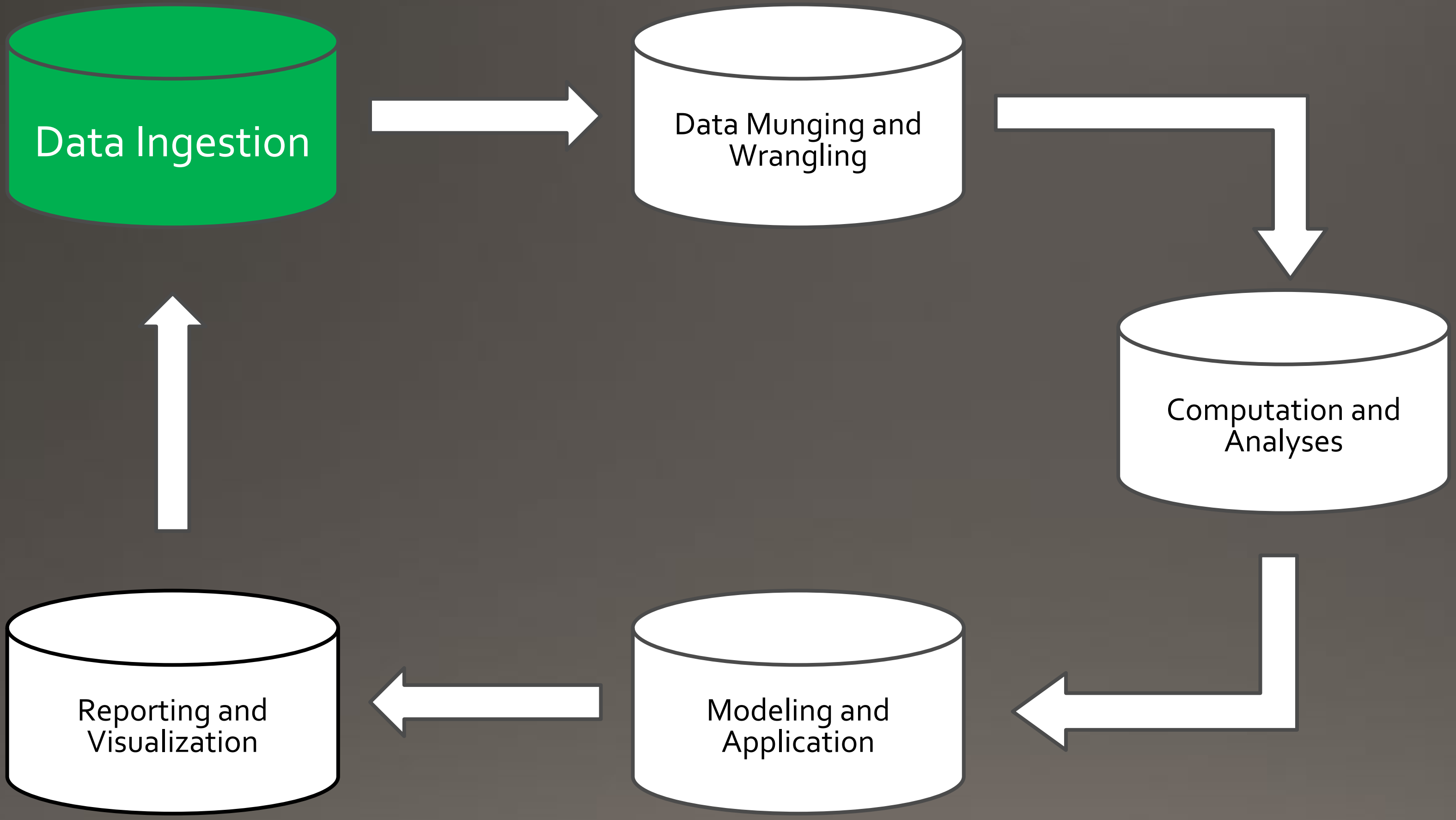


Project Description

- Apply Next City progress model on Washington D.C.
- Apply Machine Learning Clustering Models on the same data
  - Added monthly data
  - Added features
- Hypothesis: Prove that our implementation is superior to the NextCity model

https://nextcity.org/features/view/philadelphia-neighborhoods-gentrification-mapping-growth

Data Ingestion



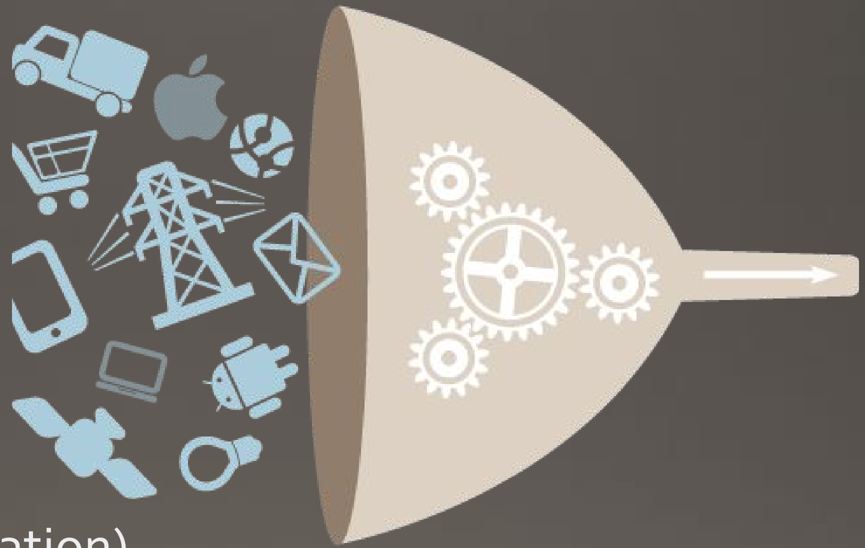
Data Ingestion

Data Sources

- Census.gov
- Opendata.dc.gov
- Zillow.com
- USBoundary.com

Time Range: 2011-2015

- Mean Household Income ($)
- Median Home Price ($)
- Population (race, age, gender, native born, education)
- Crime (total crimes, violent & theft)
- Poverty (pre-tax income)



Data Ingestion

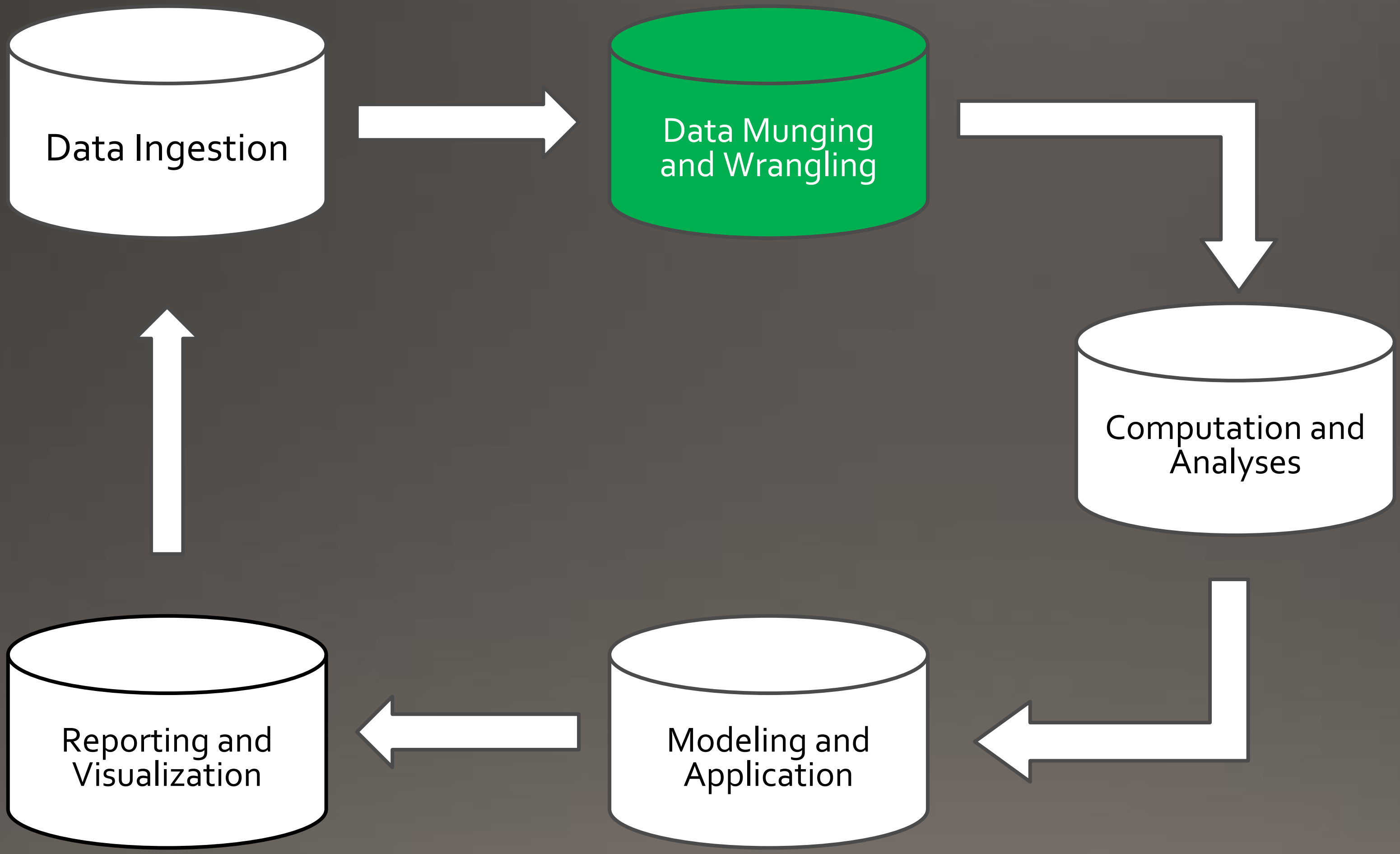
Data Store

- AWS RDS (Amazon Web Services)
- PostgreSQL
- Beautiful Soup
- User accounts, permissions (tables had to be public access)

Worm Store

- Drop box

Data Wrangling



Data Wrangling

- Key Objectives
  - Aggregate to 46 neighborhoods
  - Generate instances for data frame
- Aggregation
  - Identify boundary space
  - 1 to many Census Tracts, Point data, and Zillow into named neighborhood
- Transformation
  - Dollar values, Frequency of Occurrence, Ratio, Ranges, Merge
  - Impute months from yearly
  - Compare mean values to DC mean

```python
import shapefile
from shapely.geometry import Point, shape
"""
Returns neighborhood_cluster value as well as the description of the neighb
"""
def getNeighborhoodClusterLatLon(path, lat, lon):
    # read your shapefile
    sf = shapefile.Reader(path)
    num_shapes = sf.numRecords
    # get the shapes
    shapes = sf.shapes()
    # format the point
    point = Point(lon, lat)
    # iterate through the shapes and check each polygon for the point
    for i in range(num_shapes):
        polygon = shape(shapes[i])
        if (polygon.contains(point)):
            cluster = sf.record(i)
            # return both the cluster neighborhood and neighborhood names
            return(cluster[2], cluster[3])
    return "Neighborhood not found."
```

Data Wrangling

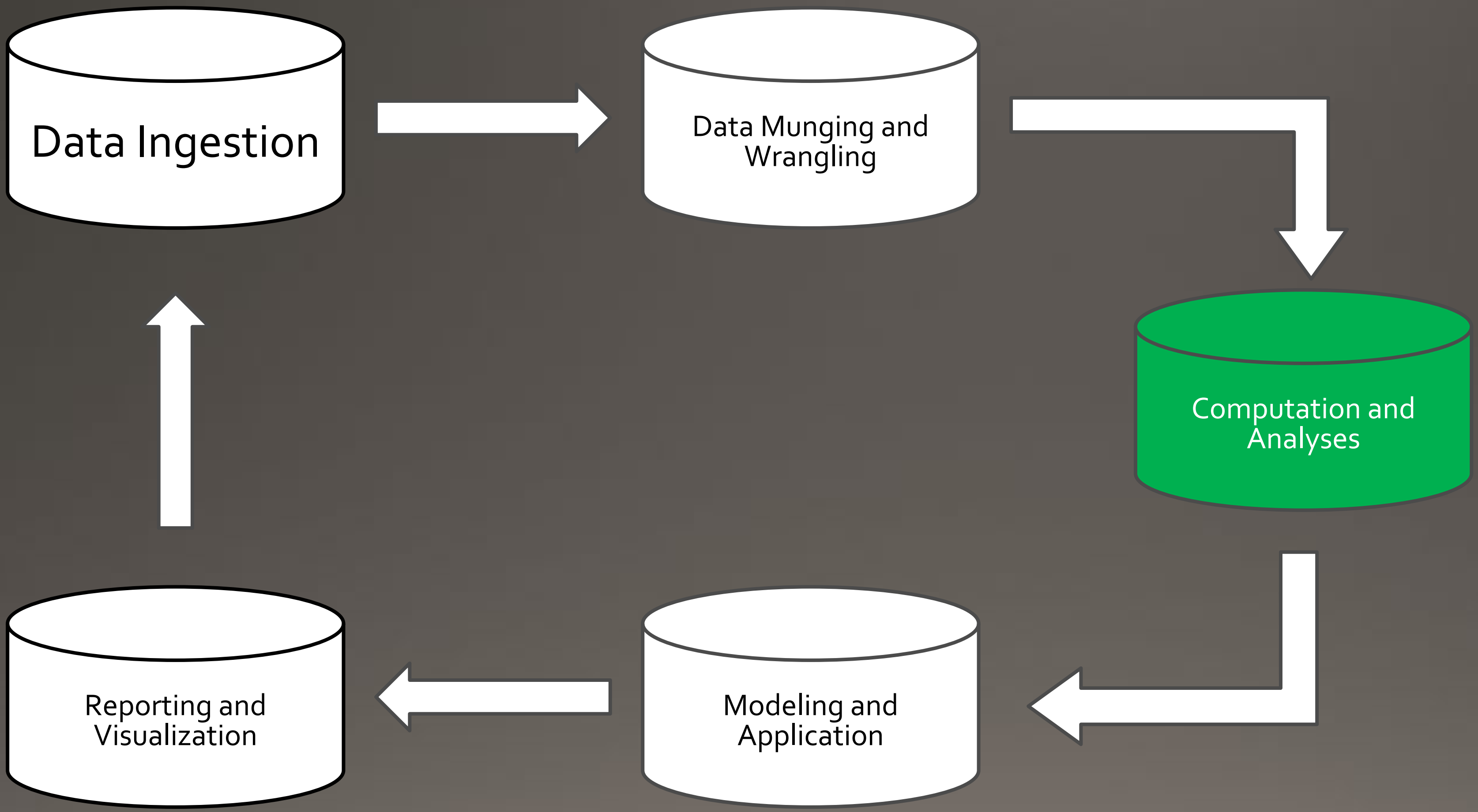
- Data Frame Preparation
  - 2640 instances per feature (12 months X 5 years x 44 NBH)
  - 40+ features refined to 21
  - Data sparsity – dropped National Mall & Arboretum

```
In [10]: df.head()
Out[10]:
```

Date	Cluster	Population	White	Black	Asian/Pacific	Native American	Dependency Ratio	M/F Ratio	Own/Rent Ratio	HS Max %	College Educated %	Native Born %	Naturalized %	No Citizen %
2011-01	Cluster 1	17222.0	12990.0	2070.0	1202.0	74.0	0.188872	0.999768	0.694571	0.102557	0.897443	0.784462	0.078562	0.136976
2011-02	Cluster 1	17222.0	12990.0	2070.0	1202.0	74.0	0.188872	0.999768	0.694571	0.102557	0.897443	0.784462	0.078562	0.136976
2011-03	Cluster 1	17222.0	12990.0	2070.0	1202.0	74.0	0.188872	0.999768	0.694571	0.102557	0.897443	0.784462	0.078562	0.136976
2011-04	Cluster 1	17222.0	12990.0	2070.0	1202.0	74.0	0.188872	0.999768	0.694571	0.102557	0.897443	0.784462	0.078562	0.136976
2011-05	Cluster 1	17222.0	12990.0	2070.0	1202.0	74.0	0.188872	0.999768	0.694571	0.102557	0.897443	0.784462	0.078562	0.136976

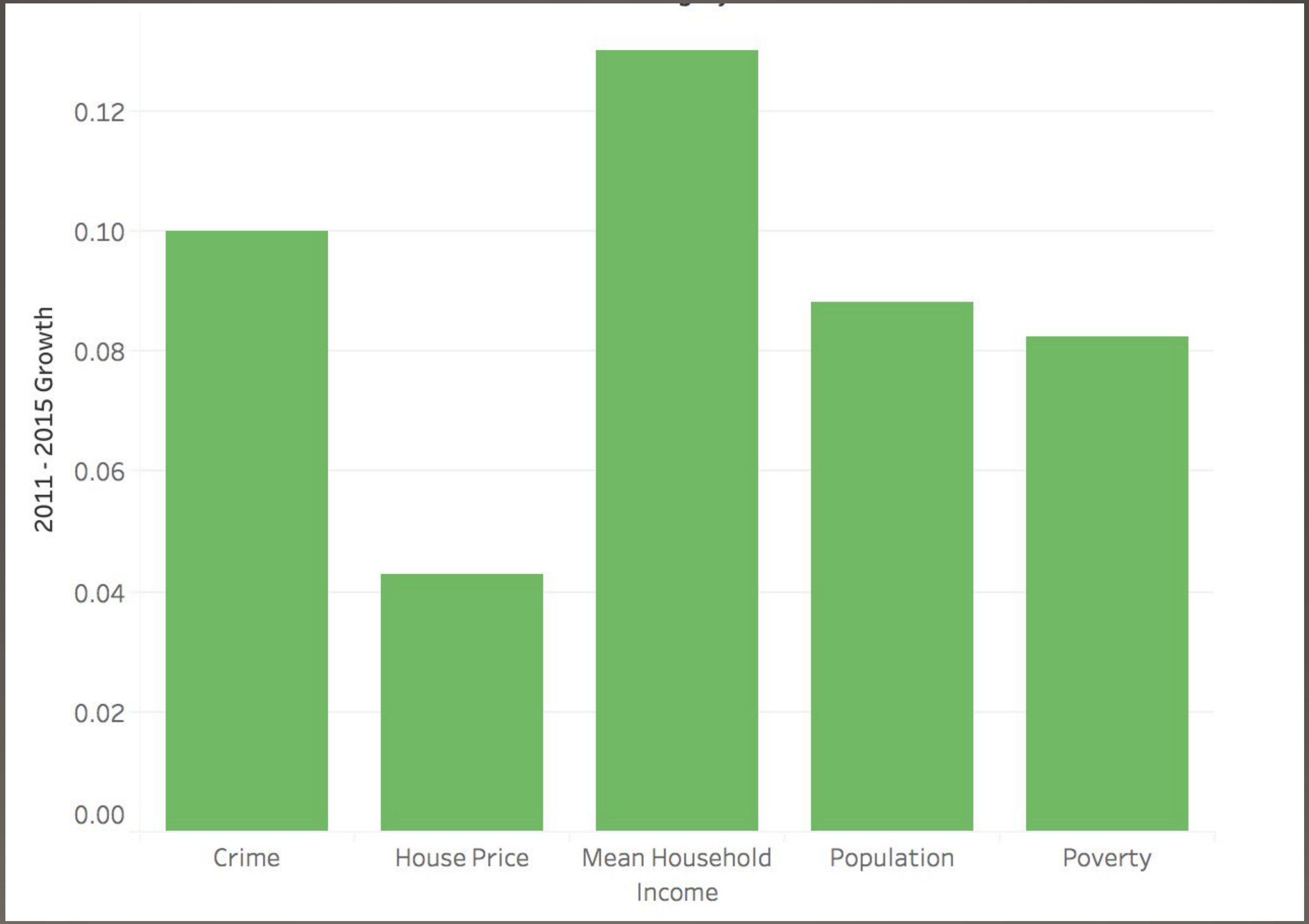
Date	Cluster	Poverty Below 100	Poverty 100-149	Mean Income	Median Rent Price	Median Price Asked	Total Crimes	Violent Crimes	Theft Crimes
2011-01	Cluster 1	969.0	699.0	113868.0	1308.81	3076200.0	55	4	51
2011-02	Cluster 1	969.0	699.0	113868.0	1308.81	3042000.0	70	3	67
2011-03	Cluster 1	969.0	699.0	113868.0	1308.81	3036800.0	87	2	85
2011-04	Cluster 1	969.0	699.0	113868.0	1308.81	3066100.0	59	4	55
2011-05	Cluster 1	969.0	699.0	113868.0	1308.81	3098900.0	104	8	96

Computation and Analyses



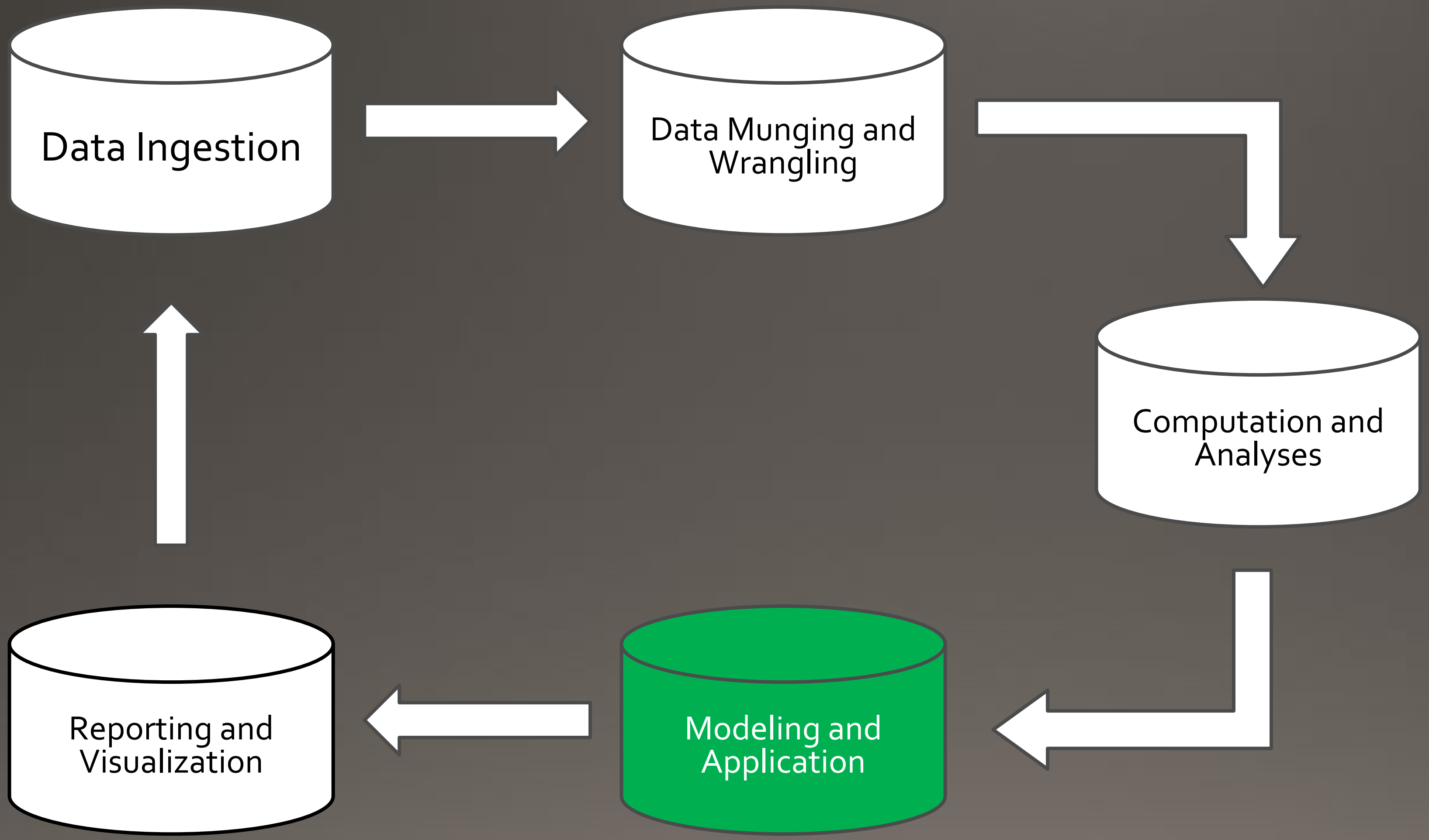
Computation and Analyses

- Time Period Covering
  - 2011 - 2015
- Growth
  - Crime **10%**
  - Home Prices **4.3%***
  - Mean household income **13%**
  - Population **8.81%**
  - Poverty **8.26%***



* Factored negatively in progress score

Modeling and Application



Modeling & Application

- Cluster data using unsupervised learning algorithms
- Label clusters according to socioeconomic characteristics of instances inside cluster
- Determine growth by measuring movement between labeled clusters

scikit learn

```python
def gmm(df, nc, n_components=2):
    """GMM clustering on PCA-reduced data with silhouette scores."""

    print('GMM clustering on PCA-reduced (' + str(n_components) + ' components) ' + 'data')

    # Standardize data
    df_tr = StandardScaler().fit_transform(df)

    # Plot explained variance ratio graph
    pca = PCA().fit(df_tr)
    plt.plot(np.cumsum(pca.explained_variance_ratio_) * 100)
    plt.xlabel('Number of components')
    plt.ylabel('Cumulative explained variance in %')
    plt.title('Cumulative Explained Variance')
    plt.show()

    # Reduce features via PCA
    pca = PCA(n_components=n_components).fit(df_tr)
    reduced_data = pca.transform(df_tr)

    # Plot PCA components composition
    plt.matshow(pca.components_, cmap='viridis')
    plt.yticks([0, 1], ["First component", "Second component"])
    plt.colorbar()
    plt.xticks(range(len(df.columns)),
               df.columns, rotation=60, ha='left')
    plt.xlabel("Features")
    plt.ylabel("Principal components")
    plt.show()

    model = GaussianMixture(n_components=nc, covariance_type='diag')
    model.fit(reduced_data)
    cluster_labels = model.predict(reduced_data)
    dfq['Cluster Labels'] = cluster_labels
```

Machine Learning Pipeline

Standard Scaler → Principal Component Analysis → Gaussian Mixture Model

Standard Scaler

- Different units
- Different order of magnitude

Principal Component Analysis

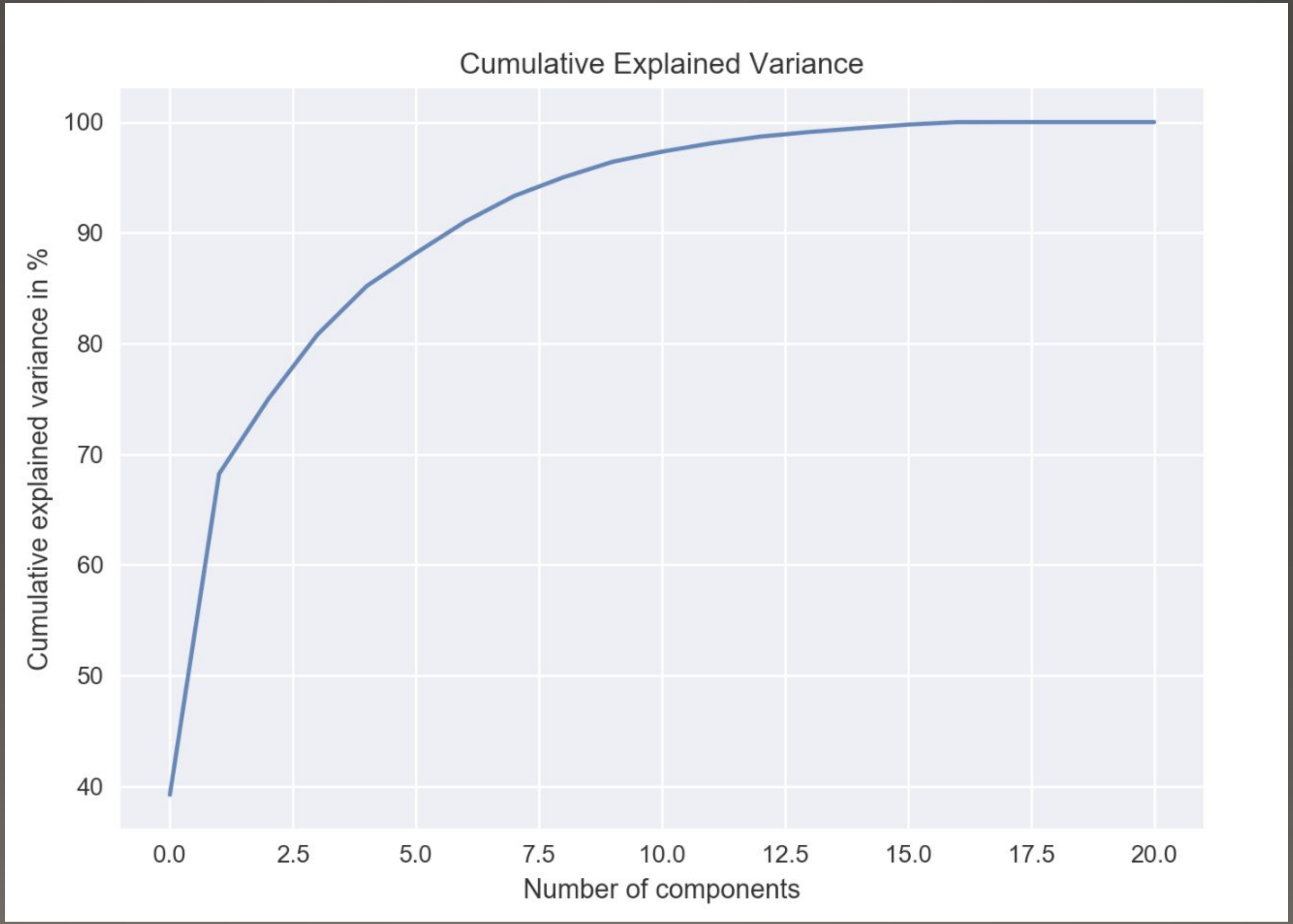
- Reduce complexity
- Prevent overfitting

Gaussian Mixture Model

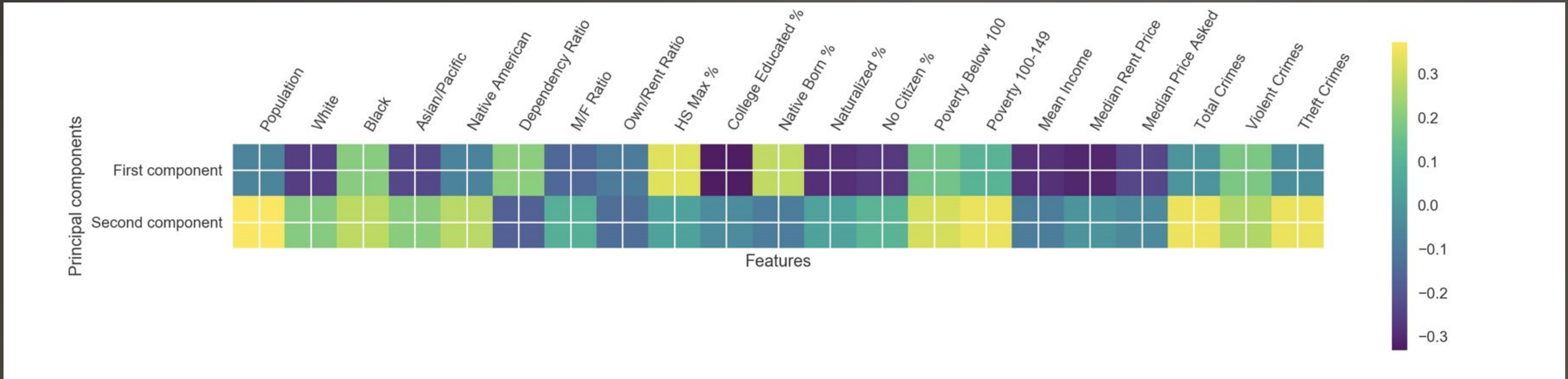
- Flexibility in covariance
- Won't bias cluster sizes

Cumulative Explained Variance

- Amount of variance retained from the original data set
- Chose 2 components with 75% explained variance



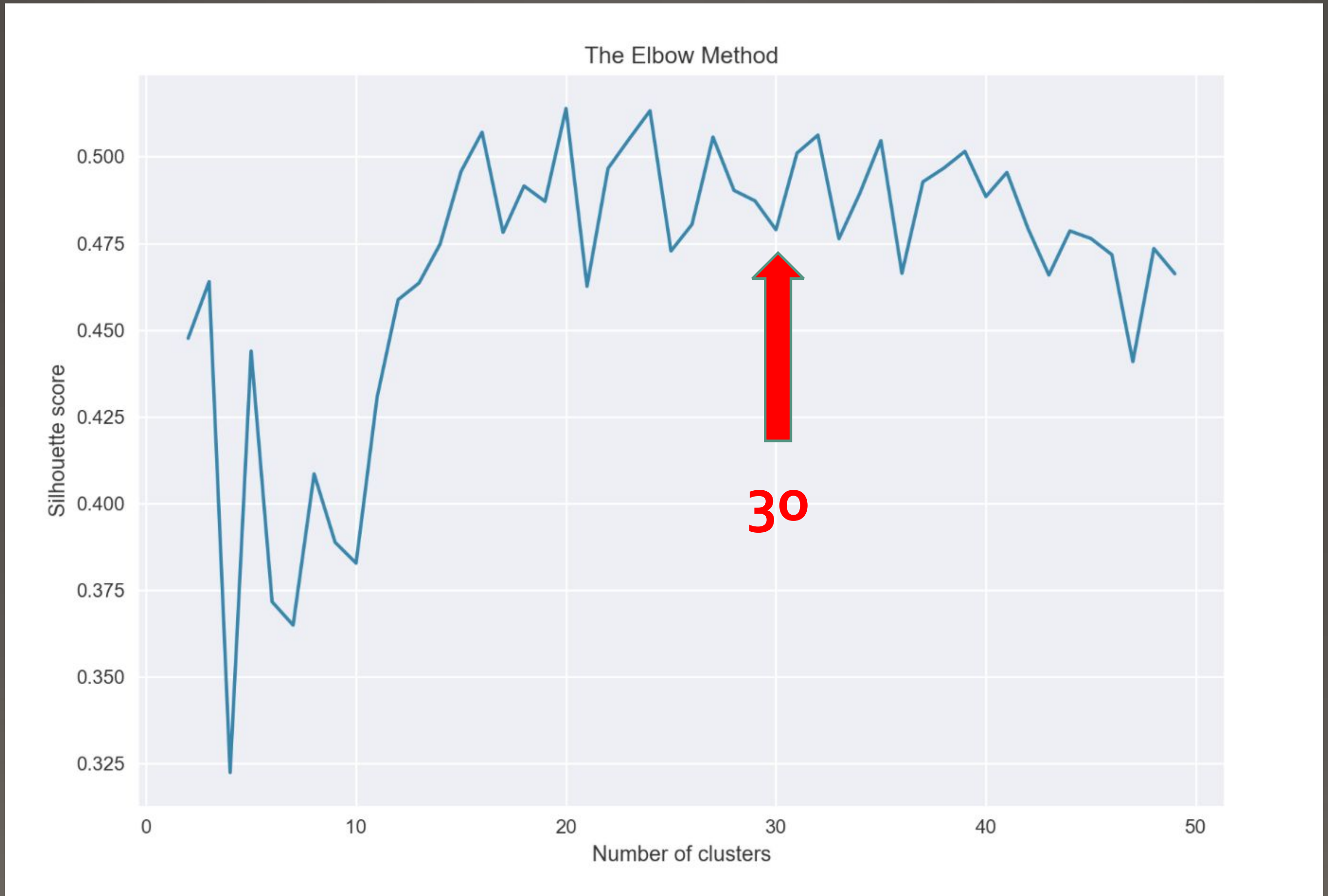
Coefficient Heat Map



- Coefficients of features inside components
- First component: HS max % and poverty
- Second component: population and total crime

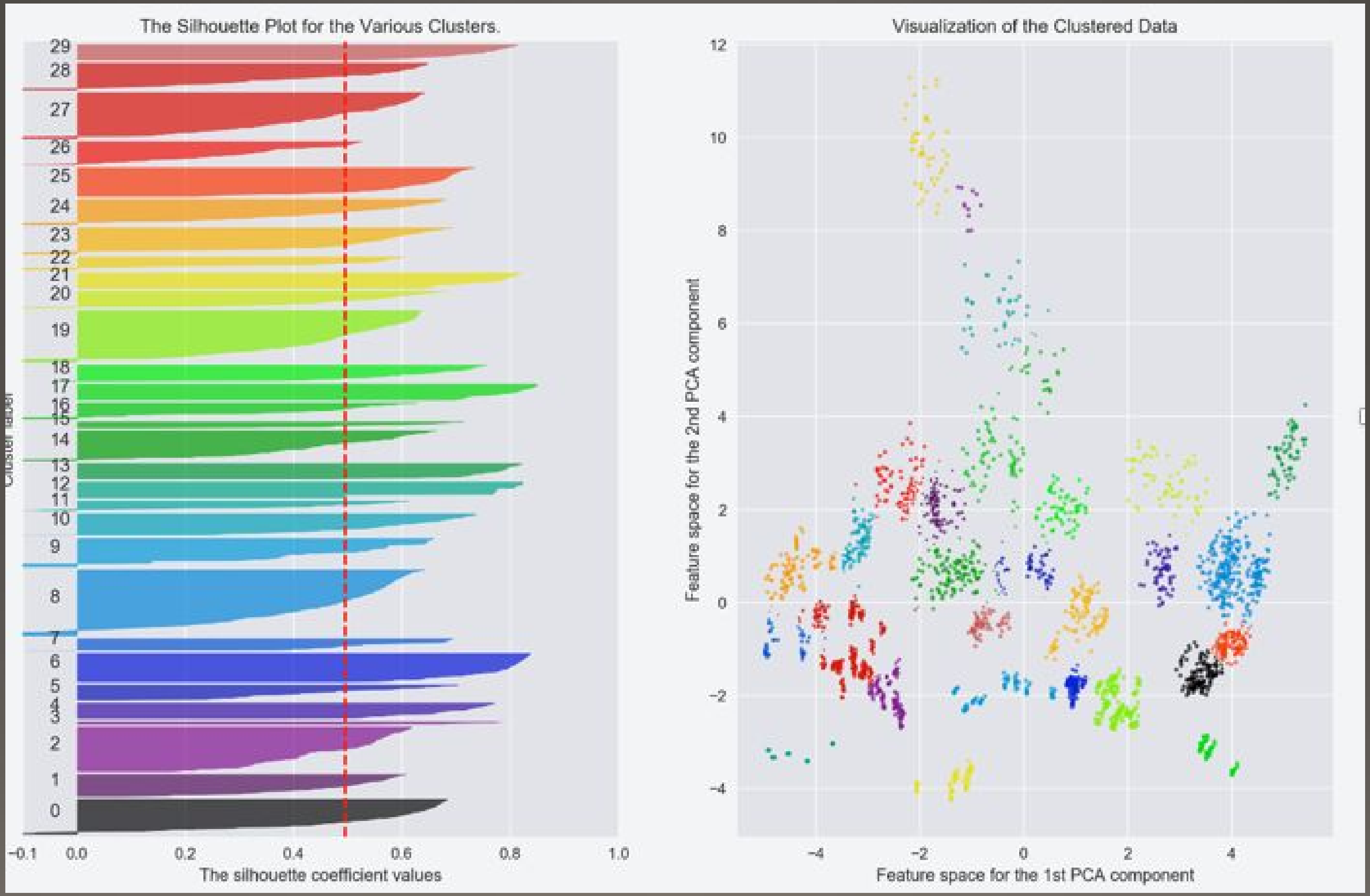
The Elbow Method

- Silhouette score = cohesion and separability of clusters
- Balance between high silhouette score and neighborhood movement between clusters
- 30 clusters chosen



Silhouette Analysis for GMM Clustering

- Clusters are greater than mean silhouette score
- Acceptable cohesion and separability

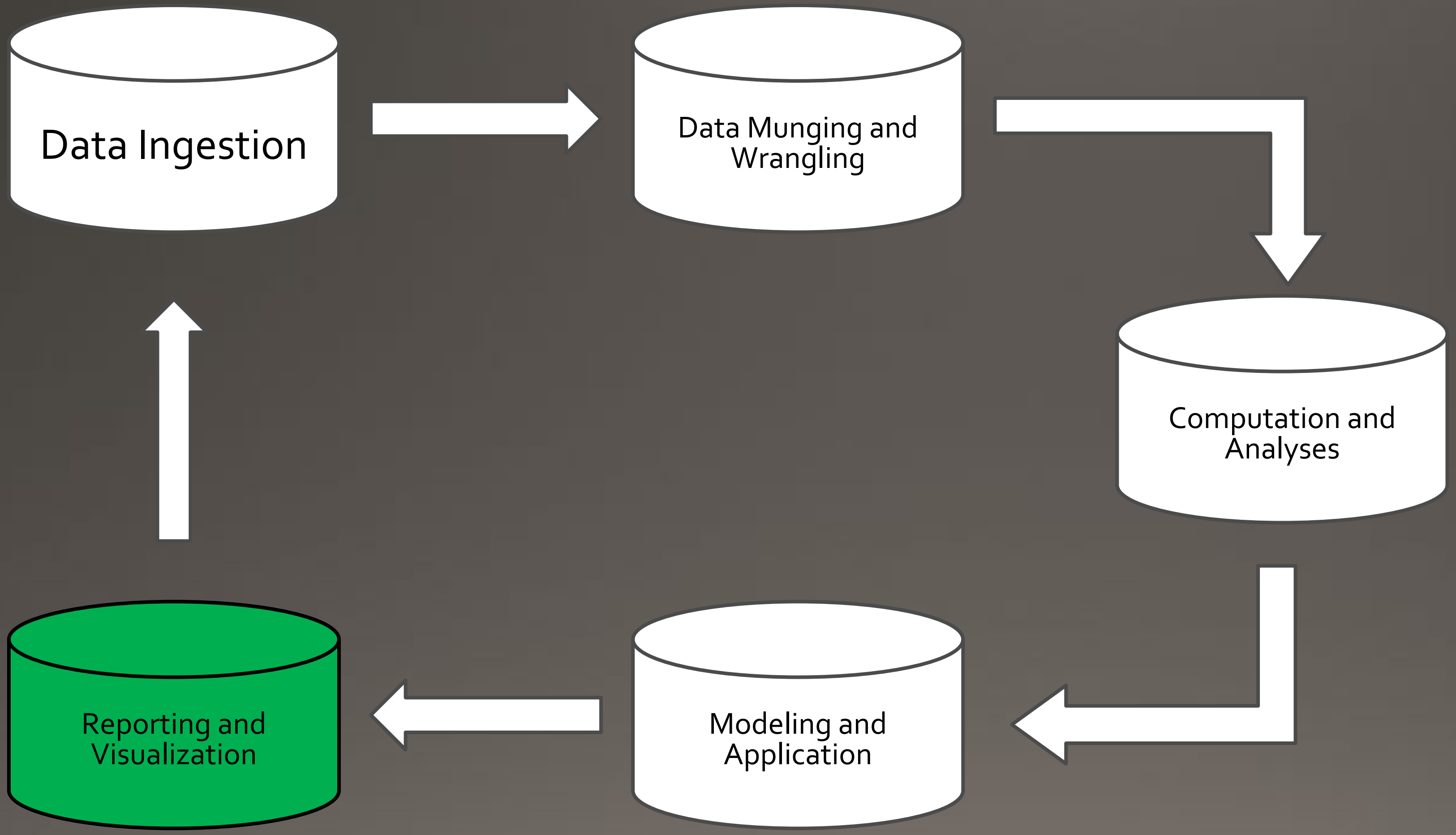


Results

- Clustering did not converge into an optimal solution
- Iterated 1000 times

Neighborhood	Total Score
Near Southeast, Navy Yard	1.496499
West End, Foggy Bottom, GWU	1.250709
Takoma, Brightwood, Manor Park	0.562196
Union Station, Stanton Park, Kingman Park	0.452762
Friendship Heights, American University Park, Tenleytown	0.351286
Cleveland Park, Woodley Park, Massachusetts Avenue Heights, Woodland-Normanstone Terrace	0.328648
Georgetown, Burleith/Hillandale	0.195585
Lamont Riggs, Queens Chapel, Fort Totten, Pleasant Hill	0.154441
North Cleveland Park, Forest Hills, Van Ness	0.132966
Woodridge, Fort Lincoln, Gateway	0.121842
Brightwood Park, Crestwood, Petworth	0.070608
Ivy City, Arboretum, Trinidad, Carver Langston	0.050899
Howard University, Le Droit Park, Cardozo/Shaw	0.012893
Capitol Hill, Lincoln Park	0.011953
Colonial Village, Shepherd Park, North Portal Estates	0.003955
Rock Creek Park	0.000497
Sheridan, Barry Farm, Buena Vista	0.000109
Edgewood, Bloomingdale, Truxton Circle, Eckington	0
Fairfax Village, Naylor Gardens, Hillcrest, Summit Park	0
Walter Reed	0
Joint Base Anacostia-Bolling	0
Cathedral Heights, McLean Gardens, Glover Park	0
Southwest Employment Area, Southwest/Waterfront, Fort McNair, Buzzard Point	0
Observatory Circle	-1.42E-17
Saint Elizabeths	-0.000383
Twining, Fairlawn, Randle Highlands, Penn Branch, Fort Davis Park, Fort Dupont	-0.000735
Brookland, Brentwood, Langdon	-0.00125
Mayfair, Hillbrook, Mahaning Heights	-0.0014
Congress Heights, Bellevue, Washington Highlands	-0.001902
Kalorama Heights, Adams Morgan, Lanier Heights	-0.002842
Eastland Gardens, Kenilworth	-0.0046
Historic Anacostia	-0.006032
Dupont Circle, Connecticut Avenue/K Street	-0.013091
Douglas, Shipley Terrace	-0.017472
North Michigan Park, Michigan Park, University Heights	-0.039576
Spring Valley, Palisades, Wesley Heights, Foxhall Crescent, Foxhall Village, Georgetown Reservoir	-0.054335
Shaw, Logan Circle	-0.085476
Hawthorne, Barnaby Woods, Chevy Chase	-0.114506
River Terrace, Benning, Greenway, Dupont Park	-0.151747
Capitol View, Marshall Heights, Benning Heights	-0.151878
Deanwood, Burrville, Grant Park, Lincoln Heights, Fairmont Heights	-0.158593
Woodland/Fort Stanton, Garfield Heights, Knox Hill	-0.160748
Columbia Heights, Mt. Pleasant, Pleasant Plains, Park View	-0.219148
Downtown, Chinatown, Penn Quarters, Mount Vernon Square, North Capitol Street	-0.249717

Reporting and Visualization



http://arcg.is/2C4zGNp

Key Insights

- What was expected, didn't happen: Crime & Poverty Increased
- Machine learning allowed us to do apples and oranges comparisons that a typical statistical model could not
- Richer data sources and a greater number of instances impacts accuracy
  - Ex. Age Dependency Data noted in the First Principle Component, has wide variance, improved clustering

Lessons Learned

- Would choose Census Tract Shapefiles over Neighborhood Shapefiles
- Re-wrangle/ETL raw data to explore different result sets i.e. more features less rows
- 80% of our time committed to Data cleaning
- Choose a wider year range and add more features dynamically for broader applications:
  - Education
  - Unemployment
  - New Construction
  - Foreclosures
  - Parks and Infrastructure
  - Jobs added/lost
  - Transportation
  - Homeless Shelters


Questions?