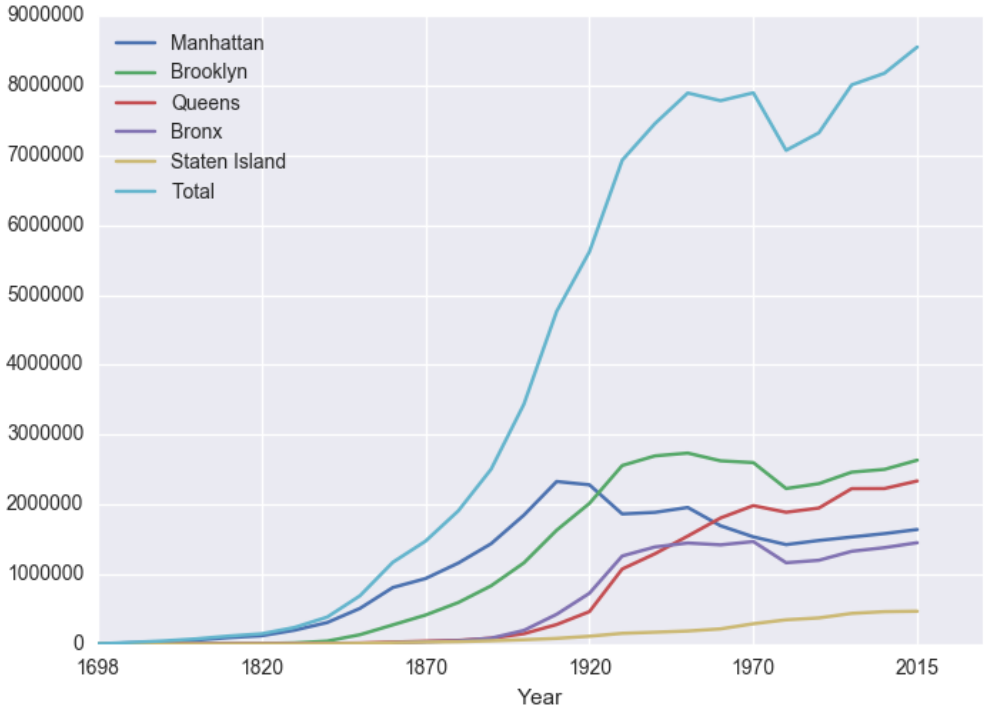
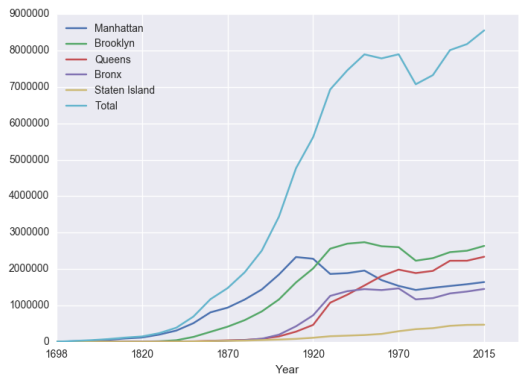
MfA: Python in the City



Katherine St. John
City University of New York
American Museum of Natural History

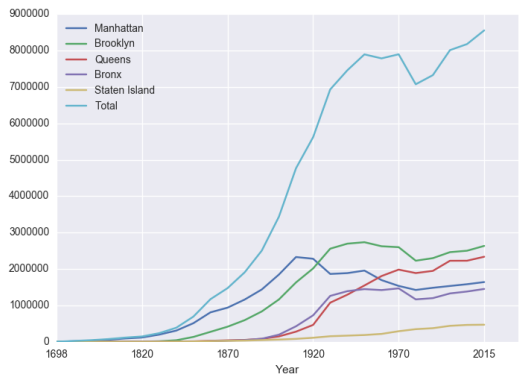
Goal: Every table have at most one from each school, one from each discipline.

Outline



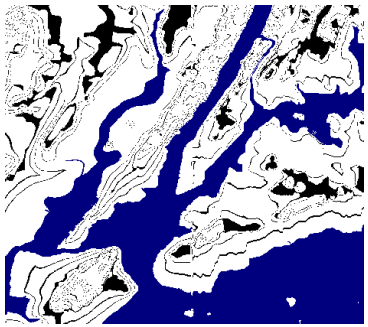
- Recap
- Design Challenge: NYC Population
- Variations on the Theme
- Design a Challenge
- Break
- Design Challenge: Parking Tickets
- Variations on the Theme
- Design a Challenge
- Wrap Up

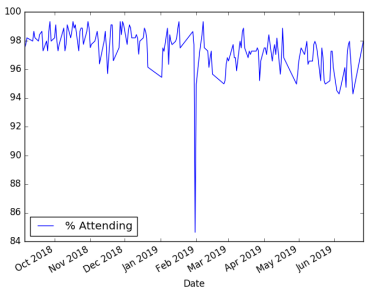
Outline

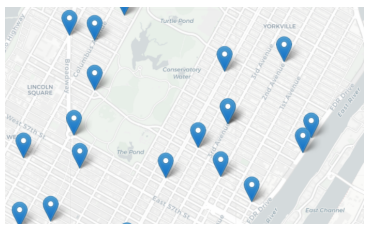


- Recap
- Design Challenge: NYC Population
- Variations on the Theme
- Design a Challenge
- Break
- Design Challenge: Parking Tickets
- Variations on the Theme
- Design a Challenge
- Wrap Up

Recap: Workshop Overview



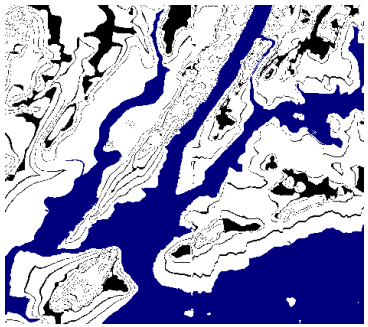


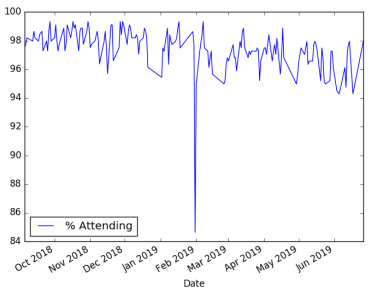


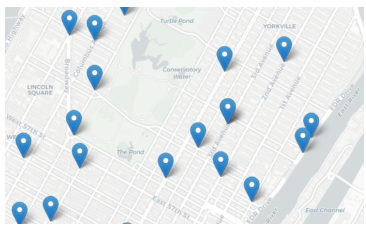
Three sessions:

1. Flood Maps (arrays & images)
2. Noisiest Street (structured data, file I/O)
3. Mapping Collisions (using objects, mapping coordinates)

Recap: Workshop Overview







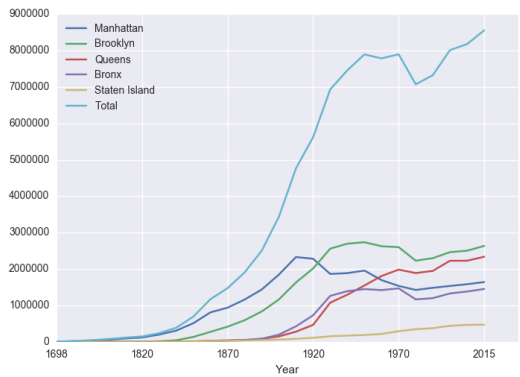
Three sessions:

1. Flood Maps (arrays & images)
2. Noisiest Street (structured data, file I/O)
3. Mapping Collisions (using objects, mapping coordinates)

Each session:

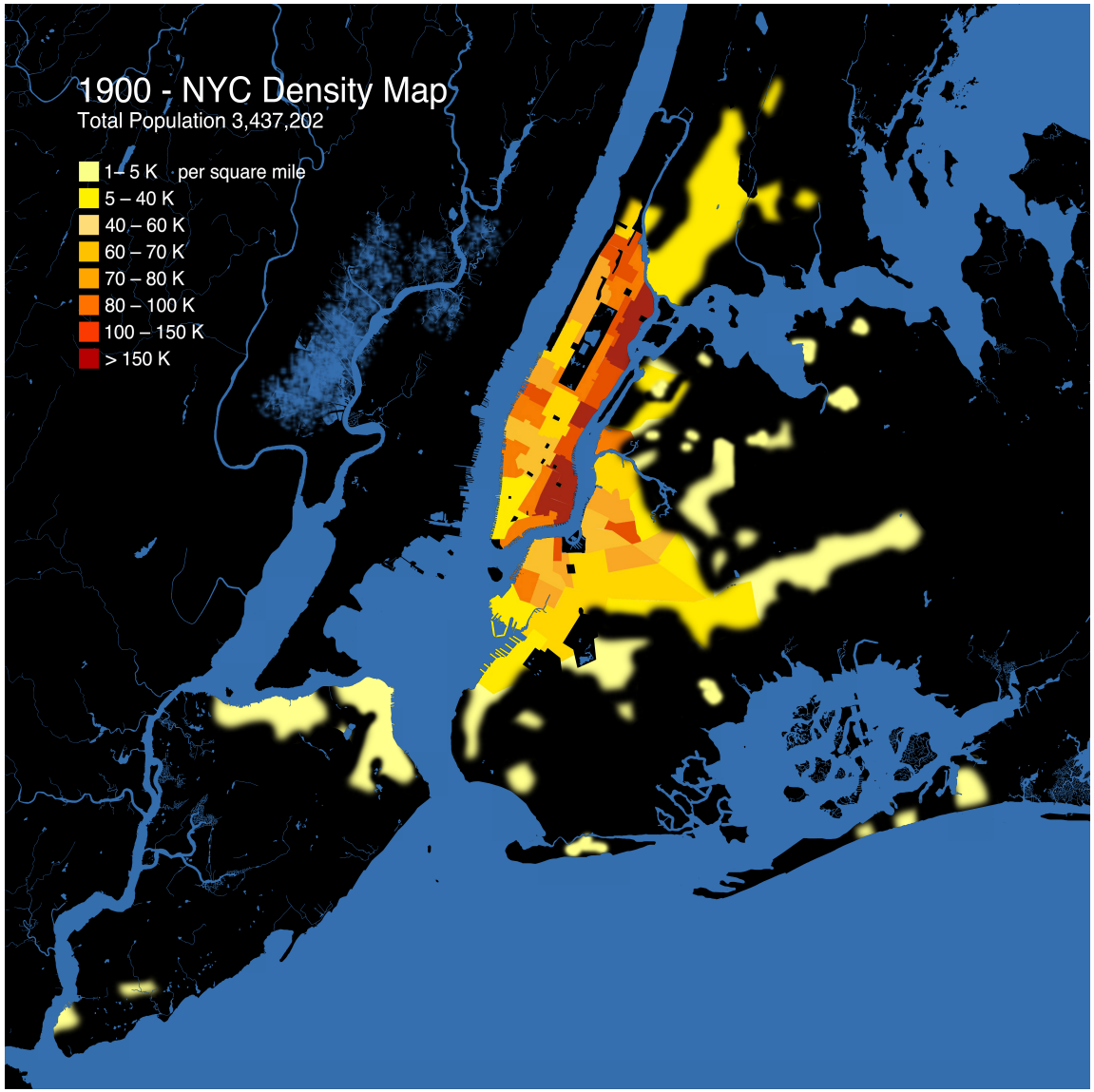
- Design Challenge
  - Analyze a publicly available dataset
  - Introduce computing concepts & packages
  - Write a program to solve the problem
- Variations on the theme
- Design a Challenge

Outline



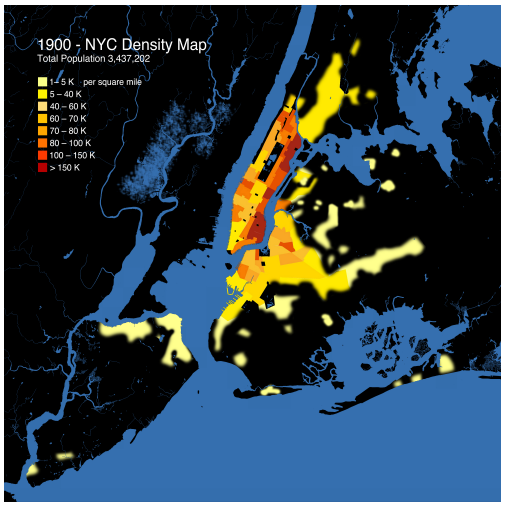
- Recap
- Design Challenge: NYC Population
- Variations on the Theme
- Design a Challenge
- Break
- Design Challenge: Parking Tickets
- Variations on the Theme
- Design a Challenge
- Wrap Up

Design Challenge: NYC Population



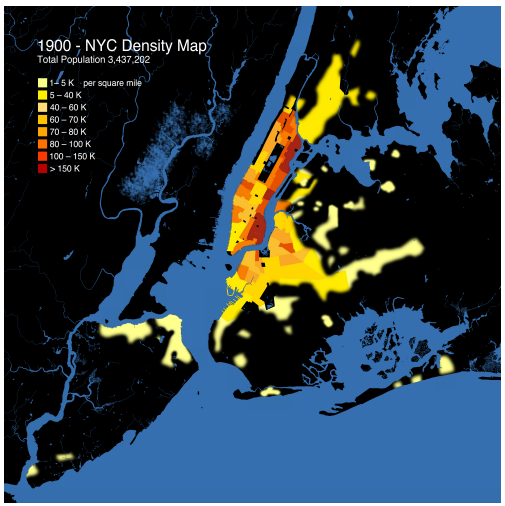
(Myles Zhang, wiki)

Design Challenge: NYC Population



(Myles Zhang, wiki)

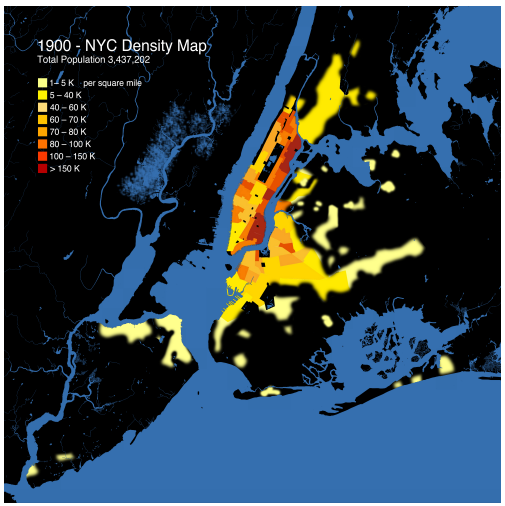
Design Challenge: NYC Population



(Myles Zhang, wiki)

Working in Groups: graph NYC population (and growth) since 1900.

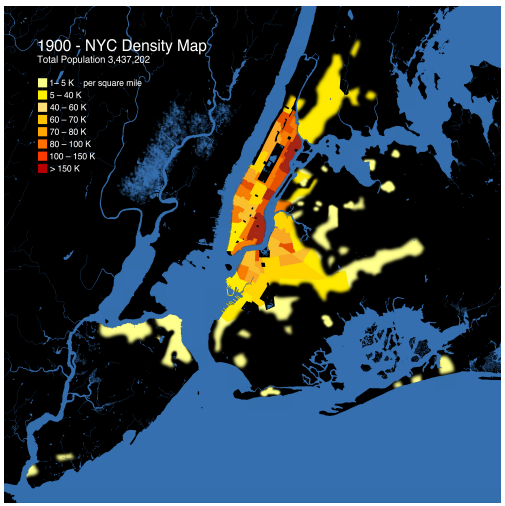
Design Challenge: NYC Population



(Myles Zhang, wiki)

Working in Groups: graph NYC population (and growth) since 1900.

Design Challenge: NYC Population

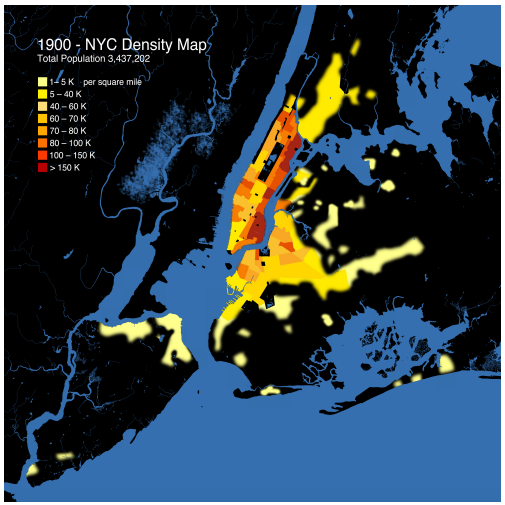


(Myles Zhang, wiki)

Working in Groups: graph NYC population (and growth) since 1900.

- Input: What data do you need?
- Process: How can you compute the growth?
- Output: How can you present your data?

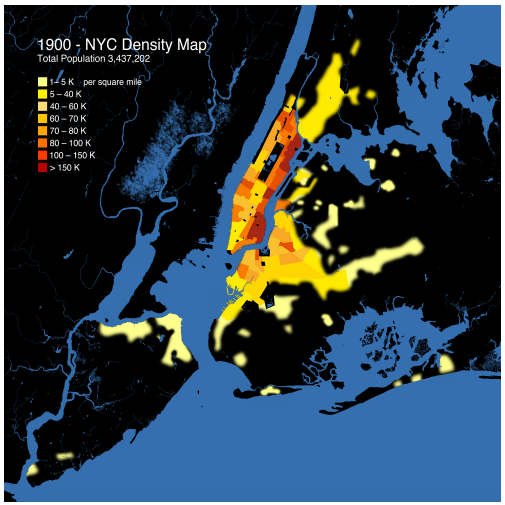
Data: Population



(Myles Zhang, wiki)

Input: What data do you need?

Data: Population

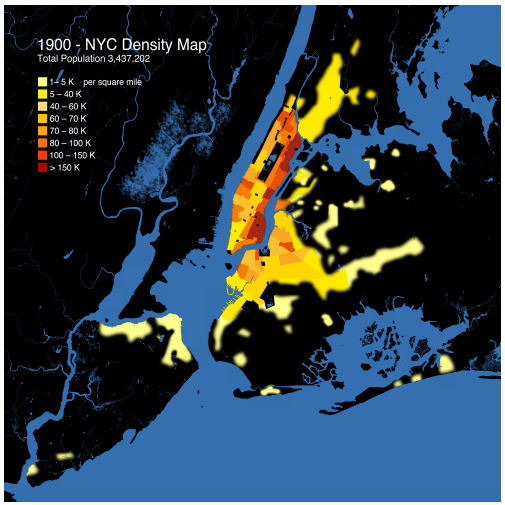


(Myles Zhang, wiki)

Input: What data do you need?

- Populations of the city in 1900 to today.

Data: Population

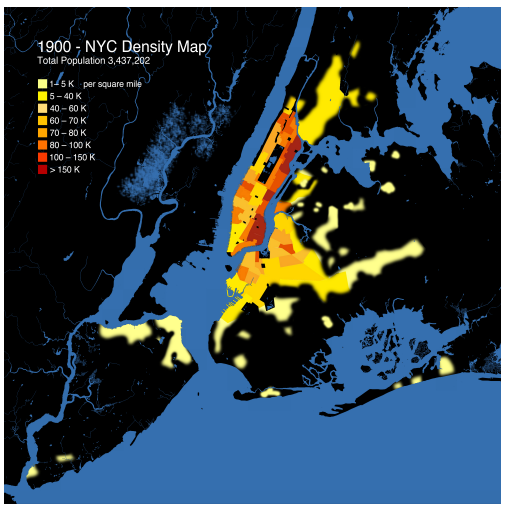


(Myles Zhang, wiki)

Input: What data do you need?

- Populations of the city in 1900 to today.
- Wikipedia page on NYC Historical Population.

Data: Population

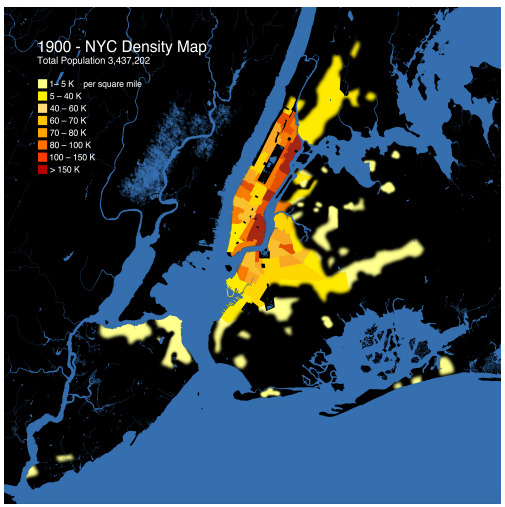


(Myles Zhang, wiki)

Input: What data do you need?

- Populations of the city in 1900 to today.
- Wikipedia page on NYC Historical Population.
- Saved as a CSV file at:
  `stjohn.github.io/service/mfa/f19.html`

Data: Population



(Myles Zhang, wiki)

Input: What data do you need?

- Populations of the city in 1900 to today.
- Wikipedia page on NYC Historical Population.
- Saved as a CSV file at:
  `stjohn.github.io/service/mfa/f19.html`
  (Download to your computer to use for the following slides.)

CSV File: stjohn.github.io/service/mfa/f19.html

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

nycHistPop.csv

CSV Files

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

- Excel .xls files have much extra formatting.

CSV Files

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

- Excel .xls files have much extra formatting.
- The text file version is called **CSV** for comma separated values.

CSV Files

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

- Excel .xls files have much extra formatting.
- The text file version is called **CSV** for comma separated values.
- Each row is a line in the file.

CSV Files

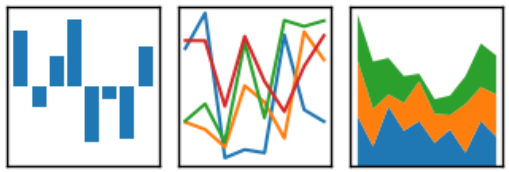
```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

- Excel .xls files have much extra formatting.
- The text file version is called **CSV** for comma separated values.
- Each row is a line in the file.
- Columns are separated by commas on each line.

Structured Data

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

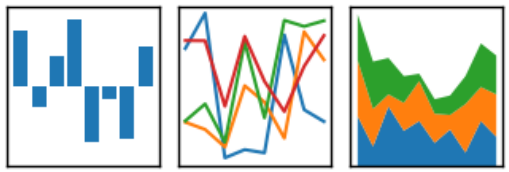


- We will use the popular Python Data Analysis Library (**Pandas**).

Structured Data

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

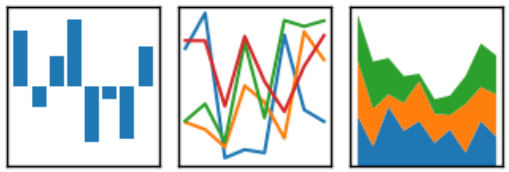


- We will use the popular Python Data Analysis Library (**Pandas**).
- Open source and freely available (part of anaconda distribution).

Structured Data

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



- We will use the popular Python Data Analysis Library (**Pandas**).
- Open source and freely available (part of anaconda distribution).
- To use, add to the top of your file:

```
import pandas as pd
```

Reading in CSV Files

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

- To read in a CSV file: `myVar = pd.read_csv("myFile.csv")`

Reading in CSV Files

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

- To read in a CSV file: `myVar = pd.read_csv("myFile.csv")`
- Pandas has its own type, **DataFrame**, that is perfect for holding a sheet of data.

Reading in CSV Files

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

- To read in a CSV file: `myVar = pd.read_csv("myFile.csv")`
- Pandas has its own type, **DataFrame**, that is perfect for holding a sheet of data.
- Often abbreviated: `df`.

Reading in CSV Files

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

- To read in a CSV file: `myVar = pd.read_csv("myFile.csv")`
- Pandas has its own type, **DataFrame**, that is perfect for holding a sheet of data.
- Often abbreviated: `df`.
- It also has **Series**, that is perfect for holding a row or column of data.

Your Turn: Reading in CSV Files

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

`nycHistPop.csv`

Your Turn: Reading in CSV Files

```
import matplotlib.pyplot as plt
import pandas as pd
```

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

`nycHistPop.csv`

Your Turn: Reading in CSV Files

```
import matplotlib.pyplot as plt
import pandas as pd

pop = pd.read_csv('nycHistPop.csv',skiprows=5)
```

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

nycHistPop.csv

Your Turn: Reading in CSV Files

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

nycHistPop.csv

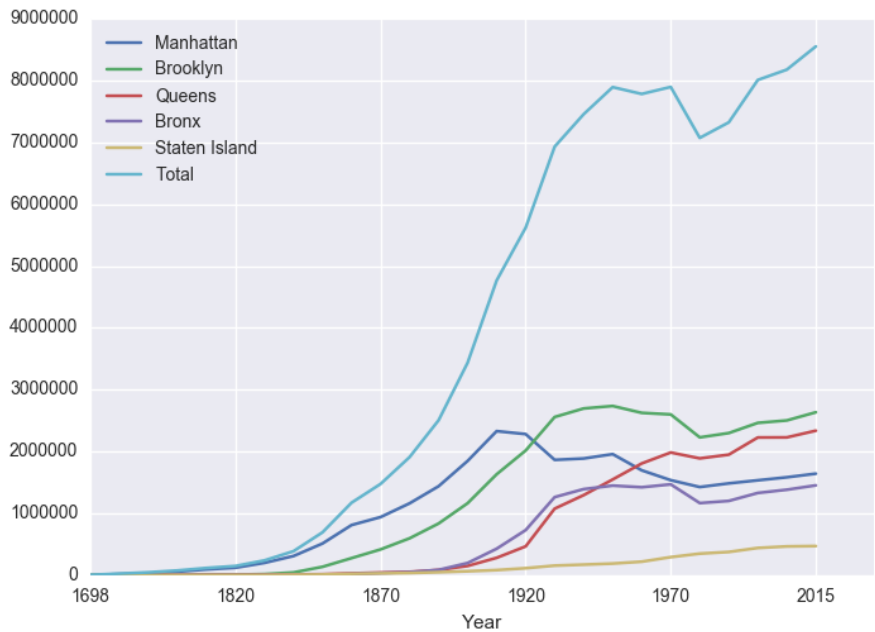
```
import matplotlib.pyplot as plt
import pandas as pd

pop = pd.read_csv('nycHistPop.csv',skiprows=5)

pop.plot(x="Year")
plt.show()
```

Your Turn: Reading in CSV Files

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

nycHistPop.csv

```
import matplotlib.pyplot as plt
import pandas as pd

pop = pd.read_csv('nycHistPop.csv',skiprows=5)

pop.plot(x="Year")
plt.show()
```

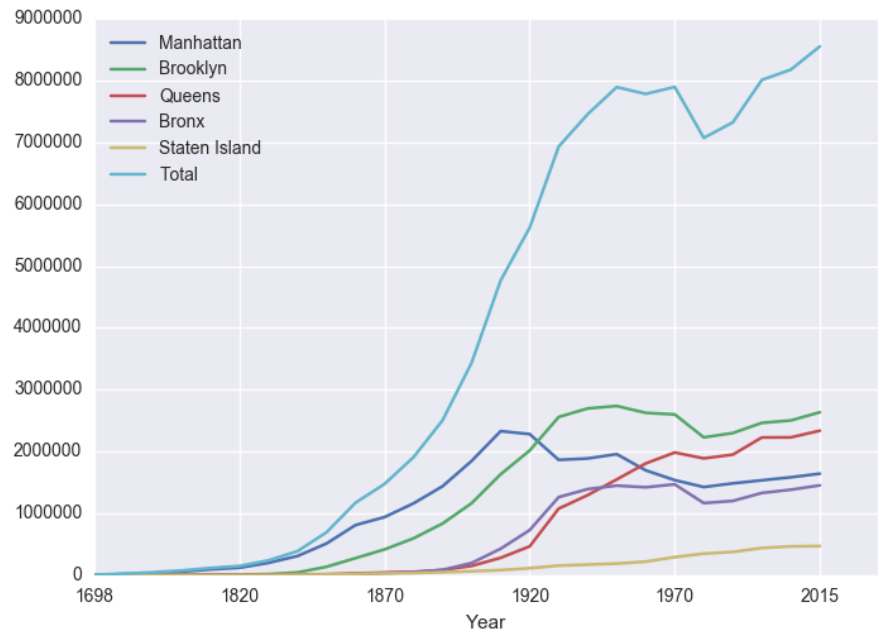


Your Turn: Reading in CSV Files

```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
1771,21863,3623,,,2847,28423
1790,33131,4549,6159,1781,3827,49447
1800,60515,5740,6642,1755,4563,79215
1810,96373,8303,7444,2267,5347,119734
1820,123706,11187,8246,2782,6135,152056
1830,202589,20535,9049,3023,7082,242278
1840,312710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,696115
1860,813669,279122,32903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1850093,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,7781984
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230936,1891325,1168972,352121,7071639
1990,1487536,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585873,2504700,2230722,1385108,468730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550405
```

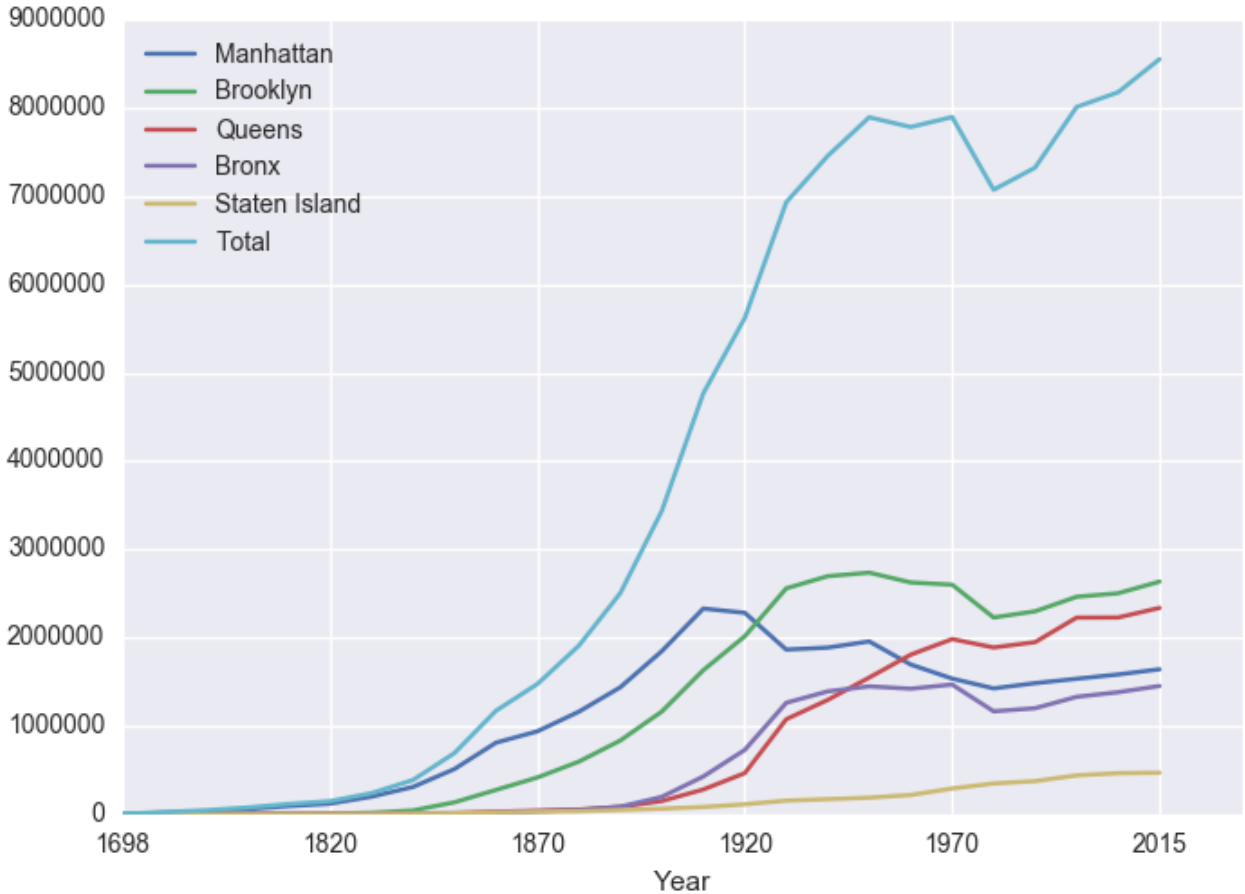
nycHistPop.csv

```
import matplotlib.pyplot as plt
import pandas as pd

pop = pd.read_csv('nycHistPop.csv',skiprows=5)

pop.plot(x="Year")
plt.show()
```

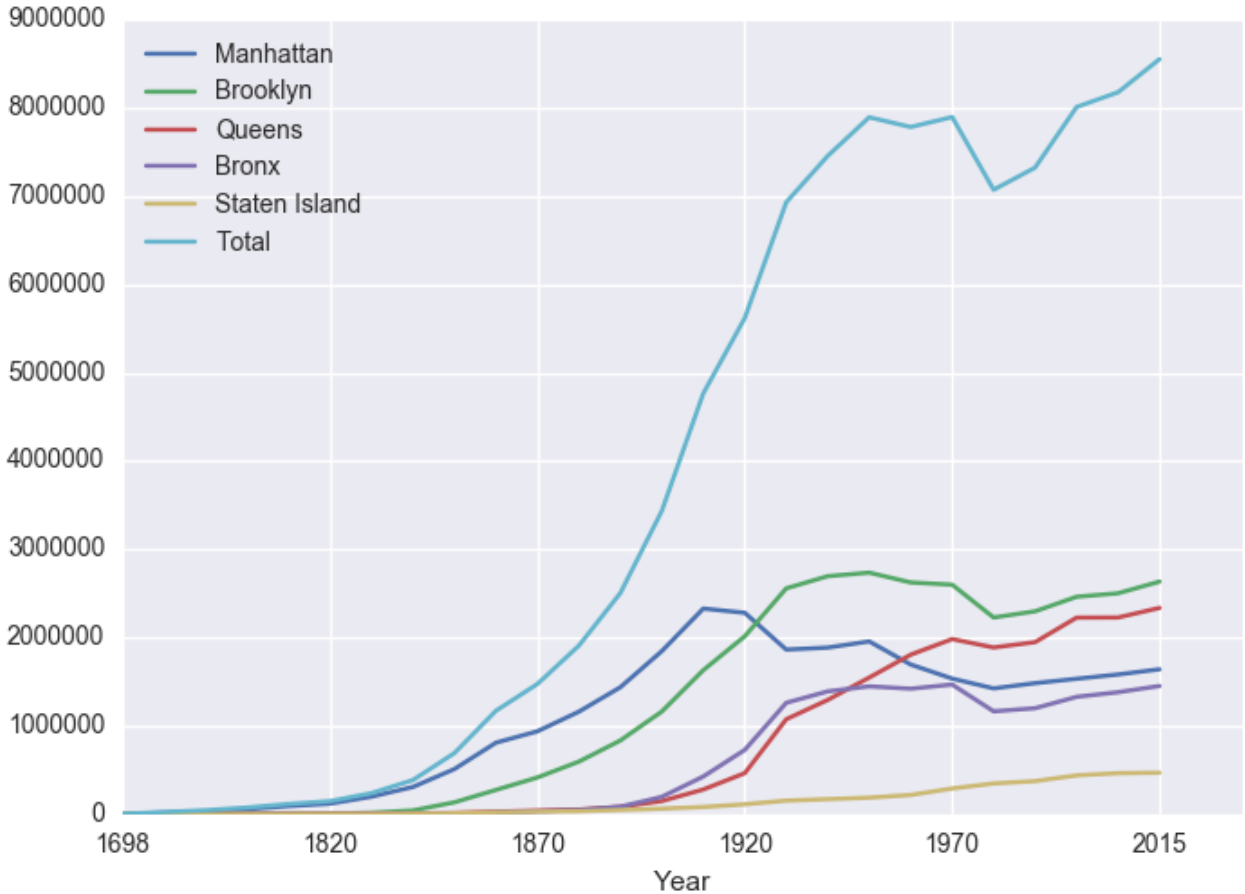


(To get above color scheme, `import seaborn`.)

Nice Graph, But Doesn't Answer the Question

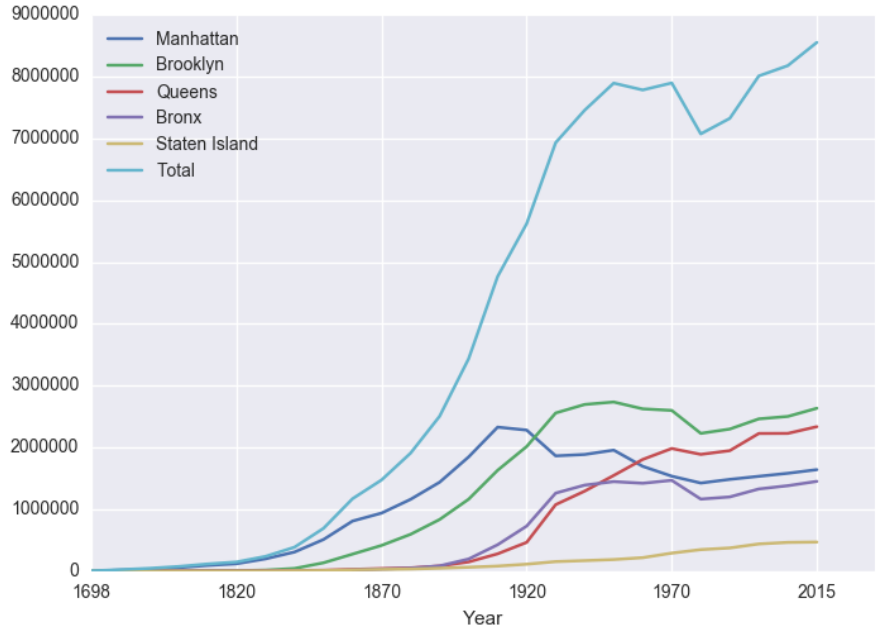


Nice Graph, But Doesn't Answer the Question



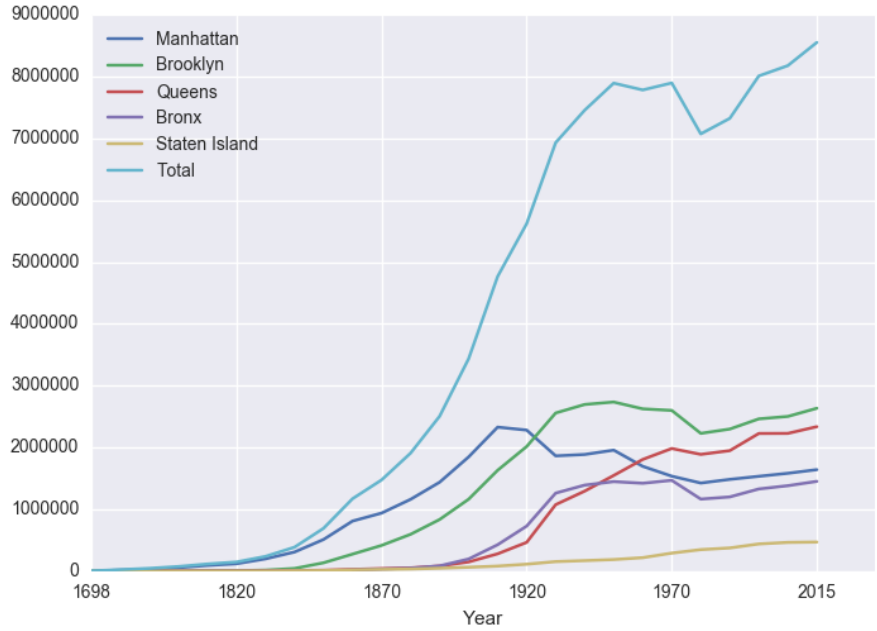
Let's survey what else you can do with pandas....

Series in Pandas



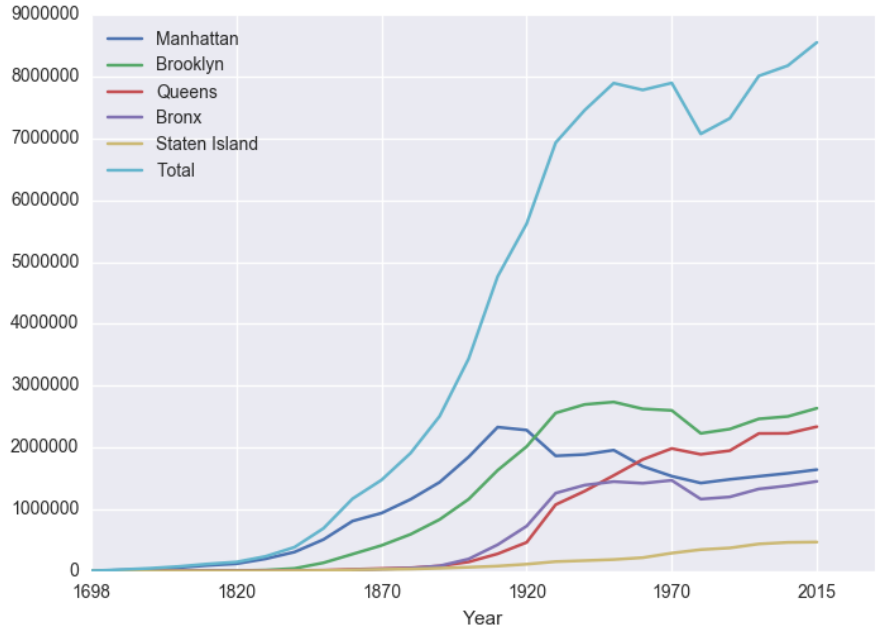
- Series can store a column or row of a DataFrame.

Series in Pandas



- Series can store a column or row of a DataFrame.
- Example: `pop["Manhattan"]` is the Series corresponding to the column of Manhattan data.

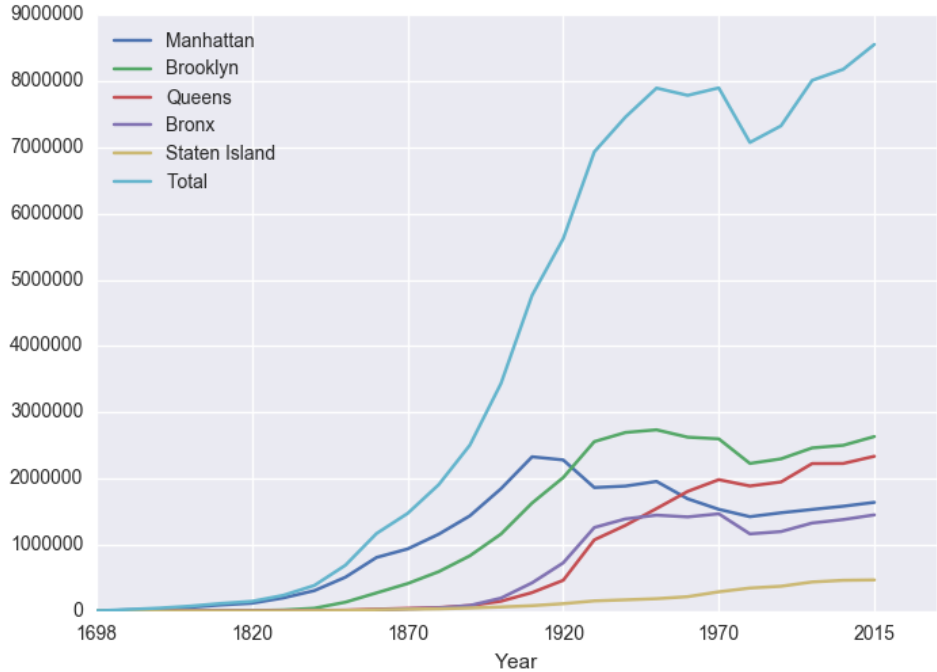
Series in Pandas



- Series can store a column or row of a DataFrame.
- Example: `pop["Manhattan"]` is the Series corresponding to the column of Manhattan data.
- Example:

```
print("The largest number living in the Bronx is",\
pop["Bronx"].max())
```

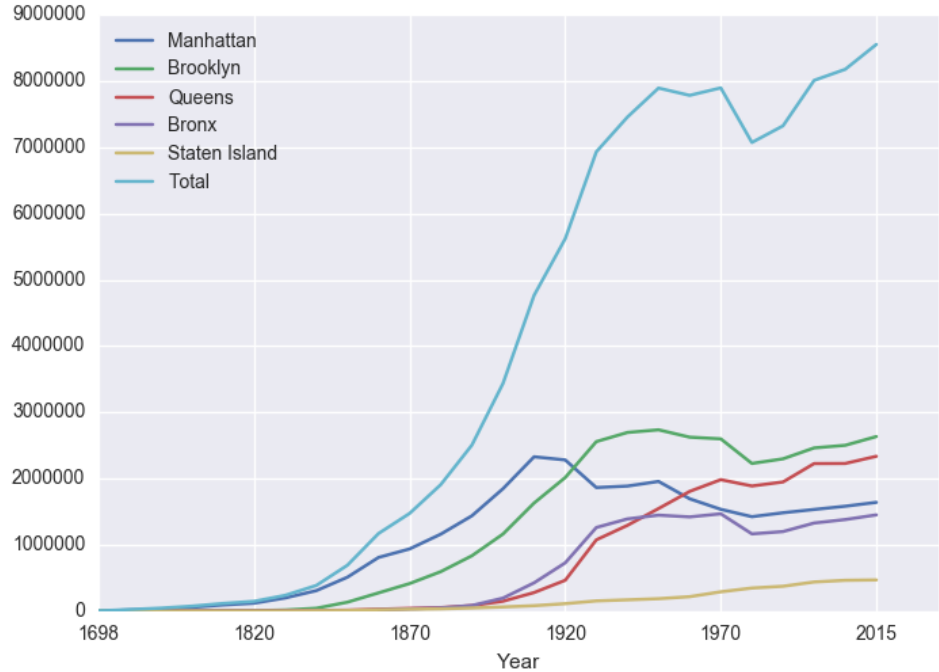
In Groups



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`

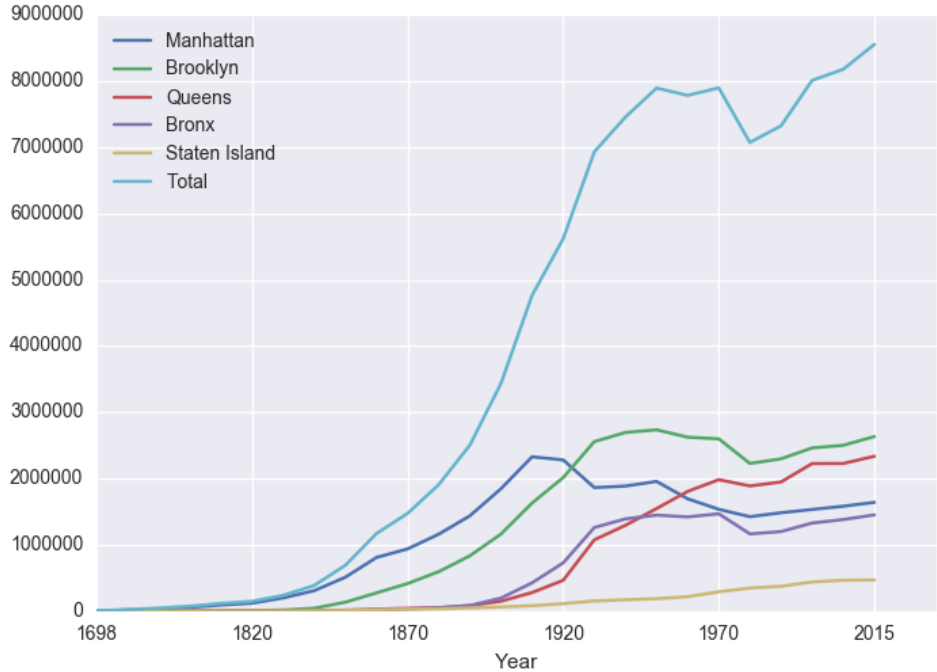
In Groups



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
- `print("S I:", pop["Staten Island"].mean())`

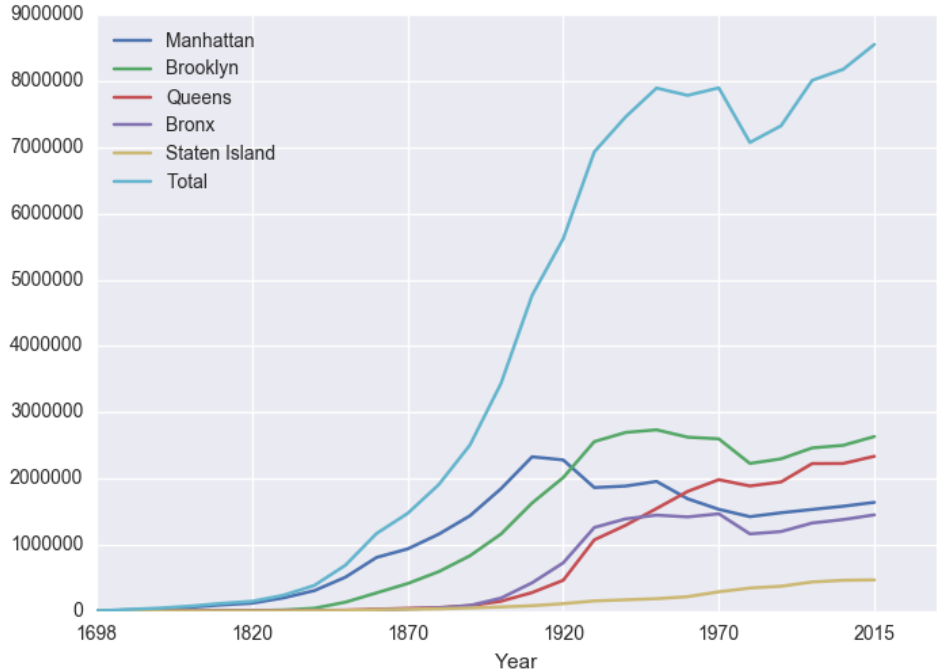
In Groups



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
- `print("S I:", pop["Staten Island"].mean())`
- `print("S I:", pop["Staten Island"].std())`

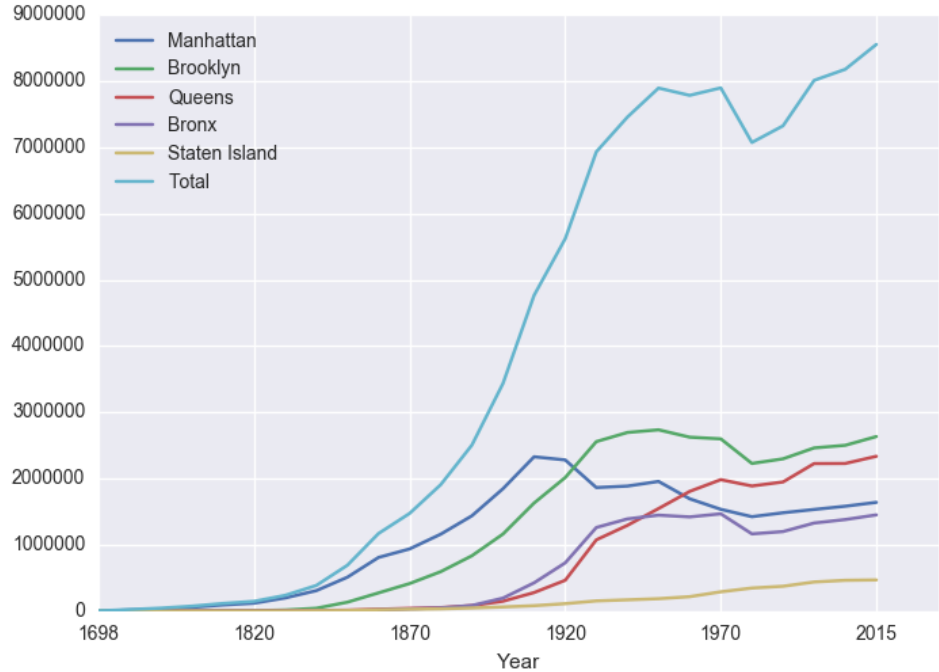
In Groups



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
- `print("S I:", pop["Staten Island"].mean())`
- `print("S I:", pop["Staten Island"].std())`
- `pop.plot.bar(x="Year")`

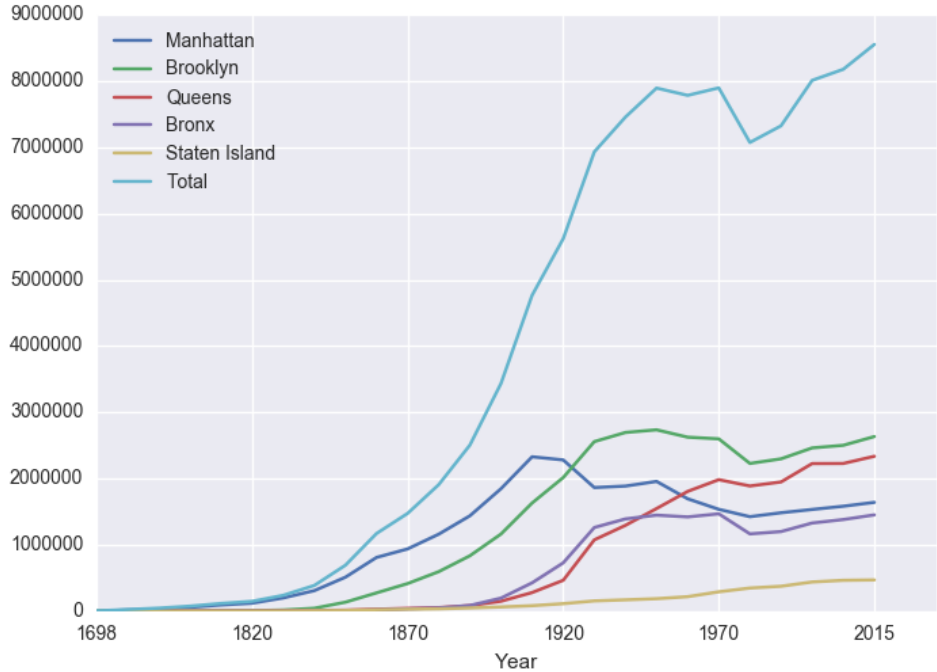
In Groups



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
- `print("S I:", pop["Staten Island"].mean())`
- `print("S I:", pop["Staten Island"].std())`
- `pop.plot.bar(x="Year")`
- `pop.plot.scatter(x="Brooklyn", y= "Total")`

In Groups



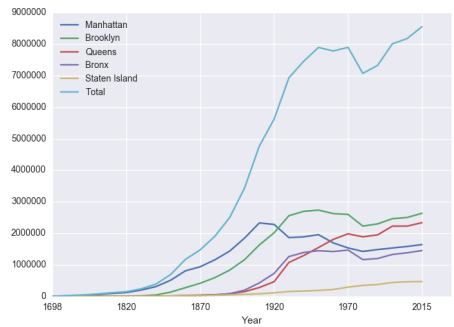
Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
- `print("S I:", pop["Staten Island"].mean())`
- `print("S I:", pop["Staten Island"].std())`
- `pop.plot.bar(x="Year")`
- `pop.plot.scatter(x="Brooklyn", y= "Total")`
- `pop["Fraction"] = pop["Bronx"]/pop["Total"]`

Solutions

Predict what the following will do:

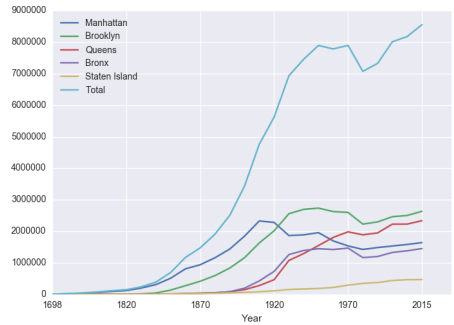
- `print("Queens:", pop["Queens"].min())`



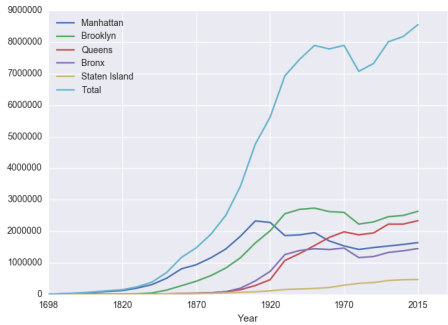
Solutions

Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
  Minimum value in the column with label "Queens".



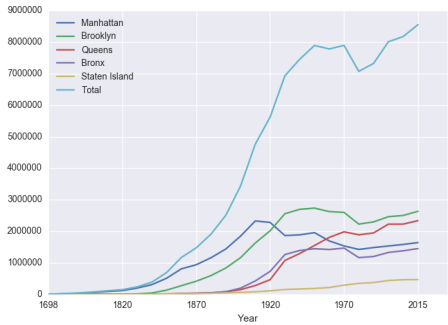
Solutions



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
  Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`

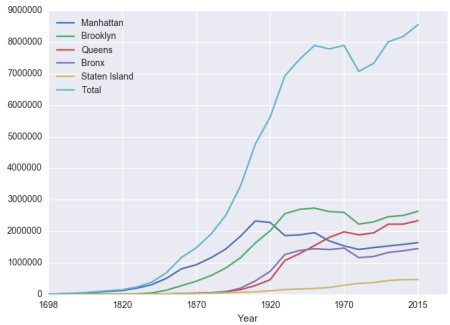
Solutions



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
  Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`
  Average of values in the column "Staten Island".

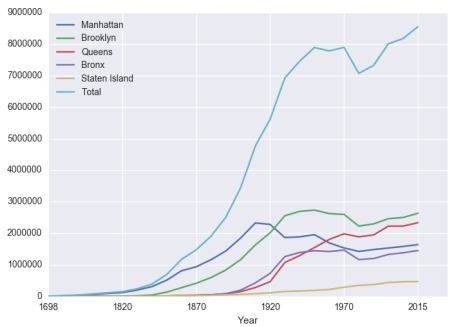
Solutions



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
  Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`
  Average of values in the column "Staten Island".
- `print("S I :", pop["Staten Island"].std())`

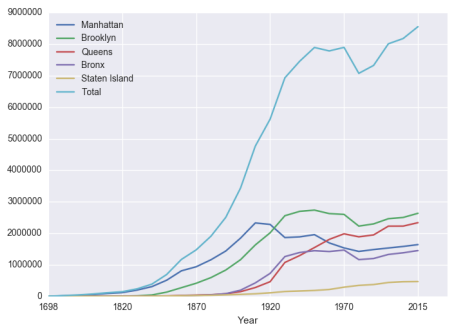
Solutions



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
  Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`
  Average of values in the column "Staten Island".
- `print("S I :", pop["Staten Island"].std())`
  Standard deviation of values in the column "Staten Island".

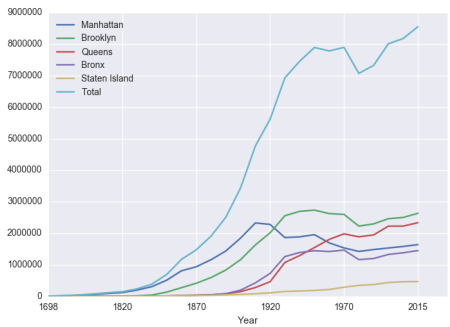
Solutions



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
  Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`
  Average of values in the column "Staten Island".
- `print("S I :", pop["Staten Island"].std())`
  Standard deviation of values in the column "Staten Island".
- `pop.plot.bar(x="Year")`

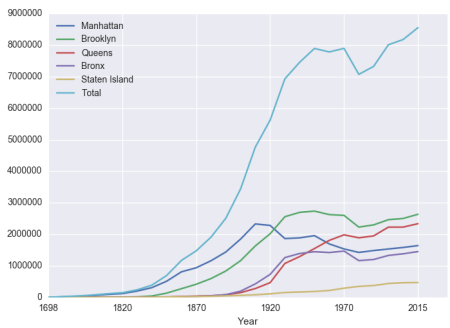
Solutions



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
  Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`
  Average of values in the column "Staten Island".
- `print("S I :", pop["Staten Island"].std())`
  Standard deviation of values in the column "Staten Island".
- `pop.plot.bar(x="Year")`
  Bar chart with x-axis "Year".

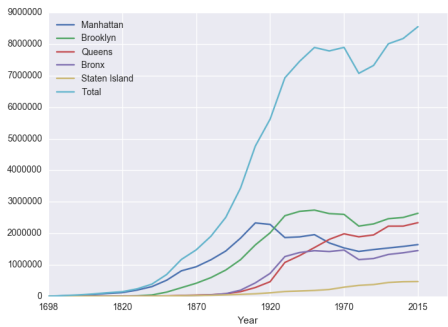
Solutions



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
  Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`
  Average of values in the column "Staten Island".
- `print("S I :", pop["Staten Island"].std())`
  Standard deviation of values in the column "Staten Island".
- `pop.plot.bar(x="Year")`
  Bar chart with x-axis "Year".
- `pop.plot.scatter(x="Brooklyn", y= "Total")`

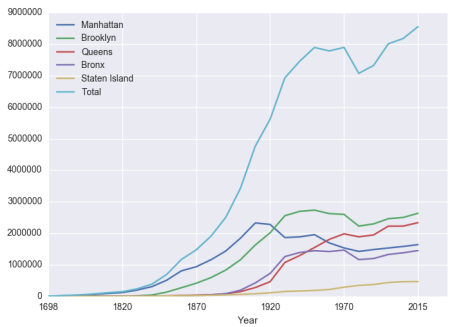
Solutions



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
  Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`
  Average of values in the column "Staten Island".
- `print("S I :", pop["Staten Island"].std())`
  Standard deviation of values in the column "Staten Island".
- `pop.plot.bar(x="Year")`
  Bar chart with x-axis "Year".
- `pop.plot.scatter(x="Brooklyn", y= "Total")`
  Scatter plot of Brooklyn versus Total values.

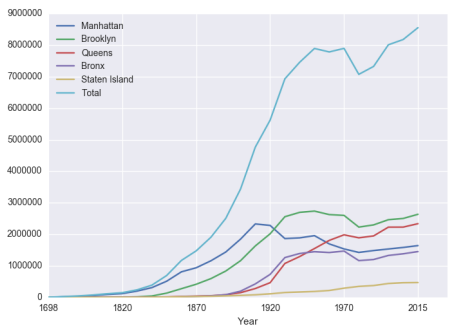
Solutions



Predict what the following will do:

- `print("Queens:", pop["Queens"].min())`
  Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`
  Average of values in the column "Staten Island".
- `print("S I :", pop["Staten Island"].std())`
  Standard deviation of values in the column "Staten Island".
- `pop.plot.bar(x="Year")`
  Bar chart with x-axis "Year".
- `pop.plot.scatter(x="Brooklyn", y= "Total")`
  Scatter plot of Brooklyn versus Total values.
- `pop["Fraction"] = pop["Bronx"]/pop["Total"]`

Solutions



Predict what the following will do:

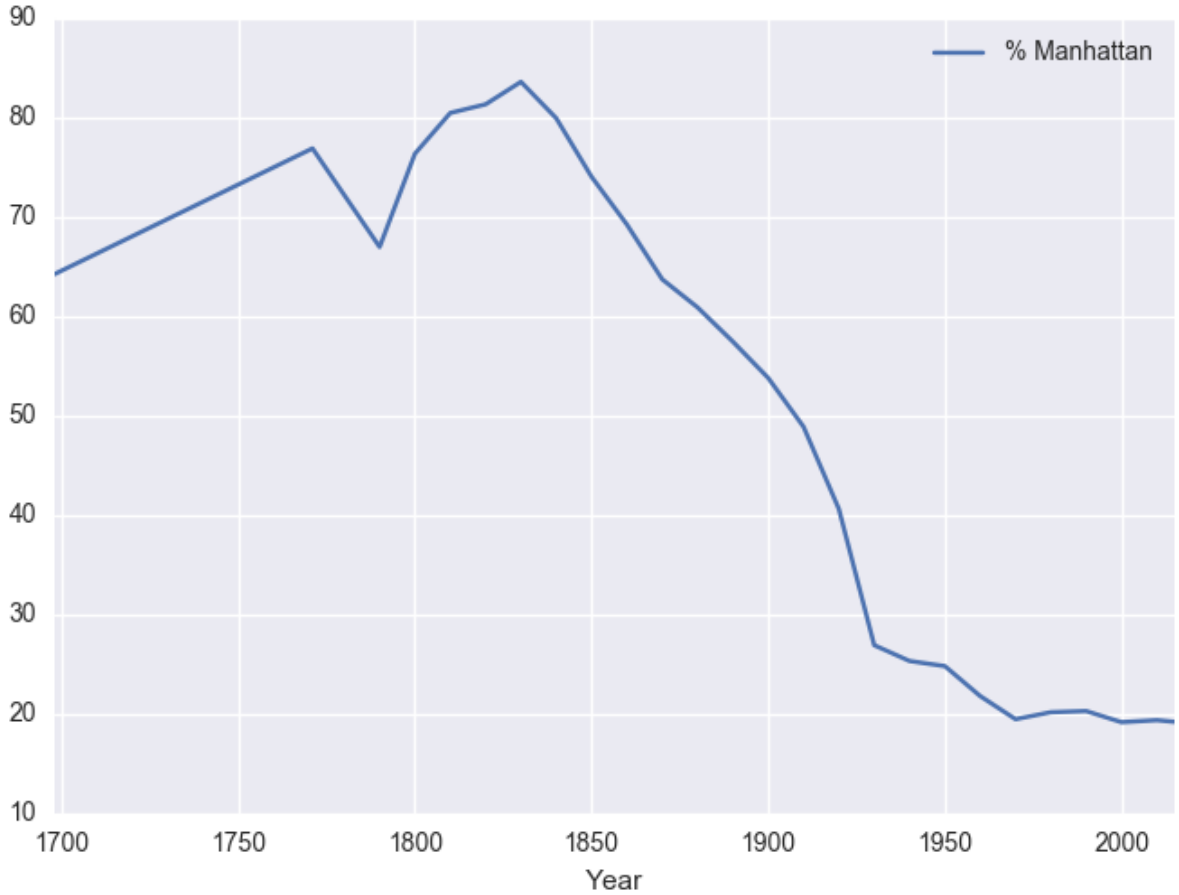
- `print("Queens:", pop["Queens"].min())`
  Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`
  Average of values in the column "Staten Island".
- `print("S I :", pop["Staten Island"].std())`
  Standard deviation of values in the column "Staten Island".
- `pop.plot.bar(x="Year")`
  Bar chart with x-axis "Year".
- `pop.plot.scatter(x="Brooklyn", y= "Total")`
  Scatter plot of Brooklyn versus Total values.
- `pop["Fraction"] = pop["Bronx"]/pop["Total"]`
  New column with the fraction of population that lives in the Bronx.

In Groups

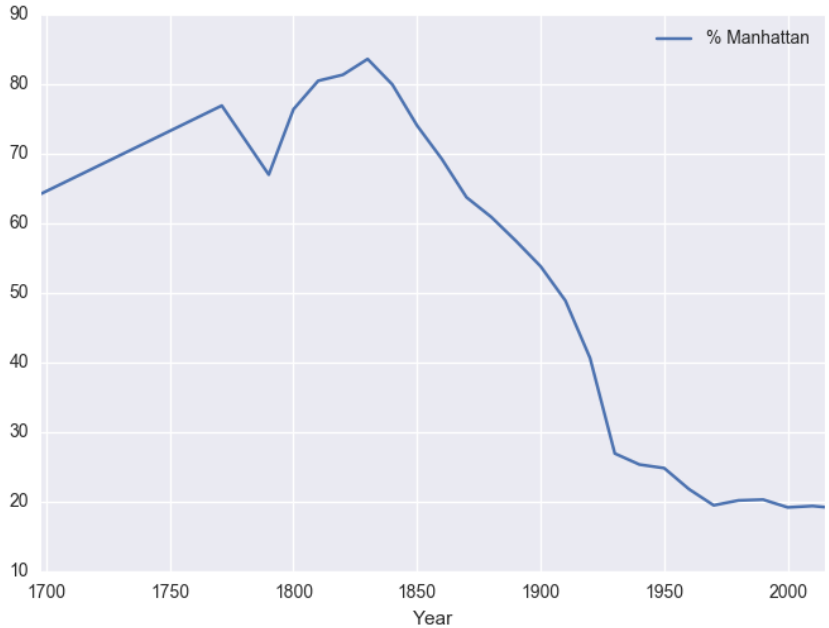
Graph the percentage of the total population that live in Manhattan:

In Groups

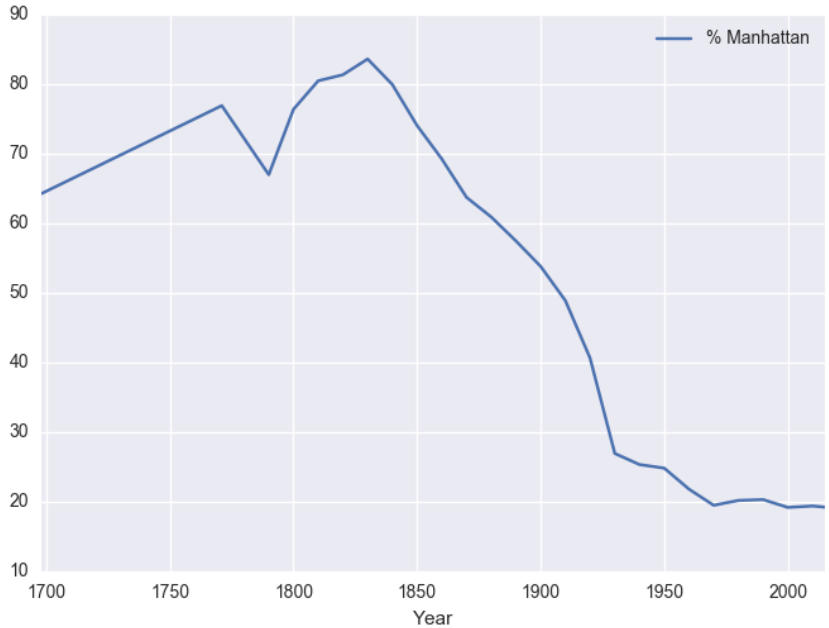
Graph the percentage of the total population that live in Manhattan:



Percent of Population in Manhattan, Over Time

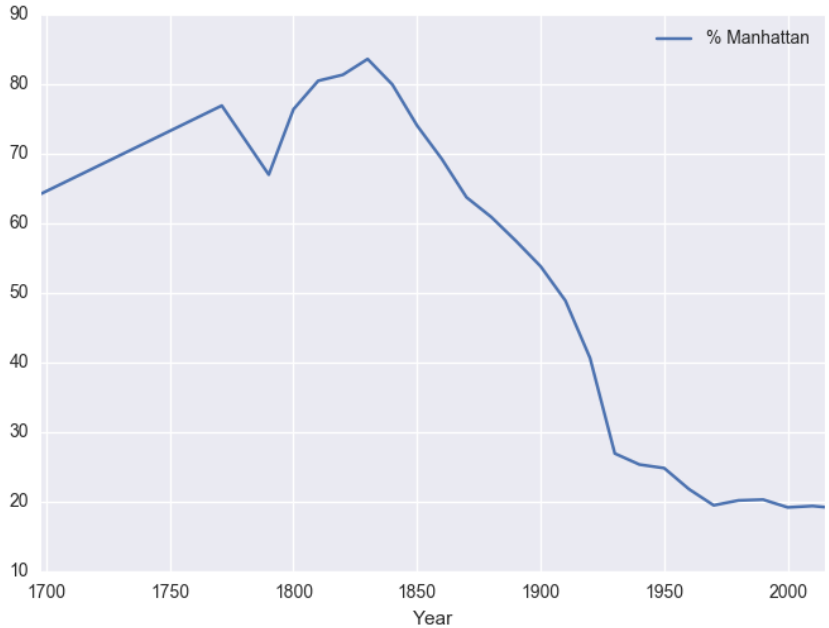


Percent of Population in Manhattan, Over Time



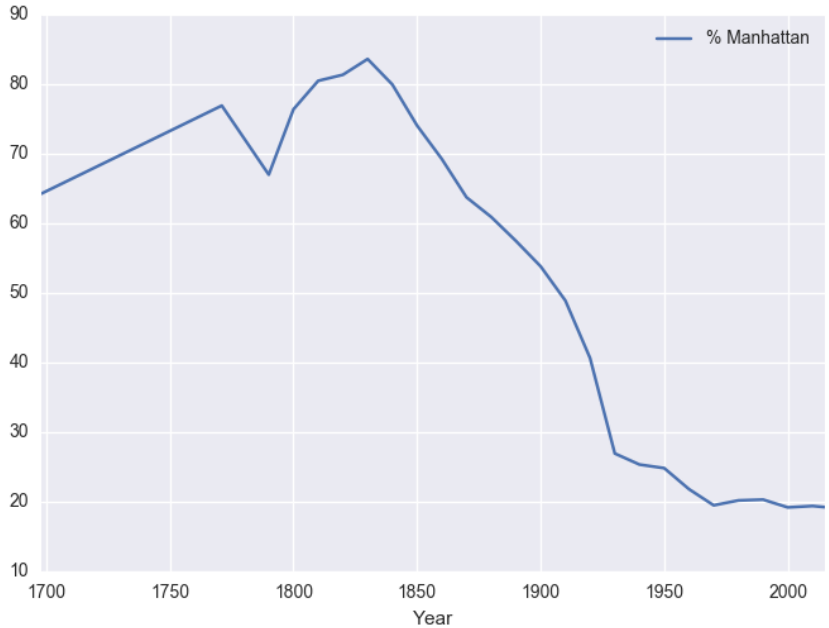

```
import matplotlib.pyplot as plt
import pandas as pd
```

Percent of Population in Manhattan, Over Time



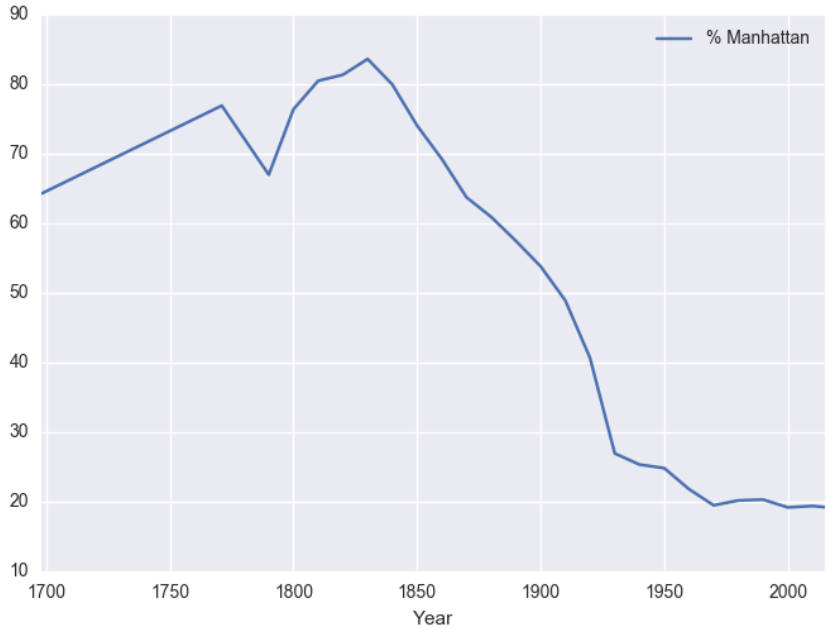

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn
```

Percent of Population in Manhattan, Over Time



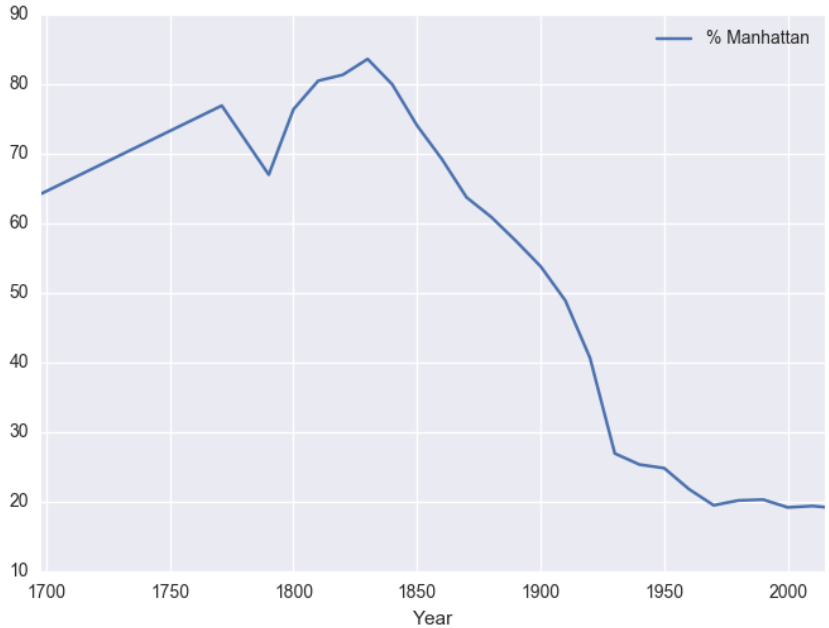

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn
pop = pd.read_csv('nycHistPop.csv',skiprows=5)
```

Percent of Population in Manhattan, Over Time



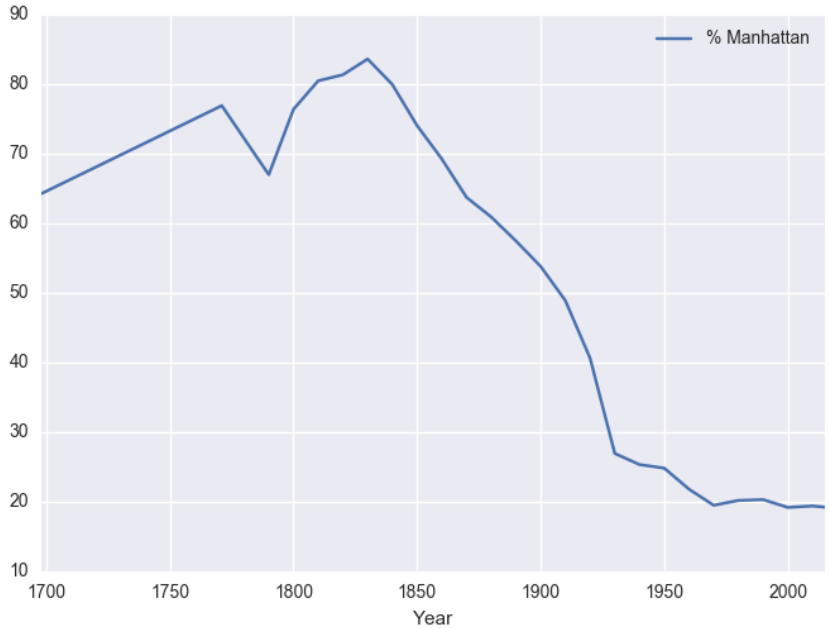

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn
pop = pd.read_csv('nycHistPop.csv',skiprows=5)
pop["% Manhattan"] = 100*pop["Manhattan"]/pop["Total"]
```

Percent of Population in Manhattan, Over Time



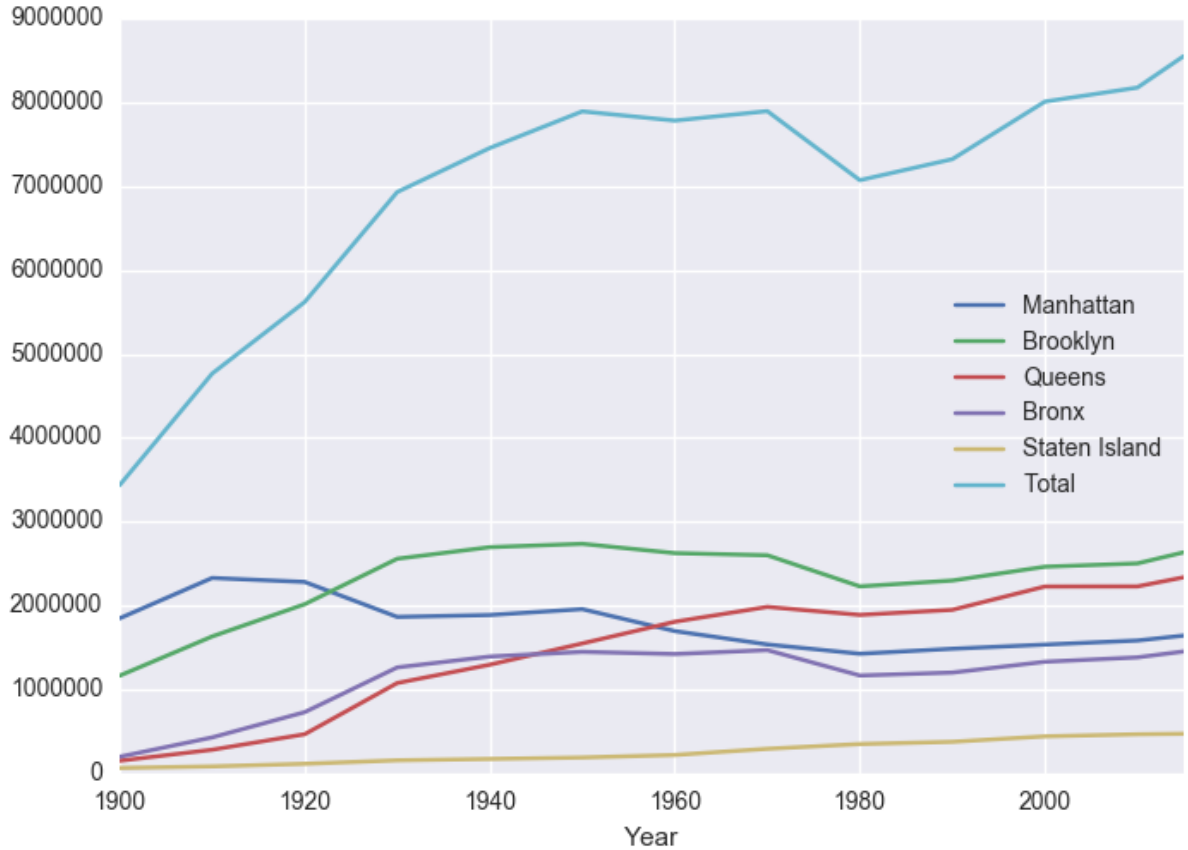

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn
pop = pd.read_csv('nycHistPop.csv',skiprows=5)
pop["% Manhattan"] = 100*pop["Manhattan"]/pop["Total"]
pop.plot(x="Year",y="% Manhattan")
```

Percent of Population in Manhattan, Over Time

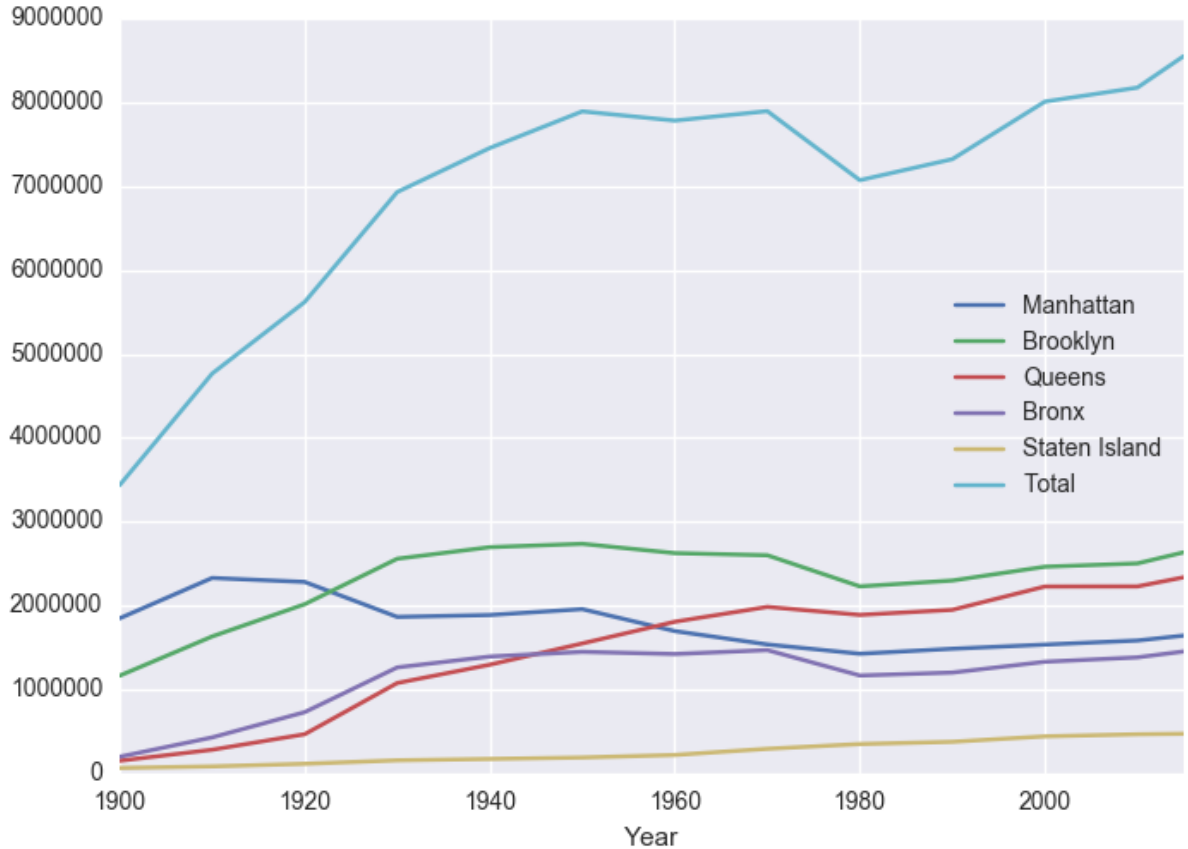



```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn
pop = pd.read_csv('nycHistPop.csv',skiprows=5)
pop["% Manhattan"] = 100*pop["Manhattan"]/pop["Total"]
pop.plot(x="Year",y="% Manhattan")
plt.show()
```

Population Since 1900

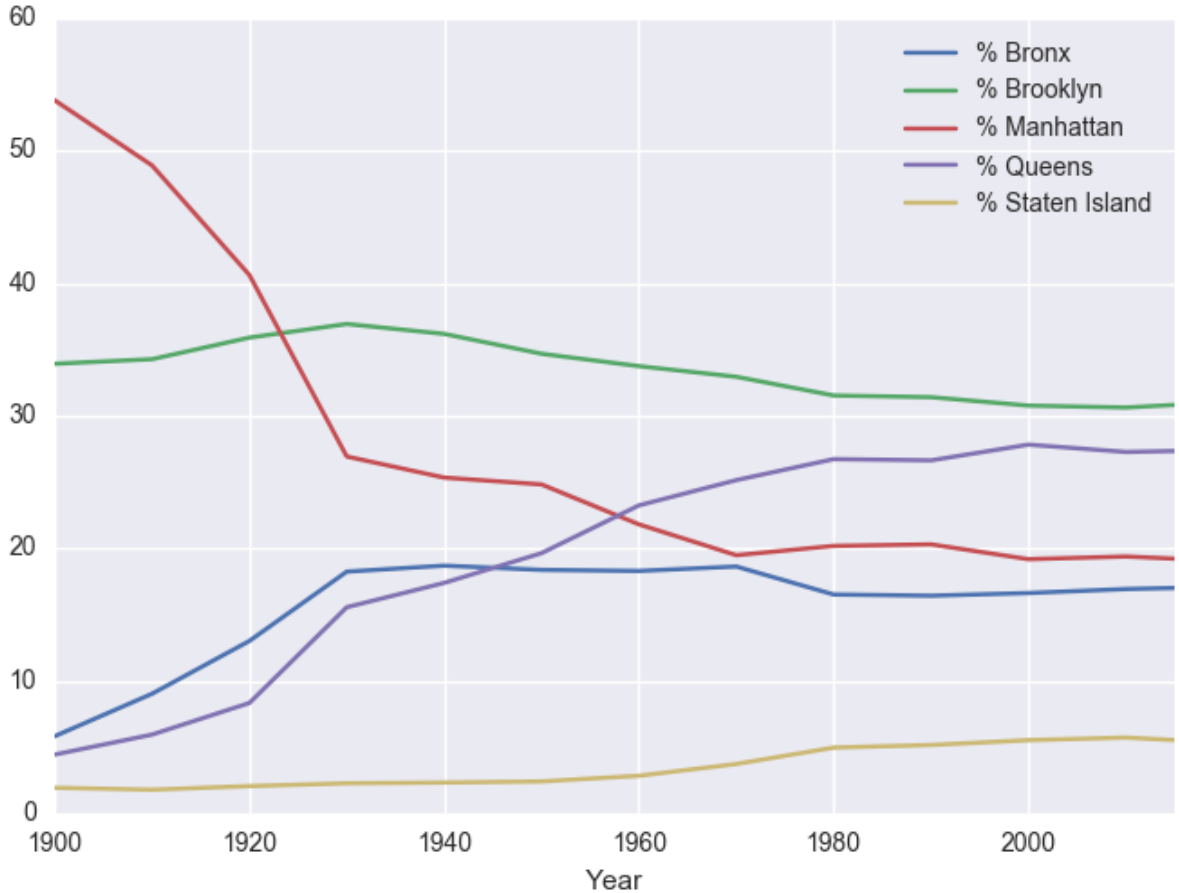


Population Since 1900

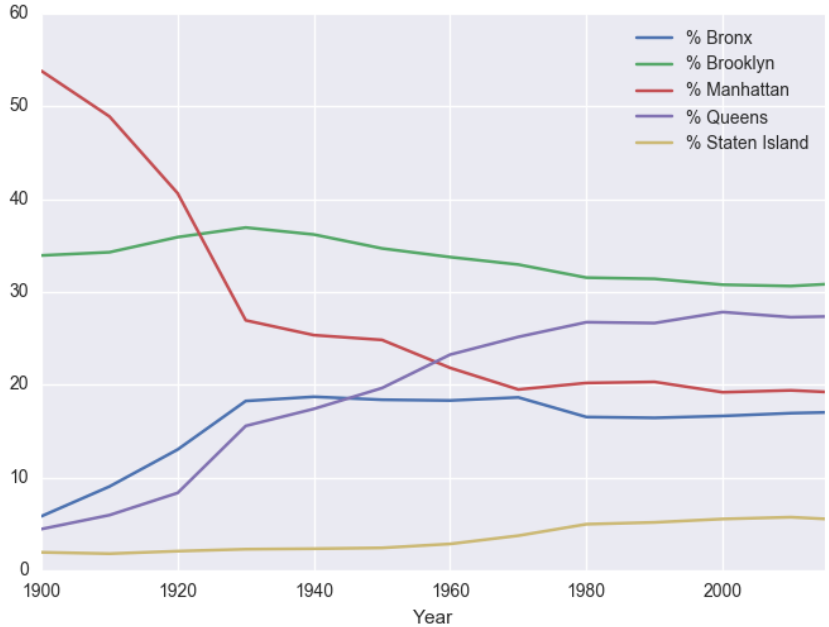



```
pop1900 = pop[pop['Year'] >= 1900]
pop1900.plot(x="Year")
plt.show()
```

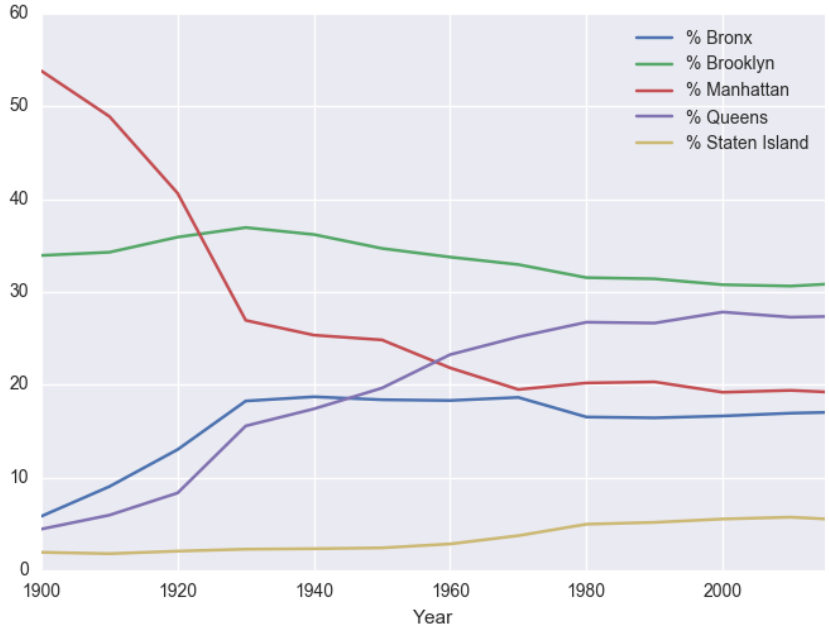
Population Percentage by Borough



Population Percentage by Borough

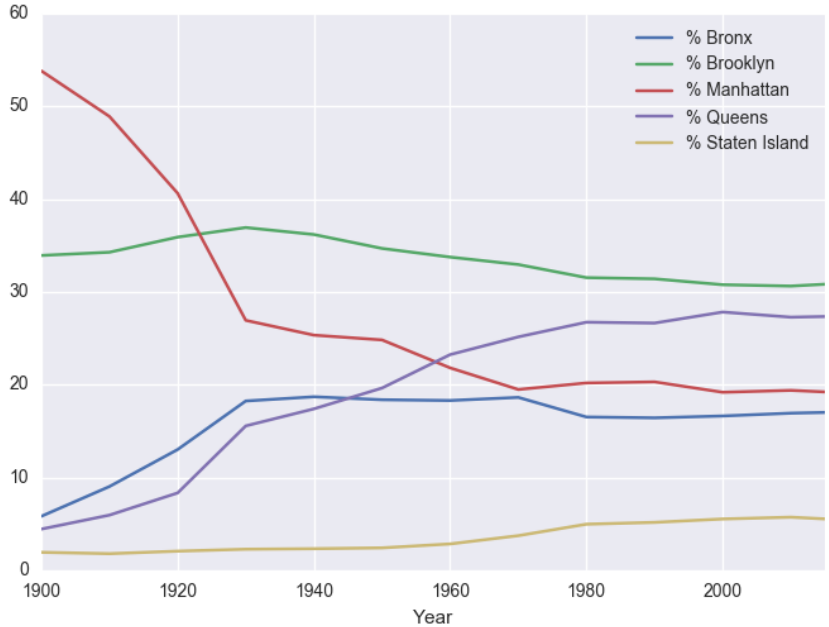


Population Percentage by Borough



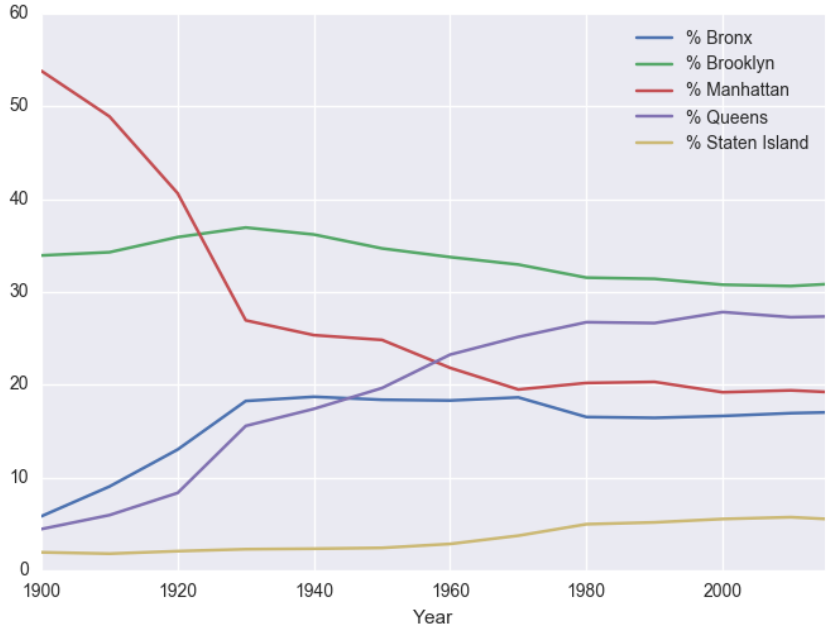

```
boros = ["Bronx","Brooklyn","Manhattan","Queens","Staten Island"]
```

Population Percentage by Borough



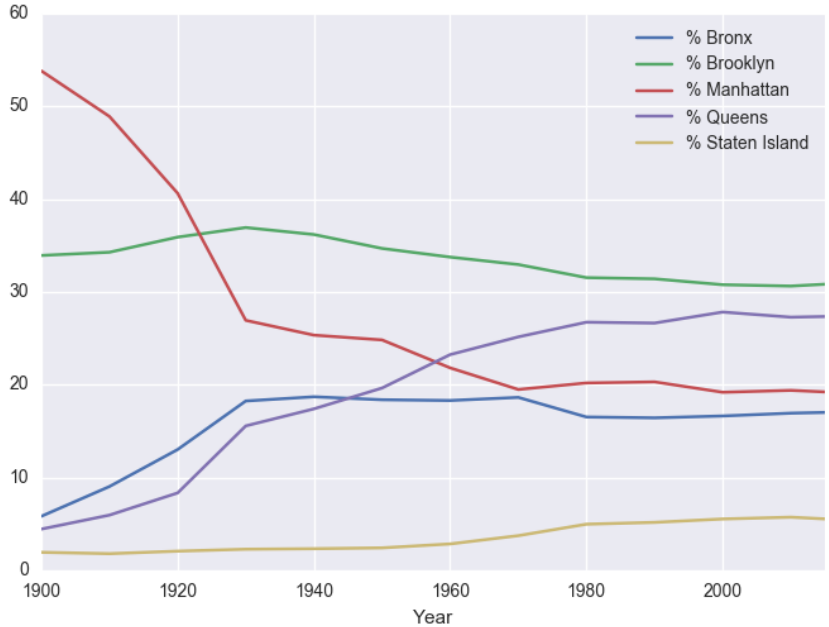

```
boros = ["Bronx","Brooklyn","Manhattan","Queens","Staten Island"]
percentCol = ["% "+ boro for boro in boros] #List comprehension
```

Population Percentage by Borough



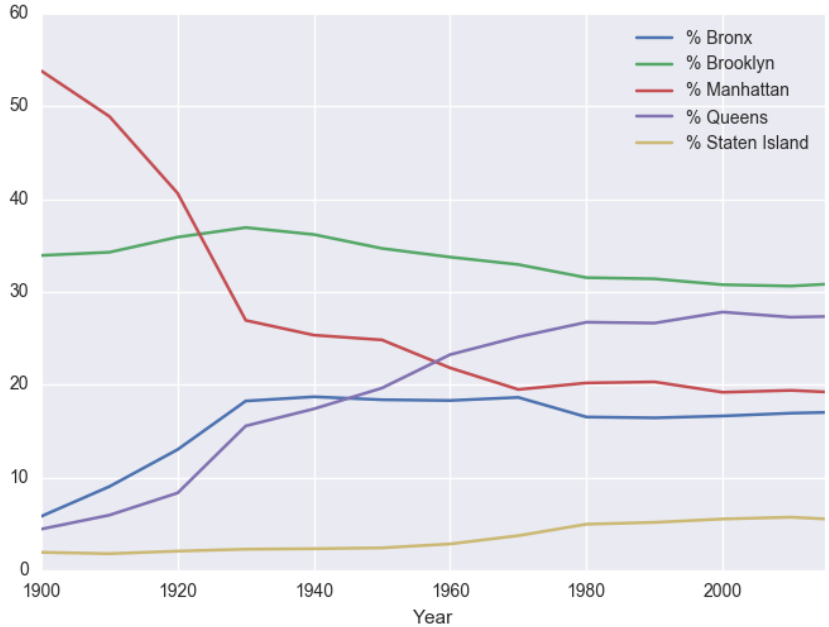

```
boros = ["Bronx","Brooklyn","Manhattan","Queens","Staten Island"]
percentCol = ["% "+ boro for boro in boros] #List comprehension
for boro in boros:
```

Population Percentage by Borough



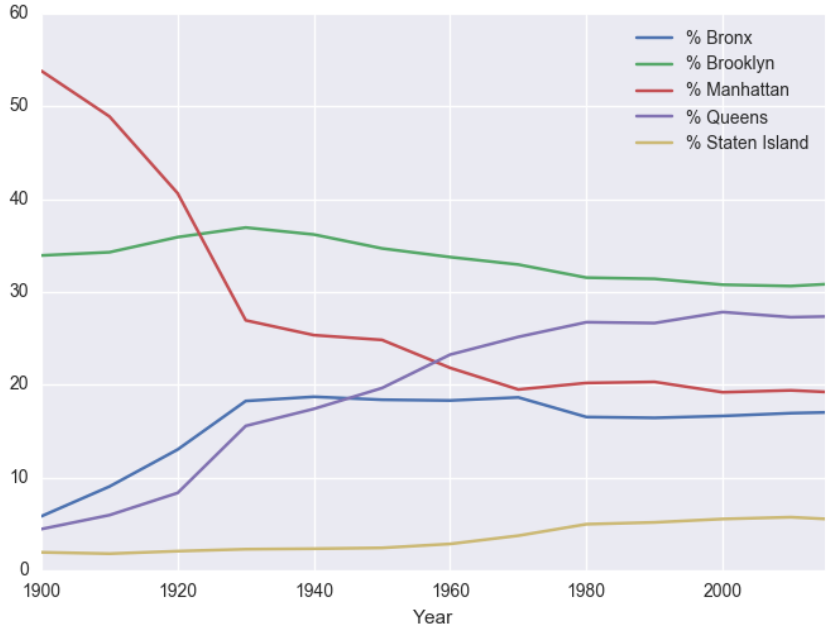

```
boros = ["Bronx","Brooklyn","Manhattan","Queens","Staten Island"]
percentCol = ["% "+ boro for boro in boros] #List comprehension
for boro in boros:
    print('Computing percentage for', boro)
    pop["% " + boro] = 100*pop2[boro]/pop["Total"]
```

Population Percentage by Borough



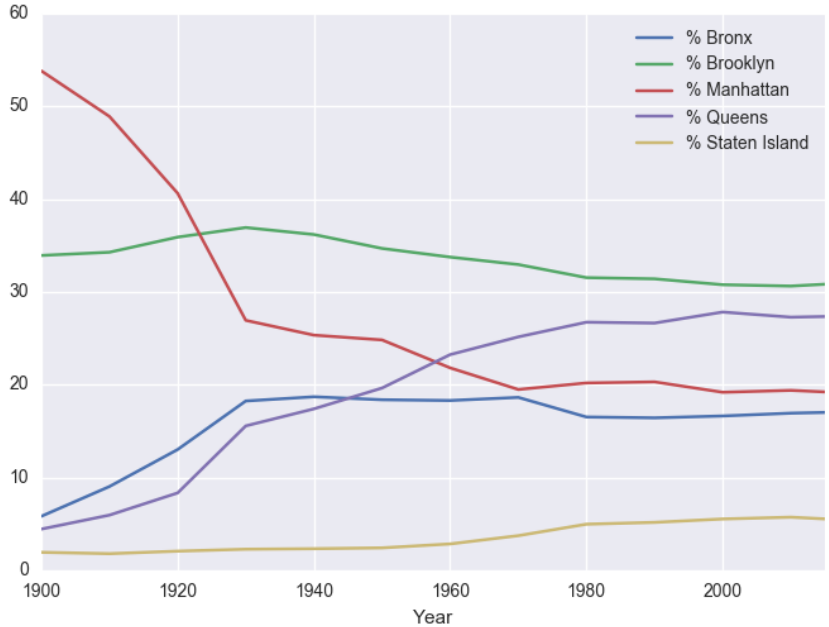

```
boros = ["Bronx","Brooklyn","Manhattan","Queens","Staten Island"]
percentCol = ["% "+ boro for boro in boros] #List comprehension
for boro in boros:
    print('Computing percentage for', boro)
    pop["% " + boro] = 100*pop2[boro]/pop["Total"]
pop = pop.drop(boros, axis = 1)
```

Population Percentage by Borough



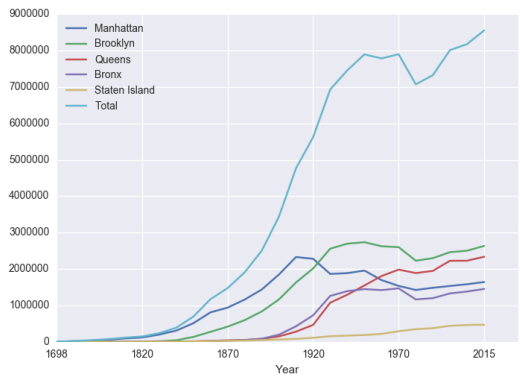

```
boros = ["Bronx","Brooklyn","Manhattan","Queens","Staten Island"]
percentCol = ["% "+ boro for boro in boros] #List comprehension
for boro in boros:
    print('Computing percentage for', boro)
    pop["% " + boro] = 100*pop2[boro]/pop["Total"]
pop = pop.drop(boros, axis = 1)
pop = pop.drop("Total", axis = 1)
```

Population Percentage by Borough



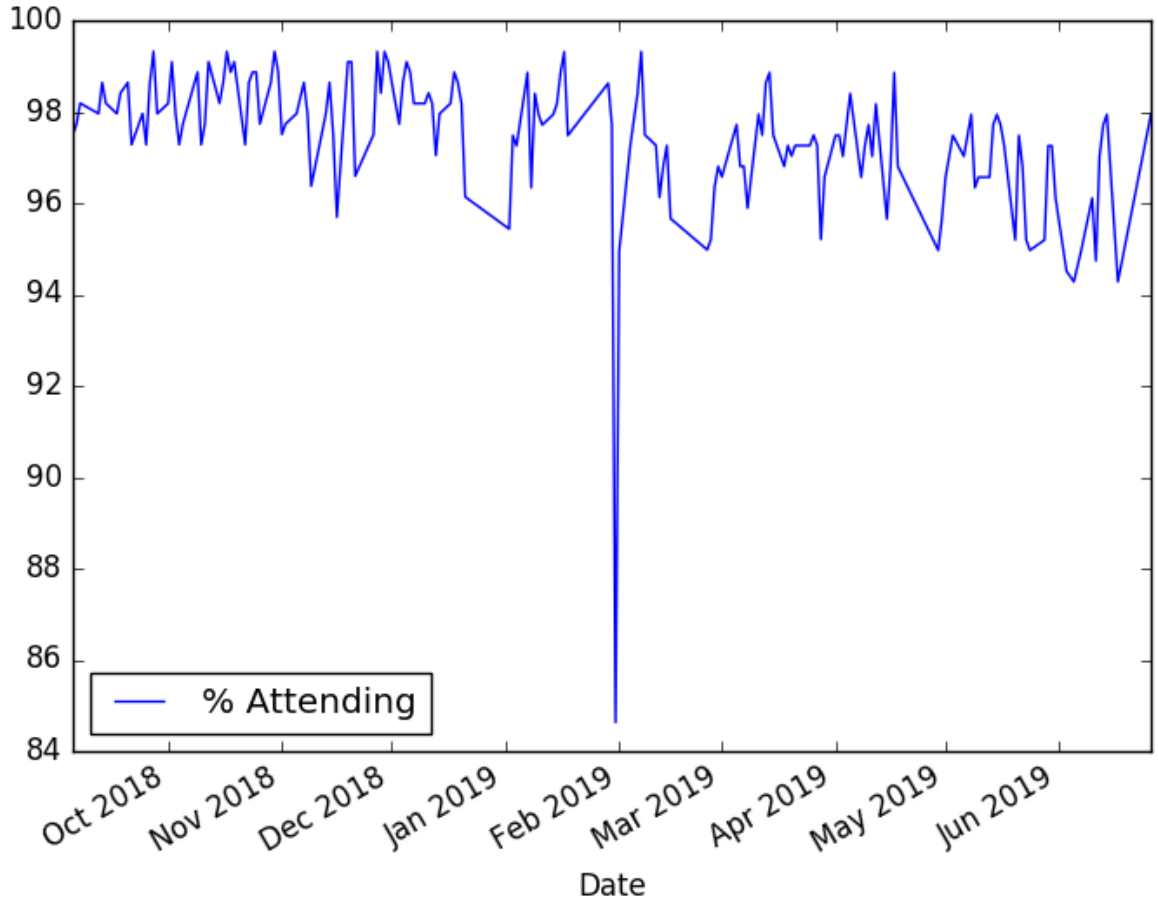

```
boros = ["Bronx","Brooklyn","Manhattan","Queens","Staten Island"]
percentCol = ["% "+ boro for boro in boros] #List comprehension
for boro in boros:
    print('Computing percentage for', boro)
    pop["% " + boro] = 100*pop2[boro]/pop["Total"]
pop = pop.drop(boros, axis = 1)
pop = pop.drop("Total", axis = 1)
pop.plot(x = "Year")
plt.show()
```

Outline



- Recap
- Design Challenge: NYC Population
- Variations on the Theme
- Design a Challenge
- Break
- Design Challenge: Parking Tickets
- Variations on the Theme
- Design a Challenge
- Wrap Up

Variations on the Theme: School Attendance



Manhattan Hunter HS, 2018-2019

Accessing Structured Data: NYC Open Data

- Freely available source of data.

Accessing Structured Data: NYC Open Data

- Freely available source of data.
- Maintained by the NYC data analytics team.

Accessing Structured Data: NYC Open Data

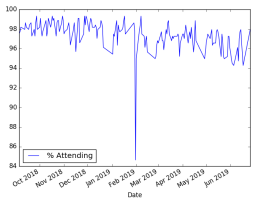
- Freely available source of data.
- Maintained by the NYC data analytics team.
- Will use `pandas`, `pyplot` & `folium` libraries to analyze, visualize and map the data.

Accessing Structured Data: NYC Open Data

- Freely available source of data.
- Maintained by the NYC data analytics team.
- Will use `pandas`, `pyplot` & `folium` libraries to analyze, visualize and map the data.
- More on downloading NYC OpenData datasets after break.

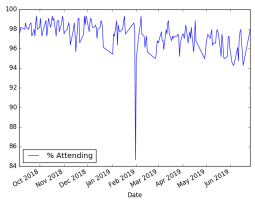
Variations on the Theme: School Attendance

- NYC OpenData has daily population counts for schools.



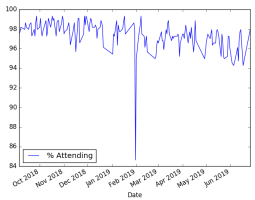
Variations on the Theme: School Attendance

- NYC OpenData has daily population counts for schools.
- 2018-2019 data for Manhattan Hunter linked on webpage.

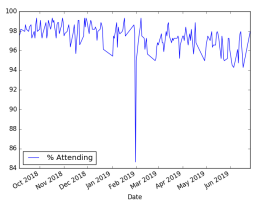


Variations on the Theme: School Attendance

- NYC OpenData has daily population counts for schools.
- 2018-2019 data for Manhattan Hunter linked on webpage. Download now to work through the exercise.

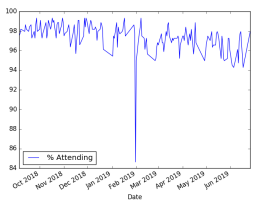


Variations on the Theme: School Attendance



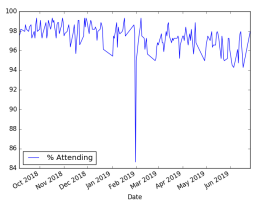
- NYC OpenData has daily population counts for schools.
- 2018-2019 data for Manhattan Hunter linked on webpage. Download now to work through the exercise.
- Can download your own from OpenData NYC (view data and filter by "School DBN").

Variations on the Theme: School Attendance



- NYC OpenData has daily population counts for schools.
- 2018-2019 data for Manhattan Hunter linked on webpage. Download now to work through the exercise.
- Can download your own from OpenData NYC (view data and filter by "School DBN").
- Dates need to be converted from `'YYYYMMDD'` to a datetime format:

Variations on the Theme: School Attendance

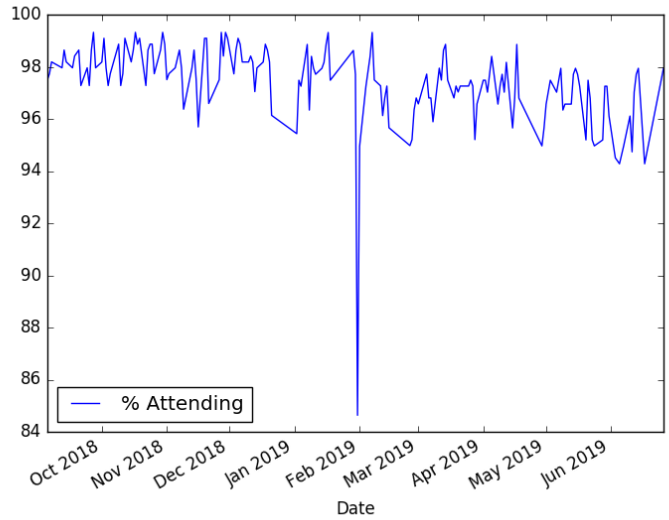


- NYC OpenData has daily population counts for schools.
- 2018-2019 data for Manhattan Hunter linked on webpage.
  Download now to work through the exercise.
- Can download your own from OpenData NYC
  (view data and filter by "School DBN").
- Dates need to be converted from `'YYYYMMDD'` to a datetime format:

```
df["Date"] = pd.to_datetime(df["Date"].apply(str))
```

- Goal: Make a plot of daily attendance
  (as percentage of enrolled).

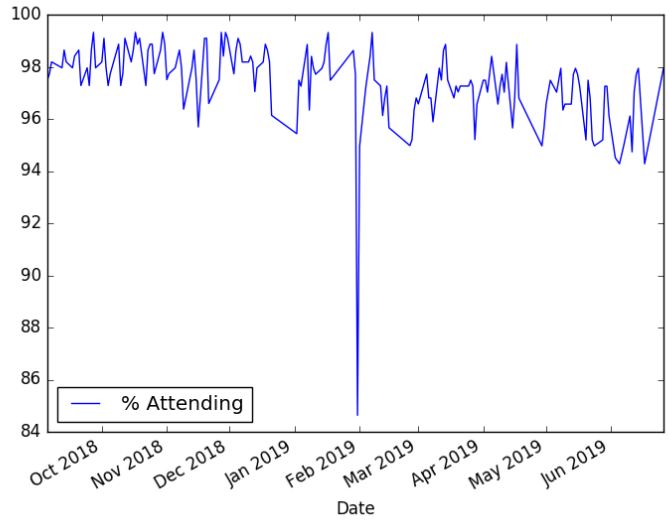
Variations on the Theme: School Attendance



c

```
df = pd.read_csv('dailyAttendance.csv') #Read file to a dataframe
```

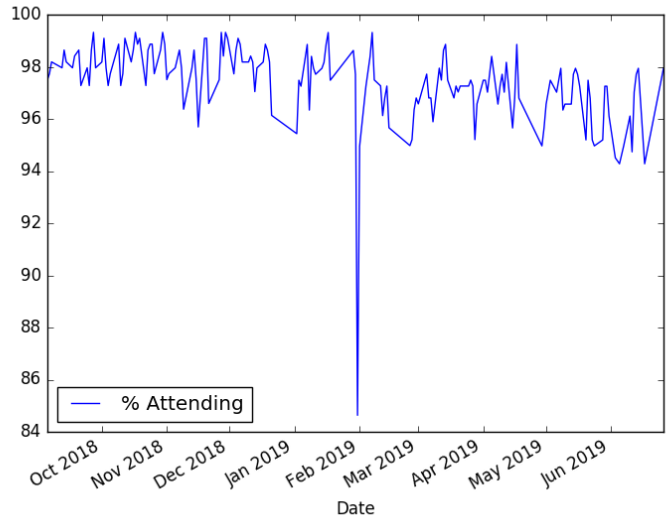
Variations on the Theme: School Attendance



c

```
df = pd.read_csv('dailyAttendance.csv') #Read file to a dataframe
df["Date"] = pd.to_datetime(df["Date"].apply(str))
```

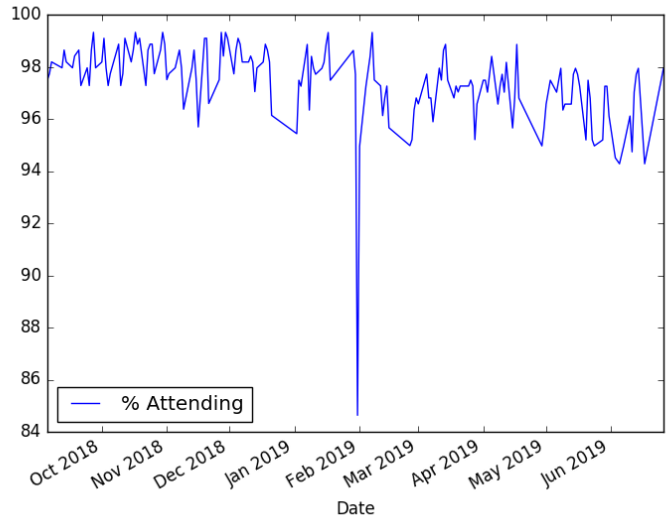
Variations on the Theme: School Attendance



c

```
df = pd.read_csv('dailyAttendance.csv') #Read file to a dataframe
df["Date"] = pd.to_datetime(df["Date"].apply(str))
df["% Attending"] = 100*df["Present"]/df["Enrolled"]
```

Variations on the Theme: School Attendance

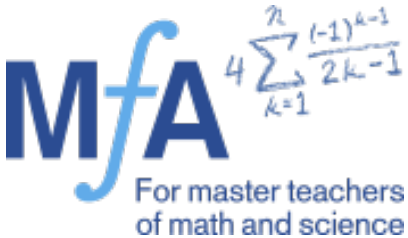


c

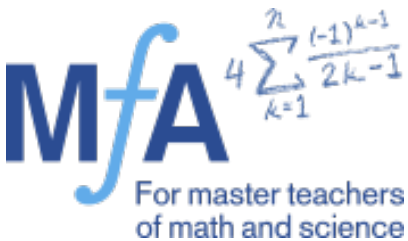
```
df = pd.read_csv('dailyAttendance.csv') #Read file to a dataframe
df["Date"] = pd.to_datetime(df["Date"].apply(str))
df["% Attending"] = 100*df["Present"]/df["Enrolled"]
df.plot(x='Date',y="% Attending")
```

Design a Challenge

- Find an interesting data set for a challenge.

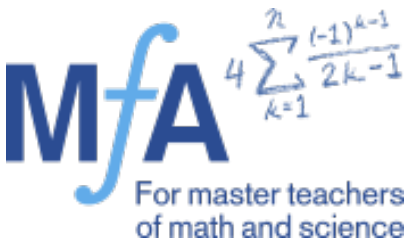


Design a Challenge



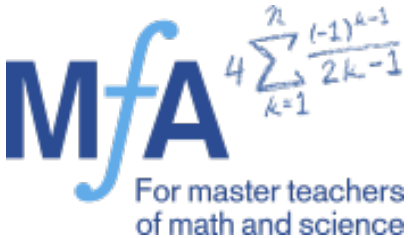
- Find an interesting data set for a challenge.
- Suggested places to look:

Design a Challenge



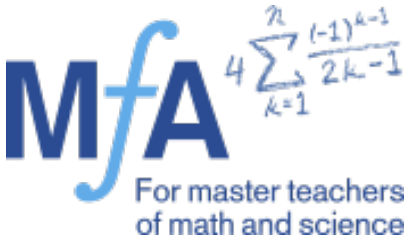
- Find an interesting data set for a challenge.
- Suggested places to look:
  - NYC OpenData.

Design a Challenge



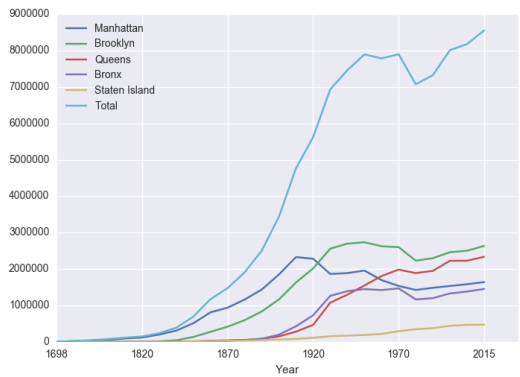
- Find an interesting data set for a challenge.
- Suggested places to look:
  - NYC OpenData.
  - Weather Underground (for historical weather data).

Design a Challenge



- Find an interesting data set for a challenge.
- Suggested places to look:
  - NYC OpenData.
  - Weather Underground (for historical weather data).
  - Kaggle Open Datasets: data, code, and competitions for data science.

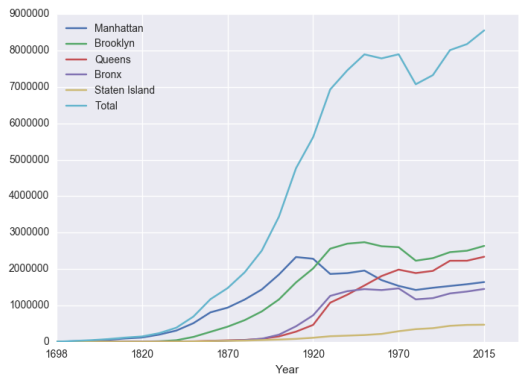
Outline



- Recap
- Design Challenge: NYC Population
- Variations on the Theme
- Design a Challenge
- Break
- Design Challenge: Parking Tickets
- Variations on the Theme
- Design a Challenge
- Wrap Up

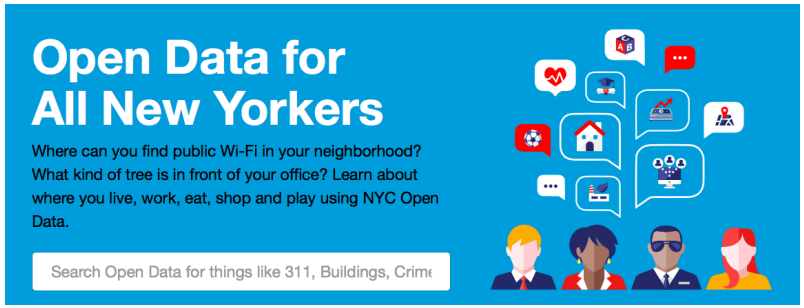
Break

Outline



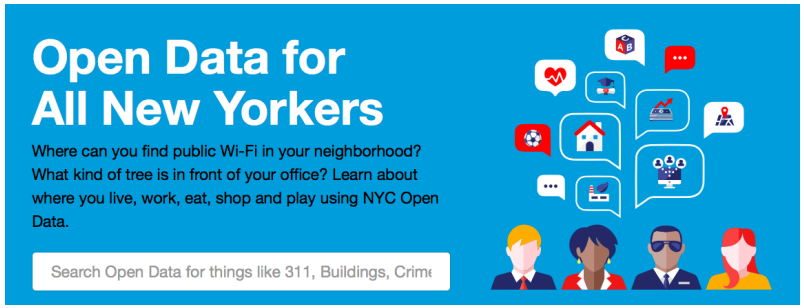
- Recap
- Design Challenge: NYC Population
- Variations on the Theme
- Design a Challenge
- Break
- Design Challenge: Parking Tickets
- Variations on the Theme
- Design a Challenge
- Wrap Up

Variations on the Theme: Binning Data: Parking Tickets



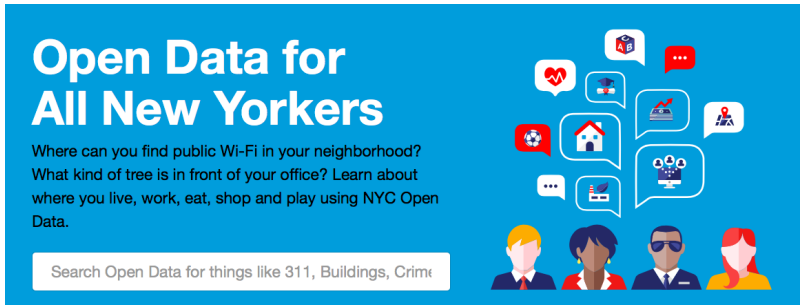
- Open Data has archived recent NYC parking tickets.

Variations on the Theme: Binning Data: Parking Tickets



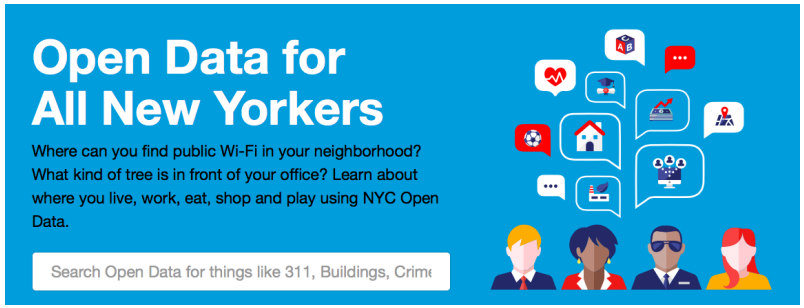
- Open Data has archived recent NYC parking tickets.
  - We will use a small version (1000 lines).

Variations on the Theme: Binning Data: Parking Tickets



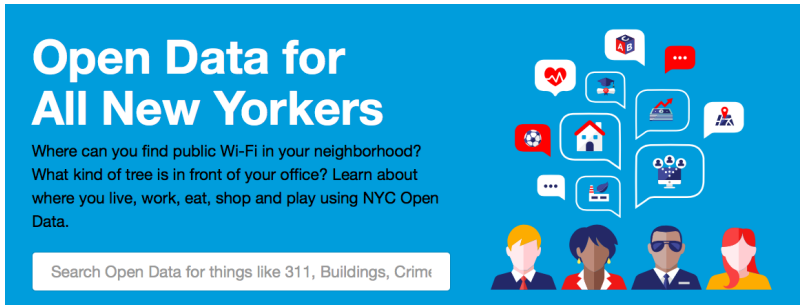
- Open Data has archived recent NYC parking tickets.
  - We will use a small version (1000 lines).
  - You are welcome to use any neighborhood in the city.

Variations on the Theme: Binning Data: Parking Tickets



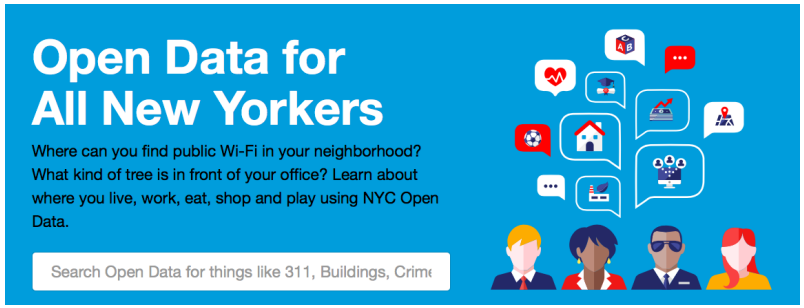
- Open Data has archived recent NYC parking tickets.
  - We will use a small version (1000 lines).
  - You are welcome to use any neighborhood in the city.
  - Suggest restricting to a single year, since can be quite large.

Variations on the Theme: Binning Data: Parking Tickets



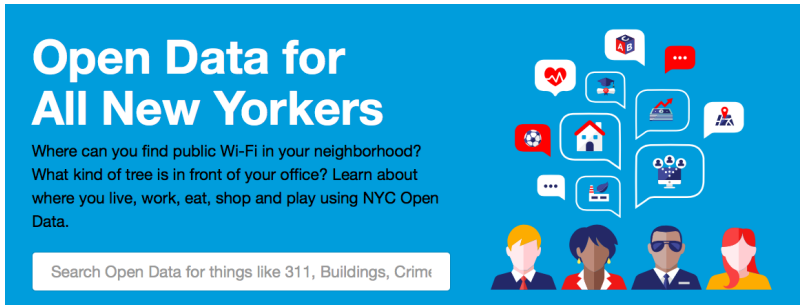
- Open Data has archived recent NYC parking tickets.
  - We will use a small version (1000 lines).
  - You are welcome to use any neighborhood in the city.
  - Suggest restricting to a single year, since can be quite large.
- A simple, but very powerful, technique is binning data: grouping data into the number of occurrences for each categories.

Variations on the Theme: Binning Data: Parking Tickets



- Open Data has archived recent NYC parking tickets.
  - We will use a small version (1000 lines).
  - You are welcome to use any neighborhood in the city.
  - Suggest restricting to a single year, since can be quite large.
- A simple, but very powerful, technique is binning data: grouping data into the number of occurrences for each categories.
- Can often show patterns that individual data points do not.

Variations on the Theme: Binning Data: Parking Tickets



- Open Data has archived recent NYC parking tickets.
  - We will use a small version (1000 lines).
  - You are welcome to use any neighborhood in the city.
  - Suggest restricting to a single year, since can be quite large.
- A simple, but very powerful, technique is binning data: grouping data into the number of occurrences for each categories.
- Can often show patterns that individual data points do not.
- We will bin parking tickets by attributes: license plate number, car color, etc.

Parking Ticket Data Format

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

Parking Ticket Data Format

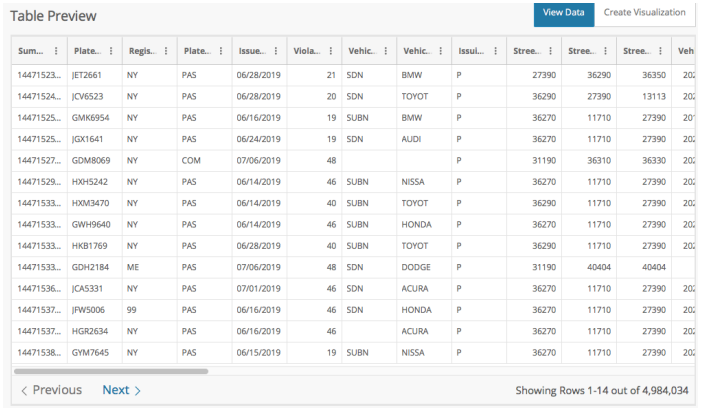


Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

Parking Ticket Data Format

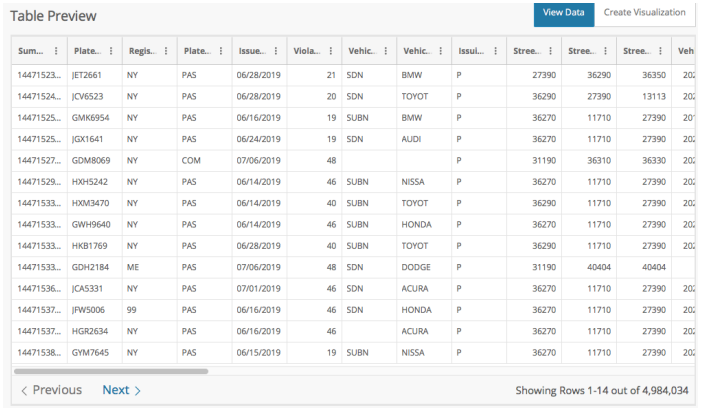


Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

- Instead of zipcode, classified by the issuing police precinct.

Parking Ticket Data Format

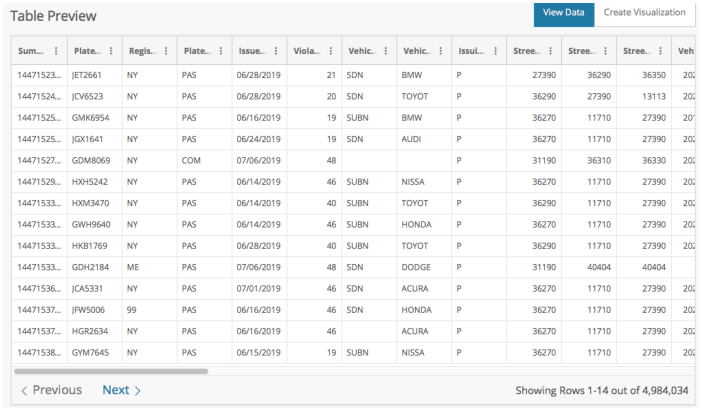


Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

- Instead of zipcode, classified by the issuing police precinct.
- To limit to a precinct, filter by “Violation Precinct”.

Parking Ticket Data Format

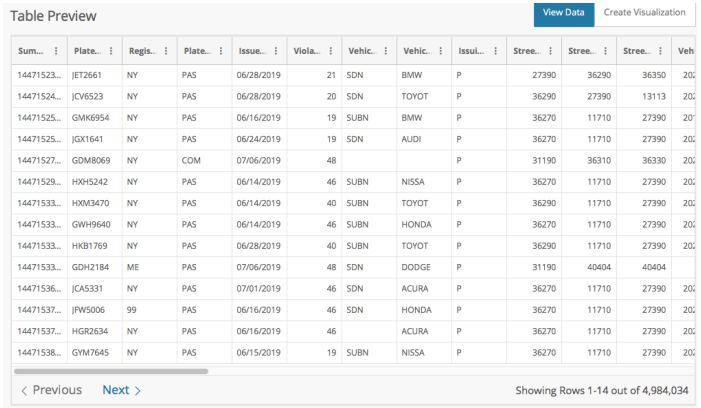


Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

- Instead of zipcode, classified by the issuing police precinct.
- To limit to a precinct, filter by “Violation Precinct”.
- The first line gives the entries in the order they occur in the rows.

Parking Ticket Data Format

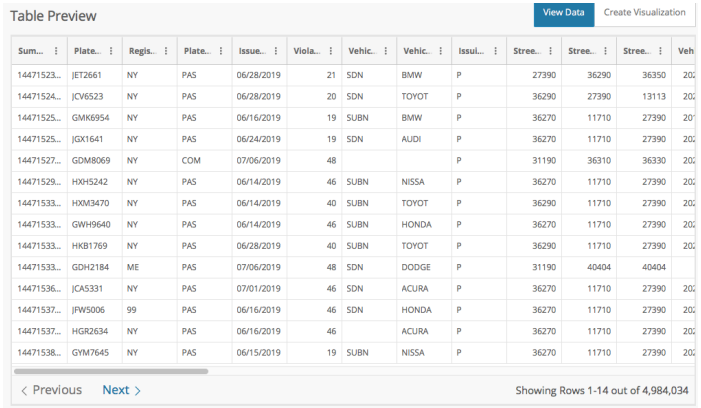


Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

- Instead of zipcode, classified by the issuing police precinct.
- To limit to a precinct, filter by “Violation Precinct”.
- The first line gives the entries in the order they occur in the rows.
- Each entry begins with a unique identifier to look up the ticket.

Parking Ticket Data Format

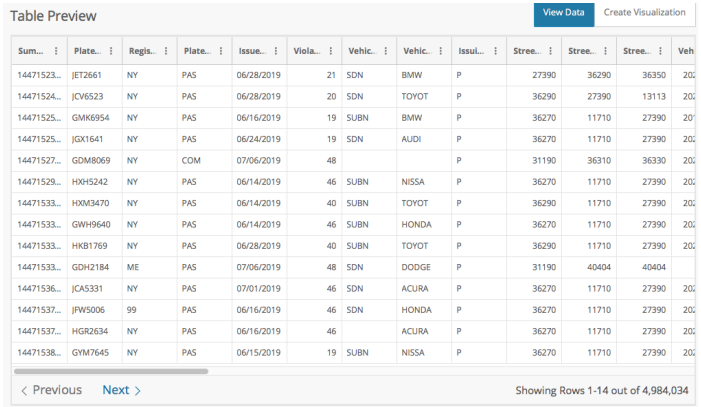


Table Preview — View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next > Showing Rows 1-14 out of 4,984,034

- Instead of zipcode, classified by the issuing police precinct.
- To limit to a precinct, filter by "Violation Precinct".
- The first line gives the entries in the order they occur in the rows.
- Each entry begins with a unique identifier to look up the ticket.
- Sample line of CSV file:

```
1335632335,L040HZ,FL,PAS,06/09/2015,46,SUBN,NISSA,X,35430,14510,15710,0,0020,20,74,921167,E074,0000,1213
83 ST,,0,408,C,,BBBBBBB,ALL,ALL,RED,0,0,-,0,,,,,
```

Parking Ticket Data Format

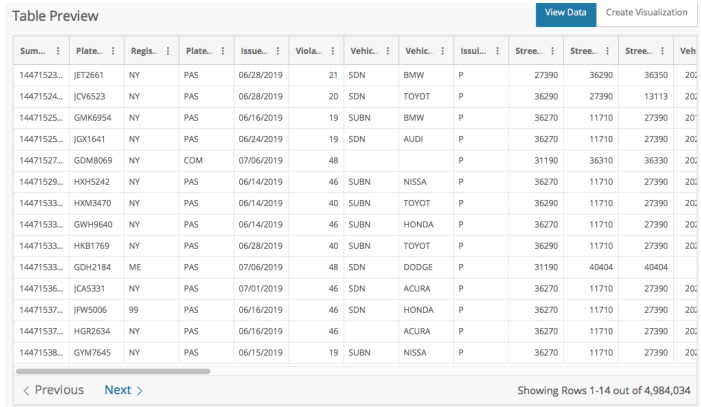


Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

- Instead of zipcode, classified by the issuing police precinct.
- To limit to a precinct, filter by "Violation Precinct".
- The first line gives the entries in the order they occur in the rows.
- Each entry begins with a unique identifier to look up the ticket.
- Sample line of CSV file:

```
1335632335,L040HZ,FL,PAS,06/09/2015,46,SUBN,NISSA,X,35430,14510,15710,0,0020,20,74,921167,E074,0000,1213
83 ST,,0,408,C,,BBBBBBB,ALL,ALL,RED,0,0,-,0,,,,,
```

- Issued on June 9, 2015 to a passenger car with Florida plates, L040HZ. The red Nissan SUV received it on W 83rd Street.

Binning Data: Parking Tickets

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

Binning Data: Parking Tickets

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

In groups, brainstorm about how to answer:

Binning Data: Parking Tickets

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

In groups, brainstorm about how to answer:

- Which car got the most tickets?

Binning Data: Parking Tickets

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

In groups, brainstorm about how to answer:

- Which car got the most tickets?
- What color of car is most likely to get a ticket?

Binning Data: Parking Tickets

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

In groups, brainstorm about how to answer:

- Which car got the most tickets?
- What color of car is most likely to get a ticket?
- What type of license gets the most tickets?

Binning Data: Parking Tickets

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

In groups, brainstorm about how to answer:

- Which car got the most tickets?
- What color of car is most likely to get a ticket?
- What type of license gets the most tickets?
- Are all states equally represented in license plates that get tickets?

Binning Data: Parking Tickets

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

In groups, brainstorm about how to answer:

- Which car got the most tickets?
- What color of car is most likely to get a ticket?
- What type of license gets the most tickets?
- Are all states equally represented in license plates that get tickets?
- Which location yields the most tickets?

Counting Tickets per Car

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

Counting Tickets per Car

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

How can tell which car got the most tickets?

Counting Tickets per Car

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

How can tell which car got the most tickets?

- Need to a unique way to identify different cars.

Counting Tickets per Car

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

How can tell which car got the most tickets?

- Need to a unique way to identify different cars.
  Luckily, cars almost always have license plates– unique by state.

Counting Tickets per Car

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

How can tell which car got the most tickets?

- Need to a unique way to identify different cars.
  Luckily, cars almost always have license plates– unique by state.
  (For this simple exercise, assume each license plate ID is unique– not unreasonable since every state has a different schema for assigning numbers, but to be more accurate should keep track of license plate number and issuing state.)

Counting Tickets per Car

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

How can tell which car got the most tickets?

- Need to a unique way to identify different cars.
  Luckily, cars almost always have license plates– unique by state.
  (For this simple exercise, assume each license plate ID is unique– not unreasonable since every state has a different schema for assigning numbers, but to be more accurate should keep track of license plate number and issuing state.)
- Want to "bin" tickets by license plates ("Plate ID"),

Counting Tickets per Car

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

How can tell which car got the most tickets?

- Need to a unique way to identify different cars.
  Luckily, cars almost always have license plates– unique by state.
  (For this simple exercise, assume each license plate ID is unique– not unreasonable since every state has a different schema for assigning numbers, but to be more accurate should keep track of license plate number and issuing state.)
- Want to "bin" tickets by license plates ("Plate ID"),
  and then count the size of bins.

Your Turn: Binning Data

- Sample program:

Your Turn: Binning Data

- Sample program:
  ```
  import pandas as pd
  ```

Your Turn: Binning Data

- Sample program:

```
import pandas as pd
tickets = pd.read_csv('tickets.csv')
```

Your Turn: Binning Data

- Sample program:

```
import pandas as pd
tickets = pd.read_csv('tickets.csv')
#Print out the data frame:
print(tickets)
```

Your Turn: Binning Data

- Sample program:

```
import pandas as pd
tickets = pd.read_csv('tickets.csv')
#Print out the data frame:
print(tickets)
#Print out licence plates:
print(tickets["Plate ID"])
```

Your Turn: Binning Data

- Sample program:

```
import pandas as pd
tickets = pd.read_csv('tickets.csv')
#Print out the data frame:
print(tickets)
#Print out licence plates:
print(tickets["Plate ID"])
#Print out plates & number of tickets each got:
print(tickets["Plate ID"].value_counts())
```

Your Turn: Binning Data

- Sample program:

```
import pandas as pd
tickets = pd.read_csv('tickets.csv')
#Print out the data frame:
print(tickets)
#Print out licence plates:
print(tickets["Plate ID"])
#Print out plates & number of tickets each got:
print(tickets["Plate ID"].value_counts())
#Print 10 worst & number of tickets:
print(tickets["Plate ID"].value_counts()[:10])
```

Your Turn: Binning Data

- Sample program:

```
import pandas as pd
tickets = pd.read_csv('tickets.csv')
#Print out the data frame:
print(tickets)
#Print out licence plates:
print(tickets["Plate ID"])
#Print out plates & number of tickets each got:
print(tickets["Plate ID"].value_counts())
#Print 10 worst & number of tickets:
print(tickets["Plate ID"].value_counts()[:10])
```

- For the sample data set, there were few cars that got more than a ticket a day...

Binning Data: Parking Tickets

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

Binning Data: Parking Tickets

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

< Previous Next >

Showing Rows 1-14 out of 4,984,034

In groups, write programs for:

Binning Data: Parking Tickets

Table Preview

View Data | Create Visualization

Sum...	Plate...	Regis...	Plate...	Issue...	Viola...	Vehic...	Vehic...	Issui...	Stree...	Stree...	Stree...	Veh
14471523...	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	202
14471524...	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	202
14471525...	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	201
14471525...	JGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	202
14471527...	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	202
14471529...	HXH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	202
14471533...	HXM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GWH9640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	202
14471533...	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	202
14471533...	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40404	40404	
14471536...	JCA5331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	202
14471537...	JFW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	202
14471537...	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	202
14471538...	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	202

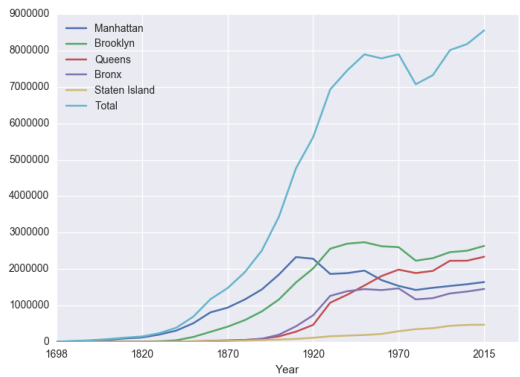
< Previous Next >

Showing Rows 1-14 out of 4,984,034

In groups, write programs for:

- Which car got the most tickets?
- What color of car is most likely to get a ticket?
- What type of license gets the most tickets?
- Are all states equally represented in license plates that get tickets?
- Which location yields the most tickets?

Outline



- Recap
- Design Challenge: NYC Population
- Variations on the Theme
- Design a Challenge
- Break
- Design Challenge: Parking Tickets
- Variations on the Theme
- Design a Challenge
- Wrap Up

Variations on the Theme: OpenData Film Permits

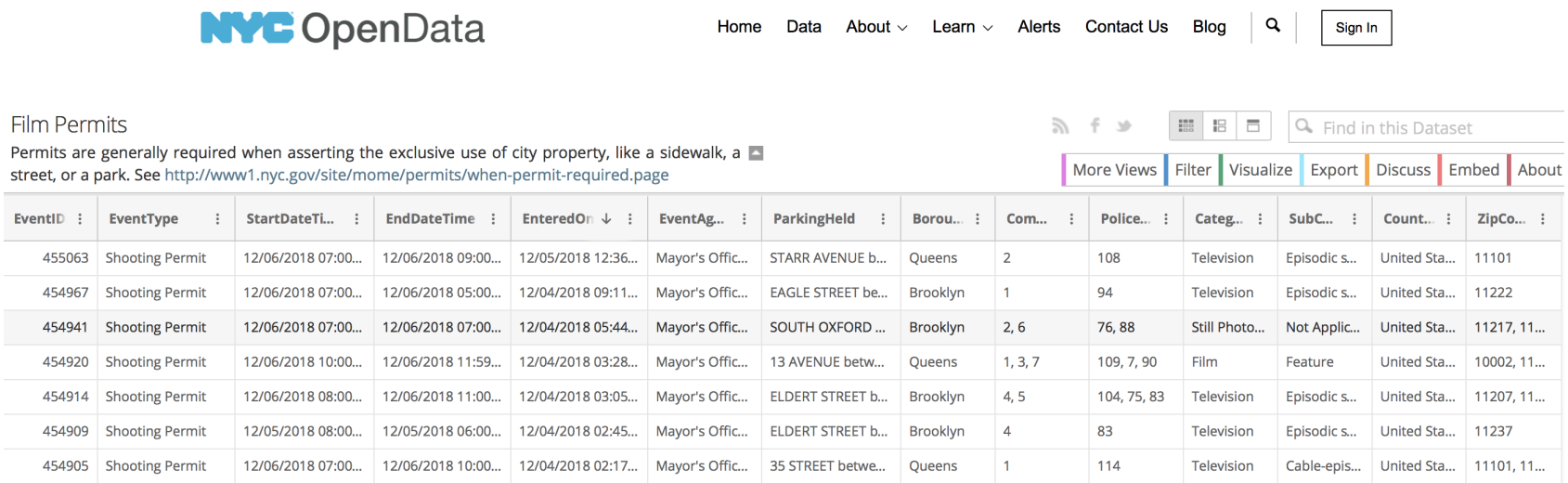
Home Data About Learn

Film Permits

Permits are generally required when asserting the exclusive use of city property, like a sidewalk, a street, or a park. See http://www1.nyc.gov/site/mome/permits/when-permit-required.page

EventID	EventType	StartDateTi...	EndDateTime	EnteredOn	EventAg...	ParkingHeld	Borou...
455063	Shooting Permit	12/06/2018 07:00...	12/06/2018 09:00...	12/05/2018 12:36...	Mayor's Offic...	STARR AVENUE b...	Queens
454967	Shooting Permit	12/06/2018 07:00...	12/06/2018 05:00...	12/04/2018 09:11...	Mayor's Offic...	EAGLE STREET be...	Brooklyn
454941	Shooting Permit	12/06/2018 07:00...	12/06/2018 07:00...	12/04/2018 05:44...	Mayor's Offic...	SOUTH OXFORD ...	Brooklyn
454920	Shooting Permit	12/06/2018 10:00...	12/06/2018 11:59...	12/04/2018 03:28...	Mayor's Offic...	13 AVENUE betw...	Queens
454914	Shooting Permit	12/06/2018 08:00...	12/06/2018 11:00...	12/04/2018 03:05...	Mayor's Offic...	ELDERT STREET b...	Brooklyn
454909	Shooting Permit	12/05/2018 08:00...	12/05/2018 06:00...	12/04/2018 02:45...	Mayor's Offic...	ELDERT STREET b...	Brooklyn
454905	Shooting Permit	12/06/2018 07:00...	12/06/2018 10:00...	12/04/2018 02:17...	Mayor's Offic...	35 STREET betwe...	Queens

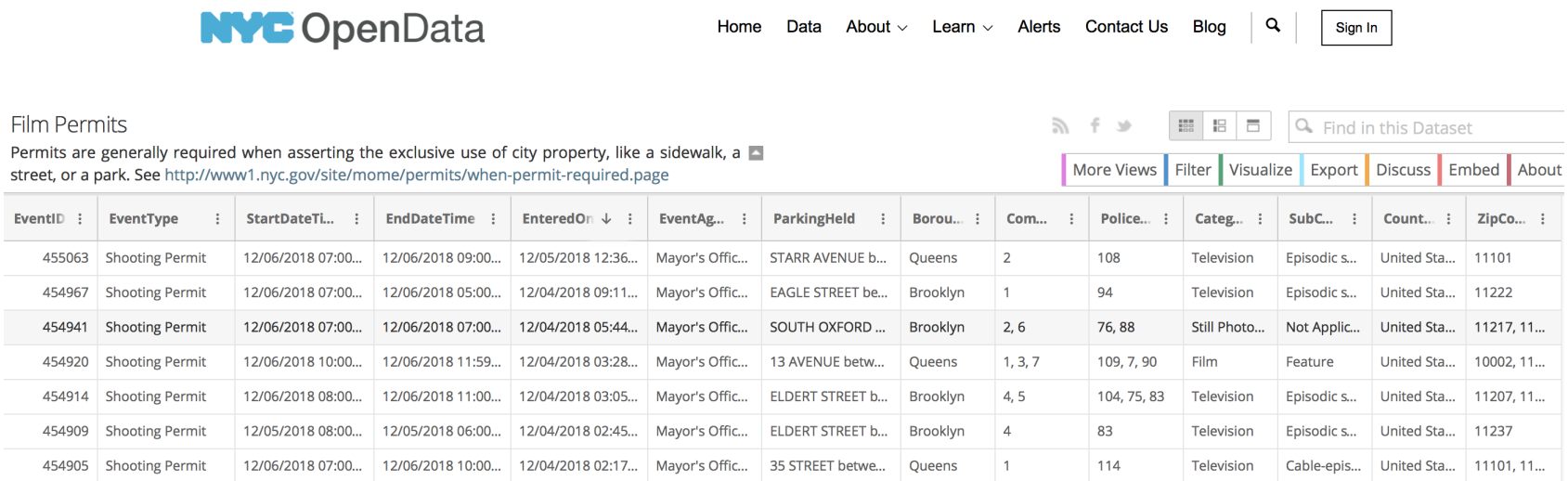
Example: OpenData Film Permits



EventID	EventType	StartDateTi...	EndDateTime	EnteredOn	EventAg...	ParkingHeld	Borou...	Com...	Police...	Categ...	SubC...	Count...	ZipCo...
455063	Shooting Permit	12/06/2018 07:00...	12/06/2018 09:00...	12/05/2018 12:36...	Mayor's Offic...	STARR AVENUE b...	Queens	2	108	Television	Episodic s...	United Sta...	11101
454967	Shooting Permit	12/06/2018 07:00...	12/06/2018 05:00...	12/04/2018 09:11...	Mayor's Offic...	EAGLE STREET be...	Brooklyn	1	94	Television	Episodic s...	United Sta...	11222
454941	Shooting Permit	12/06/2018 07:00...	12/06/2018 07:00...	12/04/2018 05:44...	Mayor's Offic...	SOUTH OXFORD ...	Brooklyn	2, 6	76, 88	Still Photo...	Not Applic...	United Sta...	11217, 11...
454920	Shooting Permit	12/06/2018 10:00...	12/06/2018 11:59...	12/04/2018 03:28...	Mayor's Offic...	13 AVENUE betw...	Queens	1, 3, 7	109, 7, 90	Film	Feature	United Sta...	10002, 11...
454914	Shooting Permit	12/06/2018 08:00...	12/06/2018 11:00...	12/04/2018 03:05...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4, 5	104, 75, 83	Television	Episodic s...	United Sta...	11207, 11...
454909	Shooting Permit	12/05/2018 08:00...	12/05/2018 06:00...	12/04/2018 02:45...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4	83	Television	Episodic s...	United Sta...	11237
454905	Shooting Permit	12/06/2018 07:00...	12/06/2018 10:00...	12/04/2018 02:17...	Mayor's Offic...	35 STREET betwe...	Queens	1	114	Television	Cable-epis...	United Sta...	11101, 11...

- What's the most popular street for filming?

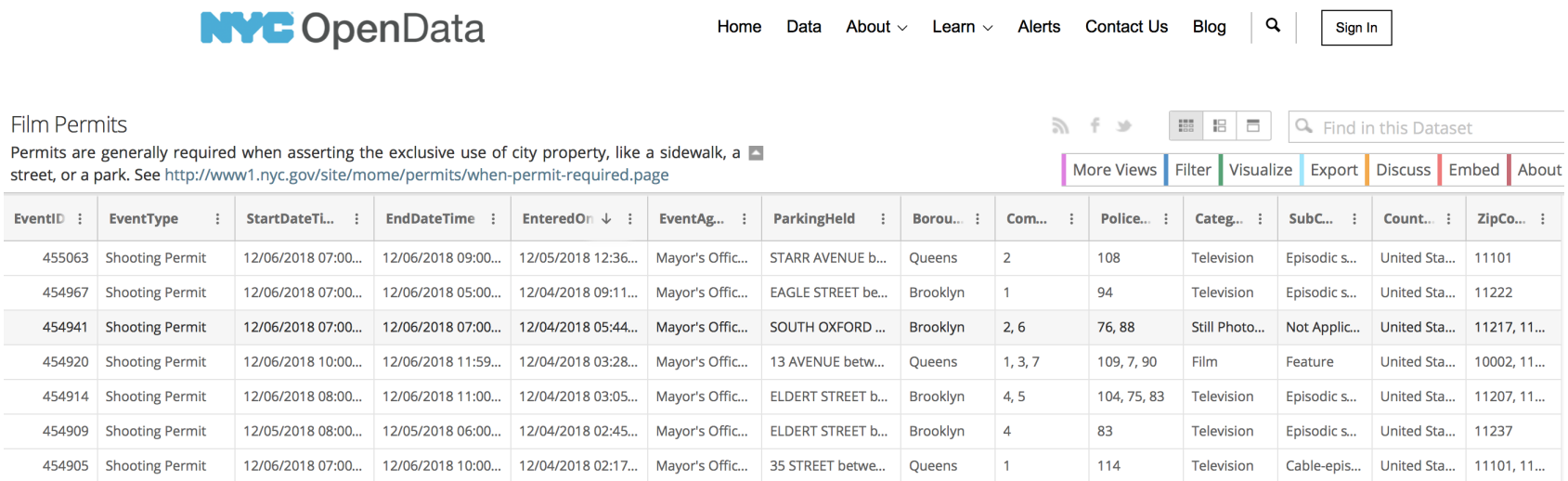
Example: OpenData Film Permits



EventID	EventType	StartDateTi...	EndDateTime	EnteredOn	EventAg...	ParkingHeld	Borou...	Com...	Police...	Categ...	SubC...	Count...	ZipCo...
455063	Shooting Permit	12/06/2018 07:00...	12/06/2018 09:00...	12/05/2018 12:36...	Mayor's Offic...	STARR AVENUE b...	Queens	2	108	Television	Episodic s...	United Sta...	11101
454967	Shooting Permit	12/06/2018 07:00...	12/06/2018 05:00...	12/04/2018 09:11...	Mayor's Offic...	EAGLE STREET be...	Brooklyn	1	94	Television	Episodic s...	United Sta...	11222
454941	Shooting Permit	12/06/2018 07:00...	12/06/2018 07:00...	12/04/2018 05:44...	Mayor's Offic...	SOUTH OXFORD ...	Brooklyn	2, 6	76, 88	Still Photo...	Not Applic...	United Sta...	11217, 11...
454920	Shooting Permit	12/06/2018 10:00...	12/06/2018 11:59...	12/04/2018 03:28...	Mayor's Offic...	13 AVENUE betw...	Queens	1, 3, 7	109, 7, 90	Film	Feature	United Sta...	10002, 11...
454914	Shooting Permit	12/06/2018 08:00...	12/06/2018 11:00...	12/04/2018 03:05...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4, 5	104, 75, 83	Television	Episodic s...	United Sta...	11207, 11...
454909	Shooting Permit	12/05/2018 08:00...	12/05/2018 06:00...	12/04/2018 02:45...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4	83	Television	Episodic s...	United Sta...	11237
454905	Shooting Permit	12/06/2018 07:00...	12/06/2018 10:00...	12/04/2018 02:17...	Mayor's Offic...	35 STREET betwe...	Queens	1	114	Television	Cable-epis...	United Sta...	11101, 11...

- What's the most popular street for filming?
- What's the most popular borough?

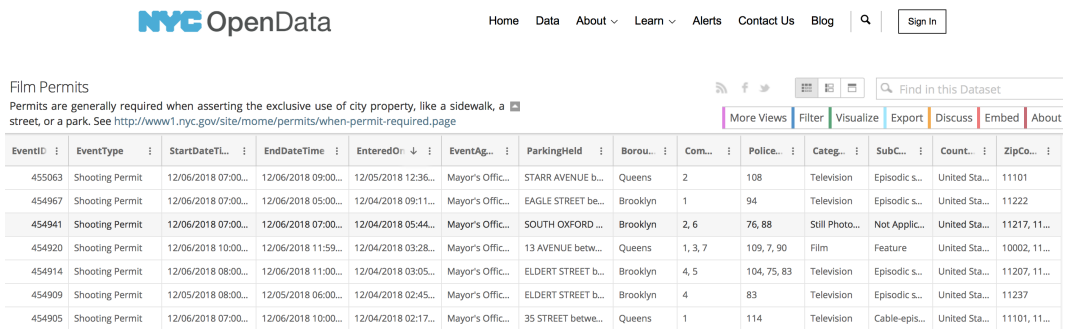
Example: OpenData Film Permits



EventID	EventType	StartDateTi...	EndDateTime	EnteredOn	EventAg...	ParkingHeld	Borou...	Com...	Police...	Categ...	SubC...	Count...	ZipCo...
455063	Shooting Permit	12/06/2018 07:00...	12/06/2018 09:00...	12/05/2018 12:36...	Mayor's Offic...	STARR AVENUE b...	Queens	2	108	Television	Episodic s...	United Sta...	11101
454967	Shooting Permit	12/06/2018 07:00...	12/06/2018 05:00...	12/04/2018 09:11...	Mayor's Offic...	EAGLE STREET be...	Brooklyn	1	94	Television	Episodic s...	United Sta...	11222
454941	Shooting Permit	12/06/2018 07:00...	12/06/2018 07:00...	12/04/2018 05:44...	Mayor's Offic...	SOUTH OXFORD ...	Brooklyn	2, 6	76, 88	Still Photo...	Not Applic...	United Sta...	11217, 11...
454920	Shooting Permit	12/06/2018 10:00...	12/06/2018 11:59...	12/04/2018 03:28...	Mayor's Offic...	13 AVENUE betw...	Queens	1, 3, 7	109, 7, 90	Film	Feature	United Sta...	10002, 11...
454914	Shooting Permit	12/06/2018 08:00...	12/06/2018 11:00...	12/04/2018 03:05...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4, 5	104, 75, 83	Television	Episodic s...	United Sta...	11207, 11...
454909	Shooting Permit	12/05/2018 08:00...	12/05/2018 06:00...	12/04/2018 02:45...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4	83	Television	Episodic s...	United Sta...	11237
454905	Shooting Permit	12/06/2018 07:00...	12/06/2018 10:00...	12/04/2018 02:17...	Mayor's Offic...	35 STREET betwe...	Queens	1	114	Television	Cable-epis...	United Sta...	11101, 11...

- What's the most popular street for filming?
- What's the most popular borough?
- How many TV episodes were filmed?

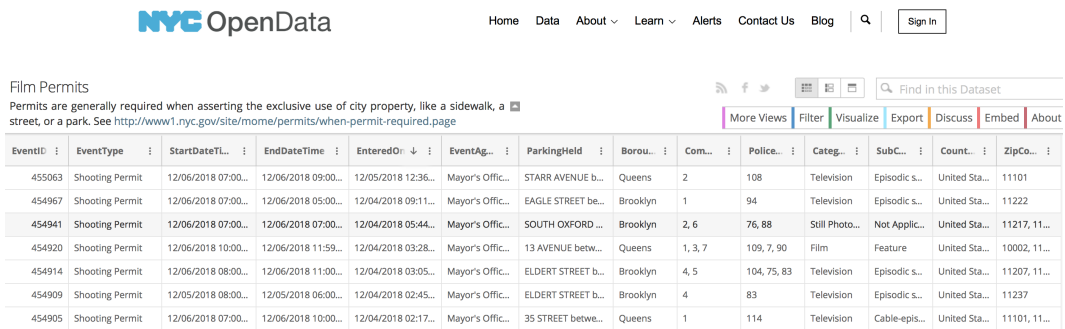
Example: OpenData Film Permits



EventID	EventType	StartDateTi...	EndDateTime	EnteredOn	EventAg...	ParkingHeld	Borou...	Com...	Police...	Categ...	SubC...	Count...	ZipCo...
455063	Shooting Permit	12/06/2018 07:00...	12/06/2018 09:00...	12/05/2018 12:36...	Mayor's Offic...	STARR AVENUE b...	Queens	2	108	Television	Episodic s...	United Sta...	11101
454967	Shooting Permit	12/06/2018 07:00...	12/06/2018 05:00...	12/04/2018 09:11...	Mayor's Offic...	EAGLE STREET be...	Brooklyn	1	94	Television	Episodic s...	United Sta...	11222
454941	Shooting Permit	12/06/2018 07:00...	12/06/2018 07:00...	12/04/2018 05:44...	Mayor's Offic...	SOUTH OXFORD ...	Brooklyn	2, 6	76, 88	Still Photo...	Not Applic...	United Sta...	11217, 11...
454920	Shooting Permit	12/06/2018 10:00...	12/06/2018 11:59...	12/04/2018 03:28...	Mayor's Offic...	13 AVENUE betw...	Queens	1, 3, 7	109, 7, 90	Film	Feature	United Sta...	10002, 11...
454914	Shooting Permit	12/06/2018 08:00...	12/06/2018 11:00...	12/04/2018 03:05...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4, 5	104, 75, 83	Television	Episodic s...	United Sta...	11207, 11...
454909	Shooting Permit	12/05/2018 08:00...	12/05/2018 06:00...	12/04/2018 02:45...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4	83	Television	Episodic s...	United Sta...	11237
454905	Shooting Permit	12/06/2018 07:00...	12/06/2018 10:00...	12/04/2018 02:17...	Mayor's Offic...	35 STREET betwe...	Queens	1	114	Television	Cable-epis...	United Sta...	11101, 11...

- Download the data as a CSV file and store on your computer.

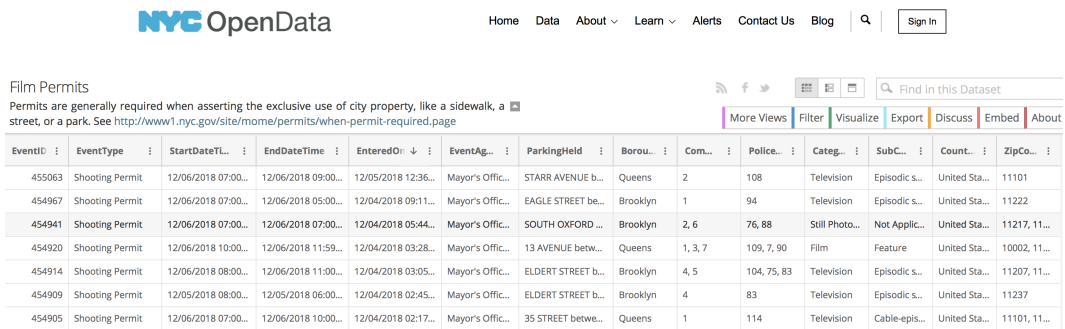
Example: OpenData Film Permits



- Download the data as a CSV file and store on your computer.
- Python program:

```
#CSci 127 Teaching Staff
#March 2019
#OpenData Film Permits

#Import pandas for reading and analyzing CSV data:
import pandas as pd
csvFile = "filmPermits.csv"   #Name of the CSV file
tickets = pd.read_csv(csvFile)#Read in the file to a dataframe
```

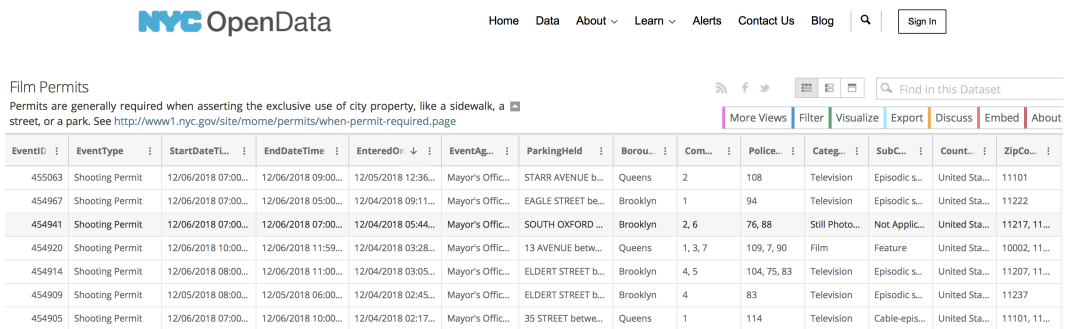
Example: OpenData Film Permits



- Download the data as a CSV file and store on your computer.
- Python program:

```
#CSci 127 Teaching Staff
#March 2019
#OpenData Film Permits

#Import pandas for reading and analyzing CSV data:
import pandas as pd
csvFile = "filmPermits.csv"    #Name of the CSV file
tickets = pd.read_csv(csvFile)#Read in the file to a dataframe
print(tickets)                 #Print out the dataframe
```

Example: OpenData Film Permits

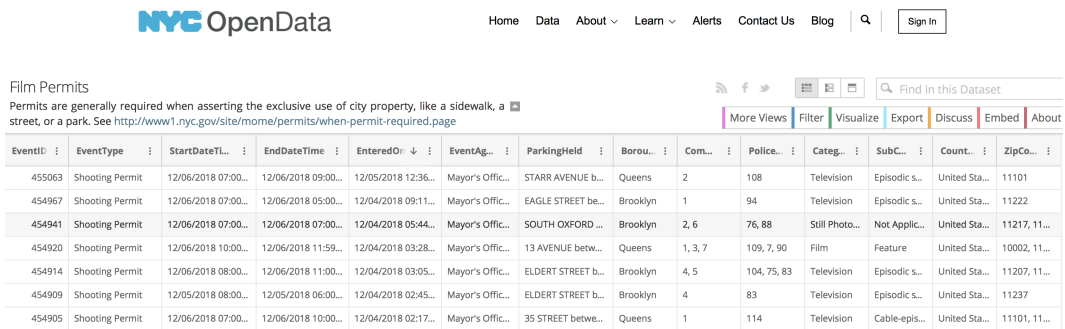


EventID	EventType	StartDateTi...	EndDateTime	EnteredOn	EventAg...	ParkingHeld	Borou...	Com...	Police...	Categ...	SubC...	Count...	ZipCo...
455063	Shooting Permit	12/06/2018 07:00...	12/06/2018 09:00...	12/05/2018 12:36...	Mayor's Offic...	STARR AVENUE b...	Queens	2	108	Television	Episodic s...	United Sta...	11101
454967	Shooting Permit	12/06/2018 07:00...	12/06/2018 05:00...	12/04/2018 09:11...	Mayor's Offic...	EAGLE STREET be...	Brooklyn	1	94	Television	Episodic s...	United Sta...	11222
454941	Shooting Permit	12/06/2018 07:00...	12/06/2018 07:00...	12/04/2018 05:44...	Mayor's Offic...	SOUTH OXFORD ...	Brooklyn	2, 6	76, 88	Still Photo...	Not Applic...	United Sta...	11217, 11...
454920	Shooting Permit	12/06/2018 10:00...	12/06/2018 11:59...	12/04/2018 03:28...	Mayor's Offic...	13 AVENUE betw...	Queens	1, 3, 7	109, 7, 90	Film	Feature	United Sta...	10002, 11...
454914	Shooting Permit	12/06/2018 08:00...	12/06/2018 11:00...	12/04/2018 03:05...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4, 5	104, 75, 83	Television	Episodic s...	United Sta...	11207, 11...
454909	Shooting Permit	12/05/2018 08:00...	12/05/2018 06:00...	12/04/2018 02:45...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4	83	Television	Episodic s...	United Sta...	11237
454905	Shooting Permit	12/06/2018 07:00...	12/06/2018 10:00...	12/04/2018 02:17...	Mayor's Offic...	35 STREET betwe...	Queens	1	114	Television	Cable-epis...	United Sta...	11101, 11...

- Download the data as a CSV file and store on your computer.
- Python program:

```
#CSci 127 Teaching Staff
#March 2019
#OpenData Film Permits

#Import pandas for reading and analyzing CSV data:
import pandas as pd
csvFile = "filmPermits.csv"    #Name of the CSV file
tickets = pd.read_csv(csvFile)#Read in the file to a dataframe
print(tickets)                 #Print out the dataframe
print(tickets["ParkingHeld"]) #Print out streets (multiple times)
```

Example: OpenData Film Permits

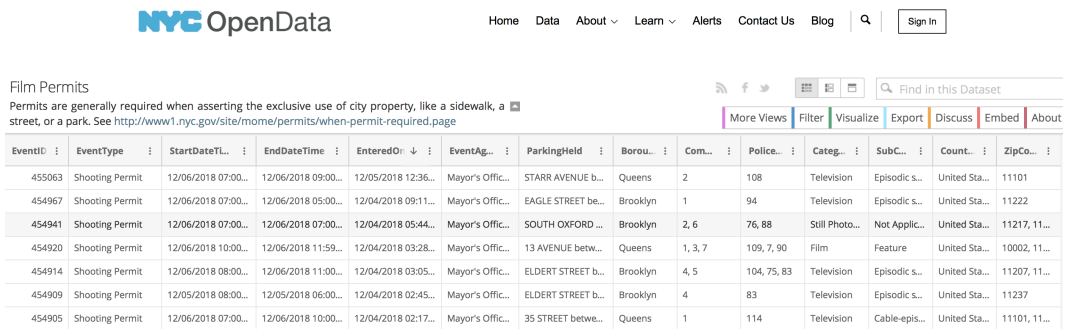


EventID	EventType	StartDateTi...	EndDateTime	EnteredOn	EventAg...	ParkingHeld	Borou...	Com...	Police...	Categ...	SubC...	Count...	ZipCo...
455063	Shooting Permit	12/06/2018 07:00...	12/06/2018 09:00...	12/05/2018 12:36...	Mayor's Offic...	STARR AVENUE b...	Queens	2	108	Television	Episodic s...	United Sta...	11101
454967	Shooting Permit	12/06/2018 07:00...	12/06/2018 05:00...	12/04/2018 09:11...	Mayor's Offic...	EAGLE STREET be...	Brooklyn	1	94	Television	Episodic s...	United Sta...	11222
454941	Shooting Permit	12/06/2018 07:00...	12/06/2018 07:00...	12/04/2018 05:44...	Mayor's Offic...	SOUTH OXFORD ...	Brooklyn	2, 6	76, 88	Still Photo...	Not Applic...	United Sta...	11217, 11...
454920	Shooting Permit	12/06/2018 10:00...	12/06/2018 11:59...	12/04/2018 03:28...	Mayor's Offic...	13 AVENUE betw...	Queens	1, 3, 7	109, 7, 90	Film	Feature	United Sta...	10002, 11...
454914	Shooting Permit	12/06/2018 08:00...	12/06/2018 11:00...	12/04/2018 03:05...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4, 5	104, 75, 83	Television	Episodic s...	United Sta...	11207, 11...
454909	Shooting Permit	12/05/2018 08:00...	12/05/2018 06:00...	12/04/2018 02:45...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4	83	Television	Episodic s...	United Sta...	11237
454905	Shooting Permit	12/06/2018 07:00...	12/06/2018 10:00...	12/04/2018 02:17...	Mayor's Offic...	35 STREET betwe...	Queens	1	114	Television	Cable-epis...	United Sta...	11101, 11...

- Download the data as a CSV file and store on your computer.
- Python program:

```
#CSci 127 Teaching Staff
#March 2019
#OpenData Film Permits

#Import pandas for reading and analyzing CSV data:
import pandas as pd
csvFile = "filmPermits.csv"    #Name of the CSV file
tickets = pd.read_csv(csvFile)#Read in the file to a dataframe
print(tickets)                 #Print out the dataframe
print(tickets["ParkingHeld"]) #Print out streets (multiple times)
print(tickets["ParkingHeld"].value_counts()) #Print out streets & number of times used
```

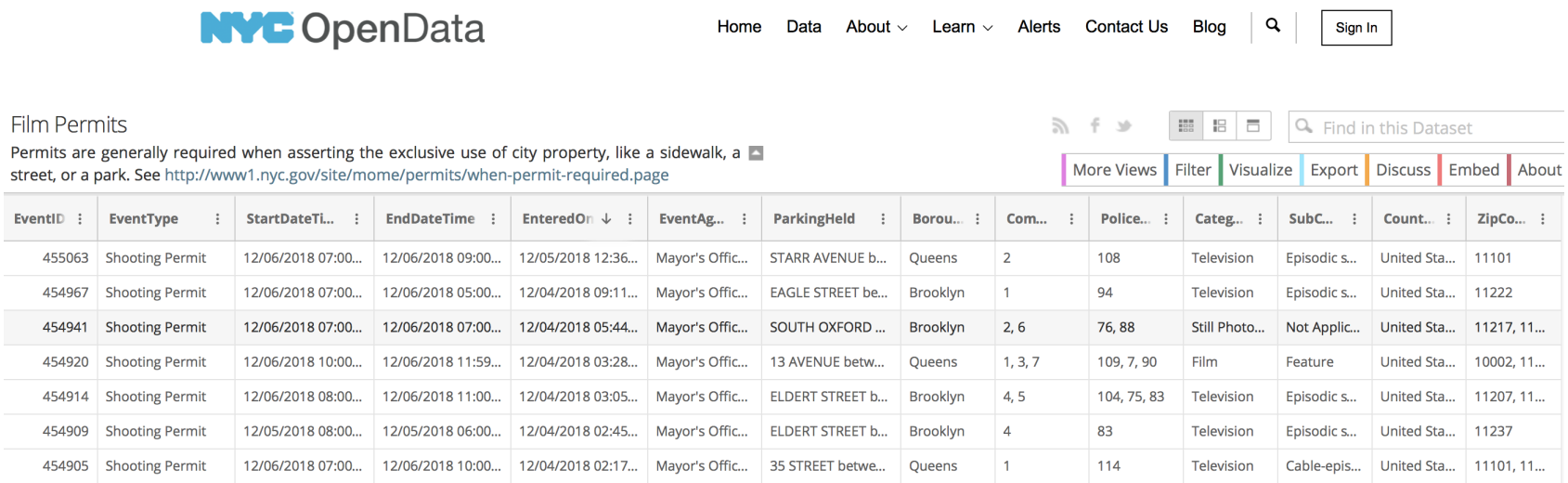
Example: OpenData Film Permits



- Download the data as a CSV file and store on your computer.
- Python program:

```
#CSci 127 Teaching Staff
#March 2019
#OpenData Film Permits

#Import pandas for reading and analyzing CSV data:
import pandas as pd
csvFile = "filmPermits.csv"   #Name of the CSV file
tickets = pd.read_csv(csvFile)#Read in the file to a dataframe
print(tickets)                #Print out the dataframe
print(tickets["ParkingHeld"]) #Print out streets (multiple times)
print(tickets["ParkingHeld"].value_counts()) #Print out streets & number of times used
print(tickets["ParkingHeld"].value_counts()[:10])  #Print 10 most popular
```

In Groups: OpenData Film Permits

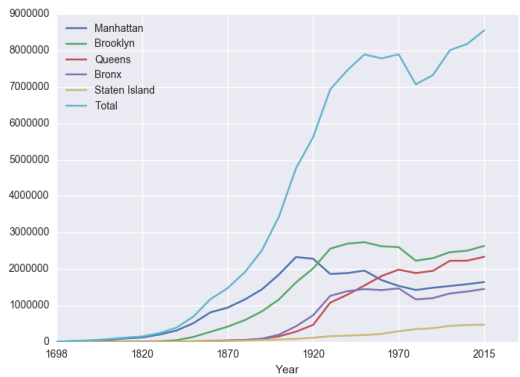


EventID	EventType	StartDateTi...	EndDateTime	EnteredOn	EventAg...	ParkingHeld	Borou...	Com...	Police...	Categ...	SubC...	Count...	ZipCo...
455063	Shooting Permit	12/06/2018 07:00...	12/06/2018 09:00...	12/05/2018 12:36...	Mayor's Offic...	STARR AVENUE b...	Queens	2	108	Television	Episodic s...	United Sta...	11101
454967	Shooting Permit	12/06/2018 07:00...	12/06/2018 05:00...	12/04/2018 09:11...	Mayor's Offic...	EAGLE STREET be...	Brooklyn	1	94	Television	Episodic s...	United Sta...	11222
454941	Shooting Permit	12/06/2018 07:00...	12/06/2018 07:00...	12/04/2018 05:44...	Mayor's Offic...	SOUTH OXFORD ...	Brooklyn	2, 6	76, 88	Still Photo...	Not Applic...	United Sta...	11217, 11...
454920	Shooting Permit	12/06/2018 10:00...	12/06/2018 11:59...	12/04/2018 03:28...	Mayor's Offic...	13 AVENUE betw...	Queens	1, 3, 7	109, 7, 90	Film	Feature	United Sta...	10002, 11...
454914	Shooting Permit	12/06/2018 08:00...	12/06/2018 11:00...	12/04/2018 03:05...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4, 5	104, 75, 83	Television	Episodic s...	United Sta...	11207, 11...
454909	Shooting Permit	12/05/2018 08:00...	12/05/2018 06:00...	12/04/2018 02:45...	Mayor's Offic...	ELDERT STREET b...	Brooklyn	4	83	Television	Episodic s...	United Sta...	11237
454905	Shooting Permit	12/06/2018 07:00...	12/06/2018 10:00...	12/04/2018 02:17...	Mayor's Offic...	35 STREET betwe...	Queens	1	114	Television	Cable-epis...	United Sta...	11101, 11...

Can approach the other questions in the same way:

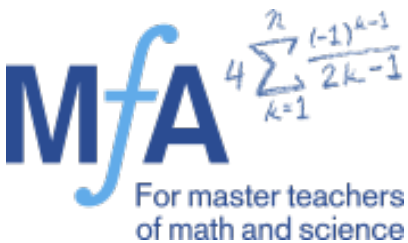
- What's the most popular street for filming?
- What's the most popular borough?
- How many TV episodes were filmed?

Outline



- Recap
- Design Challenge: NYC Population
- Variations on the Theme
- Design a Challenge
- Break
- Design Challenge: Parking Tickets
- Variations on the Theme
- Design a Challenge
- Wrap Up

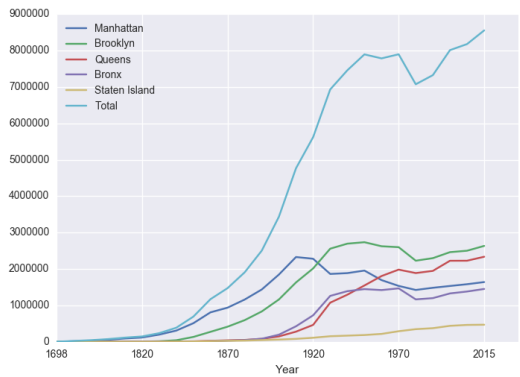
Design a Challenge



With your group, brainstorm about a design challenge that:

- An interesting publicly available data set,
- To analyze with the pandas commands we've discussed.

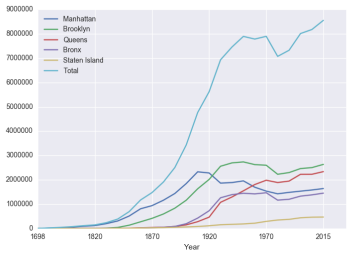
Outline



- Recap
- Design Challenge: NYC Population
- Variations on the Theme
- Design a Challenge
- Break
- Design Challenge: Parking Tickets
- Variations on the Theme
- Design a Challenge
- Wrap Up

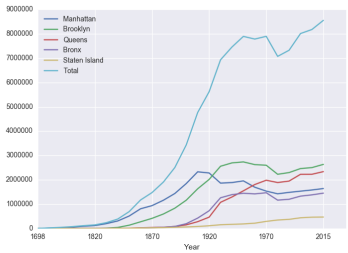
Wrap Up

- Introduced pandas for analyzing structured data.

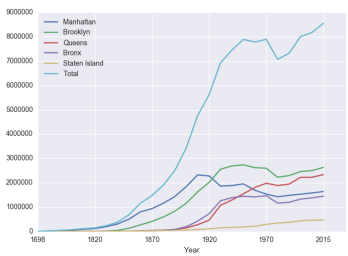


Wrap Up

- Introduced pandas for analyzing structured data.
  - Plotting, simple stats functions, and slicing.

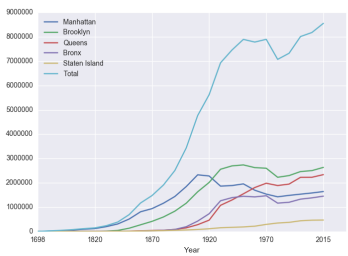


Wrap Up



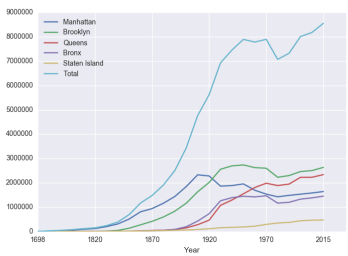
- Introduced pandas for analyzing structured data.
  - Plotting, simple stats functions, and slicing.
  - Didn't cover: accessing rows, joining/merging tables, applying functions, ...

Wrap Up



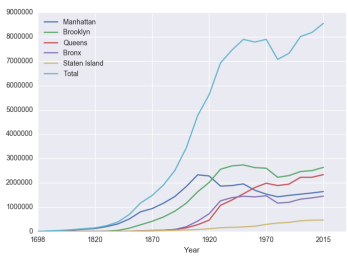
- Introduced pandas for analyzing structured data.
  - Plotting, simple stats functions, and slicing.
  - Didn't cover: accessing rows, joining/merging tables, applying functions, ...
- Used publicly available data:

Wrap Up



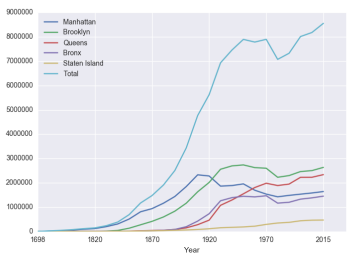
- Introduced pandas for analyzing structured data.
  - Plotting, simple stats functions, and slicing.
  - Didn't cover: accessing rows, joining/merging tables, applying functions, ...
- Used publicly available data:
  - Great source: NYC Open Data.

Wrap Up



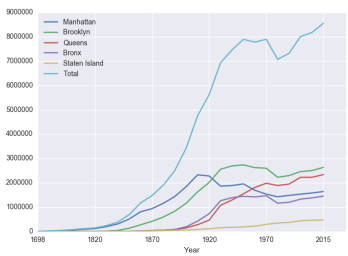
- Introduced pandas for analyzing structured data.
  - Plotting, simple stats functions, and slicing.
  - Didn't cover: accessing rows, joining/merging tables, applying functions, ...
- Used publicly available data:
  - Great source: NYC Open Data.
  - Examined population data, school attendance, parking tickets, film permits.

Wrap Up



- Introduced pandas for analyzing structured data.
  - Plotting, simple stats functions, and slicing.
  - Didn't cover: accessing rows, joining/merging tables, applying functions, ...
- Used publicly available data:
  - Great source: NYC Open Data.
  - Examined population data, school attendance, parking tickets, film permits.
- If time: share challenges you designed...

Wrap Up



- Introduced pandas for analyzing structured data.
  - Plotting, simple stats functions, and slicing.
  - Didn't cover: accessing rows, joining/merging tables, applying functions, ...
- Used publicly available data:
  - Great source: NYC Open Data.
  - Examined population data, school attendance, parking tickets, film permits.
- If time: share challenges you designed...
- See you in three weeks!