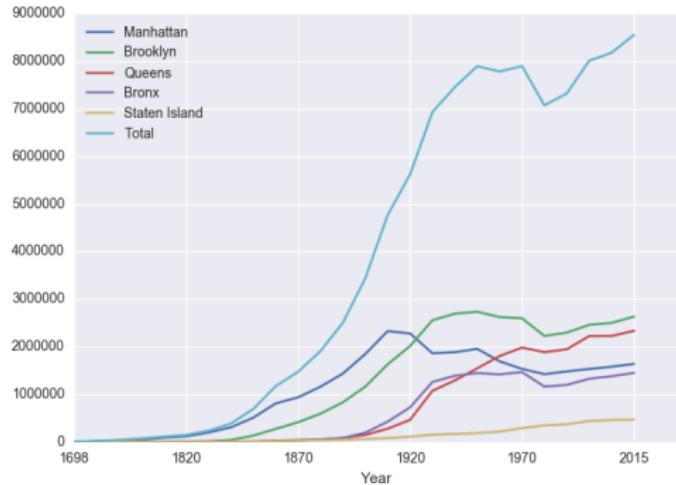


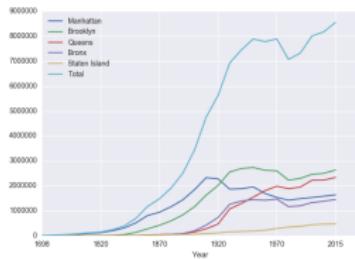
MfA: Python in the City



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

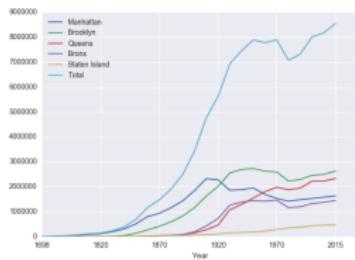
Goal: Every table have at most one from each school, one from each discipline.

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

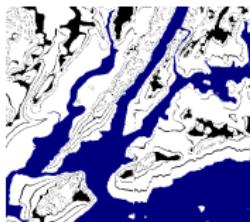
Outline



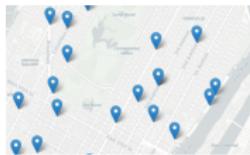
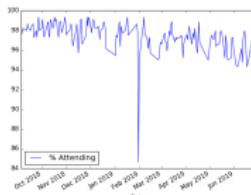
- Recap
- Design Challenge: NYC Population
- Variations on the Theme
- Design a Challenge
- Break
- Design Challenge: Parking Tickets
- Variations on the Theme
- Design a Challenge
- Wrap Up

Recap: Workshop Overview

Three sessions:

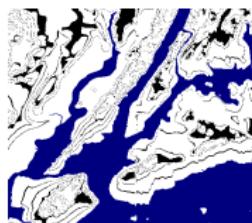


- ① Flood Maps (arrays & images)
- ② Noisiest Street (structured data, file I/O)
- ③ Mapping Collisions (using objects, mapping coordinates)

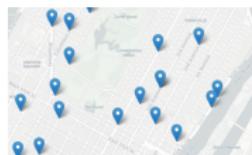
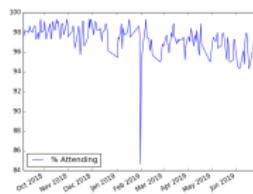


Recap: Workshop Overview

Three sessions:



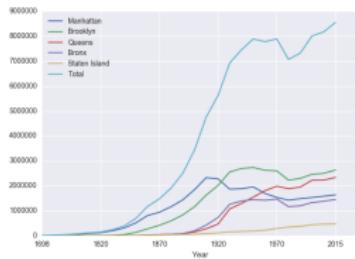
- ① Flood Maps (arrays & images)
- ② Noisiest Street (structured data, file I/O)
- ③ Mapping Collisions (using objects, mapping coordinates)



Each session:

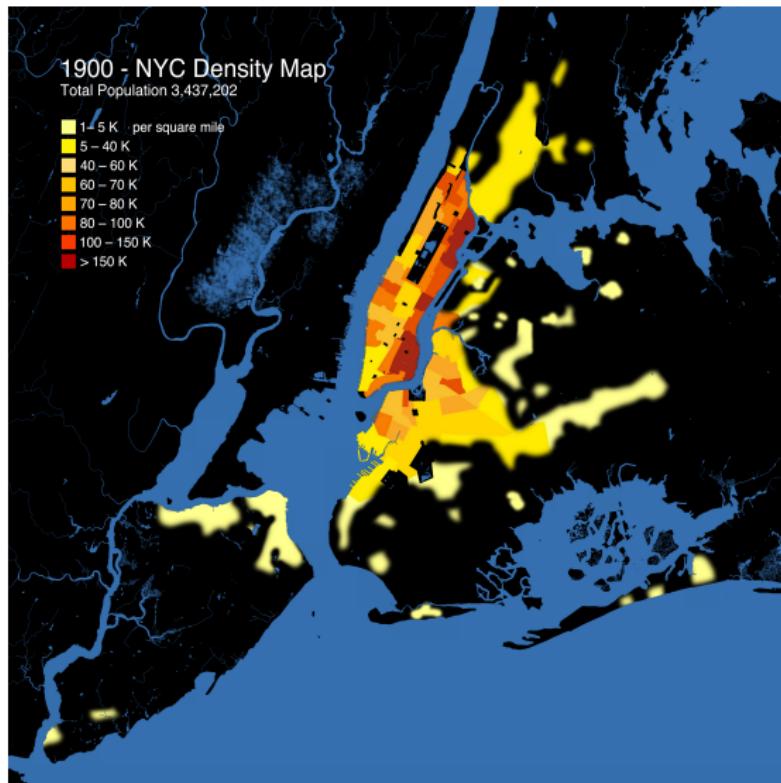
- Design Challenge
 - ▶ Analyze a publicly available dataset
 - ▶ Introduce computing concepts & packages
 - ▶ Write a program to solve the problem
- Variations on the theme
- Design a Challenge

Outline



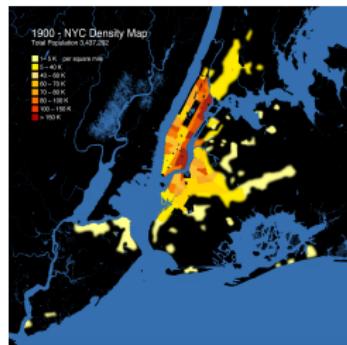
- Recap
- Design Challenge: NYC Population
- Variations on the Theme
- Design a Challenge
- Break
- Design Challenge: Parking Tickets
- Variations on the Theme
- Design a Challenge
- Wrap Up

Design Challenge: NYC Population



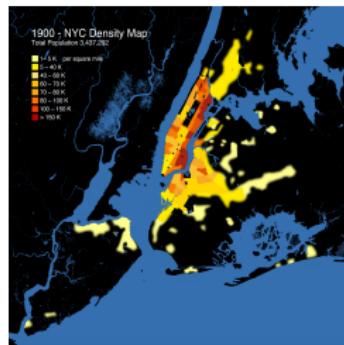
(Myles Zhang, wiki)

Design Challenge: NYC Population



(Myles Zhang, wiki)

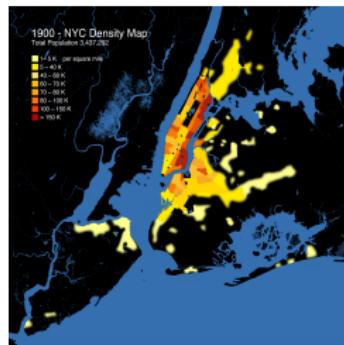
Design Challenge: NYC Population



(Myles Zhang, wiki)

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

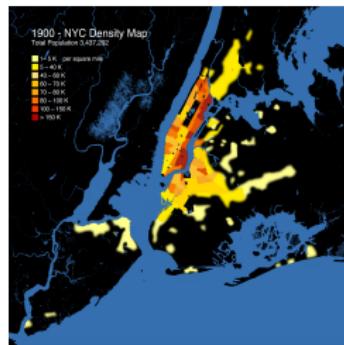
Design Challenge: NYC Population



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

(Myles Zhang, wiki)

Design Challenge: NYC Population

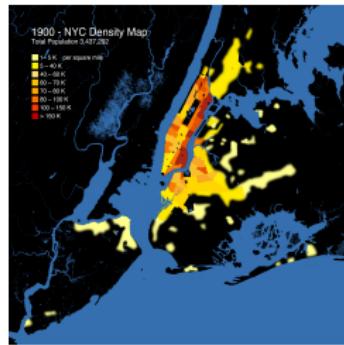


(Myles Zhang, wiki)

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?

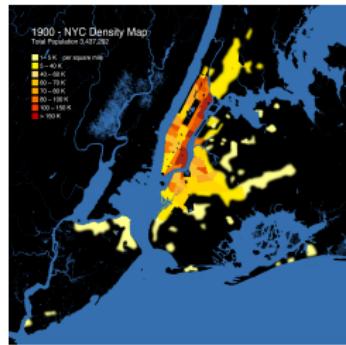
Data: Population



Input: What data do you need?

(Myles Zhang, wiki)

Data: Population

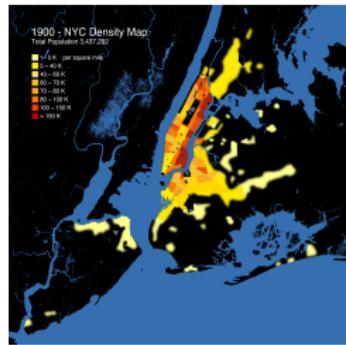


(Myles Zhang, wiki)

Input: What data do you need?

- Populations of the city in 1900 to today.

Data: Population

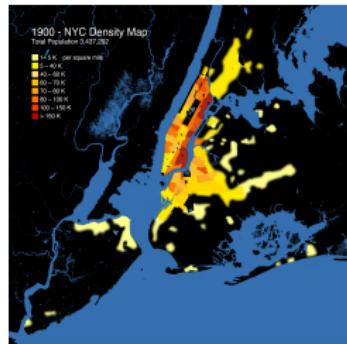


(Myles Zhang, wiki)

Input: What data do you need?

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

Data: Population

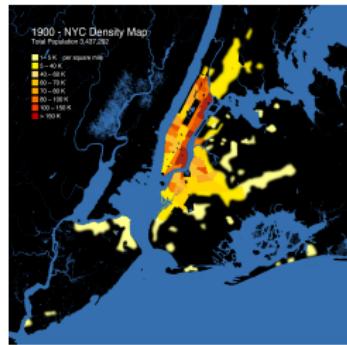


(Myles Zhang, wiki)

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
1699,4937,2017,,727,7681
1771,21863,3623,,,2847,2843
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
1830,148716,11803,9031,8246,6135,152056
1840,212710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,496115
1860,813669,279122,12903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1800093,1166580,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2852112,2180401,308041,430980,85969,5207485
1930,1867312,2560401,1079129,1265289,158046,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,778198
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230930,1891325,1168972,352121,7071639
1990,1487536,233664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585073,2504700,2230722,1385100,460730,8175133
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
1699,4937,2017,,727,7681
1771,21863,3623,,,2847,2843
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
1830,151716,121821,9121,8246,5346,189111
1840,212710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,496115
1860,813669,279122,12903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1800993,1166580,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2852112,2230932,1891325,1168972,352121,7071639
1930,1867312,2560401,1079129,1265239,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,778198
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2330932,1891325,1168972,352121,7071639
1990,1487536,233664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585073,2504700,2230722,1385100,460730,8175133
2015,16444518,2636735,2339150,1455444,474558,8550405

- 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
1699,4937,2017,,727,7681
1771,21863,3623,,,2847,2843
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
1830,148712,125401,9831,10965,391114
1840,212710,47613,14480,5346,10965,391114
1850,515547,138882,18593,8032,15061,496115
1860,813669,279122,12903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1860093,1166580,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2802000,2180000,3200000,380000,500000,600000
1930,1867312,2560401,1079129,1265239,158346,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,778198
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2330930,1891325,1168972,352121,7071639
1990,1487536,233664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585073,2504700,2230722,1385100,460730,8175133
2015,16444518,2636735,2339150,1455444,474558,8550405

- 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

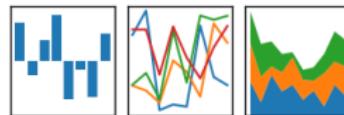
Source: https://en.wikipedia.org/wiki/Demographics_of_New_York_City,.....
All population figures are consistent with present-day boundaries.....
First census after the consolidation of the five boroughs.....
.....
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1699,4937,2017,,727,7681
1771,21863,3623,,,2847,2843
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
1830,148712,128882,10593,8032,15061,496115
1840,212710,47613,14480,5346,10965,391114
1850,515547,138882,10593,8032,15061,496115
1860,813669,279122,12903,23593,25492,1174779
1870,942292,419921,45468,37393,33029,1478103
1880,1164673,599495,56559,51980,38991,1911698
1890,1441216,838547,87050,88908,51693,2507414
1900,1860093,1166580,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2802641,2230930,318978,443728,7071639,58148
1930,1867312,2560401,1079129,1265289,158046,6930446
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1960101,2738175,1550849,1451277,191555,7891957
1960,1698281,2627319,1809578,1424815,221991,778198
1970,1539233,2602012,1986473,1471701,295443,7894862
1980,1428285,2230930,1891325,1168972,352121,7071639
1990,1487536,233664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332650,443728,8008278
2010,1585073,2504700,2230722,1385100,460730,8175133
2015,16444518,2636735,2339150,1455444,474558,8550405

- 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

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

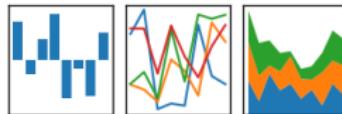


- We will use the popular Python Data Analysis Library (**Pandas**).

Structured Data

pandas

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

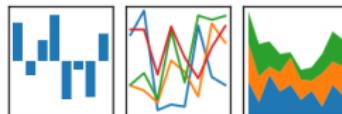


- We will use the popular Python Data Analysis Library (**Pandas**).
- Open source and freely available (part of anaconda distribution).

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
1782,23131,3623,1111,27,49447
1800,60515,5740,6642,1785,4563,79215
1810,96373,8202,7444,2267,5347,119734
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
1860,813669,279122,32803,23593,25492,1174105
1870,948154,249202,30095,37319,2019,1478103
1880,1144471,25095,5558,1990,3891,11999,11998
1890,1441216,818547,87050,88908,51691,2507414
1900,1850993,1166582,152999,200507,67021,3437202
1910,2331542,1634351,284041,430980,85969,4766883
1920,2284103,2018356,469042,732016,116531,5620048
1930,1867312,2560401,1079129,1265258,150346,693046
1940,1889924,2698285,1297634,1394711,174441,7454995
1950,1964001,2738175,1550849,1451277,191555,7891957
1960,1888981,2610219,1809578,1617115,221193,7781987
1970,1533981,2009013,1710111,147176,211111,7781982
1980,1428285,2230936,1891295,1160972,352121,7071639
1990,1487526,2300664,1951598,1203789,378977,7322564
2000,1537195,2465326,2229379,1332450,443728,8008278
2010,1505073,2504700,2230722,1385108,460730,8175133
2015,1644518,2636735,2339150,1455444,474558,8550403
```

- 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  
1771,21853,3623,,2847,28423  
1790,33111,52001,10111,11227,49447  
1800,60515,5740,6642,1785,4563,79215  
1810,96373,8202,7444,2267,5347,119734  
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  
1860,813669,279122,32803,23593,25492,1174105  
1870,948154,249201,30095,37319,21919,1478103  
1880,1144671,25095,5555,3990,3891,109198  
1890,1441216,818547,87050,88908,51591,2507414  
1900,1850993,1166582,153999,200507,67021,3437202  
1910,2331542,1634351,284041,430980,85969,4766683  
1920,2284103,2018356,469042,732016,116531,5620048  
1930,1867312,2560401,1079129,1265258,158346,6930446  
1940,1889924,2698285,1297634,1394711,174441,7454995  
1950,1966101,2738175,1550849,1451277,191555,7891957  
1960,1828981,2610159,1895978,16115,221954,781987  
1970,1553000,2000104,1711111,147178,4143,7781962  
1980,1428285,2230936,1891295,1160972,352211,7071639  
1990,1487536,2300664,1951298,1203789,378977,7322564  
2000,1537195,2465326,2229379,1332450,443728,8008278  
2010,1505073,2504700,2230722,1385108,460730,8175133  
2015,1644518,2636735,2339150,1455444,474558,8550403
```

- 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  
1771,21853,3623,,2847,28423  
1790,23131,3623,1111,1227,49447  
1800,60515,5740,6642,1785,4563,79215  
1810,96373,8202,7444,2267,5347,119734  
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  
1860,813669,279122,32803,23593,25492,1174799  
1870,940154,249221,30095,237319,21919,1478103  
1880,1144671,52095,55559,1990,38911,109198  
1890,1441216,818547,87050,88908,51692,2507414  
1900,1850993,1166582,152999,200507,67021,3437202  
1910,2331542,1634351,284041,430980,85969,4766683  
1920,2284103,2018356,469042,732016,116531,5620048  
1930,1867312,2560401,1079129,1265258,150346,6930446  
1940,1889924,2698285,1297634,1394711,174441,7454995  
1950,1966101,2738175,1550849,1451277,191555,7891957  
1960,1828981,2610159,1809578,16115,221954,781987  
1970,1553800,2000104,1711111,147178,143143,7781962  
1980,1428285,2230936,1891295,1160972,352211,7071639  
1990,1487536,2300664,1951298,1203789,378977,7322564  
2000,1537195,2465326,2229379,1332450,443728,8008278  
2010,1505073,2504700,2230722,1385108,460730,8175133  
2015,1644518,2636735,2339150,1455444,474558,8550400
```

- 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  
1771,21563,3623,,2847,28423  
1790,23314,3623,1111,1227,49447  
1800,60515,5740,6642,1785,4563,79215  
1810,96573,8202,7444,2267,5347,119734  
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  
1860,813669,279122,32803,23593,25492,1174779  
1870,948154,249223,30095,237319,23199,1478103  
1880,1164874,52995,55587,1990,38911,1199198  
1890,1441216,818547,87050,88908,516921,2507414  
1900,1850993,1166582,152999,200507,67021,3437202  
1910,2331542,1634351,284041,430980,85969,4766883  
1920,2284103,2018356,469042,732016,116531,5620048  
1930,1867312,2560401,1079129,1265258,1580346,6930446  
1940,1889924,2698285,1297634,1394711,174441,7454995  
1950,1964601,2738175,1550849,1451277,191555,7891957  
1960,1828981,2610519,1809578,161515,221954,7819867  
1970,1700000,2500000,1900000,1700000,2100000,7781962  
1980,1428285,2230936,1891295,1160972,352211,7071239  
1990,1487536,2300664,1951298,1203789,378977,7322564  
2000,1537195,2465326,2229379,1332450,443726,8008278  
2010,1505073,2504700,2230722,1385108,460730,8175133  
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

Sources: 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
1699, 4937, 2017, 727, 7648
1771, 21863, 36323, 2847, 28423
1790, 33131, 4549, 6159, 1781, 3827, 48447
1800, 33131, 4549, 6159, 1781, 3827, 48447
1810, 96371, 8103, 7444, 2267, 5347, 119734
1820, 123706, 11187, 8246, 2702, 6135, 15205
1830, 202548, 20535, 9049, 3023, 7082, 242278
1840, 244216, 218852, 108852, 18933, 8032, 15681, 694114
1850, 515547, 138882, 18593, 8032, 15681, 694115
1860, 813649, 279122, 32903, 23593, 25492, 1174779
1870, 942649, 419921, 45468, 37393, 33029, 1479182
1880, 1144721, 510821, 50082, 40082, 35082, 1699198
1890, 1441216, 838547, 87050, 86909, 51693, 2507414
1900, 1850939, 1166582, 152999, 200507, 67021, 3437202
1910, 2331542, 1634351, 284041, 430986, 8589, 4746883
1920, 2842161, 218882, 232903, 23593, 25492, 1174779
1930, 1867112, 2560401, 1079129, 1265258, 158246, 6930446
1940, 1884924, 2698285, 1297634, 1394711, 174443, 7454995
1950, 1885101, 2738103, 158050, 145171, 131055, 761957
1960, 1898233, 2409338, 158050, 145171, 131055, 761946
1970, 1539233, 2402012, 1986473, 1471701, 235443, 7894862
1980, 1428285, 2230398, 1891328, 1168972, 352121, 7071639
1990, 1371195, 218882, 2329378, 1532650, 443728, 8008484
2000, 1537195, 2465324, 2329378, 1532650, 443728, 8008278
2010, 1585473, 2504700, 2230722, 1385108, 468730, 8175133
2015, 1644518, 2636735, 2230159, 1455444, 474558, 8550405
```

nycHistPop.csv

Your Turn: Reading in CSV Files

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

Sources: 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  
1690, 4937, 2017, 727, 7648  
1771, 21863, 36323, 2847, 28423  
1790, 33131, 4549, 6159, 1781, 3827, 48447  
1800, 35131, 4882, 7000, 2232, 79215  
1810, 96371, 8103, 7444, 2267, 5347, 119734  
1820, 123706, 11187, 8246, 2702, 6135, 15205  
1830, 202506, 20535, 9049, 3022, 7082, 242278  
1840, 244706, 24082, 10341, 21301, 7082, 35914  
1850, 515547, 138882, 18593, 8032, 15861, 698115  
1860, 813649, 279122, 32903, 23593, 25492, 1174779  
1870, 942649, 419921, 45468, 37393, 33299, 1478152  
1880, 1144473, 519921, 53299, 41392, 35299, 1799198  
1890, 1441216, 838547, 87050, 86990, 51693, 2507414  
1900, 1850939, 1166582, 152999, 200507, 67621, 3437202  
1910, 2331542, 1634351, 284041, 430986, 8589, 4746883  
1920, 2850000, 2180000, 3000000, 3000000, 3000000, 10000000  
1930, 1867112, 2560401, 1079129, 1265258, 158246, 6930446  
1940, 1884924, 2698285, 1297634, 1394711, 174443, 7454995  
1950, 1951101, 2738103, 158050, 145157, 131055, 761957  
1960, 1898233, 2402012, 1986473, 1358578, 148121, 781984  
1970, 1539233, 2402012, 1986473, 135443, 7894862  
1980, 1428285, 2230398, 1891328, 1168972, 352121, 7071639  
1990, 1371185, 2180700, 2230722, 1385108, 447728, 8008484  
2000, 1537135, 2445324, 2239378, 1332450, 443728, 8028278  
2010, 1585473, 2504700, 2230722, 1385108, 448730, 8175133  
2015, 1644518, 2636735, 2230159, 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)
```

Sources: 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
1699, 4937, 2017, 727, 7648
1771, 21863, 36323, 2847, 28423
1790, 33131, 4549, 6159, 1781, 3827, 48447
1800, 33131, 4549, 6159, 1781, 3827, 48447
1810, 96371, 8103, 17444, 2267, 5347, 119734
1820, 123706, 11187, 8246, 2702, 6135, 152058
1830, 202504, 20535, 9049, 3023, 7082, 242278
1840, 233142, 20535, 9049, 3023, 7082, 242278
1850, 515547, 138882, 18593, 8032, 15861, 694115
1860, 813649, 279122, 32903, 23593, 25492, 1174779
1870, 942649, 419921, 85468, 37393, 33029, 1478152
1880, 1144737, 510000, 100000, 40000, 40000, 197998
1890, 1441216, 838547, 87050, 86990, 51693, 2507414
1900, 185093, 1166582, 152999, 200507, 67621, 3437202
1910, 2331542, 1634351, 284041, 430986, 8589, 4746883
1920, 2831112, 2560401, 1107129, 1265258, 158246, 6930446
1930, 1867112, 2560401, 1107129, 1265258, 158246, 6930446
1940, 1889294, 2698285, 1297634, 1394711, 174443, 7454995
1950, 1981101, 2738103, 158050, 145177, 131055, 761957
1960, 1698233, 2402012, 1986473, 1471701, 1328181, 781946
1970, 1539233, 2402012, 1986473, 1395443, 7894862
1980, 1428285, 2230936, 1891328, 1168972, 1352121, 7071639
1990, 1337135, 2145324, 2239378, 1332450, 1443728, 8008484
2000, 1237135, 2145324, 2239378, 1332450, 1443728, 8008278
2010, 1585473, 2304700, 2230722, 1385108, 468730, 8175133
2015, 1644518, 2363735, 2230159, 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)

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

Sources: 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.....First census.....Year..... Manhattan, Brooklyn, Queens, Bronx, Staten Island, Total
1699, 4937, 2017, , 727, 7648
1771, 21863, 36323, , 2847, 28423
1790, 33131, 4549, 6159, 1781, 3627, 45447
1800, 36131, 5049, 6159, 1781, 3627, 47925
1810, 96371, 8130, 17446, 2267, 5347, 119734
1820, 123706, 11187, 8246, 2702, 6135, 15205
1830, 202546, 20535, 9049, 3023, 7082, 242278
1840, 233142, 163435, 10892, 3023, 7082, 242278
1850, 515547, 138882, 18593, 8032, 15681, 698115
1860, 813649, 279122, 32903, 23593, 25492, 1174779
1870, 942649, 419921, 85468, 37393, 33029, 1479182
1880, 1144737, 510921, 10890, 47812, 33029, 1799198
1890, 1441216, 838547, 87050, 86990, 51693, 2507414
1900, 1850939, 1166582, 152999, 200507, 67621, 3437202
1910, 2331542, 1634351, 284041, 430986, 8589, 4746883
1920, 2831542, 1634351, 284041, 430986, 8589, 4746883
1930, 1867112, 2560401, 1079129, 1265258, 158246, 6930446
1940, 1884924, 2698285, 1297634, 1394711, 174443, 7454995
1950, 1885101, 2738105, 1589233, 1451571, 191055, 760957
1960, 1898233, 2698285, 1318333, 1578578, 148121, 781946
1970, 1539233, 2402012, 1986473, 1471701, 235443, 7894862
1980, 1428285, 2230936, 1891328, 1168972, 352121, 7071639
1990, 1371185, 2184732, 2239378, 1332650, 443728, 8008484
2000, 1371185, 2465324, 2239378, 1332650, 443728, 8008278
2010, 1585873, 2504700, 2230722, 1385108, 468730, 8175133
2015, 1644518, 2636735, 2230159, 1455444, 474558, 8550405
```

nycHistPop.csv

Your Turn: Reading in CSV Files

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

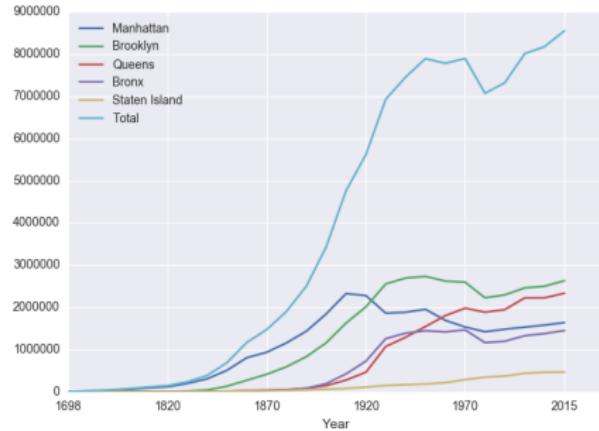
```
pop = pd.read_csv('nycHistPop.csv', skiprows=5)
```

Sources: 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

Year	Manhattan	Brooklyn	Queens	Bronx	Staten Island	Total
1690	4937	2017	..	727	7681	1690
1771	21863	3823	..	2847	28423	1771
1790	33131	4549	6159	1781	3827	48447
1800	36200	5000	6159	1781	79215	48250
1810	96371	8103	17444	2267	5347	119734
1820	123706	11187	8246	2702	6135	152058
1830	202546	20535	9049	3023	7082	242278
1840	260000	210000	100000	30000	100000	570000
1850	515547	138882	18593	8032	15661	694115
1860	813649	279122	32903	23593	25492	1174779
1870	940000	419921	45468	37393	33029	1479182
1880	1144473	510000	510000	410000	370000	2500000
1890	1441216	838547	87050	86909	51693	2507414
1900	185093	1166582	152999	200507	67621	3473202
1910	2331542	1634351	284041	430986	8569	4746883
1920	2800000	2000000	2000000	2000000	2000000	4000000
1930	1867112	2560401	1079129	1265258	158246	6930446
1940	188924	2698285	1297634	1394711	174443	7454995
1950	1951101	2738036	158050	145157	171055	7697957
1960	1898200	2800000	2130318	1881578	1881578	781984
1970	1539233	2402012	1986473	1471701	239443	7894862
1980	1428285	2230936	1891328	1168972	2352121	7071639
1990	1371185	2445324	2239378	1881481	2239378	7044844
2000	1371185	2445324	2239378	1881481	2239378	8082878
2010	1585473	2304700	2230722	1385108	4487730	8175133
2015	1644518	2636735	2130159	1455444	4745598	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)
```

Sources: 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.....

.....

..... Manhattan, Brooklyn, Queens, Bronx, Staten Island, Total

1690, 4937, 2017, , 727, 7648

1771, 21863, 36323, , 2847, 28423

1790, 33131, 4549, 6159, 1781, 3627, 48447

1800, 40000, 6000, 10000, 2000, 79215

1810, 96371, 8103, 17444, 2267, 5347, 119734

1820, 123706, 11187, 8246, 2702, 6135, 15205

1830, 202546, 20535, 9049, 3023, 7082, 242278

1840, 300000, 30000, 60000, 120000, 30000, 1114

1850, 515547, 138882, 18593, 8032, 15861, 69415

1860, 813649, 279122, 32903, 23593, 25492, 1174779

1870, 940000, 419921, 85468, 37393, 33229, 1478125

1880, 1144473, 500000, 100000, 200000, 300000, 1998

1890, 1441216, 838547, 87050, 89909, 51693, 2507414

1900, 185093, 1166582, 152999, 200507, 67621, 3437202

1910, 2331542, 1634351, 284041, 430986, 85869, 4746883

1920, 2850000, 2000000, 1500000, 1000000, 1000000, 1000000

1930, 1867112, 2560401, 1079129, 1265258, 158246, 6930446

1940, 188924, 2698285, 1297634, 1394711, 174443, 7454995

1950, 210101, 273803, 158050, 145157, 171955, 76957

1960, 268923, 2400313, 1889318, 1881818, 1881818, 783946

1970, 1539233, 2402012, 1986473, 1471701, 2395443, 7894862

1980, 1428285, 2230936, 1891328, 1168972, 1352121, 7071639

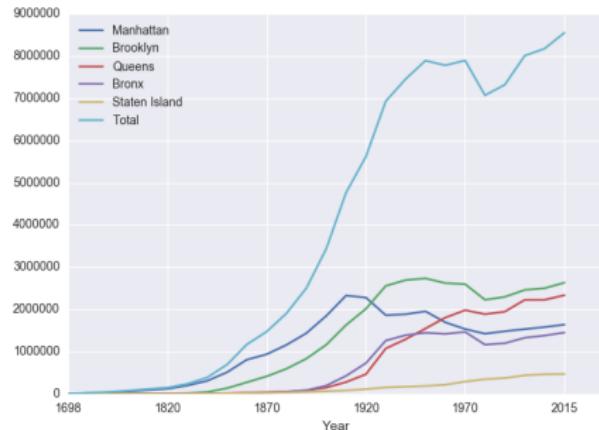
1990, 1371185, 2145324, 2239378, 1385108, 1468736, 684844

2000, 1537185, 2445324, 2239378, 1385108, 1468736, 8008278

2010, 1585473, 2304700, 2230722, 1385108, 1468736, 8175133

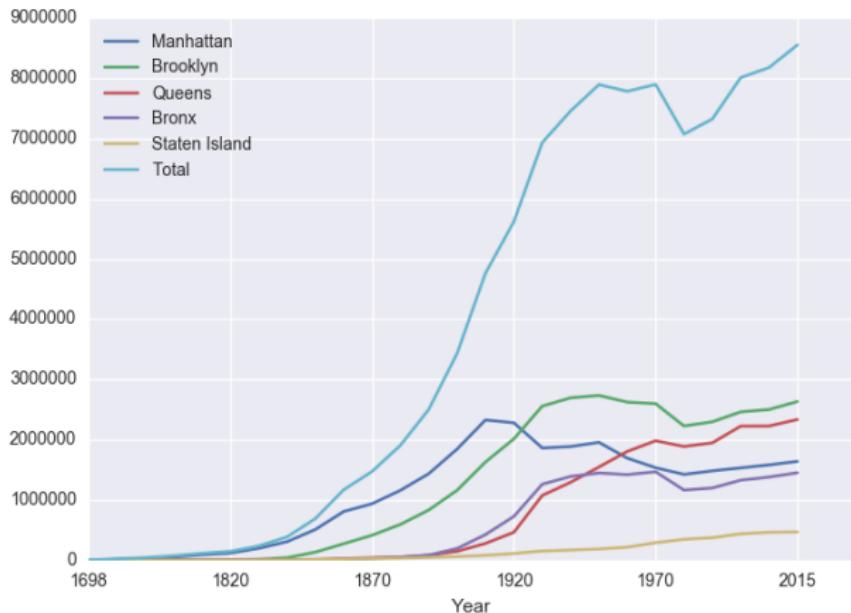
2015, 1644518, 2636735, 2130159, 1455444, 4745598, 8550405

nycHistPop.csv

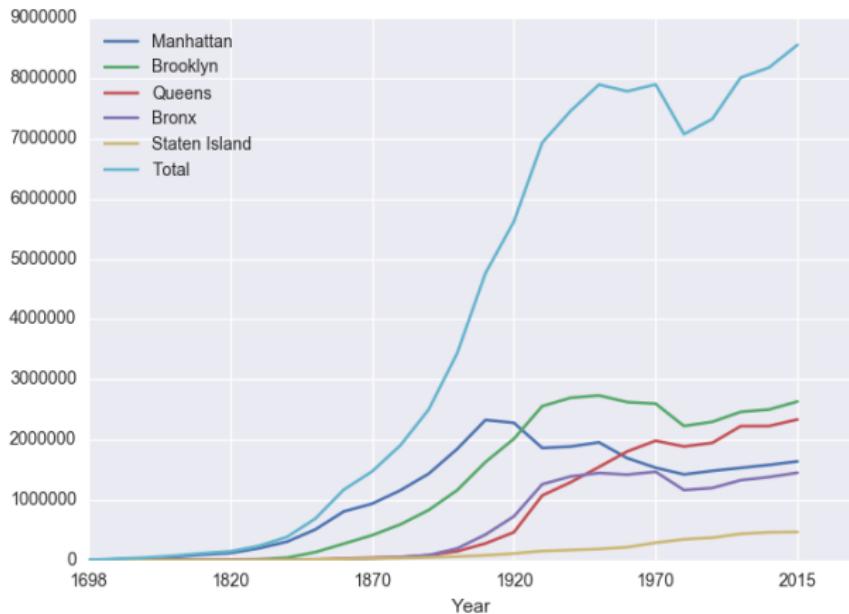


(To get above color scheme, import seaborn.)

Nice Graph, But Doesn't Answer the Question

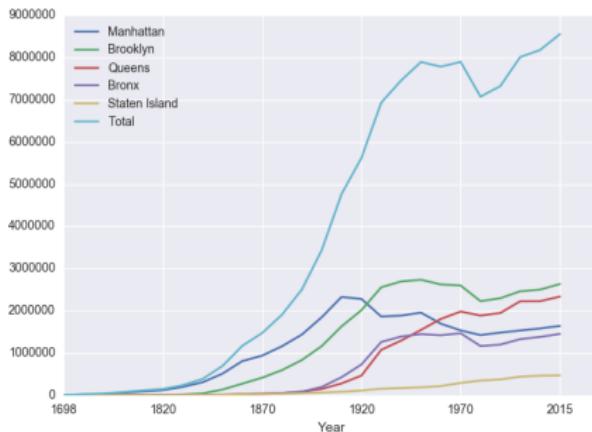


Nice Graph, But Doesn't Answer the Question



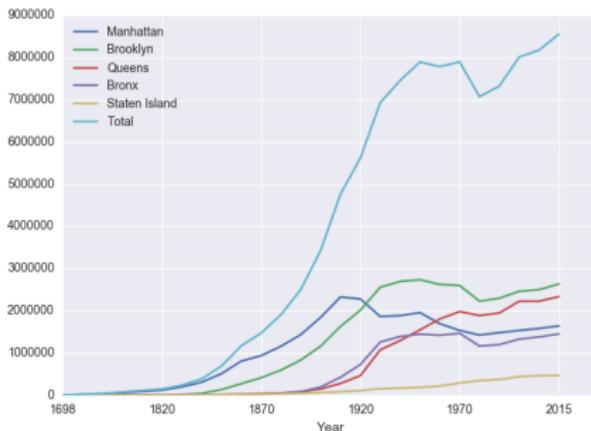
Let's survey what else you can do with pandas....

Series in Pandas



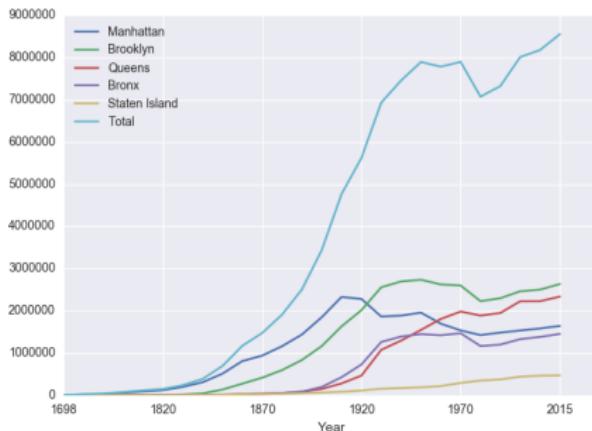
- Series can store a column or row of a DataFrame.

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.

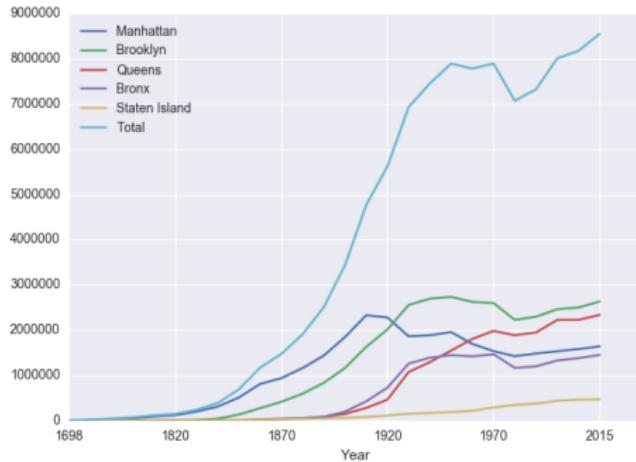
Series in Pandas



- 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",\n      pop["Bronx"].max())
```

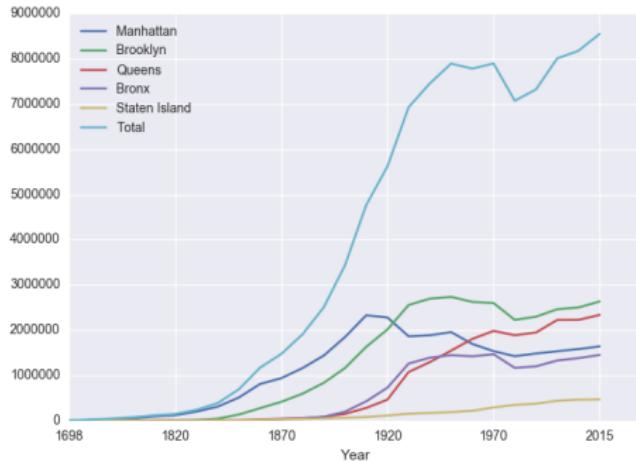
In Groups



Predict what the following will do:

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

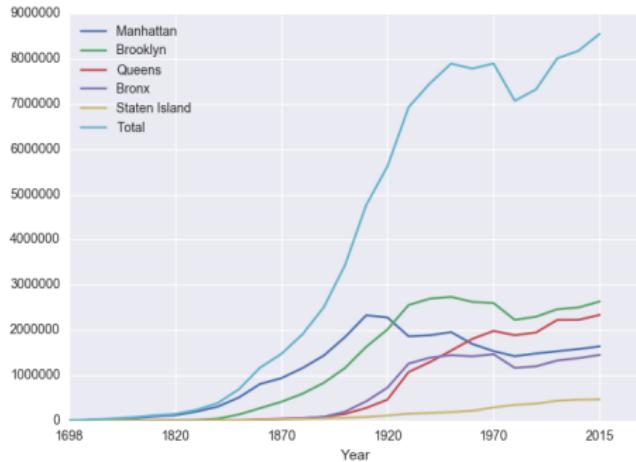
In Groups



Predict what the following will do:

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

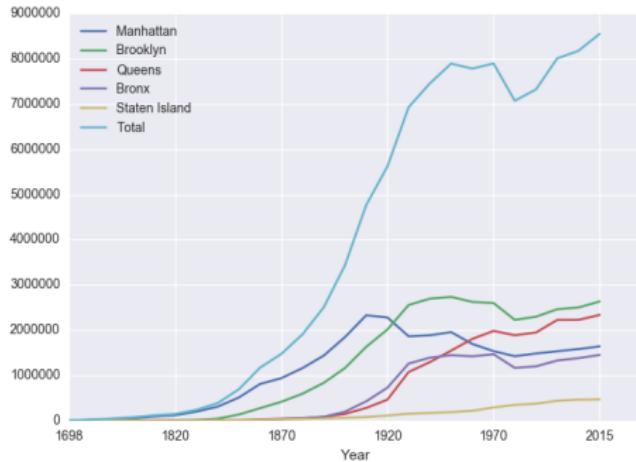
In Groups



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())`

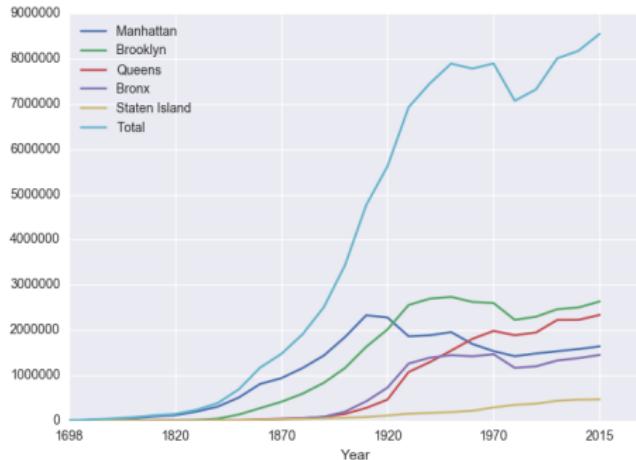
In Groups



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")`

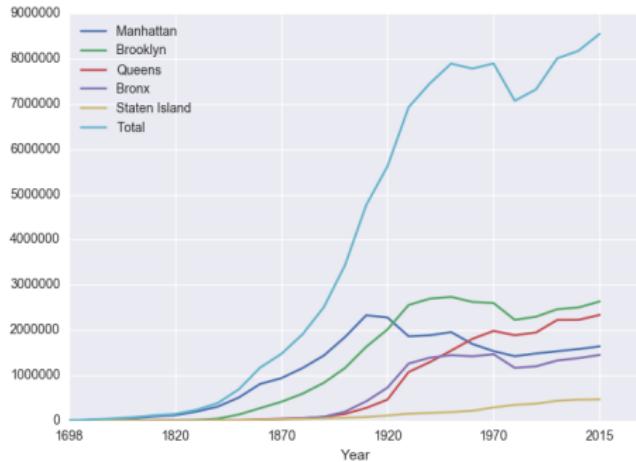
In Groups



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



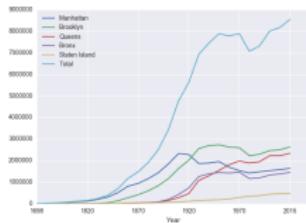
Predict what the following will do:

- `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:

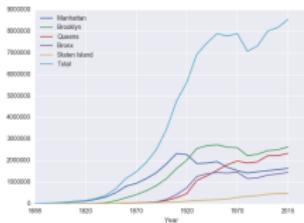
- `print("Queens:", pop["Queens"].min())`



Solutions

Predict what the following will do:

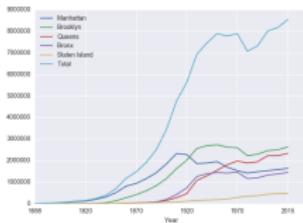
- `print("Queens:", pop["Queens"].min())`
Minimum value in the column with label "Queens".



Solutions

Predict what the following will do:

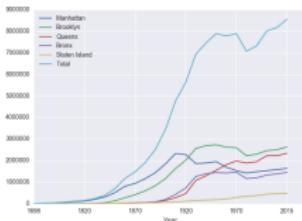
- `print("Queens:", pop["Queens"].min())`
Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`



Solutions

Predict what the following will do:

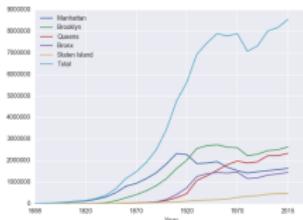
- `print("Queens:", pop["Queens"].min())`
Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`
Average of values in the column "Staten Island".



Solutions

Predict what the following will do:

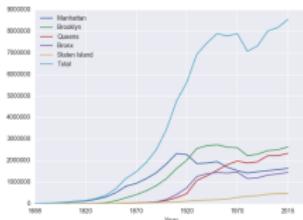
- `print("Queens:", pop["Queens"].min())`
Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`
Average of values in the column "Staten Island".
- `print("S I :", pop["Staten Island"].std())`



Solutions

Predict what the following will do:

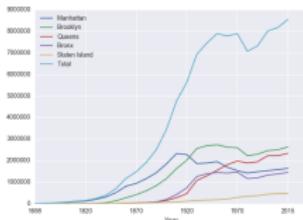
- `print("Queens:", pop["Queens"].min())`
Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`
Average of values in the column "Staten Island".
- `print("S I :", pop["Staten Island"].std())`
Standard deviation of values in the column "Staten Island".



Solutions

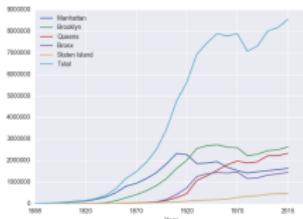
Predict what the following will do:

- `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

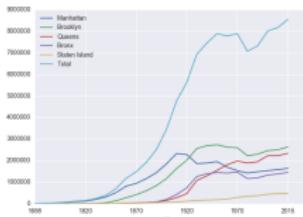
Predict what the following will do:



- `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

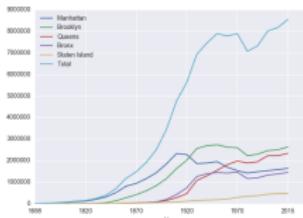
Predict what the following will do:



- `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

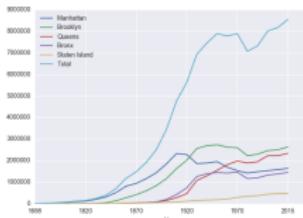
Predict what the following will do:



- `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

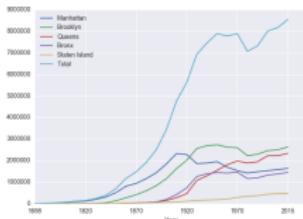
Predict what the following will do:



- `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:



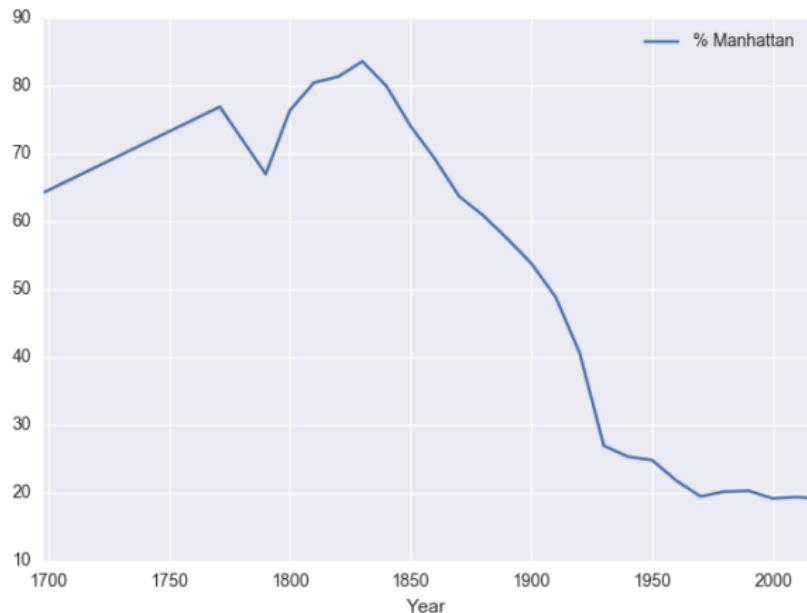
- `print("Queens:", pop["Queens"].min())`
Minimum value in the column with label "Queens".
- `print("S I:", pop["Staten Island"].mean())`
Average of values in the column "Staten Island".
- `print("S I :", pop["Staten Island"].std())`
Standard deviation of values in the column "Staten Island".
- `pop.plot.bar(x="Year")`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

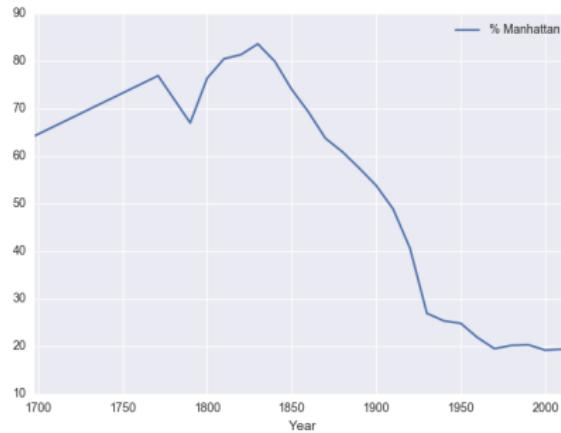
Graph the percentage of the total population that live in Manhattan:

In Groups

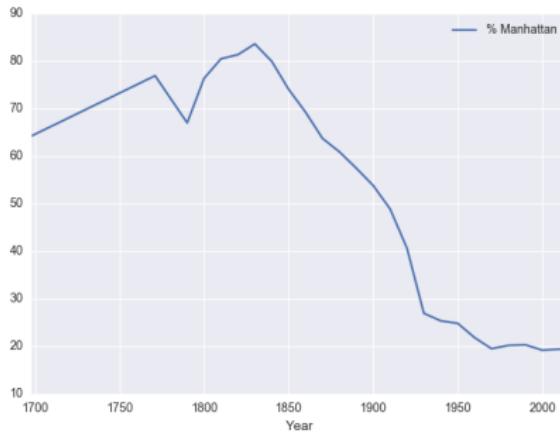
Graph the percentage of the total population that live in Manhattan:



Percent of Population in Manhattan, Over Time

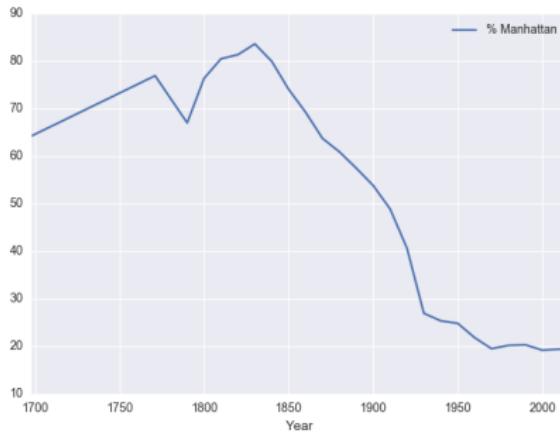


Percent of Population in Manhattan, Over Time



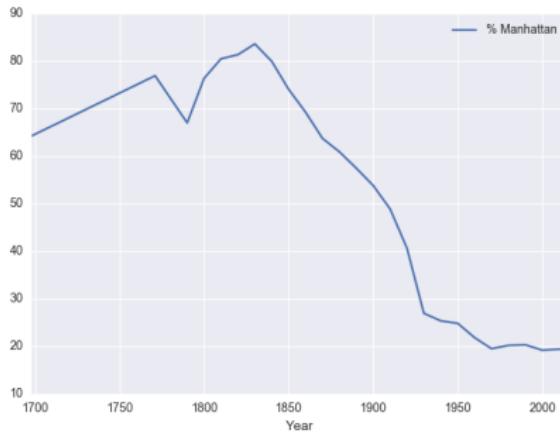
```
import matplotlib.pyplot as plt  
import pandas as pd
```

Percent of Population in Manhattan, Over Time



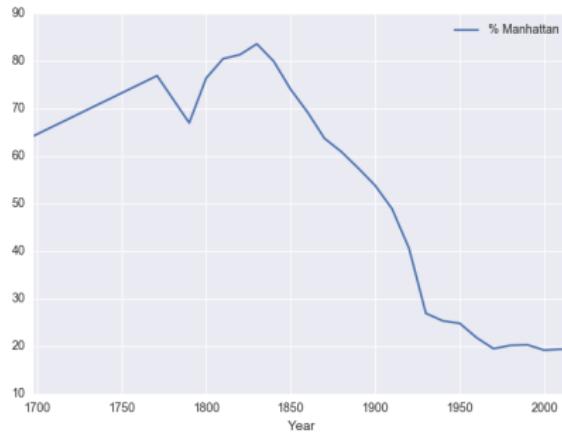
```
import matplotlib.pyplot as plt  
import pandas as pd  
import seaborn
```

Percent of Population in Manhattan, Over Time



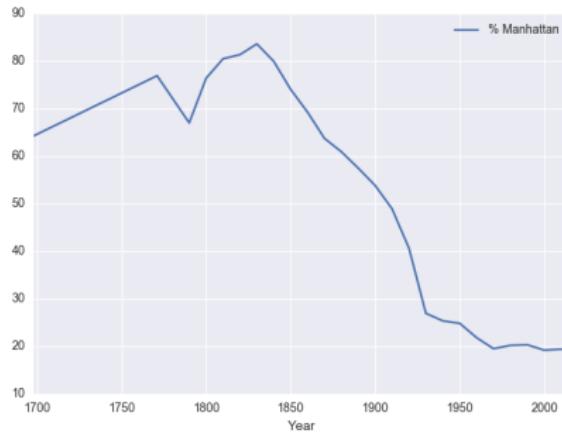
```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn
pop = pd.read_csv('nycHistPop.csv', skiprows=5)
```

Percent of Population in Manhattan, Over Time



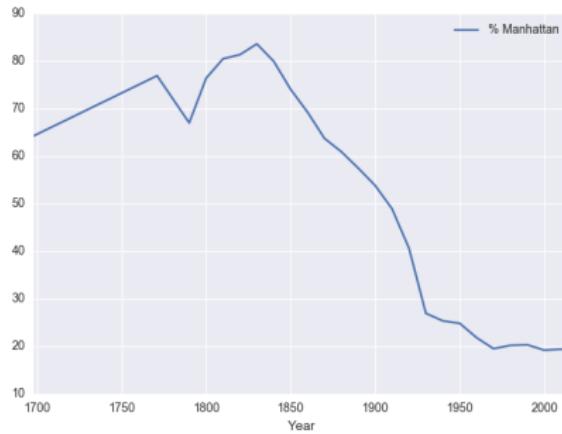
```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn
pop = pd.read_csv('nycHistPop.csv', skiprows=5)
pop['% Manhattan'] = 100*pop['Manhattan']/pop['Total']
```

Percent of Population in Manhattan, Over Time



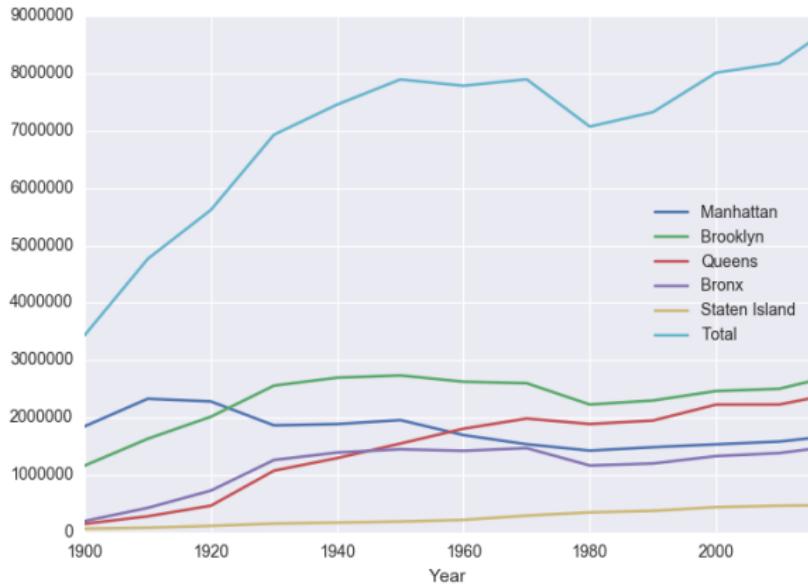
```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn
pop = pd.read_csv('nycHistPop.csv', skiprows=5)
pop['% Manhattan'] = 100*pop['Manhattan']/pop['Total']
pop.plot(x="Year",y="% Manhattan")
```

Percent of Population in Manhattan, Over Time

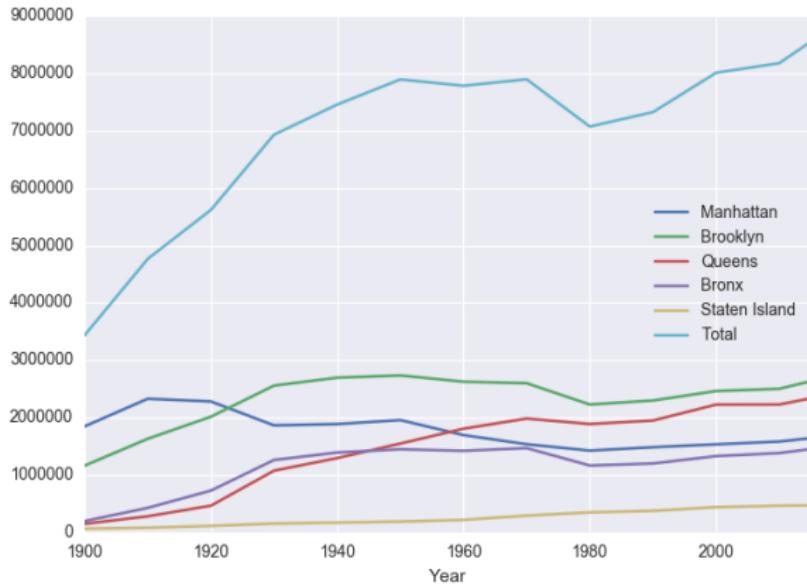


```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn
pop = pd.read_csv('nycHistPop.csv', skiprows=5)
pop['% Manhattan'] = 100*pop['Manhattan']/pop['Total']
pop.plot(x="Year",y="% Manhattan")
plt.show()
```

Population Since 1900

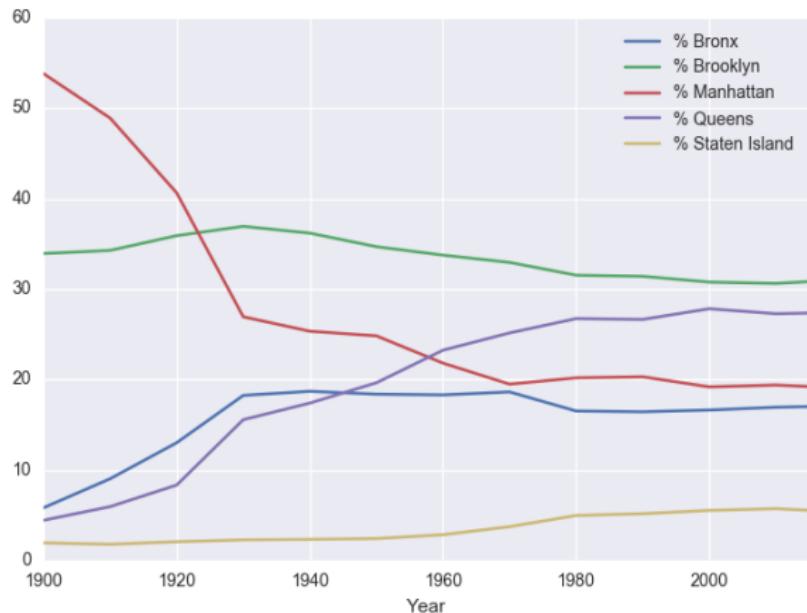


Population Since 1900

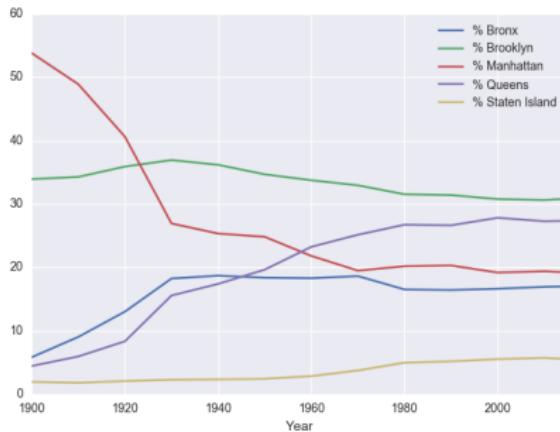


```
pop1900 = pop[pop['Year'] >= 1900]
pop1900.plot(x="Year")
plt.show()
```

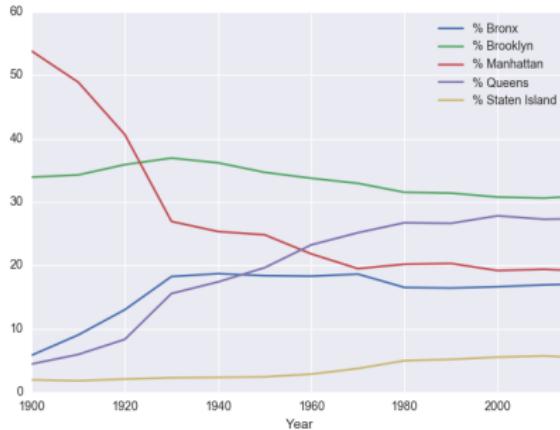
Population Percentage by Borough



Population Percentage by Borough

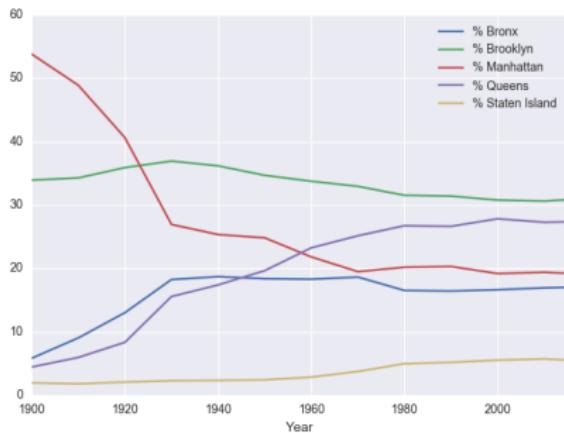


Population Percentage by Borough



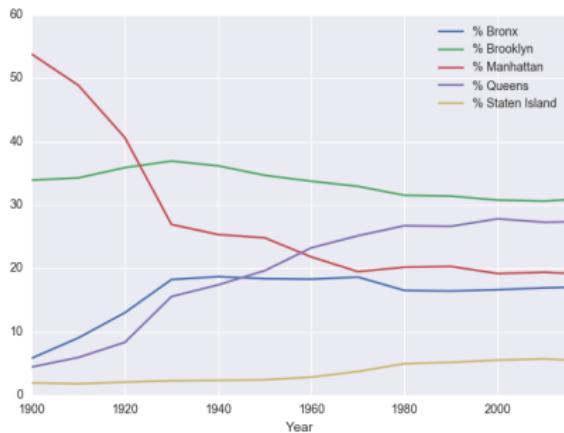
```
boros = ["Bronx", "Brooklyn", "Manhattan", "Queens", "Staten Island"]
```

Population Percentage by Borough



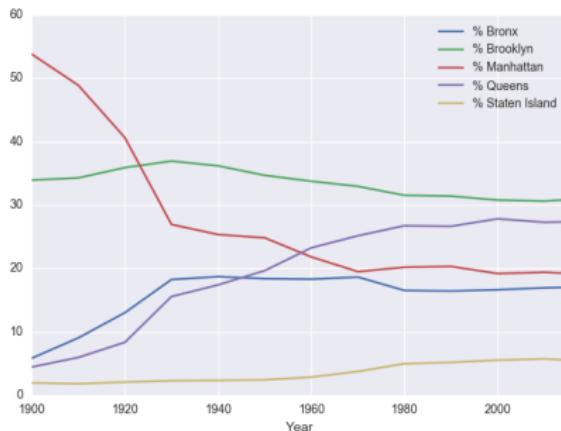
```
boros = ["Bronx", "Brooklyn", "Manhattan", "Queens", "Staten Island"]  
percentCol = [% "+ boro for boro in boros] #List comprehension
```

Population Percentage by Borough



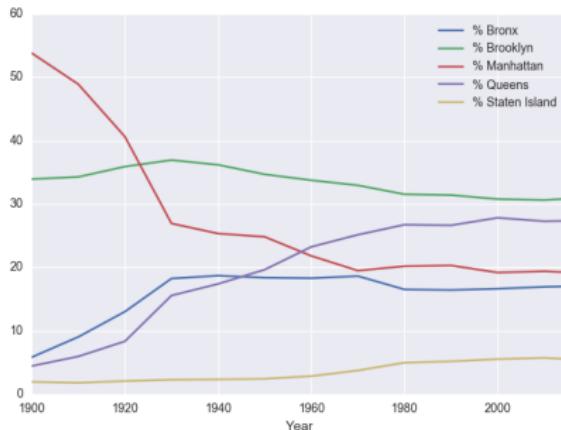
```
boros = ["Bronx", "Brooklyn", "Manhattan", "Queens", "Staten Island"]
percentCol = [% "+ boro for boro in boros] #List comprehension
for boro in boros:
```

Population Percentage by Borough



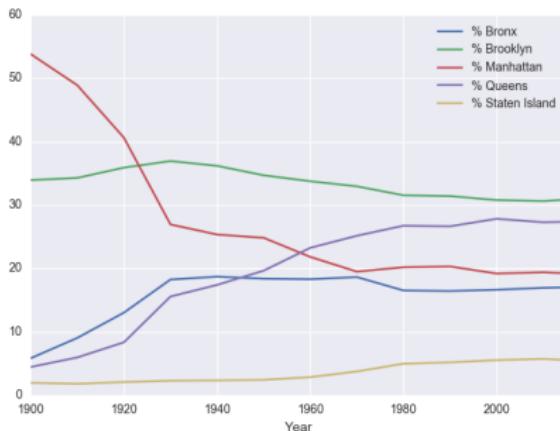
```
boros = ["Bronx", "Brooklyn", "Manhattan", "Queens", "Staten Island"]
percentCol = ["%" + boro for boro in boros] #List comprehension
for boro in boros:
    print('Computing percentage for', boro)
    pop["%" + boro] = 100*pop2[boro]/pop["Total"]
```

Population Percentage by Borough



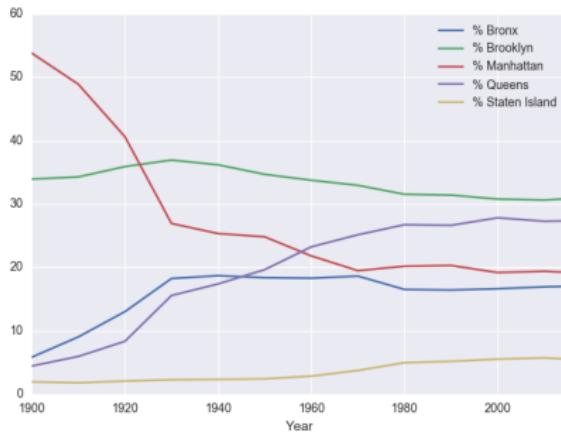
```
boros = ["Bronx", "Brooklyn", "Manhattan", "Queens", "Staten Island"]
percentCol = ["%" + boro for boro in boros] #List comprehension
for boro in boros:
    print('Computing percentage for', boro)
    pop["%" + boro] = 100*pop2[boro]/pop["Total"]
pop = pop.drop(boros, axis = 1)
```

Population Percentage by Borough



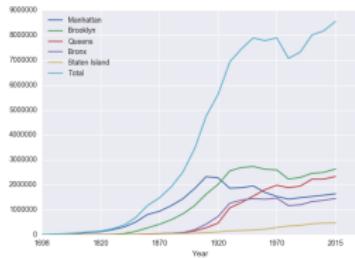
```
boros = ["Bronx", "Brooklyn", "Manhattan", "Queens", "Staten Island"]
percentCol = ["%" + boro for boro in boros] #List comprehension
for boro in boros:
    print('Computing percentage for', boro)
    pop["%" + boro] = 100*pop2[boro]/pop["Total"]
pop = pop.drop(boros, axis = 1)
pop = pop.drop("Total",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

Population Percentage by Borough



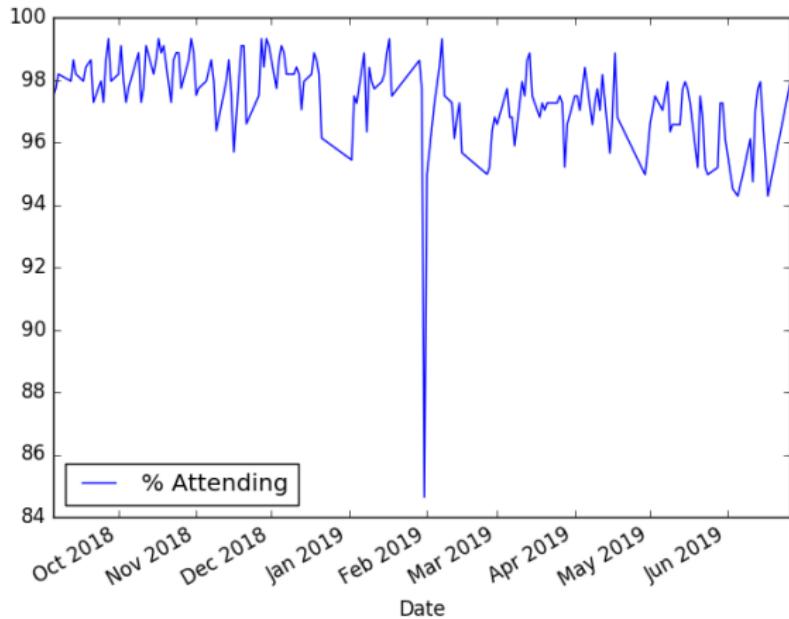
```
boros = ["Bronx", "Brooklyn", "Manhattan", "Queens", "Staten Island"]
percentCol = ["%" + boro for boro in boros] #List comprehension
for boro in boros:
    print('Computing percentage for', boro)
    pop[%" + boro] = 100*pop2[boro]/pop["Total"]
pop = pop.drop(boros, axis = 1)
pop = pop.drop("Total",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

Open Data for All New Yorkers

Where can you find public Wi-Fi in your neighborhood?
What kind of tree is in front of your office? Learn about
where you live, work, eat, shop and play using NYC Open
Data.

Search Open Data for things like 311, Buildings, Crime



- Freely available source of data.

Accessing Structured Data: NYC Open Data

Open Data for All New Yorkers

Where can you find public Wi-Fi in your neighborhood?
What kind of tree is in front of your office? Learn about where you live, work, eat, shop and play using NYC Open Data.

Search Open Data for things like 311, Buildings, Crime

The slide features a blue header with white text. Below the header is a white search bar containing placeholder text. To the right of the search bar is a cluster of colorful speech bubbles containing icons related to city life and data. At the bottom right are four stylized human figures representing different ethnicities and gender identities.

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

Accessing Structured Data: NYC Open Data

Open Data for All New Yorkers

Where can you find public Wi-Fi in your neighborhood?
What kind of tree is in front of your office? Learn about where you live, work, eat, shop and play using NYC Open Data.

Search Open Data for things like 311, Buildings, Crime

- 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

Open Data for All New Yorkers

Where can you find public Wi-Fi in your neighborhood?
What kind of tree is in front of your office? Learn about where you live, work, eat, shop and play using NYC Open Data.

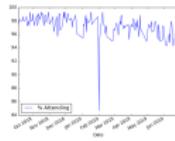
Search Open Data for things like 311, Buildings, Crime

The slide features a blue header with white text. Below the header is a white rectangular search bar containing the text "Search Open Data for things like 311, Buildings, Crime". To the right of the search bar is a cluster of colorful speech bubbles containing icons related to various city services and data points. At the bottom right is a row of four stylized human figures representing different demographic groups.

- 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.



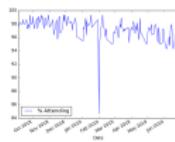
Variations on the Theme: School Attendance

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



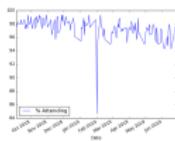
Variations on the Theme: School Attendance

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

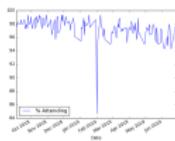


Variations on the Theme: School Attendance

- 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”).

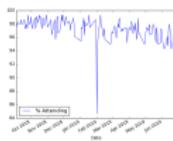


Variations on the Theme: School Attendance



- 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:

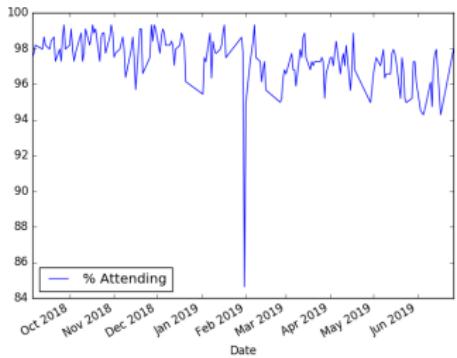
Variations on the Theme: School Attendance



- 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).

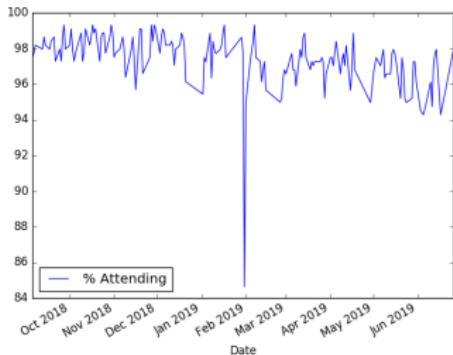
Variations on the Theme: School Attendance



C

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

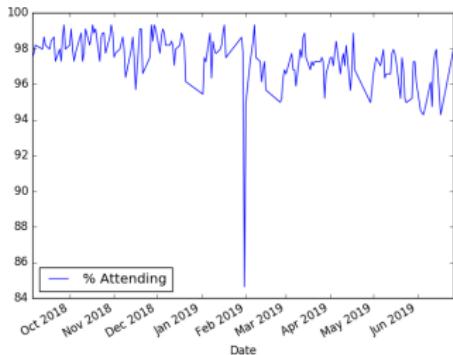
Variations on the Theme: School Attendance



C

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

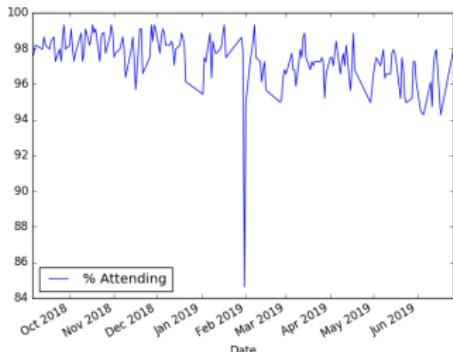
Variations on the Theme: School Attendance



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

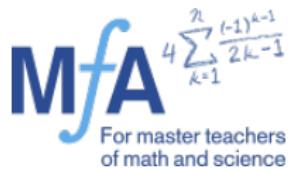


C

```
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")
```

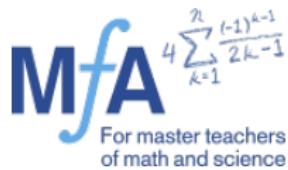
Design a Challenge

- Find an interesting data set for a challenge.

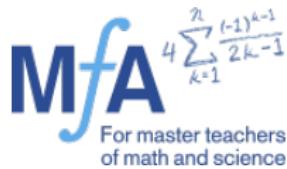


Design a Challenge

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

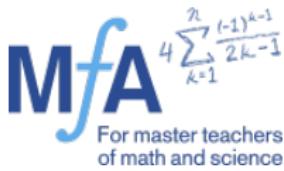


Design a Challenge



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

Design a Challenge



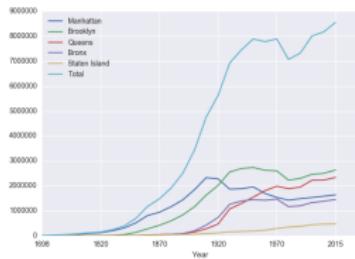
- Find an interesting data set for a challenge.
- Suggested places to look:
 - ▶ NYC OpenData.
 - ▶ Weather Underground
(for historical weather data).

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

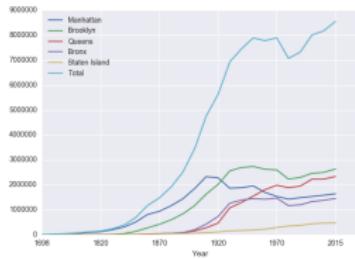


- Recap
- Design Challenge: NYC Population
- Variations on the Theme
- Design a Challenge
- Break
- Design Challenge: Parking Tickets
- Variations on the Theme
- Design a Challenge
- Wrap Up

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... ID	Plate... Number	Regis... State	Plate... Type	Issue... Date	Viola... Code	Vehic... Type	Vehic... Model	Issui... Status	Stree... Address	Stree... Address	Stree... Address	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	202
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](#)

Sanc...	I	Plate...	I	Regis...	I	Plat...	I	Issue...	I	Vehic...	I	Vehicle...	I	Insu...	I	Strea...	I	Street...	I	Strea...	I	Whe...
14471125...	175661	NY	FNS	06/28/2019	21	SON	BMW	P	27990	36290	30500	20										
14471126...	IVH5222	NY	FNS	06/28/2019	20	SON	TENST	P	36290	27990	12113	20										
14471126...	0AB9898	NY	FNS	06/28/2019	19	SUH	BMW	P	36270	11710	27990	20										
14471126...	0EF1947	NY	FNS	06/28/2019	19	SON	AUDI	P	36270	11710	27990	20										
14471127...	G0B8863	NY	CDM	07/05/2019	48	SUH		P	21160	36290	36290	20										
14471128...	HWB0432	NY	FNS	06/14/2019	48	SUH	MBSA	P	36270	11710	27990	20										
14471129...	YAH5470	NY	FNS	06/14/2019	46	SUH	TENST	P	36290	11710	27990	20										
14471130...	GWV9946	NY	FNS	06/14/2019	46	SUH	HENDR	P	36270	11710	27990	20										
14471130...	HWB7769	NY	FNS	06/28/2019	48	SUH	TENST	P	36290	11710	27990	20										
14471131...	03H2184	ME	FNS	05/06/2019	48	SON	DODGE	P	31190	40404	40404	20										
14471132...	EA5331	NY	FNS	05/10/2019	46	SON	ACURA	P	36270	11710	27990	20										
14471132...	JWV0808	NY	FNS	06/16/2019	48	SON	HENDR	P	36270	11710	27990	20										
14471133...	HQD954	NY	FNS	06/16/2019	46	ACURA	P	36270	11710	27990	20											
14471134...	GWV1545	NY	FNS	06/15/2019	19	SUH	MBSA	P	36270	11710	27990	20										

[< Previous](#) [Next >](#) Showing Rows 1-14 out of 4,984,034

Parking Ticket Data Format

Table Preview

View Data Create Visualization

Sanc._I	Plate._I	Regis._I	Platc._I	Issue._I	Vehc._I	Vehic._I	Insu._I	Insr._I	Street._I	Street._I	Wkt.	
14471125_	BT56861	NY	FNS	06/28/2019	21	SUN	BMW	P	27990	36290	30500	20
14471124_	IVH5222	NY	FNS	06/28/2019	20	SUN	TONST	P	36290	27990	12113	20
14471126_	0AB9898	NY	FNS	06/28/2019	19	SUN	BMW	P	36270	11710	27990	20
14471125_	0EF1947	NY	FNS	06/24/2019	19	SUN	AUDR	P	36270	11710	27990	20
14471127_	G0H8663	NY	CDM	07/05/2019	19	SUN		P	21160	36290	36290	20
14471128_	HWB0442	NY	FNS	06/14/2019	46	SUN	MBSA	P	36270	11710	27990	20
14471125_	HYA5470	NY	FNS	06/14/2019	46	SUN	TONST	P	36290	11710	27990	20
14471126_	GWV9664	NY	FNS	06/14/2019	46	SUN	HENDR	P	36270	11710	27990	20
14471128_	HWB7769	NY	FNS	06/28/2019	46	SUN	TONST	P	36260	11710	27990	20
14471125_	0SD1784	ME	FNS	05/06/2019	46	SUN	DODGE	P	31190	40404	40404	20
14471126_	EAS331	NY	FNS	07/01/2019	46	SUN	ACURA	P	36270	11710	27990	20
14471127_	JWV8084	NY	FNS	06/16/2019	46	SUN	HENDR	P	36270	11710	27990	20
14471128_	HQD954	NY	FNS	06/16/2019	46	SUN	ACURA	P	36270	11710	27990	20
14471126_	GYH1545	NY	FNS	06/15/2019	19	SUN	MBSA	P	36270	11710	27990	20

< Previous Next >

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

Parking Ticket Data Format

Table Preview

View Data Create Visualization

Sanc..._I	Plate..._I	Regis..._I	Plat..._I	Issue..._I	Viola..._I	Vehic..._I	Vehicle..._I	Insu..._I	Strea..._I	Street..._I	Strea..._I	Whe...
1A471125...	8756881	NY	PMS	06/28/2019	21 SON	BMW	P	27990	36290	30350	20	
1A471124...	IVH9322	NY	PMS	06/28/2019	20 SON	TONST	P	26260	27990	12113	20	
1A471126...	0AB9898	NY	PMS	06/28/2019	19 SUM	BMW	P	26270	11710	27990	20	
1A471125...	0EF1947	NY	PMS	06/24/2019	19 SON	AUDR	P	26270	11710	27990	20	
1A471127...	G0B8863	NY	CDM	07/05/2019			P	21160	36290	36290	20	
1A471126...	HWB0432	NY	PMS	06/14/2019	46 SUM	MBSA	P	26270	11710	27990	20	
1A471125...	HAW5470	NY	PMS	06/14/2019	46 SUM	TONST	P	26260	11710	27990	20	
1A471126...	GWV9964	NY	PMS	06/14/2019	46 SUM	HENDR	P	26270	11710	27990	20	
1A471126...	HWB7769	NY	PMS	06/28/2019	46 SUM	TONST	P	26260	11710	27990	20	
1A471125...	0SD1784	ME	PMS	05/06/2019	46 SON	DODGE	P	31190	40404	40404	20	
1A471126...	EA5331	NY	PMS	05/10/2019	46 SON	ACURA	P	26270	27990	20		
1A471127...	JWW8884	NY	PMS	06/16/2019	46 SON	HENDR	P	26270	11710	27990	20	
1A471127...	HQD934	NY	PMS	06/16/2019	46	ACURA	P	26270	11710	27990	20	
1A471126...	GYH1545	NY	PMS	06/15/2019	19 SUM	MBSA	P	26270	11710	27990	20	

< 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

Sanc...	Plat...	Regis...	Plat...	Issue...	Viol...	Vehic...	Vehicle...	Insu...	Strea...	Street...	Strea...	Whe...
14471125...	BT59881	NY	FNS	06/28/2019	21	SUN	BMW	P	27990	36290	30350	20
14471124...	IVH9322	NY	FNS	06/28/2019	20	SUN	TONST	P	36290	27990	12113	20
14471126...	0AB9898	NY	FNS	06/28/2019	19	SUN	BMW	P	98770	11710	27990	20
14471125...	0EF1947	NY	FNS	06/24/2019	19	SUN	AUDR	P	98770	11710	27990	20
14471127...	G0B8863	NY	CDM	07/05/2019	48	SUND		P	21190	36290	36290	20
14471128...	HWB0432	NY	FNS	06/14/2019	48	SUND	MBSA	P	98770	11710	27990	20
14471125...	HAD5470	NY	FNS	06/14/2019	48	SUND	TONST	P	36290	11710	27990	20
14471126...	GWV9964	NY	FNS	06/14/2019	48	SUND	HENDR	P	36270	11710	27990	20
14471126...	HWB7769	NY	FNS	06/28/2019	48	SUND	TONST	P	98690	11710	27990	20
14471123...	0SD1784	ME	FNS	05/06/2019	48	SUN	DODGE	P	31190	40404	40404	20
14471126...	EA5331	NY	FNS	07/01/2019	48	SUN	ACURA	P	36270	27990	27990	20
14471127...	JWW888	NY	FNS	06/16/2019	48	SUN	HENDR	P	98770	11710	27990	20
14471127...	H0D954	NY	FNS	06/16/2019	48	SUN	ACURA	P	36270	11710	27990	20
14471126...	GYH1545	NY	FNS	06/15/2019	19	SUND	MBSA	P	36270	11710	27990	20

< 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

Sanc...	Plat...	Regis...	Plat...	Issue...	Viola...	Vehic...	Vehicle...	Issue...	Stree...	Street...	Stree...	Whe...
14471525...	BT59881	NY	FNS	06/28/2019	21	SUN	BMW	P	27990	36290	30350	20
14471526...	JV95222	NY	FNS	06/28/2019	20	SUN	TONST	P	36290	27990	12113	20
14471527...	0489898	NY	FNS	06/28/2019	19	SUN	BMW	P	98770	11710	27990	20
14471528...	05E1947	NY	FNS	06/28/2019	18	SUN	AUDR	P	98770	11710	27990	20
14471529...	G098863	NY	CDM	07/05/2019	48	SUND		P	21160	36290	36290	20
14471530...	HHR0442	NY	FNS	06/14/2019	48	SUND	MBSA	P	98770	11710	27990	20
14471531...	HYA5470	NY	FNS	06/14/2019	46	SUND	TONST	P	36290	11710	27990	20
14471532...	GWY9964	NY	FNS	06/14/2019	46	SUND	HENDR	P	98770	11710	27990	20
14471533...	HWB7769	NY	FNS	06/28/2019	48	SUND	TONST	P	98770	11710	27990	20
14471534...	03H2184	NY	FNS	05/06/2019	48	SUN	DODGE	P	31190	40404	40404	20
14471535...	EA5331	NY	FNS	05/10/2019	46	SUN	ACURA	P	98770	11710	27990	20
14471536...	JWW888	NY	FNS	06/16/2019	48	SUN	HENDR	P	98770	11710	27990	20
14471537...	H0D954	NY	FNS	06/16/2019	46	SUND	ACURA	P	98770	11710	27990	20
14471538...	GYH1545	NY	FNS	06/15/2019	19	SUND	MBSA	P	98770	11710	27990	20

< 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

Sanc..._I	Plate..._I	Regis..._I	Plate..._I	Issue..._I	Violen..._I	Vehic..._I	Vehicle..._I	Insu..._I	Strea..._I	Street..._I	Strea..._I	Wri...
14471125...	BT56861	NY	PAS	06/28/2019	21	SUN	BMW	P	27990	36290	30350	20
14471126...	JV95222	NY	PAS	06/28/2019	20	SUN	TENST	P	36290	27990	12113	20
14471126...	0A98989	NY	PAS	06/28/2019	19	SUN	BMW	P	98770	11710	27990	20
14471126...	0E97647	NY	PAS	06/24/2019	19	SUN	AUDR	P	98770	11710	27990	20
14471127...	G098863	NY	CDM	07/05/2019	48	SUND	P	21160	36290	36290	20	
14471128...	HHR0442	NY	PAS	06/14/2019	48	SUND	NISSA	P	98770	11710	27990	20
14471129...	HYA5470	NY	PAS	06/14/2019	46	SUND	TENST	P	36290	11710	27990	20
14471130...	GWY9964	NY	PAS	06/14/2019	46	SUND	HONDA	P	36270	11710	27990	20
14471130...	HHR7769	NY	PAS	06/28/2019	48	SUND	TENST	P	98690	11710	27990	20
14471131...	03H2184	NY	PAS	05/06/2019	48	SUN	DODGE	P	31190	40404	40404	20
14471132...	EA5331	NY	PAS	05/01/2019	46	SUN	ACURA	P	36270	11710	27990	20
14471132...	JWW888	NY	PAS	06/16/2019	48	SUN	HONDA	P	36270	11710	27990	20
14471133...	HQD834	NY	PAS	06/16/2019	48	SUN	ACURA	P	36270	11710	27990	20
14471134...	GYH1545	NY	PAS	06/15/2019	19	SUND	NISSA	P	36270	11710	27990	20

< 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

Sanc...	I	Plate...	I	Regis...	I	Plat...	I	Issue...	I	Vehic...	I	Vehicle...	I	Offic...	I	Insu...	I	Strea...	I	Street...	I	Strea...	I	Whe...
14471125...	175661	NY	PAS	06/28/2019	21	SON	BMW	P	27990	36290	30350	20												
14471126...	JV95222	NY	PAS	06/28/2019	20	SON	TENST	P	36290	27990	12113	20												
14471126...	0489889	NY	PAS	06/28/2019	19	SUBN	BMW	P	36270	11710	27990	20												
14471126...	05E7647	NY	PAS	06/28/2019	19	SUN	AUDR	P	36270	11710	27990	20												
14471126...	G098663	NY	CDM	07/05/2019	48	SUN	P	31190	36290	36290	20													
14471126...	HHR0422	NY	PAS	06/14/2019	48	SUBN	NISSA	P	36270	11710	27990	20												
14471126...	YAM5470	NY	PAS	06/14/2019	46	SUBN	TENST	P	36290	11710	27990	20												
14471126...	GWY9964	NY	PAS	06/14/2019	46	SUBN	HENDR	P	36270	11710	27990	20												
14471126...	HWB7769	NY	PAS	06/28/2019	48	SUBN	TENST	P	36290	11710	27990	20												
14471126...	05H2186	NY	PAS	05/06/2019	48	SON	DODGE	P	31190	40404	40404	20												
14471126...	EA5331	NY	PAS	05/01/2019	46	SON	ACURA	P	36270	11710	27990	20												
14471126...	JWW8084	NY	PAS	06/16/2019	48	SON	HENDR	P	36270	11710	27990	20												
14471126...	HQD854	NY	PAS	06/16/2019	48	ACURA	P	36270	11710	27990	20													
14471126...	GYH1545	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27990	20												

< 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...	Issu...	Stree...	Stree...	Stree...	Veh
14471523..	JE72681	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JOV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GMW6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471525..	JGX1641	NY	PAS	06/04/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471528..	H9H5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471533..	HOM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWGHW640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GOH2184	ME	PAS	07/06/2019	48	SDN	DOOGIE	P	31190	40404	40404	20
14471536..	ICAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	JWSN006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR0634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

[< 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...	Issu...	Stree...	Stree...	Stree...	Veh
14471523..	JE72681	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JO6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GMW6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471525..	JGX1641	NY	PAS	06/04/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471528..	HWH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471533..	HOM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWHT640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GOH2184	ME	PAS	07/06/2019	48	SDN	DOOGIE	P	31190	40404	40404	20
14471536..	ICAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	JWSN006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR0634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

[< 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...	Issu...	Stree...	Stree...	Stree...	Veh
14471523..	JE72681	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GMW6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471525..	JGX1641	NY	PAS	06/04/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471528..	HKH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471533..	HOM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWGH640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GOH2184	ME	PAS	07/06/2019	48	SDN	DOOGIE	P	31190	40404	40404	20
14471536..	ICAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	JPSW506	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

[< 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...	Issu...	Stree...	Stree...	Stree...	Veh
14471523..	JE72681	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GMW6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471525..	JGX1641	NY	PAS	06/04/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471528..	HKH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471533..	HOM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWGK640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GOH2184	ME	PAS	07/06/2019	48	SDN	DOOGIE	P	31190	40404	40404	20
14471536..	ICAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	JPW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

< 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...	Issu...	Stree...	Stree...	Stree...	Veh...
14471523..	JE72681	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GMW6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471525..	JGX1641	NY	PAS	06/04/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471528..	HKH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471533..	HDM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWHT640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GOH2184	ME	PAS	07/06/2019	48	SDN	DOOGE	P	31190	40404	40404	20
14471536..	ICAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	JPN5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

[< 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...	Issu...	Stree...	Stree...	Stree...	Veh
14471523..	JE72681	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JCV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GMW6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471525..	JGX1641	NY	PAS	06/04/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471528..	HKH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471533..	HDM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWHT640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GOH2184	ME	PAS	07/06/2019	48	SDN	DOOGE	P	31190	40404	40404	20
14471536..	ICAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	JPSW506	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

[< 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...	Issu...	Stree...	Stree...	Stree...	Veh...
14471523..	JE72681	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JOV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GMW6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471525..	JGX1641	NY	PAS	06/04/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471528..	HKH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471533..	HOM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWHT640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GOH2184	ME	PAS	07/06/2019	48	SDN	DOOGE	P	31190	40404	40404	20
14471536..	ICAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	JPSW506	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR0634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

[< 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..	Violat..	Vehic..	Vehic..	Issu..	Stree..	Stree..	Stree..	Veh
14471523..	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JOV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471526..	IGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDMB069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471529..	HGH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471530..	HRM340	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471531..	GWHP640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471532..	HBH1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40484	40404	20
14471536..	JGAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	IPW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GVM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

< 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..	Violat..	Vehic..	Vehic..	Issu..	Stree..	Stree..	Stree..	Veh..
14471523..	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JOV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471526..	IGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDMB069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471529..	HGH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471530..	HRM340	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWHP640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40484	40404	20
14471536..	JGAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	IPW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

< 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..	Violat..	Vehic..	Vehic..	Issu..	Stree..	Stree..	Stree..	Veh
14471523..	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JOV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471526..	IGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDMB069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471529..	H9H5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471530..	HRM340	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471531..	GWHP640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471532..	HRB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40484	40404	20
14471536..	JGAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	IPW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471538..	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471539..	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

< 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...	Issu...	Stree...	Stree...	Stree...	Veh
14471523..	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JOV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GNMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471526..	IGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GOMB069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471529..	H0H5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471530..	HMK340	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWHT940	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HBH1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40484	40404	20
14471536..	JGAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	IPW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

< 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_	Violat_	Vehic_	Vehic_	Issu_	Stree_	Stree_	Stree_	Veh
14471523..	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	32390	36290	36350	20
14471524..	JOV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GNK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471526..	IGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDMMB69	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471528..	HGH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471533..	HJM340	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWHT940	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HBH1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GHD2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40484	40404	20
14471536..	JGAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	IPW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GVM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

< 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...	Issu...	Stree...	Stree...	Stree...	Veh...
14471523..	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	32790	36290	36350	205
14471524..	JOV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	205
14471525..	GNMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	205
14471526..	IGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	205
14471527..	GDMB069	NY	COM	07/06/2019	48			P	31190	36310	36330	205
14471528..	H0H5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	205
14471529..	HXM340	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	205
14471530..	GWWH940	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	205
14471531..	HBH1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	205
14471532..	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40484	40404	205
14471533..	JGAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	205
14471534..	IPW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	205
14471535..	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	205
14471536..	GVM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	205

< 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_	Violat_	Vehic_	Vehic_	Issu_	Stree_	Stree_	Stree_	Veh
14471523..	JET2661	NY	PAS	06/28/2019	21	SDN	BMW	P	32390	36290	36350	20
14471524..	JOV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GNMK6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471526..	IGX1641	NY	PAS	06/24/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDMB069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471528..	H0H5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471533..	HXM340	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWWH940	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HBH1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GDH2184	ME	PAS	07/06/2019	48	SDN	DODGE	P	31190	40484	40404	20
14471536..	JGAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	IPW5006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GVM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

< 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...	Issu...	Stree...	Stree...	Stree...	Veh
14471523..	JE72681	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JOV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GMW6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471525..	JGX1641	NY	PAS	06/04/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471528..	H9H5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471533..	HOM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWGHW640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GOH2184	ME	PAS	07/06/2019	48	SDN	DOOGIE	P	31190	40404	40404	20
14471536..	ICAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	JWSN006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR0634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

[< 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...	Issu...	Stree...	Stree...	Stree...	Veh
14471523..	JE72681	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JQ6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GMW6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471525..	JGX1641	NY	PAS	06/04/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471528..	H9H5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471533..	HDM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWHT640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GOH2184	ME	PAS	07/06/2019	48	SDN	DOOGIE	P	31190	40404	40404	20
14471536..	ICAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	JWSN006	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR0634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

[< 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...	Issu...	Stree...	Stree...	Stree...	Veh...
14471523..	JE72681	NY	PAS	06/28/2019	21	SDN	BMW	P	27390	36290	36350	20
14471524..	JOV6523	NY	PAS	06/28/2019	20	SDN	TOYOT	P	36290	27390	13113	20
14471525..	GMW6954	NY	PAS	06/16/2019	19	SUBN	BMW	P	36270	11710	27390	20
14471525..	JGX1641	NY	PAS	06/04/2019	19	SDN	AUDI	P	36270	11710	27390	20
14471527..	GDM8069	NY	COM	07/06/2019	48			P	31190	36310	36330	20
14471528..	HMH5242	NY	PAS	06/14/2019	46	SUBN	NISSA	P	36270	11710	27390	20
14471533..	HOM3470	NY	PAS	06/14/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GWHT640	NY	PAS	06/14/2019	46	SUBN	HONDA	P	36270	11710	27390	20
14471533..	HKB1769	NY	PAS	06/28/2019	40	SUBN	TOYOT	P	36290	11710	27390	20
14471533..	GOH2184	ME	PAS	07/06/2019	48	SDN	DOOGE	P	31190	40404	40404	20
14471536..	ICAS331	NY	PAS	07/01/2019	46	SDN	ACURA	P	36270	11710	27390	20
14471537..	JPSW506	99	PAS	06/16/2019	46	SDN	HONDA	P	36270	11710	27390	20
14471537..	HGR2634	NY	PAS	06/16/2019	46		ACURA	P	36270	11710	27390	20
14471538..	GYM7645	NY	PAS	06/15/2019	19	SUBN	NISSA	P	36270	11710	27390	20

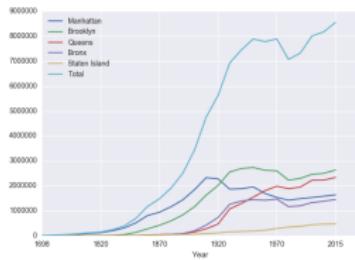
[< Previous](#) [Next >](#)

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

NYC OpenData Home Data About Learn Alerts Contact Us Blog Sign In

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	StartDateTLM	EndDateTime	EnteredOn	EventAge	ParkingHeld	Borough	Comm	Police	Category	SubCategory	Count	ZipCode
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



Home Data About Learn Alerts Contact Us Blog



Sign In

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	StartTimeL...	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



Home Data About Learn Alerts Contact Us Blog



Sign In

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>

The screenshot shows a data visualization interface for the NYC OpenData "Film Permits" dataset. At the top, there are navigation links for Home, Data, About, Learn, Alerts, Contact Us, and Blog, along with a search bar and a sign-in button. Below the header, a descriptive text explains that permits are required for exclusive use of city property like sidewalks, streets, or parks, with a link to the official page. The main content is a table with 15 columns and 8 rows of data. The columns are: EventID, EventType, StartDateTLocal, EndDateTime, EnteredOn, EventAge, ParkingHeld, Borough, Com..., Police..., Category, SubCategory, Count..., and ZipCode. The data rows show various shooting permit details, including dates, locations (e.g., Starr Avenue, Eagle Street), and associated police precincts and categories like "Television" and "Feature".

EventID	EventType	StartDateTLocal	EndDateTime	EnteredOn	EventAge	ParkingHeld	Borough	Com...	Police...	Category	SubCategory	Count...	ZipCode
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...	ELDER 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...	ELDER 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

NYC OpenData

Film Permits

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

EventID	EventType	StartDateTm	EndDateTm	EnteredOn	EventMgt.	ParkingHeld	Boro...	Cou...	Permit...	Compl...	SubCo...	Crater...	ZipCo...
455063	Shooting Permit	12/05/2018 0700..	12/06/2018 0900..	12/05/2018 1230..	Mayor's Off...	STARR AVENUE Bl..	Queens	2	108	Television	Episodic s...	United Sta...	11101
454967	Shooting Permit	12/06/2018 0700..	12/06/2018 0500..	12/06/2018 0201..	Mayor's Off...	SAGGS STREET Bl..	Brooklyn	1	84	Television	Episodic s...	United Sta...	11222
454941	Shooting Permit	12/06/2018 0700..	12/06/2018 0700..	12/06/2018 0544..	Mayor's Off...	SOUTH OXFORD ..	Brooklyn	2, 8	76, 88	Still Photo..	Not Applic...	United Sta...	11217, 11...
454920	Shooting Permit	12/06/2018 1100..	12/06/2018 1150..	12/06/2018 0232..	Mayor's Off...	13 AVENUE betw..	Queens	1, 3, 7	108, 7, 98	Film	Feature	United Sta...	10802, 11...
454914	Shooting Permit	12/06/2018 0800..	12/06/2018 1100..	12/06/2018 0359..	Mayor's Off...	ELDERT STREET Bl..	Brooklyn	4, 5	104, 75, 89	Television	Episodic s...	United Sta...	11203, 11...
454909	Shooting Permit	12/05/2018 0800..	12/05/2018 0600..	12/04/2018 0245..	Mayor's Off...	ELDERT STREET Bl..	Brooklyn	4	83	Television	Episodic s...	United Sta...	11237
454905	Shooting Permit	12/06/2018 0700..	12/06/2018 1030..	12/06/2018 0217..	Mayor's Off...	36 STREET betwe..	Queens	1	114	Television	Color+sp...	United Sta...	11101, 11...

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

Example: OpenData Film Permits

EventID	EventType	StartDate	EndDate	EnteredOn	EventMg.	ParkingHeld	Block#	Comm.	Police	Comp.	SubCn.	Create	ZipCode
455083	Shooting Permit	12/05/2018 0700..	12/06/2018 0900..	12/05/2018 1230..	Mayor's Offc.	STARK AVENUE Bl..	Queens	2	108	Television	Episodic s...	United Sta...	11101
454967	Shooting Permit	12/06/2018 0700..	12/06/2018 0500..	12/06/2018 0201..	Mayor's Offc.	SAGGS STREET Bl..	Brooklyn	1	84	Television	Episodic s...	United Sta...	11232
454941	Shooting Permit	12/06/2018 0700..	12/06/2018 0700..	12/06/2018 0944..	Mayor's Offc.	SOUTH OXFORD ..	Brooklyn	2, 9	76, 88	Still Photo	Not Applic...	United Sta...	11217, 11...
454920	Shooting Permit	12/06/2018 1000..	12/06/2018 1150..	12/06/2018 0232..	Mayor's Offc.	13 AVENUE betw..	Queens	1, 3, 7	108, 7, 98	Film	Feature	United Sta...	10002, 11...
454914	Shooting Permit	12/06/2018 0800..	12/06/2018 1100..	12/06/2018 0359..	Mayor's Offc.	ELDERY STREET Bl..	Brooklyn	4, 5	104, 75, 89	Television	Episodic s...	United Sta...	11201, 11...
454909	Shooting Permit	12/05/2018 0800..	12/05/2018 0600..	12/04/2018 0245..	Mayor's Offc.	ELDERY STREET Bl..	Brooklyn	4	83	Television	Episodic s...	United Sta...	11237
454905	Shooting Permit	12/06/2018 0700..	12/06/2018 1030..	12/06/2018 0217..	Mayor's Offc.	36 STREET betwe..	Queens	1	114	Television	Cable-sys...	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
```

Example: OpenData Film Permits

NYC OpenData

Film Permits

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

EventID	EventType	StartDateL1	EndDate/Time	EnteredOn	EventMgt.	ParkingHeld	Boro...	Com...	Police...	Categ...	SubCate...	CrimeType	ZipCode	Lat	Long
455063	Shooting Permit	12/05/2018 0700..	12/06/2018 0900..	12/05/2018 1230..	Mayor's Off...	STARK AVENUE Bl..	Queens	2	108	Television	Ephemeral s...	United Sta...	11101	40.7122	-74.0059
454967	Shooting Permit	12/06/2018 0700..	12/06/2018 0500..	12/06/2018 0901..	Mayor's Off...	SAGGS STREET Bl..	Brooklyn	1	84	Television	Ephemeral s...	United Sta...	11232	40.6875	-74.0125
454941	Shooting Permit	12/06/2018 0700..	12/06/2018 0700..	12/06/2018 0944..	Mayor's Off...	SOUTH OXFORD Bl..	Brooklyn	2, 9	76, 68	Still Photo	Not Applicable	United Sta...	11217, 112...	40.6875	-74.0125
454920	Shooting Permit	12/06/2018 1100..	12/06/2018 1150..	12/06/2018 0232..	Mayor's Off...	13 AVENUE betw..	Queens	1, 3, 7	108, 7, 98	Film	Feature	United Sta...	10002, 11...	40.7122	-74.0059
454914	Shooting Permit	12/06/2018 0600..	12/06/2018 1100..	12/06/2018 0239..	Mayor's Off...	ELDERY STREET Bl..	Brooklyn	4, 5	104, 75, 89	Television	Ephemeral s...	United Sta...	11203, 11...	40.6875	-74.0125
454909	Shooting Permit	12/05/2018 0800..	12/05/2018 0600..	12/04/2018 0245..	Mayor's Off...	ELDERY STREET Bl..	Brooklyn	4	83	Television	Ephemeral s...	United Sta...	11237	40.6875	-74.0125
454965	Shooting Permit	12/06/2018 0700..	12/06/2018 1030..	12/06/2018 0217..	Mayor's Off...	36 STREET betwe..	Queens	1	114	Television	Color-equal	United Sta...	11101, 11...	40.7122	-74.0059

- 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	StartDate/Time	EndDate/Time	EnteredOn	EventMg.	ParkingHeld	Block#	Comm.	Police	Comp.	SubCom.	Create	ZipCode
455083	Shooting Permit	12/05/2018 0700..	12/06/2018 0900..	12/05/2018 1230..	Mayor's Offc.	STARK AVENUE Bl..	Queens	2	108	Television	Epidemic s..	United Sta..	11101
454967	Shooting Permit	12/06/2018 0700..	12/06/2018 0500..	12/06/2018 0901..	Mayor's Offc.	SAGGS STREET Bl..	Brooklyn	1	84	Television	Epidemic s..	United Sta..	11232
454941	Shooting Permit	12/06/2018 0700..	12/06/2018 0700..	12/06/2018 0944..	Mayor's Offc.	SOUTH OXFORD ..	Brooklyn	2, 9	76, 88	Still Photo..	Not Applic..	United Sta..	11217, 11..
454922	Shooting Permit	12/06/2018 1000..	12/06/2018 1150..	12/06/2018 0232..	Mayor's Offc.	13 AVENUE betw..	Queens	1, 3, 7	106, 7, 98	Film	Feature	United Sta..	10002, 11..
454914	Shooting Permit	12/06/2018 0800..	12/06/2018 1100..	12/06/2018 0239..	Mayor's Offc.	ELDERY STREET Bl..	Brooklyn	4, 5	104, 75, 89	Television	Epidemic s..	United Sta..	11203, 11..
454909	Shooting Permit	12/05/2018 0800..	12/05/2018 0600..	12/04/2018 0245..	Mayor's Offc.	ELDERY STREET Bl..	Brooklyn	4	83	Television	Epidemic s..	United Sta..	11237
454905	Shooting Permit	12/06/2018 0700..	12/06/2018 1030..	12/06/2018 0217..	Mayor's Offc.	36 STREET betwe..	Queens	1	114	Television	Cable-epic..	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

The screenshot shows the NYC OpenData website with the 'Film Permits' dataset selected. The page includes a navigation bar with links like Home, Data, About, Learn, Alerts, Contact Us, Blog, and Sign In. Below the navigation is a search bar and a 'Find in this Dataset' button. The main content area displays a table of data with columns: EventID, EventType, StartDate, EndDate/Time, EnteredOn, EventMgt, ParkingHeld, Block#, Comm., Police, Comp., SubCn, Create, and ZipCode. The table contains 10 rows of permit information, such as shooting permits issued at various addresses in Brooklyn and Queens.

EventID	EventType	StartDate	EndDate/Time	EnteredOn	EventMgt	ParkingHeld	Block#	Comm.	Police	Comp.	SubCn	Create	ZipCode
454983	Shooting Permit	12/05/2018 0700..	12/06/2018 0900..	12/05/2018 1230..	Mayor's Offic...	STAIR AVENUE Bl...	Queens	2	108	Television	Ephemeric s...	United Stat...	11101
454967	Shooting Permit	12/06/2018 0700..	12/06/2018 0500..	12/06/2018 0901..	Mayor's Offic...	SAGGS STREET Bl...	Brooklyn	1	84	Television	Ephemeric s...	United Stat...	11232
454941	Shooting Permit	12/06/2018 0700..	12/06/2018 0700..	12/06/2018 0944..	Mayor's Offic...	SOUTH OXFORD Bl...	Brooklyn	2, 9	76, 88	Still Photo	Not Applicable	United Stat...	11217, 11...
454922	Shooting Permit	12/06/2018 1000..	12/06/2018 1150..	12/06/2018 0232..	Mayor's Offic...	13 AVENUE betw...	Queens	1, 3, 7	108, 7, 98	Film	Feature	United Stat...	10802, 11...
454914	Shooting Permit	12/06/2018 0800..	12/06/2018 1100..	12/06/2018 0239..	Mayor's Offic...	ELDERY STREET Bl...	Brooklyn	4, 5	104, 75, 89	Television	Ephemeric s...	United Stat...	11203, 11...
454909	Shooting Permit	12/05/2018 0800..	12/05/2018 0600..	12/04/2018 0245..	Mayor's Offic...	ELDERY STREET Bl...	Brooklyn	4	83	Television	Ephemeric s...	United Stat...	11237
454905	Shooting Permit	12/06/2018 0700..	12/06/2018 1030..	12/06/2018 0217..	Mayor's Offic...	36 STREET betwe...	Queens	1	114	Television	Cable-epis...	United Stat...	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

The screenshot shows the NYC OpenData website with the 'Film Permits' dataset selected. The page includes a header with links for Home, Data, About, Learn, Alerts, Contact Us, Blog, and Sign In. Below the header is a search bar and a 'Find in this Dataset' button. The main content area displays a table of film permit data with columns: EventID, EventType, StartDate, EndDate, EnteredOn, EventMgt, ParkingHeld, Block, Borough, Cmt., Police, Comp., SubCm., Create, and ZipCode. The table contains 10 rows of permit information.

EventID	EventType	StartDate	EndDate	EnteredOn	EventMgt	ParkingHeld	Block	Borough	Cmt.	Police	Comp.	SubCm.	Create	ZipCode
455083	Shooting Permit	12/05/2018 0700..	12/06/2018 0900..	12/05/2018 1230..	Mayor's Offc.	STARR AVENUE Bl..	Queens	2	108	Television	Epidemic s..	United Sta..	11/01	
454967	Shooting Permit	12/06/2018 0700..	12/06/2018 0500..	12/06/2018 0201..	Mayor's Offc.	SAGGS STREET Bl..	Brooklyn	1	84	Television	Epidemic s..	United Sta..	11/22	
454941	Shooting Permit	12/06/2018 0700..	12/06/2018 0700..	12/06/2018 0544..	Mayor's Offc.	SOUTH OXFORD Bl..	Brooklyn	2, 8	76, 88	Still Photo..	Not Applic..	United Sta..	11/21/17	
454920	Shooting Permit	12/06/2018 1100..	12/06/2018 1150..	12/06/2018 0232..	Mayor's Offc.	13 AVENUE betw..	Queens	1, 3, 7	106, 7, 98	Film	Feature	United Sta..	10/02/18	11..
454914	Shooting Permit	12/06/2018 0800..	12/06/2018 1100..	12/06/2018 0239..	Mayor's Offc.	ELDERT STREET Bl..	Brooklyn	4, 5	104, 75, 89	Television	Epidemic s..	United Sta..	12/02/18	11..
454909	Shooting Permit	12/05/2018 0800..	12/05/2018 0600..	12/04/2018 0245..	Mayor's Offc.	ELDERT STREET Bl..	Brooklyn	4	83	Television	Epidemic s..	United Sta..	11/27	
454905	Shooting Permit	12/06/2018 0700..	12/06/2018 1030..	12/06/2018 0217..	Mayor's Offc.	36 STREET betwe..	Queens	1	114	Television	Epidemic s..	United Sta..	11/01/18	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  
print(tickets["ParkingHeld"].value_counts()[:10]) #Print 10 most popular
```

In Groups: OpenData Film Permits

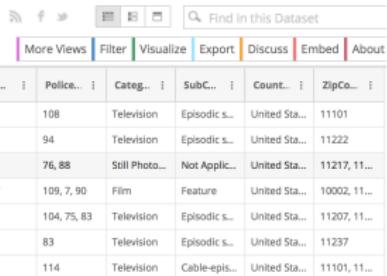


Home Data About Learn Alerts Contact Us Blog Sign In



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>



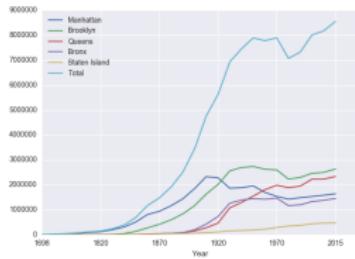
A screenshot of a data visualization interface for the NYC OpenData Film Permits dataset. The interface includes a header with navigation links (Home, Data, About, Learn, Alerts, Contact Us, Blog) and a sign-in button. Below the header is a search bar and a 'Find in this Dataset' button. The main area displays a table with 15 rows of data. The columns are: EventID, EventType, StartDateTL, EndDateTime, EnteredOn, EventAge, ParkingHeld, Borough, Com..., Police..., Categ..., SubC..., Count..., and ZipCo... . The data shows various shooting permits issued across different boroughs, with details like the date range, location, and permit type.

EventID	EventType	StartDateTL	EndDateTime	EnteredOn	EventAge	ParkingHeld	Borough	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:

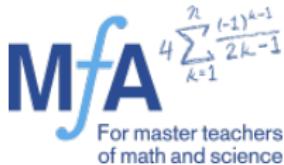
- What's the most popular street for filming?
- What's the most popular borough?
- How many TV episodes were filmed?

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

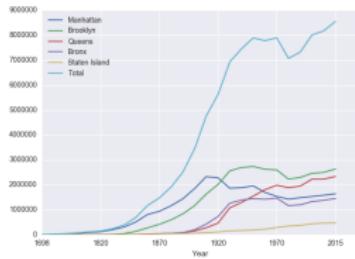
Design a Challenge



With your group, brainstorm about a design challenge that:

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

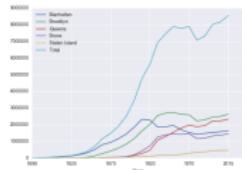
Outline



- 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](#)

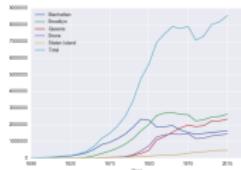
Wrap Up

- Introduced pandas for analyzing structured data.



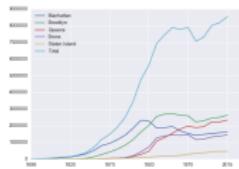
Wrap Up

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

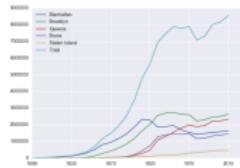


Wrap Up

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

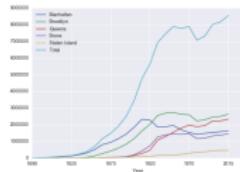


Wrap Up



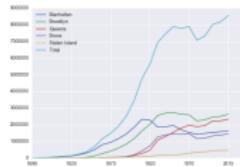
- Introduced pandas for analyzing structured data.
 - ▶ Plotting, simple stats functions, and slicing.
 - ▶ Didn't cover: accessing rows, joining/merging tables, applying functions, ...
- Used publicly available data:

Wrap Up



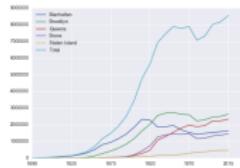
- Introduced pandas for analyzing structured data.
 - ▶ Plotting, simple stats functions, and slicing.
 - ▶ Didn't cover: accessing rows, joining/merging tables, applying functions, ...
- Used publicly available data:
 - ▶ Great source: NYC Open Data.

Wrap Up



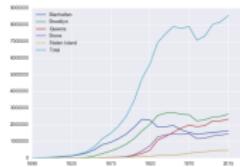
- Introduced pandas for analyzing structured data.
 - ▶ Plotting, simple stats functions, and slicing.
 - ▶ Didn't cover: accessing rows, joining/merging tables, applying functions, ...
- Used publicly available data:
 - ▶ Great source: NYC Open Data.
 - ▶ Examined population data, school attendance,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