CSci 127: Introduction to Computer Science

hunter.cuny.edu/csci

Today's Topics

- Recap: Logical Expressions & Circuits
- Design: Cropping Images
- Pandas

Recap: Logical Operators

and

in1		in2	*returns:*
False	and	False	False
False	and	True	False
True	and	False	False
True	and	True	True

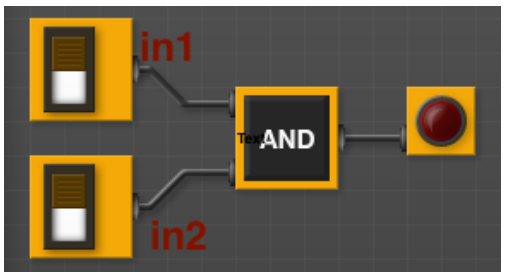
or

in1		in2	*returns:*
False	or	False	False
False	or	True	True
True	or	False	True
True	or	True	True

not

	in1	*returns:*
not	False	True
not	True	False

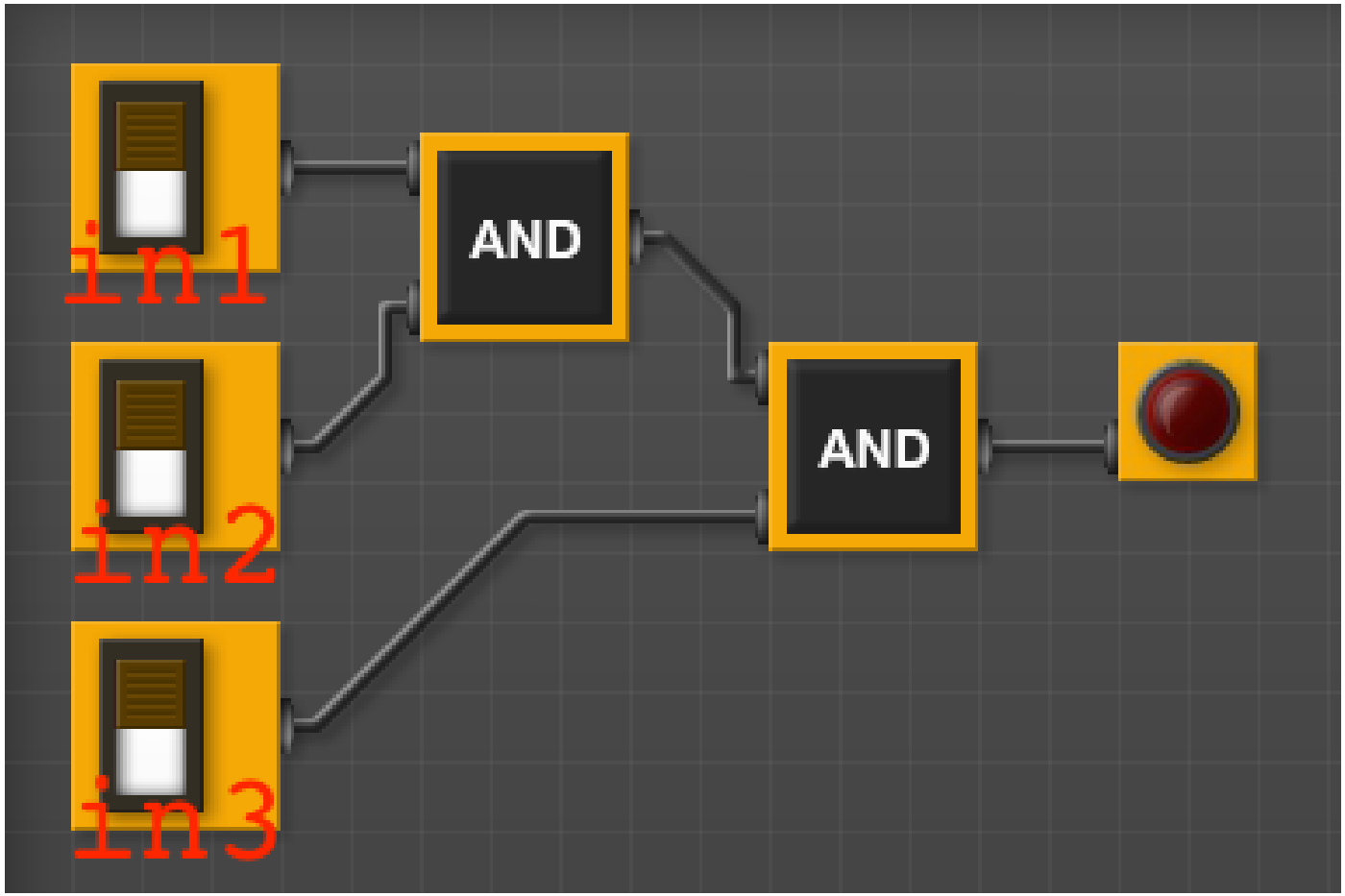
Logical Operators & Circuits



- Each logical operator (`and`, `or`, & `not`) can be used to join together expressions.

  Example: `in1 and in2`
- Each logical operator (`and`, `or`, & `not`) has a corresponding logical circuit that can be used to join together inputs.

Examples: Logical Circuit



`(in1 and in2) and in3`

Examples: Logical Expressions

Examples from last lecture:

```
origin = "Indian Ocean"
winds = 100
if (winds > 74):
    print("Major storm, called a ", end="")
    if origin == "Indian Ocean" or origin == "South Pacific":
        print("cyclone.")
    elif origin == "North Pacific":
        print("typhoon.")
    else:
        print("hurricane.")

visibility = 0.2
winds = 40
conditions = "blowing snow"
if (winds > 35) and (visibility < 0.25) and \
      (conditions == "blowing snow" or conditions == "heavy snow"):
    print("Blizzard!")
```

In Pairs or Triples:

Predict what the code will do:

```
x = 6
y = x % 4
w = y**3
z = w // 2
print(x,y,w,z)
x,y = y,w
print(x,y,w,z)
x = y / 2
print(x,y,w,z)
```

```
sports = ["Field Hockey","Swimming","Water Polo"]
mess = "Qoauxca BrletRce crcx qvBnqa ocUxk"
result = ""
for i in range(len(mess)):
    if i % 3 == 0:
        print(mess[i])
        result = result + mess[i]
print(sports[1], result)
```

Python Tutor

```
x = 6
y = x % 4
w = y**3
z = w // 2
print(x,y,w,z)
x,y = y,w
print(x,y,w,z)
x = y / 2
print(x,y,w,z)
```

(Demo with `pythonTutor`)

In Pairs or Triples: Design Question

From Final Exam, Fall 2017, V4, #6.

Design an algorithm that reads in an image and displays the lower left corner of the image.

https://mmakki11.github.io/CSCI127/images/hunterLogo2.png

Input:
Output:
Process: *(Brainstorm for a "To Do" list to accomplish this.)*

Design Question

Design a program that asks the user for an image and then display the lower left corner of the image. (First, design the pseudocode, and if time, expand to a Python program.)

How to approach this:

- Create a "To Do" list of what your program has to accomplish.
- Read through the problem, and break it into "To Do" items.
- Don't worry if you don't know how to do all the items you write down.
- Example:

1. Import libraries.
2. Ask user for an image name.
3. Read in image.
4. Figure out size of image.
5. Make a new image that's half the height and half the width.
6. Display the new image.

In Pairs or Triples: Design Question

1. Import libraries.

```
import matplotlib.pyplot as plt
import numpy as np
```

2. Ask user for an image name.

```
inF = input('Enter file name:')
```

3. Read in image.

```
img = plt.imread(inF) #Read in image from inF
```

4. Figure out size of image.

```
height = img.shape[0] #Get height
width = img.shape[1] #Get width
```

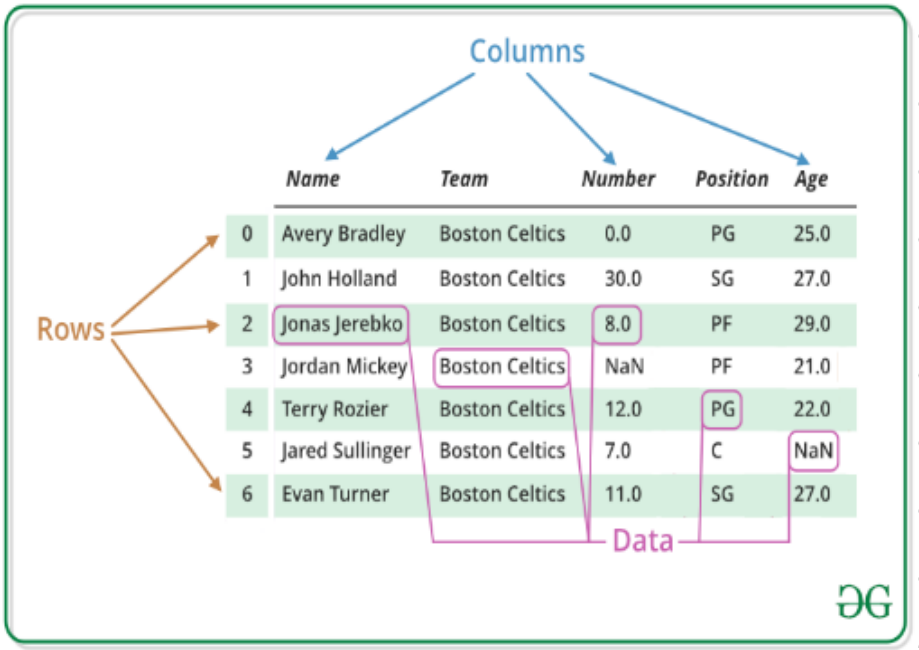
5. Make a new image that's half the height and half the width.

```
img2 = img[height//2:, :width//2] #Crop to lower left
```

6. corner Display the new image.

```
plt.imshow(img2) #Load our new image into pyplot
plt.show() #Show the image (waits until closed to continue)
```

Structured Data



		Undergraduate	
College	Full-time	Part-time	Total
Baruch	12091	3391	15482
Brooklyn	11189	3781	14970
City	10196	2834	13030
Hunter	12991	4130	17121
John Jay	11200	2546	13746
Lehman	8066	4936	13002
Queens	12532	4334	16866
Staten Island	9124	2576	11700
York	5113	3003	8116

- Common to have data structured in a spread sheet. In the example above, we have the first line that says "Undergraduate".
- Next line has the titles for the columns.
- Subsequent lines have a college and attributes about the college.
- Python has several ways to read in such data.
- We will use the popular Python Data Analysis Library (**Pandas**).

Structured Data

pandas

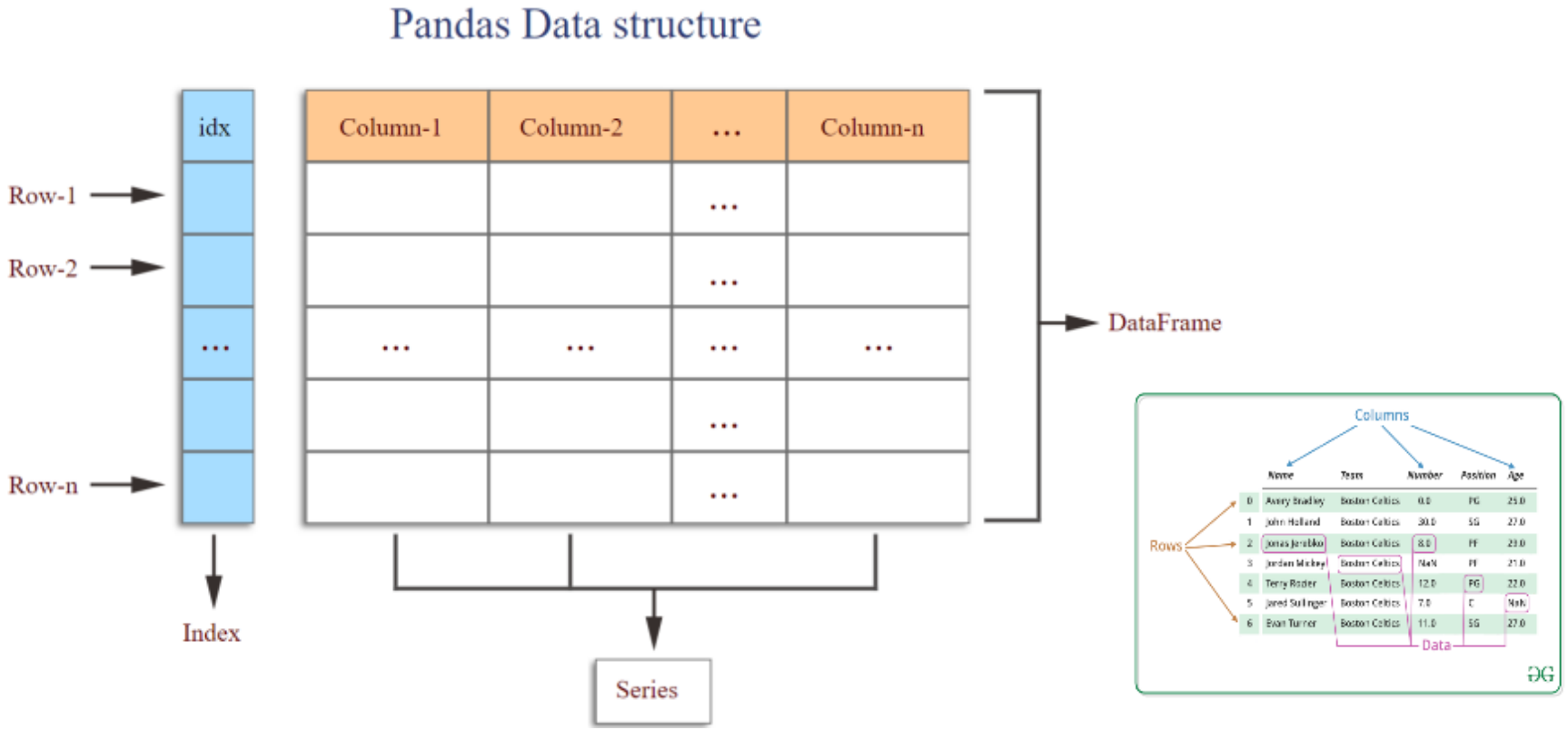
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

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

- To use, add to the top of your file:

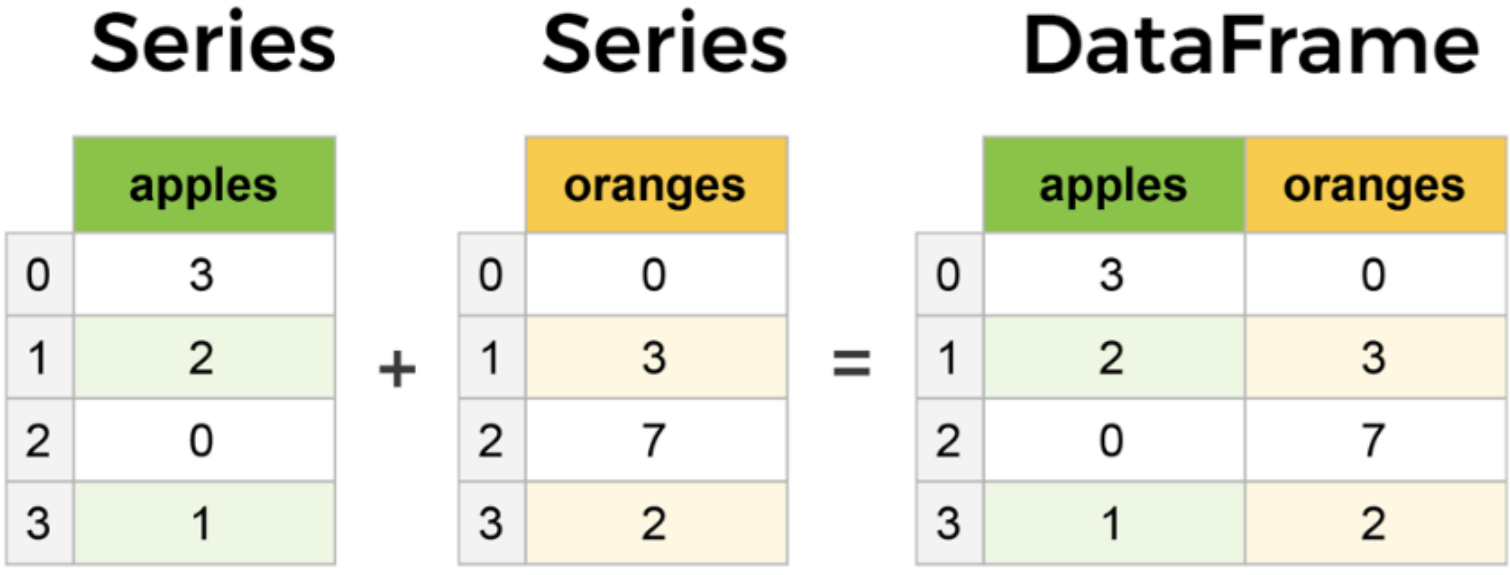
- `import pandas as pd`

Pandas



- 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.

Reading in CSV Files



CSV Files

		Undergraduate	
College	Full-time	Part-time	Total
Baruch	12091	3391	15482
Brooklyn	11189	3781	14970
City	10196	2834	13030
Hunter	12991	4130	17121
John Jay	11200	2546	13746
Lehman	8066	4936	13002
Queens	12532	4334	16866
Staten Island	9124	2576	11700
York	5113	3003	8116

- **CSV** for comma separated values.
- Each row is a line in the file.
- Columns are separated by commas on each line.

File Location

		Undergraduate	
College	Full-time	Part-time	Total
Baruch	12091	3391	15482
Brooklyn	11189	3781	14970
City	10196	2834	13030
Hunter	12991	4130	17121
John Jay	11200	2546	13746
Lehman	8066	4936	13002
Queens	12532	4334	16866
Staten Island	9124	2576	11700
York	5113	3003	8116

Write a complete Python program that reads in the file below, and produces a scatter plot of full-time versus part-time enrollment.

https://mmakki11.github.io/CSCI127/cunyEnrollment2019.csv

In Pairs or Triples

		Undergraduate	
College	Full-time	Part-time	Total
Baruch	12091	3391	15482
Brooklyn	11189	3781	14970
City	10196	2834	13030
Hunter	12991	4130	17121
John Jay	11200	2546	13746
Lehman	8066	4936	13002
Queens	12532	4334	16866
Staten Island	9124	2576	11700
York	5113	3003	8116

Write a complete Python program that reads in the file below, and produces a scatter plot of full-time versus part-time enrollment.

Solution:

1. *Include pandas & pyplot libraries.*
2. *Read in the CSV file.*
3. *Set up a scatter plot.*
4. *Display plot.*

In Pairs or Triples

College	Undergraduate		
	Full-time	Part-time	Total
Baruch	12091	3391	15482
Brooklyn	11189	3781	14970
City	10196	2834	13030
Hunter	12991	4130	17121
John Jay	11200	2546	13746
Lehman	8066	4936	13002
Queens	12532	4334	16866
Staten Island	9124	2576	11700
York	5113	3003	8116

Write a complete Python program that reads in the file, `cunyF2016.csv`, and produces a scatter plot of full-time versus part-time enrollment.

Solution:

Include pandas & pyplot libraries.

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

Read in the CSV file.

```
2 pop=pd.read_csv('cunyEnrollment2019.csv'
                  ,skiprows=1)
```

Set up a scatter plot.

```
3 pop.plot.scatter(x="Full-time"
                   ,y="Part-time")
```

Display plot.

```
4 plt.show()
```

Example: Reading in CSV Files

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

nycHistPop.csv

Example: Reading in CSV Files

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

nycHistPop.csv

In Lab 6

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

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

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

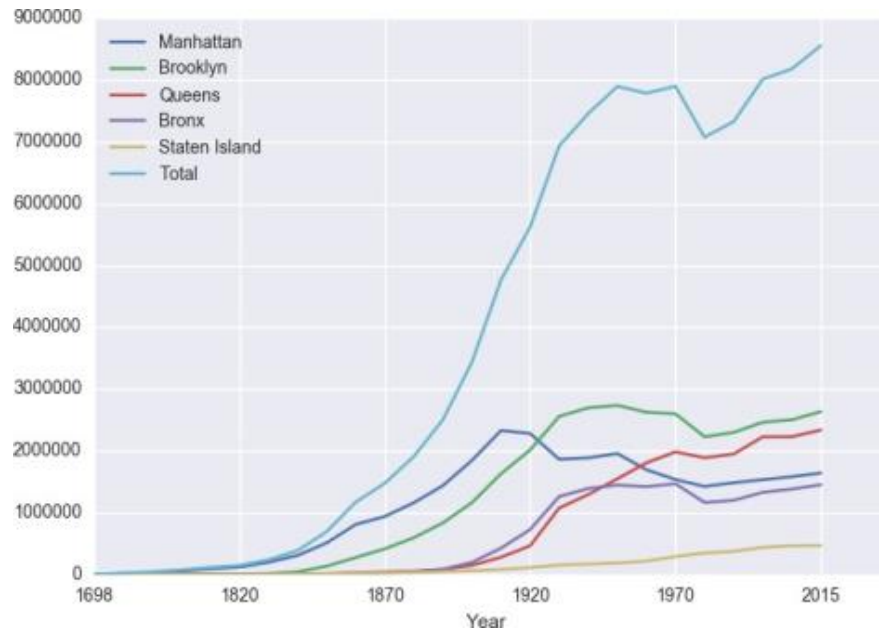
Example: Reading in CSV Files

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

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

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

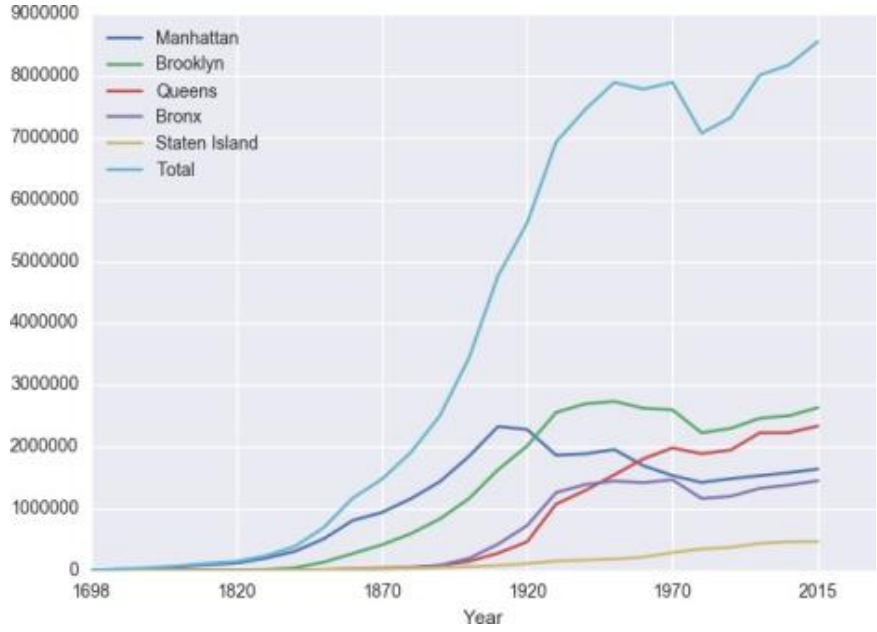
```
Source:  https://en.wikipedia.org/wiki/Demographics_of_New_York_City,,,,,,
All population figures are consistent with present-day boundaries.,,,,,,
First census after the consolidation of the five boroughs,,,,,,
,,,,,,
,,,,,,
Year,Manhattan,Brooklyn,Queens,Bronx,Staten Island,Total
1698,4937,2017,,,727,7681
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

In Lab 6



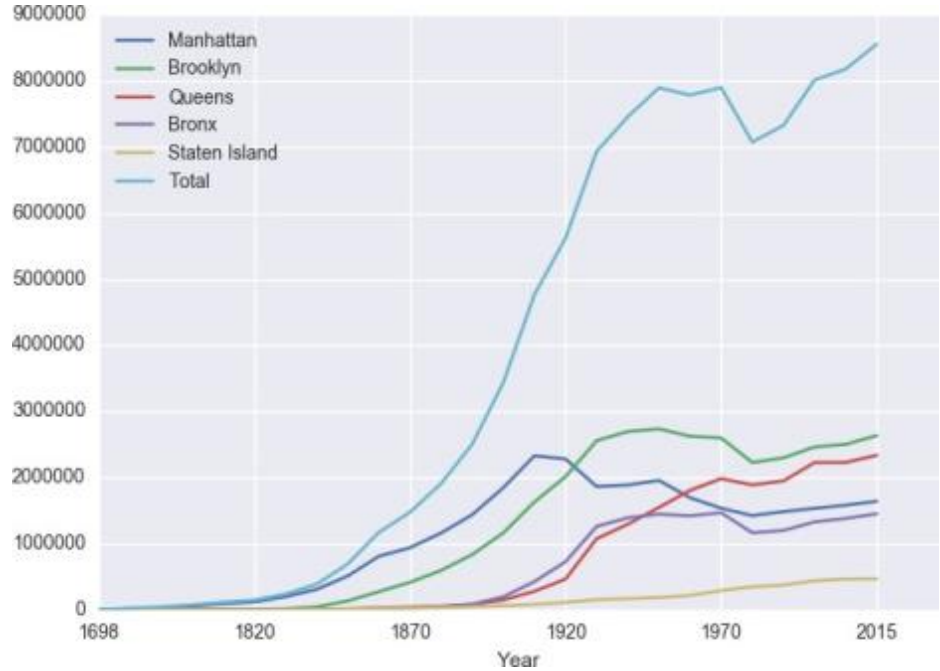
Series in Pandas



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

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

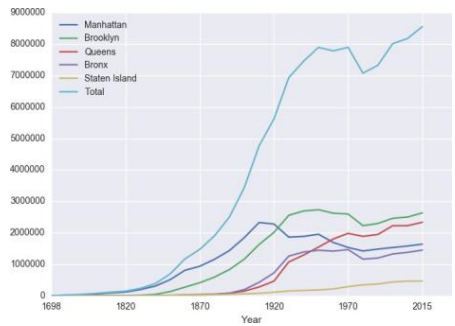
In Pairs or Triples



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())`
  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.