

# Histograms and Distributions

## Introduction

When we vote, we pick one name or another, sometimes one party or another. When voting on a handful of options, a pie chart or bar graph shows the pattern well. The largest piece, as represented by area, wins the vote. Like votes for people, some measurements also fall into categories. When you ask which color of hair is the most common, what is the most common favorite fruit, or what gear cars spend the most time in, you are comparing frequencies of **categorical variables**. Although the frequencies are numbers, the categories have no special order.

What if we were voting for or measuring a **quantitative variable** such as speed limit? We can still ask which speed limit was most frequently voted for or which speed was measured. But the speeds themselves are numbers where smaller numbers really mean something smaller (as opposed to, say, zip codes). To visualize how these votes or measurements are distributed, we use a histogram or boxplot. Is there a quantitative variable that describes something you're interested in?



Categorical data can be shown in a pie chart.



Quantitative data can be shown in a histogram.

## Materials

- Computer with Enthought Canopy distribution of the *Python*® programming language and access to Internet

## Resources

[3.1.5 SourceFiles.zip](#)

# Procedure

## Part I: Types of Measures

1. Form pairs as directed by your teacher. Meet or greet each other to practice professional skills. Set team norms.
2. There are many ways to describe a collection of people. Consider these areas of interest.
  - Physical appearance, like height or hair color
  - Physical health, like resting heart rate, best dash time, or whether the person smokes
  - Economic measures, like employment or spending and saving habits
  - Demographics, like heritage, religiosity, or gender
  - Family structure, like number of siblings or living arrangements
  - Hobbies, like favorite games or hours of TV or technology use

In the examples listed above, circle the categorical measures and place a box around the quantitative measures. Note that several of the measures could be either categorical or quantitative, depending on how it is described.

3. Consider which of the descriptions above you would be most interested in learning about your class or school. As a class, brainstorm some more questions that would be interesting, within the categories above or in other categories.
4. Bar graphs are different than histograms. Bar graphs can be used to compare any value across several categories, like average home size across people of different income groups. The y-axis is always quantitative (ordered, interval, or ratio), and the x-axis is organized into **discrete** groups. The groups might be categorical (like “City High” and “North High”). The groups might also be discrete quantitative values (like grade level). A continuous quantity (like income or temperature) can also be represented by the groups on the x-axis, but quantities will be grouped into intervals (like income groups or temperature ranges).

Think of three examples of possible bar graphs that would be interesting to you. Think of one for each type of x-measurement below. Record the x- and y-axis variables for each idea.

	Type of x-axis	Example for x-axis	Example for y-axis
<b>Example 1</b>	Categorical		
<b>Example 2</b>	Quantitative, Discrete		
<b>Example 3</b>	Quantitative, Continuous but Grouped		

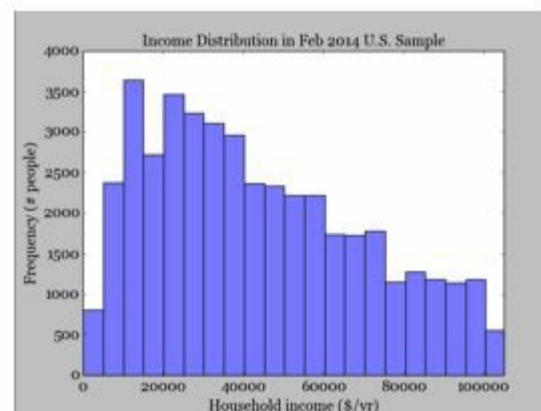
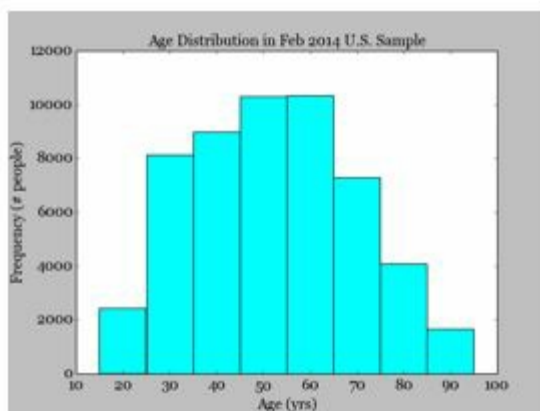
## Part II: Histograms

5. When the value on the y-axis is the **frequency** of each category and x-axis values are ordered, like grade levels or income, then the graph is called a **histogram**. The x-axis values are grouped into categories called **classes** (or equivalently called **bins** or **intervals**).

Refer to your downloadable resources for this material. Interactive

A histogram should show the following

- title
- source and date of the data
- numbers on the x-axis to indicate intervals
- numbers on the y-axis to indicate frequency
- bars touching (unless a bin has frequency 0)
- y-axis starting at zero
- labels on the x and y axes



Below are two histograms showing the U.S. age and income distribution of 50,000 householders sampled in 2014 by the Census Bureau in its monthly Current Population Survey. Critique the two histograms.

6. To create a histogram, you need data organized in one of the following two representations.
- A list of values, like those listed here

**Ages**

15

17

16

15

25

- A two-column table of interval/bin/class and frequencies, like the frequency table shown

here.

Age Interval	Frequency
15-24	2,422
25-34	8,128
35-44	8,989

The two-column table above shows the first few lines of the frequency table used to generate the histogram of age shown above on the left. Use the histogram to record two or three more lines in the frequency table, providing rounded estimates as needed.

7. Describe how the numbers in each column in the two-column table in the previous step are represented in the visualization above.
8. The Current Population Survey (CPS) reports data on about 50,000 randomly selected U.S. households each month. The age above describes the “householder” who is on the rental or ownership documents. The income above describes the combined income of everyone in the household.
  - Create the first few lines of a two-column table showing the income distribution displayed in the graph above on the right.

income	frequency

- Examine the age and income data that is contained in `age_income_feb14.csv` using Notepad++ or TextWrangler. (These simple text editors will show ASCII characters representing the raw data, whereas a spreadsheet application like MS Excel® will transform the data before displaying it by attaching meaning to characters like \$ or –.) Describe how the contents of this file are related to the height of one of the bars in the histogram above of income distribution.
9. Open `histogram_age_income.py` in Canopy and execute it. Examine the two histograms produced. A histogram can be produced by two different `plt` methods, corresponding to the two types of data shown above.
    - `AxesSubplot.hist(x)` – the argument `x` is a list of values.
    - `AxesSubplot.bar(left, height, width=1.0)` – argument `left` is a list of the left boundary of each interval, argument `height` is a list of the frequencies, and argument `width=1.0` makes the bars touch.

Which method does `histogram_age_income.py` use and why?

10. In `histogram_age_income.py`, comment out line 68 and execute the code.

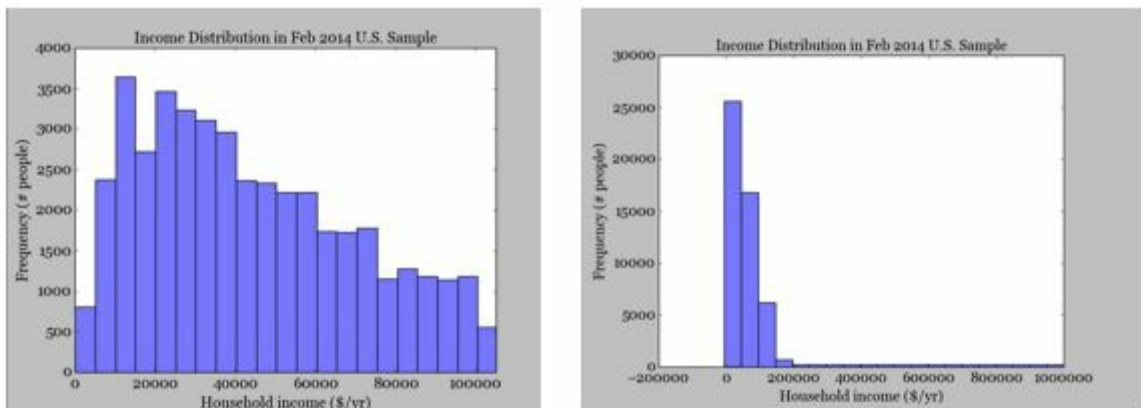
```
#ax.set_xlim(0, 105000)
```

How does the `set_xlim()` method affect the way the histogram looks?

11. Change the `bins` argument in line 67 from 200 bins to 20 bins and execute the code.

```
ax.hist(incomes, color='#7777ff', bins=20)
```

All the households represented in the histogram on the left are shown in just two bars in the histogram on the right. The appearance of a histogram can be surprisingly affected by the number of bins. Try different values for the `bins` argument and contrast two of the figures produced.



12. An important skill for working with data is transforming the data.

Refer to your downloadable resources for this material. Interactive content may not be available in the PDF edition of this course.

Data is transformed when changing from one representation to another, such as from a numeric type to a string type. Data can also be transformed by applying various functions, such as changing from dollars to cents. The data published by the Census Bureau is in `feb14pub.dat`, downloadable from the Census Bureau. The first 500 lines of that file are provided in the curriculum in `feb14pub_header.dat`. Examine this DAT file using Notepad++ or TextWrangler. These records contain about 1000 characters per person, following a code described in `January_2014_Record_Layout.doc`. The script `read_feb14.py` transforms the data by selecting four columns (age, income, state, household size) for rows where the person is identified as the household's reference person. The *Python* script writes the output to `age_income_feb14.csv` and `household_size_feb14.csv`.

- The Record Layout describes how data are represented by the Census in their DAT file. The descriptions are given in order by location, as shown below. Income, for example, is described in locations 39-40. That means the 39th and 40th character on each line reports the sampled household's income. To demonstrate that you have browsed the Record Layout document, record at least one other question asked by Census Bureau

employees, and also record the location of the answers.

NAME	SIZE	DESCRIPTION	LOCATION
HRHHID	15	HOUSEHOLD IDENTIFIER (Part 1)	1-15

- A **dictionary** is a data structure with unordered keys. A dictionary stores a value for each key. In *Python* a dictionary is written as `{key:value, key:value}`. An `int` or a `string` can be used as the key, and the value can be any data type. You can access the dictionary values with the key:

`dictionary[key]` will return value.

- Examine the Python dictionary created on lines 38-44 of `read_feb14.py`. The dictionary stores the income codes from the Record Layout. The values are 2-tuples. What does each member of the tuple represent?
- Examine the code that picks an income based on lines 109-130 in `read_feb14.py`. The result is transformed from the representation in the `feb14pub.dat` file. Describe the transformation.

13. The CSV files contain data for all 150,000 people included in the February 2014 sample. The following portion of `histogram_age_income.py` transforms the CSV data to *Python* lists `incomes` and `ages`.

```
# Build an absolute filename from directory + filename
filename = os.path.join(directory, 'age_income_feb14.csv')
datafile = open(filename, 'r')
data = datafile.readlines()

#####
# Transform the data from strings to signed integers
#####
ages = []
incomes = []
for line in data[3:]:
    age, income = line.split(',')

    # _____ the age data
    ages.append(int(age))

    # _____ the income data
    if '-' in income:
        # Do not use the first 3 characters: space, -, and $
        # and do not use the last character: \n
        incomes.append(-1*int(income[3:-1]))
    else:
        # Do not use the first 2 characters: space and $
        # and do not use the last character: \n
        incomes.append(int(income[2:-1]))
```

To understand the code above, examine the CSV data file using Notepad++.

Use the IPython session as directed below, and discuss the code and the data file with your partner.

- The method `readlines()` returns a list of strings. Use the IPython session as shown below to examine one element of `data` and to examine the effect of `split()`.

```
In []: data[5]
In []: data[5].split(',')
```

What do `readlines()` and `split()` do?

- What is the role (e.g., fixed value, accumulator, etc.) of the variable `line`?
- What is the variable type (e.g., int, string, etc.) of `line`?

```
In []: line = data[5]
```

- What is the first character of `income` on every iteration, and why?

```
In []: line = data[12] # arbitrary, just pick one
In []: age, income = line.split(',')
In []: income
```

- Line 35 slices the data by starting at element 3. Why do we want skip elements 0, 1, and 2 here?
- Lines 39, 45 and 49 transform age and income from `string` to `int`. Why do you think this is necessary prior to creating histograms?
- Examine lines 33 and 39. What is the role of the variable `ages`?
- Lines 45 and 49 drop characters at positions [0] and [1] and sometimes position [2] from `income` before transforming to an integer. Improve the comments in the code that explain this. The relevant code segments are show below, for your convenience.

```
'''
histogram_age_income.py
reads data from the age_income_feb14.csv
and creates two histograms: the age distribution and the income
distribution. The data are from U.S. Census Bureau's February 2014
Current Population Survey.
```

```
(c) 2014 Project Lead The Way
'''
```

```
####
# Transform the data from strings to signed integers
####
```

```
# _____ the income data
if '-' in income:
```

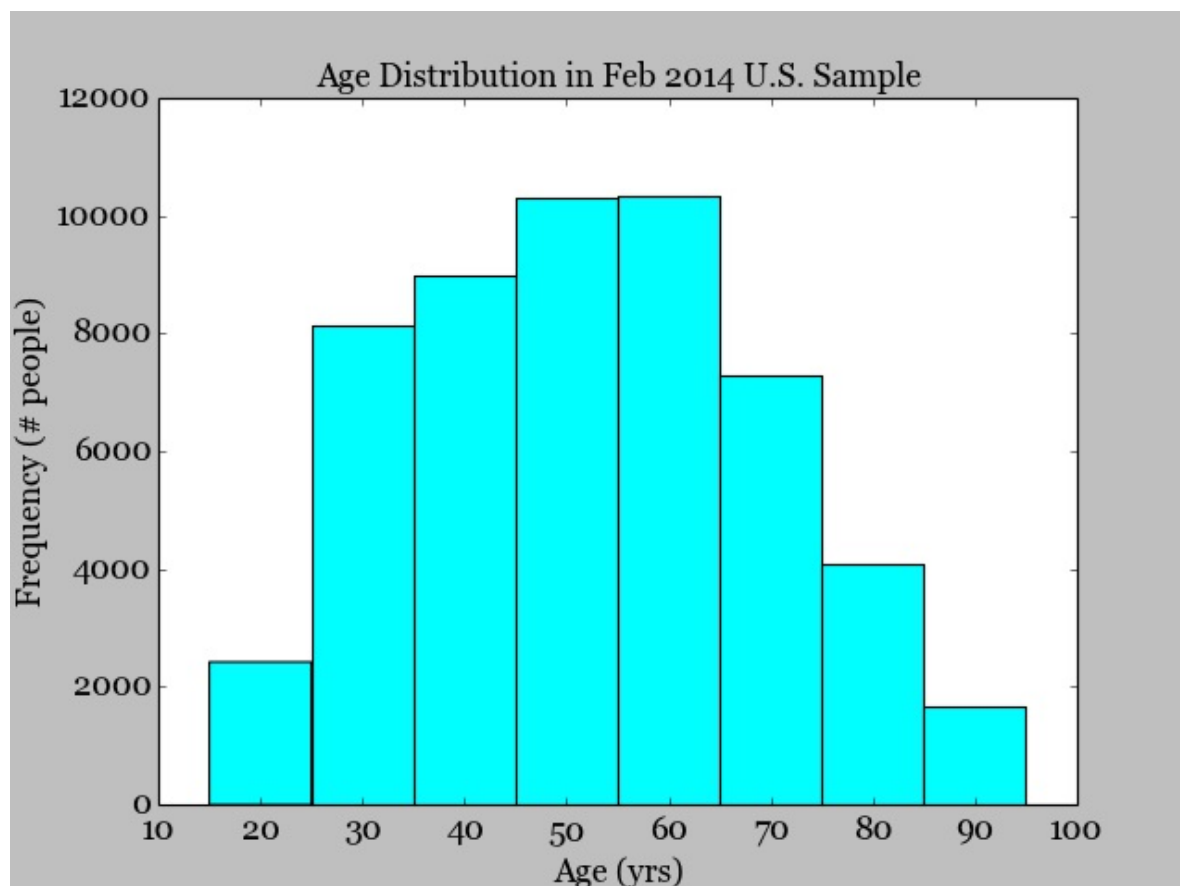
```

# Do not use the first 3 characters: space, -, and $
# and do not use the last character: \n
incomes.append(-1*int(income[3:-1]))
else:
# Do not use the first 2 characters: space and $
# and do not use the last character: \n
incomes.append(int(income[2:-1]))

```

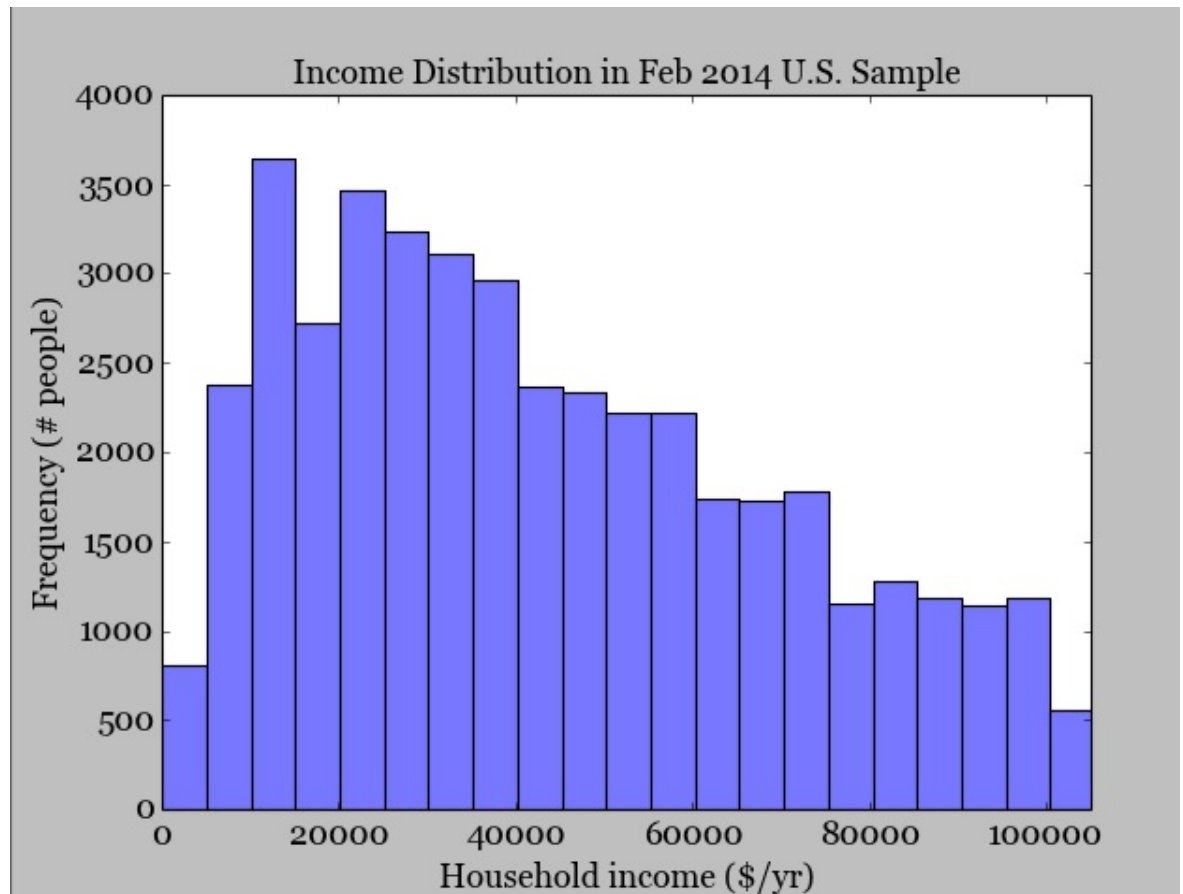
## Part III: Distributions

14. Data are usually a sample from a much larger or infinite population. The histogram shows the sample distribution of the values in the data set. The population distribution is often assumed to be a smoothed-out version of the sample's histogram. The most important characteristics of a distribution are the distribution's shape, center, and spread.
- The shape of a distribution can be one of the following three.
    - symmetric
    - negatively skewed by a heavier **tail** on the left
    - positively skewed by a heavier tail on the right
  - What is the shape of the age sample distribution?



- What is the shape of the income sample distribution?





- The center of a distribution describes the average value. The center can be described by the mean, median, or mode.
  - The mean is the sum of all elements divided by the number of elements. It can be calculated using the following two built-in functions. Record the mean of the ages and incomes.

```
In []: len(ages) # The number of list elements
In []: sum(ages) # The sum of list elements
```

- The median is the middle item in a list that has been sorted. It is the value at the 50th **percentile**; 50% of the values are below the 50 percentile. The method `sort()` will sort a list in-place, which means that no additional list is created in memory. Record the median of the ages and incomes using the 50th percentile item in the list of ages. Record the Python you used to determine the median age.

```
In []: ages.sort() # Changes ages to be ascending order
In []: ages[200] # The 201st item in the list
```

- The mode is the most common value. It is the x-value at the peak in the distribution. Estimate the modes of ages and incomes from the histograms. (These are only estimates since the intervals include multiple values.)
- Visually, the mean, median, and mode can be estimated from a distribution. The mode occurs at the peak. The median splits the area under the curve of a

distribution in half. The mean is shifted from the median toward the larger tail. Sketch the smoothed-out curve of the income distribution.

- Label the mode.
  - Label the median and shade the 50% of the area representing the lower half of income earners.
  - Label the mean.
- The **spread of a distribution** describes how widely the data vary from the center. The spread can be described by the **range** or the **interquartile range** which are nicely displayed on a **box plot**.

The following code shows a boxplot. Use boxplots to estimate the range and interquartile range of ages and incomes.

```
In []: fig, ax = plt.subplots(1, 1) # Also does show()
In []: ax.boxplot(ages)
In []: fig.canvas.draw() # Updates what is shown
```

15. A **sample distribution** is drawn from a **population distribution**, often randomly. Population distributions often have a recognizable shape. The two most important distributions are the **uniform distribution** and the **normal distribution**. Execute the following *Python* code to explore the uniform distribution.

```
import random
import matplotlib.pyplot as plt

rainfall = []
for i in range(100):
    rainfall.append(random.uniform(1,3))
fig, ax = plt.subplots(1, 1)
ax.hist(rainfall)
fig.show()
```

- Examine the histogram that is displayed and discuss what the code does. How should the y-axis be labeled?
  - Change line 5 to use a bigger number as the argument to `range()` and execute the code again. Try several values for the argument, perhaps one thousand, ten thousand, and one million. Describe how the size of the `range()` argument affects the appearance of the histogram.
  - What is the role of the variable `rainfall` in this program?
  - Describe the shape of the uniform population distribution.
16. (Optional) This is an aside to the current activity; proceed to the next step if you're not in the mood for a distraction. A key CSE concept is how to handle complexity by compartmentalizing the solution to a problem. Consider whether it would have been advantageous in the last step to use the following code instead of what was provided. Discuss in a group of two pairs.

```
import random
import matplotlib.pyplot as plt

def sample_rain(n):
    rainfall = []
    for i in range(n):
        rainfall.append(random.uniform(1,3))

    fig, ax = plt.subplots(1, 1)
    ax.hist(rainfall)
    fig.show()
```

17. The function `random.uniform(a, b)` returns a random float chosen according to a uniform distribution from `a` to `b`. To get a normal distribution, we need the `numpy.random` library. The function `numpy.random.randn(n)` returns a list of `n` random numbers chosen according to the standard normal distribution (mean 0 and standard deviation 1).

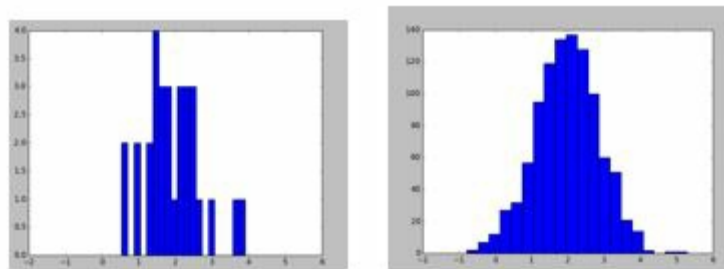
Execute the following *Python* code to sample from a normal distribution for any given mean  $\mu$  and standard deviation  $\sigma$ .

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

mean = 2
standard_deviation = 0.9
rainfall = mean + standard_deviation * np.random.randn(30)

fig, ax = plt.subplots(1, 1)
ax.hist(rainfall, bins=20)
ax.set_xlim(-2, 6)
fig.show()
```

- Examine the histogram that is displayed and discuss what the code does. How should the y-axis be labeled?
- Change line 6 to use a bigger number as the argument to `randn()` and execute the code again. Try several values for the argument, perhaps one thousand, ten thousand, and one million. Describe how the size of the `randn()` argument relates to the appearance of the sample histograms.



- What is the role of the variable `rainfall` in this program?
- Describe the shape of the standard normal distribution.

- Compare and contrast the `random.uniform(a, b)` function and the `numpy.random.randn(n)` function.
18. Generalize from your answers about histograms and sample size. How is the shape of a histogram affected by sample size?
  19. Modify code provided in this activity to use the data set `household_size_feb14.csv` to create the following. Provide your code and visualizations to your teacher as directed.
    - A histogram representing the distribution of household size
    - The mean and median of U.S. household size
    - A box plot showing the distribution of U.S. household size
    - A paragraph in which you describe the data set, describe the transformations performed on the data set, and describe the visualizations produced

## Part IV: Modeling Data

20. A model represents some of the features that describe a set of data. The following code creates two distributions of data: `uniform` and `normal_discrete_cutoff`. Which distribution more closely resembles the distribution of U.S. household sizes?

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

uniform = []
for i in range(1000):
    uniform.append(random.uniform(1,7))

mean = 3
standard_deviation = 2.5
normal = mean + standard_deviation * np.random.randn(1000)
normal_discrete = map(int, normal)
normal_discrete_cutoff = [max(0,x) for x in normal_discrete]

fig, ax = plt.subplots(1, 2)
ax[0].hist(uniform)
ax[1].hist(normal_discrete_cutoff)

fig.show()
```

## Part V: Transforming Data

21. Lines 11, 12, and 13 in the code above demonstrate ways to transform a list of data in *Python*.
  - With your partner, pick one of the techniques below.
    - *Python* operators like `+` or `*` which `numpy.array` implements (line 11)
 

```
np.array(<iterable>) + np.array(<iterable>)
```
    - The *Python* built-in `map` function (line 12)

```
map(<function>, <iterable>)
```

- The *Python* generator expression (line 13)

```
[<expression> for <walker> in <iterable>]
```

- Use the iPython session to explore and understand the technique you selected. Write an explanation of how to use the technique.
  - Use the iPython session to explain the technique to another pair of students. Those students should pick a different technique to explain to you.
22. Use the iPython session to visualize the distributions `normal`, `normal_discrete`, and `normal_discrete_cutoff`. Describe how they are similar and different.
23. Use MS Excel to transform the incomes in `age_income_feb14.csv` to adjust for inflation, reporting the data using 2010 dollars. Use the following steps.
- Open the file in MS Excel.
  - Create a new column for Income in 2010 Dollars, shown below in cell G1.

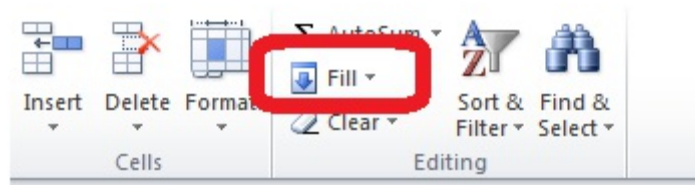
	A	B	C	D	E	F	G	H
1	Age	Income					Income (2010 Dollars)	
2	Data from February 2014 Current Population Survey							
3	Retrieved from <a href="http://thedataweb.rm.census.gov/ftp/cps_ftp.html">thedataweb.rm.census.gov/ftp/cps_ftp.html</a>							
4	76	\$94,600					=B4/1.12	
5	65	\$89,600						
6	78	\$104,100						
7	58	\$2,800						

- Create a formula that works for one of the values. Formulas in Excel begin with an equal sign. This is shown in cell G4. The B4 can be typed by hand or selected while in edit mode.

There was a total of 12% inflation from 2010 to 2014, so divide by 1.12 as shown above to reverse that 12% increase in dollar value.

- To copy that formula, Excel will use a relative reference. The reference is relative because “B4,” when used in cell G4, just means “five cells to the left.” When the formula is pasted into another cell, it will no longer use B4 but will instead still use “five cells to the left.”

Formulas are most easily copied with the “Fill” feature of MS Excel. Select all cells in the 2010 Income column from G4 to G53247. This is most easily done by selecting G4, dragging the vertical scroll bottom, and then shift-clicking G53247. Then from the **Home** ribbon in the Editing panel, select **Fill > Fill Down**.



What value is shown for the Income, in 2010 dollars, of the last person in the sample? (If the column is too narrow to display the value, it will display as #####. The column can be widened by dragging the boundary between column titles.)

- Discuss the relative advantages of *Python* and Excel for performing this task.

## Conclusion

1. What can be learned about a set of data by looking at a histogram?
2. Suppose you surveyed high school students, asking them each how many text messages they send per day.
  - What visualization would be most appropriate for that data?
  - Sketch a graphic displaying the data you hypothesize you would be likely to get. Don't forget to provide a title and labels for the axes.
  - Make another sketch, but this time show your guess about the results you would get if you surveyed college students by asking the same question.
  - What questions might you be able to answer by comparing data about the texts from these two groups? Provide as many questions as possible that you think could be addressed by comparing two large data sets of high school students' and college students' texts.
3. Describe the relationship among the terms “sample”, “population”, “sample distribution”, and “population distribution”.
4. Statistics (sample mean, sample standard deviation, . . .) describe a sample of measurements. Statistics abstract a large data set to a few numbers. The abstracted description takes up less memory than the raw data and is transmitted faster, with less power consumption— all important considerations in mobile products. The Intel Basis watch, for example, collects samples of skin resistance, skin temperature, air temperature, heart rate, and 3-D acceleration each second, processes that data, and records a summary once per minute. The graph below shows skin resistance vs. time for a video game player during a 15 minute interval. Consider what the summary recorded once per minute might look like and show the first records of the file you envision.

