DSC530-302 Data Exploration and Analysis

- Author: Chitramoy Mukherjee
- Date: 04/14/2023
- Title: "DSC530-302 Week-05 Assignment-5.1, 5.2 and 6.1"

Exercise -5.1

In the BRFSS (see Section 5.4), the distribution of heights is roughly normal with parameters μ = 178 cm and σ = 7.7 cm for men, and μ = 163 cm and σ = 7.3 cm for women. In order to join Blue Man Group, you have to be male between 5′10″ and 6′1″ (see http://bluemancasting.com). What percentage of the U.S. male population is in this range? Hint: use scipy.stats.norm.cdf.

In [13]:

```
from os.path import basename, exists


def download(url):
    filename = basename(url)
    if not exists(filename):
        from urllib.request import urlretrieve

        local, _ = urlretrieve(url, filename)
        print("Downloaded " + local)


download("https://github.com/AllenDowney/ThinkStats2/raw/master/code/thinkstats2.py")
download("https://github.com/AllenDowney/ThinkStats2/raw/master/code/thinkplot.py")
download("https://github.com/AllenDowney/ThinkStats2/raw/master/code/brfss.py")
download("https://github.com/AllenDowney/ThinkStats2/raw/master/code/CDBRFS08.ASC.gz")
import brfss
import scipy.stats
import numpy as np
import thinkstats2
import thinkplot

df = brfss.ReadBrfss()
weights = df.wtkg2.dropna()

mu = 178
sigma = 7.7
dist = scipy.stats.norm(loc=mu, scale=sigma)
type(dist)

dist.mean(), dist.std()
```

Out[13]:

```
(178.0, 7.7)
```

In [82]:

```
dist.cdf(mu - sigma)
```

Out[82]:

```
0.1586552539314574
```

In [6]:
```
# Solution

low = dist.cdf(177.8)  # 5'10"
high = dist.cdf(185.4)  # 6'1"
low, high, high - low
```

Out[6]:
```
(0.48963902786483265, 0.8317337108107857, 0.3420946829459531)
```

Exercise 5.2

To get a feel for the Pareto distribution, let's see how different the world would be if the distribution of human height were Pareto. With the parameters xm = 1 m and α = 1.7, we get a distribution with a reasonable minimum, 1 m, and median, 1.5 m. Plot this distribution. What is the mean human height in Pareto world? What fraction of the population is shorter than the mean? If there are 7 billion people in Pareto world, how many do we expect to be taller than 1 km? How tall do we expect the tallest person to be?

In [85]:
```
alpha = 1.7
xmin = 1  # meter
dist = scipy.stats.pareto(b=alpha, scale=xmin)
dist.median()
```

Out[85]:
```
1.5034066538560549
```

In [87]:
```
# Solution : What is the mean human height in Pareto world

dist.mean()
```

Out[87]:
```
2.428571428571429
```

In [88]:
```
# Solution : What fraction of the population is shorter than the mean

dist.cdf(dist.mean())
```

Out[88]:
```
0.778739697565288
```

In [90]:
```
# Solution : Out of 7 billion people, how many do we expect to be taller than 1 km?

(1 - dist.cdf(1000)) * 7e9, dist.sf(1000) * 7e9
```

Out[90]:
```
(55602.976430479954, 55602.97643069972)
```

In [89]:
```
# Solution : How tall do we expect the tallest person to be?

dist.sf(600000) * 7e9
```

Out[89]:
```
1.0525455861201714
```

Exercise 6.1

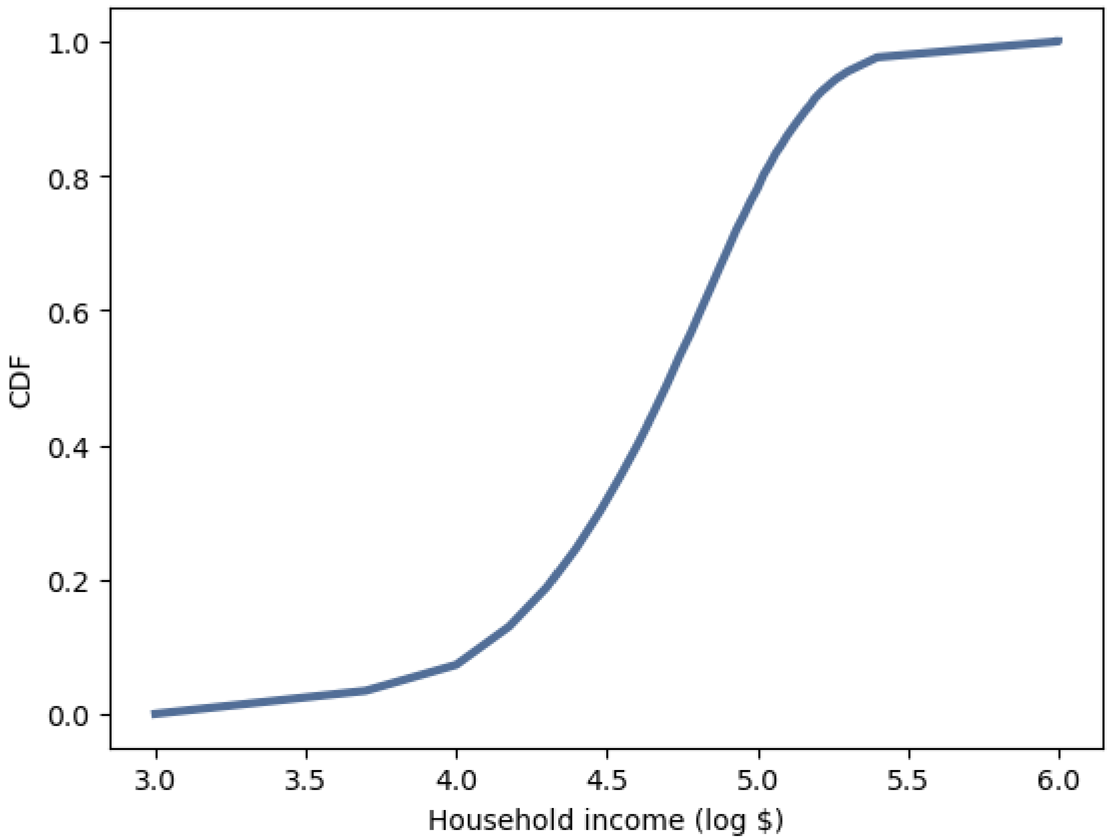
The distribution of income is famously skewed to the right. In this exercise, we'll measure how strong that skew is. The Current Population Survey (CPS) is a joint effort of the Bureau of Labor

Statistics and the Census Bureau to study income and related variables. Data collected in 2013 is available from http://www.census.gov/hhes/www/ cpstables/032013/hhinc/toc.htm. I downloaded hinc06.xls, which is an Excel spreadsheet with information about household income, and converted it to hinc06.csv, a CSV file you will find in the repository for this book. You will also find hinc2.py, which reads this file and transforms the data. The dataset is in the form of a series of income ranges and the number of respondents who fell in each range. The lowest range includes respondents who reported annual household income “Under $5000.” The highest range includes respondents who made “$250,000 or more.” To estimate mean and other statistics from these data, we have to make some assumptions about the lower and upper bounds, and how the values are distributed in each range. hinc2.py provides InterpolateSample, which shows one way to model this data. It takes a DataFrame with a column, income, that contains the upper bound of each range, and freq, which contains the number of respondents in each frame. It also takes log_upper, which is an assumed upper bound on the highest range, expressed in log10 dollars. The default value, log_upper=6.0 represents the assumption that the largest income among the respondents is 106, or one million dollars. InterpolateSample generates a pseudo-sample; that is, a sample of household incomes that yields the same number of respondents in each range as the actual data. It assumes that incomes in each range are equally spaced on a log10 scale. Compute the median, mean, skewness and Pearson’s skewness of the resulting sample. What fraction of households reports a taxable income below the mean? How do the results depend on the assumed upper bound.

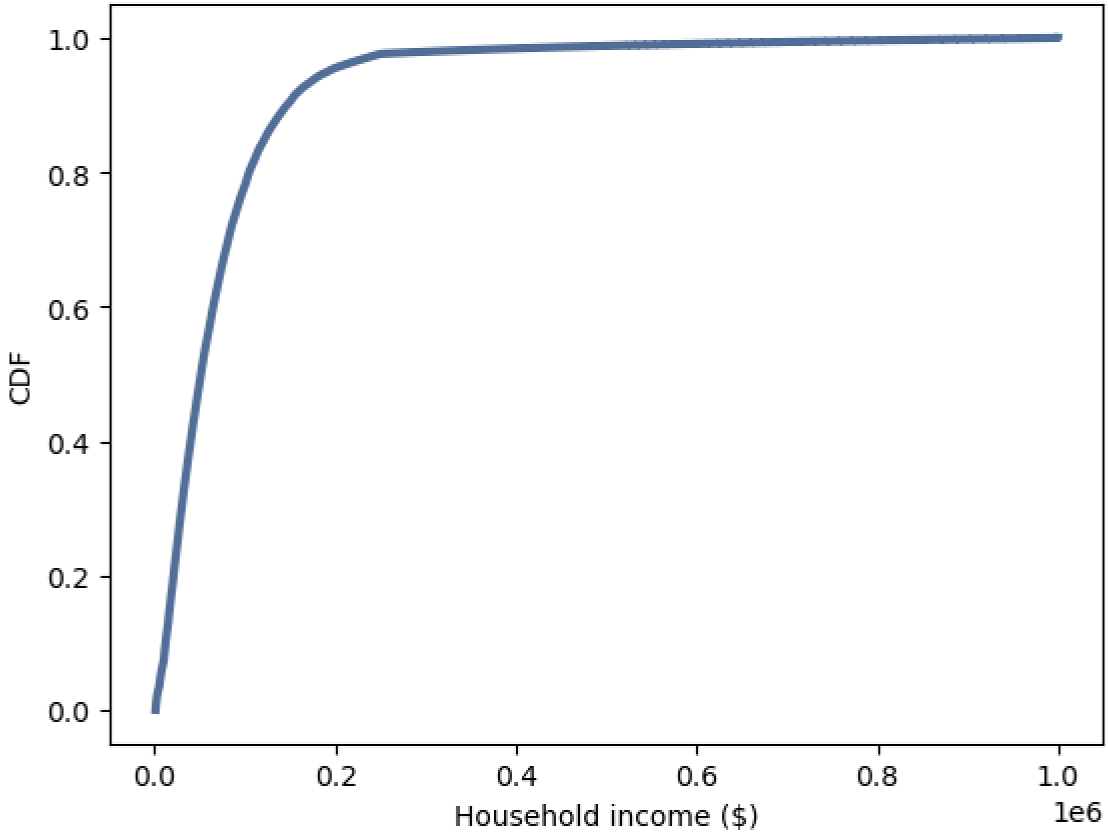
In [41]:

```
def InterpolateSample(df, log_upper=6.0):
    """Makes a sample of log10 household income.

    Assumes that log10 income is uniform in each range.

    df: DataFrame with columns income and freq
    log_upper: log10 of the assumed upper bound for the highest range

    returns: NumPy array of log10 household income
    """
    # compute the log10 of the upper bound for each range
    df['log_upper'] = np.log10(df.income)

    # get the lower bounds by shifting the upper bound and filling in
    # the first element
    df['log_lower'] = df.log_upper.shift(1)
    df.loc[0, 'log_lower'] = 3.0

    # plug in a value for the unknown upper bound of the highest range
    df.loc[41, 'log_upper'] = log_upper

    # use the freq column to generate the right number of values in
    # each range
    arrays = []
    for _, row in df.iterrows():
        vals = np.linspace(row.log_lower, row.log_upper, int(row.freq))
        arrays.append(vals)

    # collect the arrays into a single sample
```

```
    log_sample = np.concatenate(arrays)
    return log_sample

download("https://github.com/AllenDowney/ThinkStats2/raw/master/code/hinc.py")
download("https://github.com/AllenDowney/ThinkStats2/raw/master/code/hinc06.csv")

import hinc
income_df = hinc.ReadData()

log_sample = InterpolateSample(income_df, log_upper=6.0)

log_cdf = thinkstats2.Cdf(log_sample)
thinkplot.Cdf(log_cdf)
thinkplot.Config(xlabel='Household income (log $)', ylabel='CDF')
```



In [42]:
```
sample = np.power(10, log_sample)
cdf = thinkstats2.Cdf(sample)
thinkplot.Cdf(cdf)
thinkplot.Config(xlabel='Household income ($)', ylabel='CDF')
```


1.0
0.8
0.6
0.4
0.2
0.0
CDF
0.0
0.2
0.4
0.6
0.8
1.0
1e6
Household income ($)