Example looking at 1990 first name data from US Census

Thanks to Patrick LeGresley for this example.

Goal: write program to predict *male* or *female* given name

Data: Frequently Occurring Surnames from Census 1990

Algorithm:

1. If input name is in list of males, return `"M"`
2. Else if input name is in list of females, return `"F"`
3. Otherwise, return `"NA"`

Look at the files

```
$ pwd
/Users/nwh/git/cme211-notes/lecture-04
$ ls -1 *.first
dist.female.first
dist.male.first
$ head -n 5 dist.female.first
MARY            2.629  2.629      1
PATRICIA        1.073  3.702      2
LINDA           1.035  4.736      3
BARBARA         0.980  5.716      4
ELIZABETH       0.937  6.653      5
$ head -n 5 dist.male.first
JAMES           3.318  3.318      1
JOHN            3.271  6.589      2
ROBERT          3.143  9.732      3
MICHAEL         2.629 12.361      4
WILLIAM         2.451 14.812      5
```

Notes:

- the unix `head` command prints out the first number lines of a text file based on the number after the `-n` argument
- first column of the data file contains the name in uppercase
- following columns contain frequency data and rank, which we won't use in this lecture

Using sets

Exercise: write a Python script `names_set.py` to implement the name to gender algorithm specified above using the Python `set` container. Also print out some information about the data sets.

The program should take data filenames and test names from the command line. In no command line arguments are provided, the script should print out a helpful usage message.

```
$ python3 names_set.py
Usage:
  $ python3 names_set.py FEMALE_DATA MALE_DATA [TEST NAMES]
Example:
  $ python3 names_set.py dist.female.first dist.male.first Nick
```

If data filenames and test names are provided, the script should behave as follows:

```
$ python3 names_set.py dist.female.first dist.male.first Nick Sally Bicycle
There are 4275 female names and 1219 male names.
There are 331 names that appear in both sets.
Nick: M
Sally: F
Bicycle: NA
```

The word `Bicycle` does not appear in either the male or female dataset, so `NA` is printed.

Using lists

Exercise: write a Python script `names_list.py` to implement the name to gender algorithm specified above using the Python `list` container. Also print out some information about the data sets.

The script should behave the same as `names_set.py`.

Second algorithm

Some names appear in both **male** and **female** lists. Some names might not appear in either list. Let's write a new algorithm to handle this uncertainty:

Given an input name: - return `0.0` if male - return `1.0` if female - return `0.5` if uncertain or name does not appear in dataset

Exercise: write a Python script `names_dict.py` to implement the name to gender algorithm specified above using the Python `dict` container. Also print out some information about the data sets. The behavior should follow:

Usage message:

```
$ python3 names_dict.py
Usage:
  $ python3 names_dict.py FEMALE_DATA MALE_DATA [TEST NAMES]
Example:
  $ python3 names_dict.py dist.female.first dist.male.first Nick
```

`names_dict.py` in action:

```
$ python3 names_dict.py dist.female.first dist.male.first Nick Sally Billy
There are 5163 names in our reference data.
Nick: 0.0
Sally: 1.0
Billy: 0.5
```

The name "Billy" appears in both male and female datasets, so `0.5` is printed after the name to indicate uncertainty.