United States Legislators

October 25, 2017

1 United States Legislators

In the following report, we'll examine data on every legislator who has served in the history of the United States. The data set has been saved in a csv file and contains the following columns of information:

last_name: the surname of the legislator
first_name: the given name of the legislator
birthday: the birthdate of the legislator
gender: the gender of the legislator
type: whether the legislator served in the Senate or the House of Representatives
state: the state the legislator represented
party: the party the legislator belonged to
The first thing we'll do is open the csv file and read it into Python so that we can use the data.

```
In [1]: import csv, re
        f = open("legislators.csv", "r")
        legislators = list(csv.reader(f))
        for each in legislators[0:5]:
            print(*each)
```

```
last_name first_name birthday gender type state party
Bassett Richard 1745-04-02 M sen DE Anti-Administration
Bland Theodorick 1742-03-21  rep VA
Burke Aedanus 1743-06-16  rep SC
Carroll Daniel 1730-07-22 M rep MD
```

Next we'll remove the header row so that we'll manipulate only the data.

```
In [2]: legislators = legislators[1:]
        for each in legislators[0:5]:
            print(*each)
```

```
Bassett Richard 1745-04-02 M sen DE Anti-Administration
Bland Theodorick 1742-03-21  rep VA
Burke Aedanus 1743-06-16  rep SC
Carroll Daniel 1730-07-22 M rep MD
Clymer George 1739-03-16 M rep PA
```

1.1 Most Popular Name

Now that we can easily use the data, let's start to explore it. We'll write code that extrapolates the most popular given name for male legislators and female legislators.

First, the male legislators:

We'll loop through the rows of the data set, count how many times a first name appears, and add the names with their counts to a dictionary, but only if the gender is male.

```
In [3]: male_names = {}
        for row in legislators:
            if row[3] == "M":
                name = row[1]
                if name in male_names:
                    male_names[name] += 1
                else:
                    male_names[name] = 1
        print("\n".join("{}: {}".format(k, v) for k, v in male_names.items()))
```

```
Richard: 155
Daniel: 119
George: 416
Benjamin: 114
Jonathan: 38
William: 1020
Ralph: 29
Samuel: 253
Robert: 315
Michael: 60
Jeremiah: 30
Henry: 261
John: 1233
Aaron: 21
Charles: 436
Oliver: 16
Philip: 40
Moses: 21
Cornelius: 13
Joshua: 28
Roger: 13
Peter: 60
Thomas: 448
Anthony: 22
Alexander: 73
Hugh: 27
James: 710
Abraham: 29
David: 168
Frederick: 64
Joseph: 295
```

Stephen: 54
Nathaniel: 37
Andrew: 73
Timothy: 24
Isaac: 65
Christopher: 24
Elijah: 18
Jacob: 71
Josiah: 24
Nathan: 33
Theodore: 29
Matthew: 17
Mark: 24
Franklin: 22
Albert: 78
Paul: 43
Archibald: 23
Ebenezer: 21
Lewis: 44
Walter: 72
Leonard: 15
Patrick: 30
Martin: 43
Reuben: 12
Edward: 191
Stanley: 10
Herman: 17
Adam: 13
Nicholas: 18
Edwin: 53
Joel: 21
Rufus: 17
Luther: 13
Solomon: 17
Alfred: 44
Arthur: 41
Louis: 27
Edmund: 23
Harry: 54
Dudley: 13
Edgar: 13
Warren: 17
Augustus: 27
Kenneth: 14
Milton: 19
Chester: 19
Lawrence: 24
Oscar: 19

Guy: 14
Homer: 15
Donald: 34
Eugene: 19
Carl: 20
Frank: 118
Fred: 26
Howard: 22
Victor: 13
Ernest: 19
Herbert: 21
Raymond: 14
Earl: 14
Harold: 31
Rod: 2
Connie: 2
Phil: 2
Tim: 3
Carlos: 1
J.: 2
Bill: 4
Tom: 4
Ben: 4
Jay: 2
Julian: 1
Bob: 10
Rick: 4
Ron: 4
Steven: 3
Don: 1
Bruce: 1
Zell: 1
Ken: 2
Conrad: 1
Lincoln: 3
Gary: 2
Jon: 3
Greg: 1
Tony: 1
Brian: 3
Steve: 4
Dan: 3
Joe: 4
Elmer: 1
Norman: 2
Dean: 1
AnÃŋbal: 1
Wayne: 2

Cass: 1
Chris: 2
Douglas: 1
Max: 3
Brad: 2
Norm: 1
Larry: 3
Pete: 1
Calvin: 1
Jonas: 1
Porter: 1
Amory: 1
Gerald: 2
Scott: 3
Doug: 1
Jack: 1
Gordon: 1
Nick: 1
Ted: 2
Evan: 1
Sherwood: 1
Jeb: 1
Jim: 4
Ed: 1
C.: 2
Randall: 1
Byron: 1
Lane: 1
Russell: 1
Judd: 1
Gilbert: 1
Mel: 1
C.L.: 1
Major: 1
E.: 1
Arlen: 1
W.: 2
Rahm: 1
Terry: 1
Luis: 1
Vito: 1
Virgil: 1
Robin: 1
Kenny: 1
Duncan: 1
Bobby: 2
Ric: 1
Ray: 1

```
Trent: 1
H.: 1
Craig: 1
Neil: 1
Allen: 2
Anh: 1
Travis: 1
Artur: 1
Vernon: 1
Barton: 1
Parker: 1
Baron: 1
Eric: 2
Kendrick: 1
Alan: 1
Dennis: 4
Glenn: 1
Ciro: 1
Ike: 1
Zachary: 1
Bart: 1
Gene: 1
Todd: 2
Zach: 1
Roland: 1
Carte: 1
Geoff: 1
Thaddeus: 1
Jesse: 1
Barack: 1
Jeff: 2
Kent: 1
Jason: 1
Roscoe: 1
Francisco: 1
Russ: 1
Hansen: 1
Jerry: 2
Chip: 1
Barney: 1
Elton: 1
Maurice: 1
Dale: 1
R.: 1
Ronald: 1
Mike: 2
Silvestre: 1
Heath: 1
```

```
Fortney: 1
Clifford: 1
Edolphus: 1
Jo: 1
Rodney: 1
Jeffrey: 1
Melvin: 1
Trey: 1
```

Our dictionary has all the first names and the number of legislators with that name. The next step is to find out which count is the highest.

```
In [4]: highest_male_value = None
        for name in male_names:
            count = male_names[name]
            if highest_male_value is None or count > highest_male_value:
                highest_male_value = count
```

Finally, we want to add every name with the value of highest_male_value to a list. When we print this list, we will see which name(s) is the most popular for male legislators.

```
In [5]: top_male_names = []
        for name, counts in male_names.items():
            if counts == highest_male_value:
                top_male_names.append(name)

        print("\n".join(top_male_names))
        print(highest_male_value)
```

```
John
1233
```

We can use the same logic to find the most popular female names:

```
In [6]: top_female_names = []
        female_names = {}
        for row in legislators:
            if row[3] == "F":
                name = row[1]
                if name in female_names:
                    female_names[name] += 1
                else:
                    female_names[name] = 1
        print("\n".join("{}: {}".format(k, v) for k, v in female_names.items()))
```

```
Marilyn: 2
Enid: 1
```

Helen: 1
Eva: 1
Carrie: 1
Patsy: 1
Constance: 1
Lynn: 2
Marge: 1
Karen: 2
Elizabeth: 1
Jennifer: 1
Denise: 1
Katherine: 1
Melissa: 2
Nancy: 2
Sue: 2
Blanche: 1
Cynthia: 1
Anne: 1
Shelley: 2
Julia: 1
Hillary: 1
Barbara: 1
Jo Ann: 2
Thelma: 1
Darlene: 1
Stephanie: 2
Juanita: 1
Deborah: 2
Heather: 1
Virginia: 1
Kathleen: 2
Mary: 2
Carolyn: 1
Mary Jo: 1
Suzanne: 1
Betsy: 1
Hilda: 1
Ellen: 1
Diane: 1
Gabrielle: 1
Jane: 1
Kay: 1
Olympia: 1
Sandy: 1
Judy: 1
Ann Marie: 1
Nan: 1
Laura: 1

```
Jean: 1
Betty: 1
```

```
In [7]: highest_female_value = None
        for name in female_names:
            count = female_names[name]
            if highest_female_value is None or count > highest_female_value:
                highest_female_value = count

        for name, counts in female_names.items():
            if counts == highest_female_value:
                top_female_names.append(name)

        print("\n".join(top_female_names))
        print(highest_female_value)
```

```
Marilyn
Lynn
Karen
Melissa
Nancy
Sue
Shelley
Jo Ann
Stephanie
Deborah
Kathleen
Mary
2
```

1.2 Most Popular Zodiac Sign

In this portion we will examine birthdays to determine which zodiac sign is the most popular for United States legislators. Here is a rundown on where the dates break:

Aries: March 21 - April 19
Taurus:April 20 - May 20
Gemini: May 21 - June 20
Cancer: June 21 - July 22
Leo: July 23 - August 22
Virgo: August 23 - September 22
Libra: September 23 - October 22
Scorpio: October 23 - November 21
Sagittarius: November 22 - Decemebr 21
Capricorn: December 22 - January 19
Aquarius: January 20 - February 18
Pisces: February 19 - March 20

The "birthday" column of our csv file is formatted in two different ways: either YYYY-MM-DD or MM/DD/YYYY. Therefore, we have to test each row to determine which format "birthday" is in. Once we determine the format, we extract the number representing the month of the birthday and add it to a new list.

```
In [8]: birth_months = []
        for row in legislators:
            if re.search("[1][7-9][0-9]{2}", row[2]):
                parts = row[2].split("-")
                birth_months.append(parts[1])
            else:
                parts = row[2].split("/")
                birth_months.append(parts[0])
```

We do the same thing for the number representing the day of the birthday.

```
In [9]: birth_days = []
        for row in legislators:
            if re.search("[1][7-9][0-9]{2}", row[2]):
                parts = row[2].split("-")
                birth_days.append(parts[2])
            else:
                parts = row[2]
                birth_days.append(parts)
```

In order to properly work with the data, we must remove all empty strings for months and days.

```
In [10]: all_birth_months = []
         for month in birth_months:
             if re.search("[0-1][0-9]", month):
                 all_birth_months.append(month)

In [11]: all_birth_days = []
         for day in birth_days:
             if re.search("[0-3][0-9]", day):
                 all_birth_days.append(day)
```

Our next step is to convert the strings to integers.

```
In [12]: int_birth_months = []
         for month in all_birth_months:
             month = int(month)
             int_birth_months.append(month)

In [13]: int_birth_days = []
         for day in all_birth_days:
             day = int(day)
             int_birth_days.append(day)
```

We combine both lists, int_birth_months and int_birth_days, into a single list of lists. Each list represents a birthday where the first number is the month and second number is the day.

```
In [14]: legis_birthdays = zip(int_birth_months, int_birth_days)
         legis_birthdays = list(legis_birthdays)
         legis_birthdays = [list(item) for item in legis_birthdays]
```

With our list of lists, we're finally ready to do make the data meaningful. First we set up our zodiac dictionary with the twelve signs as keys and values set to zero. Then we loop through each birthday list and parse the numbers. Depending on what the numbers are, we add one to the appropriate sign.

```
In [15]: zodiac = {"Aries": 0,
                   "Taurus": 0,
                   "Gemini": 0,
                   "Cancer": 0,
                   "Leo": 0,
                   "Virgo": 0,
                   "Libra": 0,
                   "Scorpio": 0,
                   "Sagittarius": 0,
                  "Capricorn": 0,
                   "Aquarius": 0,
                   "Pisces": 0}

         for each in legis_birthdays:
             if each[0] == 1:
                 if each[1] <= 19:
                     zodiac["Capricorn"] += 1
                 else:
                     zodiac["Aquarius"] += 1
             if each[0] == 2:
                 if each[1] <= 18:
                     zodiac["Aquarius"] += 1
                 else:
                     zodiac["Pisces"] +=1
             if each[0] == 3:
                 if each[1] <= 20:
                     zodiac["Pisces"] += 1
                 else:
                     zodiac["Aries"] += 1
             if each[0] == 4:
                 if each[1] <= 19:
                     zodiac["Aries"] += 1
                 else:
                     zodiac["Taurus"] += 1
             if each[0] == 5:
                 if each[1] <= 20:
```

```
                zodiac["Taurus"] += 1
            else:
                zodiac["Gemini"] += 1
        if each[0] == 6:
            if each[1] <= 20:
                zodiac["Gemini"] += 1
            else:
                zodiac["Cancer"] += 1
        if each[0] == 7:
            if each[1] <= 22:
                zodiac["Cancer"] += 1
            else:
                zodiac["Leo"] += 1
        if each[0] == 8:
            if each[1] <= 22:
                zodiac["Leo"] += 1
            else:
                zodiac["Virgo"] += 1
        if each[0] == 9:
            if each[1] <= 22:
                zodiac["Virgo"] += 1
            else:
                zodiac["Libra"] += 1
        if each[0] == 10:
            if each[1] <= 22:
                zodiac["Libra"] += 1
            else:
                zodiac["Scorpio"] += 1
        if each[0] == 11:
            if each[1] <= 21:
                zodiac["Scorpio"] += 1
            else:
                zodiac["Sagittarius"] += 1
        if each[0] == 12:
            if each[1] <= 21:
                zodiac["Sagittarius"] += 1
            else:
                zodiac["Capricorn"] += 1
    print("\n".join("{}: {}".format(k, v) for k, v in zodiac.items()))
```

```
Aries: 930
Taurus: 872
Gemini: 843
Cancer: 871
Leo: 923
Virgo: 998
Libra: 955
Scorpio: 910
```

```
Sagittarius: 936
Capricorn: 951
Aquarius: 971
Pisces: 1026
```

As the last step, we determine which sign is the most abundant among legislators.

```
In [16]: top_zodiac_count = None
         for signs in zodiac:
             count = zodiac[signs]
             if top_zodiac_count is None or count > top_zodiac_count:
                 top_zodiac_count = count
         for sign, counts in zodiac.items():
             if counts == top_zodiac_count:
                 top_zodiac = sign
         print(top_zodiac)
```

```
Pisces
```

1.3 Most Popular Party

The third and final piece of data we'll examine is the political party. We'll count them up and see which party is represented the most in history. Please note, however, that the data set is incomplete, and several legislators, especially in the 18th Century, do not have their parties listed.

First we create a list of political parties.

```
In [17]: political_parties = []
         for row in legislators:
             party = row[6]
             political_parties.append(party)
         political_parties = list(filter(None, political_parties))
```

We count how many times each political party appears and find the most popular one.

```
In [18]: pol_party = {}
         for party in political_parties:
             if party in pol_party:
                 pol_party[party] += 1
             else:
                 pol_party[party] = 1

         top_party_count = None
         for party in pol_party:
             count = pol_party[party]
             if top_party_count is None or count > top_party_count:
                 top_party_count = count
```

```
        for party, counts in pol_party.items():
            if counts == top_party_count:
                top_party = party
        print(top_party)
        print(top_party_count)
```

```
Democrat
4920
```