The Python 3 Statistics Module and Salaries

Justus Perlwitz

2015-12-06

Contents

Python 3.4 introduced the `statistics` module. It contains helpful methods for determining basic statistical properties, such as mean, median and standard deviation of samples and populations.

In order to get a feel on how to use the methods contained in this module, I wanted to take a look at a open government dataset. The data set in question is called Employee Salaries - 2014 and contains the annual salary information for all employees of Montgomery County, Maryland paid in 2014. Yay for open government.

1 Retrieving the File

In order to follow along, make sure to download the `.csv` dataset like so

```
$ wget https://data.montgomerycountymd.gov/api/views/54rh-89p8/rows.csv
...
Length: 1490119 (1.4M) [text/csv]
...
```

This will download a 1.4 megabyte sized `.csv` file. That sounds like a good size for pure Python statistics.

In order to load the file into Python, we need to run the following:

```
from csv import DictReader
with open('rows.csv') as fd:
    reader = DictReader(fd)
    rows = tuple(filter(
        lambda s: s['Position Title'] is not 'Parttime-Regular',
        reader,
    ))

print(rows[0].keys())
```

```
['Full Name', 'Gender', 'Current Annual Salary', '2014 Gross Pay Received',
'2014 Overtime Pay', 'Department', 'Department Name', 'Division', 'Assignment
Category', 'Position Title', 'Underfilled Job Title', 'Date First Hired']
```

Note that I've discarded all part time salaries as it is hard to compare full and part time salaries. Hopefully, we will be able to extract some useful information.

2 Salary Statistics

First, I want to take a look at the mean and median salary for every employee. This can be achieved by using the `mean()` and `median()` method. In order to retrieve the “Current Annual Salary” column only we will use an `itemgetter`.

```
from statistics import mean, median
from operator import itemgetter

salaries = tuple(map(itemgetter('Current Annual Salary'), rows))
print(salaries[:10])
```

And here are the first 10 salaries

```
('$67527.83', '$95007.83', '$102153.00', '$43657.20', '$91109.00', '$63056.51',
'$43947.05', '$59000.00', '$43410.92', '$57820.00')
```

The values are not available in a float format, which is not useful for performing calculations. Therefore, we need to parse every salary value first.

```
salaries = tuple(map(lambda s: float(s[1:]), salaries))
print(salaries[:10])
```

```
(67527.83, 95007.83, 102153.0, 43657.2, 91109.0, 63056.51, 43947.05, 59000.0,
43410.92, 57820.0)
```

That looks better. Now for the mean and median salaries:

```
print("Mean: {}, Median: {}".format(mean(salaries), median(salaries)))
```

```
Mean: 73534.64393255701, Median: 68522.0
```

For some advantages of using median salaries over mean salaries for statistical analysis, read this article on payscale.com

3 Current Salary for Female and Male Employees

One of the more obvious questions a young csv statistician might ask is what the difference in salaries for female and male employees looks like. Again, we have more than one way of determining the difference. First we will look at the difference in mean and median salaries.

```
from itertools import filterfalse
is_female = lambda s: s['Gender'] == 'F'
female_employees = tuple(filter(is_female, rows))
male_employees = tuple(filterfalse(is_female, rows))

print("Number of female employees: {}".format(len(female_employees)))
print("Number of male employees: {}".format(len(male_employees)))

def get_salaries(rows):
    return map(
        lambda s: float(s[1:]),
        map(itemgetter('Current Annual Salary'), rows)
    )


print("Mean salaries female: {:.2f}, male: {:.2f}".format(
    mean(get_salaries(female_employees)),
    mean(get_salaries(male_employees)),
))

print("Median salaries female: {:.2f}, male: {:.2f}".format(
    median(get_salaries(female_employees)),
    median(get_salaries(male_employees)),
))
```

```
Number of female employees: 3022
Number of male employees: 5222
Mean salaries female: 74947.06, male: 72717.27
Median salaries female: 70758.94, male: 67527.83
```

Very interesting. First, for every 3 female employees there are around 5 male employees. Female employees have a higher mean and median salary compared to men working full time. This is contrary to the general trend of the gender pay gap according to Wikipedia. The Wikipedia article links a U.S. Department of Labor Force report Women in the Labor Force, which came up with a 0.82 female-to-male pay ratio. That means that for every USD earned by a male worker, a female worker earns 0.82 USD.

The ratio for employees of Montgomery County can be calculated as follows

```
print("Mean salaries ratio: {:.2f}".format(
    mean(get_salaries(female_employees)) /
    mean(get_salaries(male_employees))
))
```

```
Mean salaries ratio: 1.03
```

In the case of Montgomery County, a female employee earns 1.03 USD for every 1.00 USD earned by a male employee.

4 Standard Deviation of Salary for Female and Male Employees

Finally, I'd like to take a look at the difference in salary standard deviation both for female and male employees.

```
from statistics import pstdev

print("Standard deviation all employees: {:.2f}".format(
    pstdev(get_salaries(rows)),
))
print("Population standard deviation female: {:.2f}, male: {:.2f}".format(
    pstdev(get_salaries(female_employees)),
    pstdev(get_salaries(male_employees)),
))
```

```
Standard deviation all employees: 26385.30
Population standard deviation female: 25114.08, male: 27060.08
```

So, the standard deviation of salaries for female employees is below the general standard deviation, whereas males have a higher standard deviation. One reason could be that male employees have a more diverse set of position titles, whereas female employees tend to hold more or less the same position. Let's find out what the most common position title for each gender is.

```
from statistics import mode
position_title = itemgetter('Position Title')
print("Most common female employee position title: {}".format(
    mode(map(position_title, female_employees))
))
print("Most common male employee position title: {}".format(
    mode(map(position_title, male_employees))
))
```

```
Most common female employee position title: Office Services Coordinator
Most common male employee position title: Police Officer III
```

Now for the distribution of position titles for both genders. Let's consult our old friend, the `collections` module. I am interested in the 10 most common position titles for both genders.

```
from collections import Counter
female_position_titles = tuple(map(position_title, female_employees))
male_position_titles = tuple(map(position_title, male_employees))

from pprint import pprint
pprint(Counter(female_position_titles).most_common(10))
pprint(Counter(male_position_titles).most_common(10))
```

```
[('Office Services Coordinator', 184),
 ('Police Officer III', 178),
 ('Community Health Nurse II', 136),
 ('Income Assistance Program Specialist II', 126),
 ('Manager III', 125),
 ('Principal Administrative Aide', 115),
 ('Bus Operator', 107),
 ('Social Worker III', 84),
 ('Public Safety Communications Specialist III', 74),
 ('Social Worker II', 70)]
[('Police Officer III', 702),
 ('Firefighter/Rescuer III', 658),
 ('Bus Operator', 499),
 ('Master Firefighter/Rescuer', 199),
 ('Correctional Officer III (Corporal)', 183),
 ('Fire/Rescue Captain', 134),
 ('Police Sergeant', 124),
 ('Mechanic Technician II', 120),
 ('Manager III', 119),
 ('Fire/Rescue Lieutenant', 103)]
```

The male employees are really living their childhood dreams by being firefighting police bus operators. These numbers are not extremely helpful since they give absolute counts for both genders, whereas we are mostly interested in the relative probability of someone of one of the genders having a certain position title. This is an easy fix! Let's reuse the probability distribution function from a previous article about byte histograms.

```
female_distribution = probability_distribution(
    female_position_titles)
male_distribution = probability_distribution(
    male_position_titles)

def pprint_distribution(distribution, most_common=10):
    for key, value in distribution.most_common(most_common):
        print("{}: {:.2f} %".format(key, value))

print("Female Employees:")
pprint_distribution(female_distribution)
print("Male Employees:")
pprint_distribution(male_distribution)
```

```
Female Employees:
Office Services Coordinator: 6.09 %
Police Officer III: 5.89 %
Community Health Nurse II: 4.50 %
```

```
Income Assistance Program Specialist II: 4.17 %
Manager III: 4.14 %
Principal Administrative Aide: 3.81 %
Bus Operator: 3.54 %
Social Worker III: 2.78 %
Public Safety Communications Specialist III: 2.45 %
Social Worker II: 2.32 %
Male Employees:
Police Officer III: 13.44 %
Firefighter/Rescuer III: 12.60 %
Bus Operator: 9.56 %
Master Firefighter/Rescuer: 3.81 %
Correctional Officer III (Corporal): 3.50 %
Fire/Rescue Captain: 2.57 %
Police Sergeant: 2.37 %
Mechanic Technician II: 2.30 %
Manager III: 2.28 %
Fire/Rescue Lieutenant: 1.97 %
```

It seems like male employees spend a lot more time outdoors compared to female employees. Now we can find out what percentage of employees for both genders are covered by the 10 most common position titles.

```
print("""The 10 most common position titles for female employees cover {:.2f}\
 % of all employees""".format(
    sum(map(itemgetter(1), female_distribution.most_common(10)))))
print("""The 10 most common position titles for male employees cover {:.2f}\
 % of all employees""".format(
    sum(map(itemgetter(1), male_distribution.most_common(10)))))
```

```
The 10 most common position titles for female employees cover
39.68 % of all employees
The 10 most common position titles for male employees cover
54.40 % of all employees
```

That means that female employees have a more diverse range of position titles compared to men. Which makes me wonder: How can male employees have a higher standard deviation in salaries if they have more similar position titles compared to female employees? My initial hypothesis was that male employees would have more diverse job titles compared to female employees. I was wrong. This is something which I cannot answer at the moment.

EDIT: I think the explanation might be that 82 Percent of Montgomery County's 50 highest paid employees are men according to a recent article on Bethesda Magazine , retrieved on 2015-12-06. Thus, while female employees have a higher median/mean salary, the top earners being male means a higher standard deviation for male employee salaries.