# INFO370 PS3: Distributions, CLT

2022 年 4 月 19 日

## Instructions

This problemset asks you to do descriptive statistics, sampling, and some mathematical statistics. In particular, you are asked to explore a relationship between two variables, and to explore Central Limit Theorem (CLT).

- Comment and explain your results! Just numbers with no explanation will not count! Remember: your task is to convince us that you understand, not just to produce correct results!
- Include the question numbers in your solution!
- Ensure your submission is readable. Depending of the complexity of your code and the choice of variable names you may need more or less explanations. For instance, if you are asked to find largest income, then code

```
print(largestIncome)
```

  needs no additional explanations. But if you choose to call the variable "maxy", then you may need to add a comment:

```
print(maxy)  # 'maxy' is the largest income
```

As a generic recommendation–this problem set contains a number of repetitive tasks. Consider writing a function that does most of the job, and then just feed different data to this function.

Good luck!

## 1 Explore Distributions (50pt)

In this problem your task is to play with some distributions, analyze their properties, and explore what kind of inequality they describe. In the process you also have to mess with random numbers.

First, we look at Log-normal distribution. It is a popular distribution to describe unequal outcomes, such as human income. It's name, "log-normal" refers to the fact that logarithm of log-normally distributed RV is distributed normally. Log-normal has two parameters, $\mu$ and $\sigma$

(sometimes called "log-mean" and "log-variance"), in a similar fashion as normal distribution. Its pdf is given as

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-\frac{1}{2}\frac{(\log x - \mu)^2}{\sigma^2}} \tag{1}$$

Its expected value (mean) can be computed by integrating the pdf that results in

$$\text{expected value} \quad \mathbb{E}\, X = e^{\mu + \frac{1}{2}\sigma^2} \tag{2}$$

$$\text{variance} \quad \operatorname{Var} X = e^{2\mu+\sigma^2}(e^{\sigma^2} - 1). \tag{3}$$

You will encounter log-normally-distributed data again and again, so let's get a little familiar with it. See lecture notes 1.4.2, and the subsection "Log-normal distribution" there, the following section "Pareto distribution" may also be useful.

You can generate random numbers from this distribution as

```
import numpy as np
np.random.lognormal(0, 1, size=5)

## array([1.30831839, 1.36261206, 1.12708723, 0.18242973, 0.58543283])
```

(In this example $\mu = 0$ and $\sigma = 1$.)

## 1.1 Basic properties (15pt)

1. Choose your sample size S. 10,000 is a good number.

2. Explore the shape of the distribution: draw S random numbers from $LN(0, 0.2)$ (i.e. $\mu = 0$ and $\sigma = 0.2$), $LN(0, 0.5)$ and $LN(0, 1.68)$. Display all these samples on a histogram while labeling clearly which one is which one. Do the histograms in two ways: a) linear-linear scale, and b) log-linear ($x$ is log and $y$ is linear). You can just do histogram of $\log x$ instead of $x$ (you can also set log scale in matplotlib, but the bin widths will look weird).

   Comment the shape of histograms: do they look skewed? Do they look normal?

   Note: it is hard to put both histogram of $x$ and histogram of $\log x$ on the same figure. Rather do different plots.

3. Look at the histograms and tell-what do you think, which one describes the least unequal distribution, and which one the most unequal distribution?

4. Compute sample means and variance (just use `np.mean` and such functions), and compare those with what you get using the corresponding theoretical formulas (2) and (3). Are these similar?

   Hint: the less inequality, the more similar the theoretical and sample values should be. Mean should be fairly close in all cases, but variance in very inequal distribution may be off.

### 1.2 80-20 ratio (20pt)

One way to describe inequality is to compute the 20/80 ratio. As the famous example tells, 20% of people own 80% of resources. But what are the ratios in these random numbers here? Do we have that 10% of x-s "own" 90% of "values"? (very unequal) Or maybe 49% of x-s possess 51% of values (very little inequality)? The analytic solution does not exist (afaik) though one can solve the ratio numerically. But we go a simpler way and use a loop to figure out an approximate number.

1. Compute the 20/80 ratio for all these three distributions. Consult lecture notes, Section "Describing Data" and subsection "80-20 rule (Pareto principle)", page 14 for now; and python notes, Ch 7.4.1 "80-20 ratio".

   You can do it in a following way (but other solutions are ok too):

   (a) Compute the total income in your sample (i.e. sum of all values)

   (b) pick a quantile (say, upper 10%). Find the corresponding income threshold in the sample. You can use `np.percentile`, in this case it would be `np.percentile(x, 90)` for the top-10 pct threshold (this is the same as lower 90th percentile, so thats why "90").

   (c) Find the total income of the top-10 pct by just summing all income values that are larger than the threshold.

   (d) Compute the wealth share of the top-10 pct. Is this more than 90%? If yes then you should look at a smaller top percentage (e.g. 9pct). If not, look for a larger percentage (e.g 11pct).

   (e) In practice, you want to loop over top percentages (e.g. from 1% to 50%) and see where you get close to correct ratio.

   Hint: the answers are approximately 54, 60, 80 (plus/minus a pct or so).

   Hint2: it is easier for you to handle these questions here if you write the code as a function that takes the data (random numbers) in as an argument.

2. Which distribution is the most unequal one? Which one the most equal one? Does this corresponds to what did you guess based on the visual impression based on the histograms?

---

### 1.3 Inequality in data (15pt)

Now it is time to check how inequality looks in two different datasets. Your task is to analyze inequality in two distributions: citations of research papers, and income of labor market program participants (back in 1978). The datasets are as follows:

- Labor market participants: file *treatment.csv*. This contains data about individuals, some of whom did participate in certain training programs. Use the variable *re78*, real income for 1978.

- Citations: file *mag-30k-citations.csv*, citations of 30,000 research papers in Microsoft Academic Graph. The only important variable in the current context is *citations*, how many times the paper has been cited.

Now the detailed tasks. These basically repeat what you did above with random numbers.

1. Load both datasets and do basic checks—do the values of interest (number of citations and income) look reasonable?

2. Show the distribution of income and citations on a histogram. As the histogram may not look good, do it in two ways: a) histogram of income/citations and b) histogram of log income/citations. In order to avoid issues with log of zero you can do $\log(1+income)$ instead of $\log income$.

3. Compute sample mean and standard deviation for both datasets. Compare these: how much smaller (or larger) is std. dev compared to the mean?

4. Compute the 20/80 ratio for all the three distributions. You can re-use your code from PS2.

---

## 2 Explore Central Limit Theorem (50pt)

In this section you will see how does Central Limit Theorem (CLT) work. CLT states two things:

a) Means of random numbers tend to be normally distributed if the sample gets large.

b) Variance of the mean tends to be $\frac{1}{S}\operatorname{Var} X$ where $S$ is the sample size and $X$ is the random variable we are analyzing.

(This is actually a property of expectation and independence, not really CLT. But CLT is closely related to this result.)

CLT, and how variance and mean value change when sample size increases, plays a very important role in computing confidence intervals later.

The problem contains two tasks: work with Bernoulli-distributed numbers (discrete distribution), and with long-normal-distributed numbers (continuous distribution).

The task is structured in a way that you may want to create a function that takes in sample size $S$ and outputs all needed results, including the histogram. There will be quite a bit of repetitive coding otherwise.

We start with a distribution that does not look at all normal. We create a RV

$$X = \begin{cases} -1 & \text{with probability } 0.5 \\ 1 & \text{with probability } 0.5. \end{cases}$$

(You can imagine we flip a fair coin and label heads as 1 and tails as -1.) One way to sample from such RV is something like this

```
import numpy as np
np.random.randint(0,2, size=10)*2 - 1

## array([-1, -1, -1, -1, -1,  1,  1,  1,  1,  1])
```

Detailed tasks:

1. Calculate the expected value and variance of this random variable.

   Note: these are theoretical values and not related to any samples. If you use functions like `mean` or `var` here then you have misunderstood the concepts!

   Hint: read lecture notes 1.3.4 (Expected Value and Variance), and Openintro Statistics 3.4 (Random variables), in particular 3.4.2 (Variability). I recommend to use the shortcut formula $\operatorname{Var} X = \mathbb{E}\, X^2 - (\mathbb{E}\, X)^2$.

2. Choose your number of repetitions R. 1000 is a good number but you can also take 10,000 or 100,000 to get smoother histograms.

   Note: number of repetitions R *is not* the same as sample size S here. You will create samples of size S for R times below. For instance you will create $R = 1000$ times a sample of size $S = 5$. Please understand the difference, it is a fequent source of confusion!

3. Create a vector of R random realizations of X. Make a histogram of those. Comment the shape of the histogram.

   Note: in this case we have $R = 1000$ repetitions and samples are of size $S = 1$ as we look at individual realizations.

   Hint: it takes some tweaking to get nice histograms of discrete distributions. The simplest way is just to make many bars (most of which will be 0) by adding argument `bins=100` to `plt.hist`.

4. Compute and report mean and variance of the sample you created (just use `np.mean` and `np.var`). NB! Here we talk about *sample mean* and *sample variance*. Compare these numbers with the theoretical values computed at question 1 above.

5. Now create R *pairs* of random realizations of X (i.e. sample size $S = 2$). For each pair, compute its mean. You should have R means. Make the histogram. How does this look like?

   Hint: while you can do this using loops, it is more useful to create a $R \times 2$ matrix of realizations of X, where each row represents one pair. Thereafter you compute means by rows and you have R pair means. See python notes numpy statistical functions for an example.

6. Compute and report mean of the R pair means, and variance of the means. NB! we talk about *sample mean* and *sample variance* again, where sample is your sample of R pair means.

7. Compute the expected value and variance of the pair means, i.e. the theoretical concepts. This mirrors what you did in 1.

Compare the theoretical values with the sample values above. Are those fairly similar?

Note that according to CLT, the variance of a pair mean should be just 1/2 of what you got above as for pairs $\mathsf{S} = 2$.

8. Now instead of pairs of random numbers, repeat this with 5-tuples of random numbers (i.e. $\mathsf{S} = 5$ random numbers per one repetition, and still $\mathsf{R} = 1000$ repetitions in total). Compare the theoretical and sample version of mean and variance of 5-tuples. Are they similar? Do you spot any noticeable differences in the histogram compared to your previous histogram?

9. Repeat with 25-tuples...

   (Including compute the expectation and theoretical variance, and compare those with sample mean, sample variance)

10. ... and with 1000-tuples. Do not forget to compare with theoretical results.

11. Comment on the tuple size, and how the shape of the histogram changes when the tuple size increases.

12. Explain why do the histograms resemble normal distribution as $\mathsf{S}$ grows.

    In particular, explain what happens when we move from single values $\mathsf{S} = 1$ to pairs $\mathsf{S} = 2$. Why did two equal peaks turn into a "⩊"-shaped histogram?

---

## Finally...

How much time did you spend on this PS?