Statistical Distributions and Hypothesis Testing

Session-5

1

Agenda

Lecture -5: Statistical Distributions and Hypothesis Testing

- Use Python for common statistical distributions such as the normal distribution, T-distribution, and Bernoulli distribution.
- Use P values and confidence intervals to test a model
- Conduct basic hypothesis testing to determine the reliability of your conclusions

2

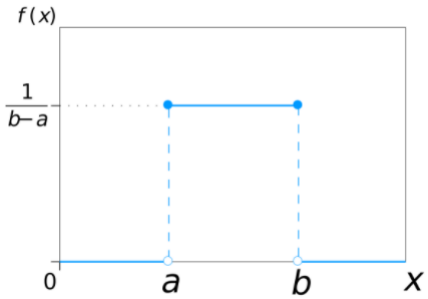
Statistical distributions

3

Uniform distribution

A uniform distribution, sometimes also known as a rectangular distribution, is a distribution that has constant probability.

The probability distribution function of the continuous uniform distribution is: $f(x) = \begin{cases} 0 & x<a \\ \frac{1}{b-a} & a \le x \le b \\ 0 & x>b \end{cases}$



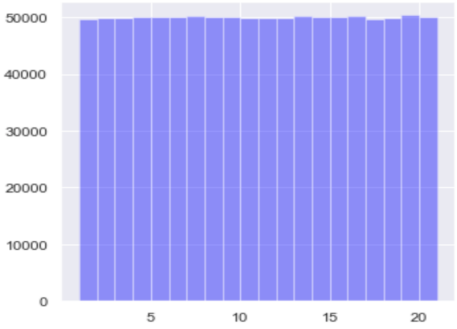
4

Uniform Function

- The uniform function generates a uniform continuous variable between the specified interval via its loc and scale arguments.
- This distribution is constant between loc and loc + scale.
- The size arguments describe the number of random varieties.
- It can be continuous or discrete

```
# import uniform distribution
from scipy.stats import uniform

# random numbers from uniform distribution
n = 100000
start = 1
width = 20
data_uniform = uniform.rvs(size=n, loc = start, scale=width)
```



5

Bernoulli Distribution

A Bernoulli distribution has only two possible outcomes, namely 1 (success) and 0 (failure), and a single trial, for example, a coin toss.

The probabilities of success and failure need not be equally likely. The Bernoulli distribution is a special case of the binomial distribution where a single trial is conducted (n=1).

Its probability mass function is given by:

$$f(k;\, p) = p^k(1-p)^{1-k} \text{ for } k \in \{0,\, 1\}$$

6

Bernoulli Function

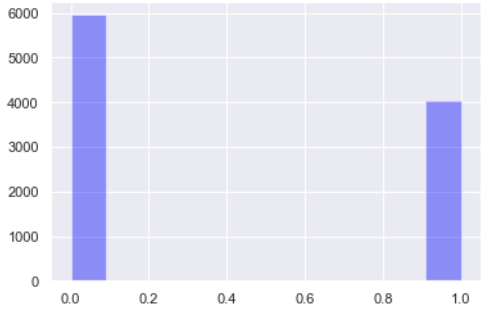
Use rvs(p, loc=0, size=1, random_state=None) method from scipy.stats.bernoulli to create a distribution for a Bernoulli discrete random variable.

```
In [16]: # import bernoulli distribution
         from scipy.stats import bernoulli
         data_bern = bernoulli.rvs(size=10000,p=0.4)

In [17]: data_bern

Out[17]: array([1, 1, 0, ..., 1, 0, 0])

In [18]: sns.distplot(data_bern,
                      kde=False,
                      color="blue")
```



7

Binomial Distribution

- ❑ Only two outcomes are possible, such as success or failure, gain or loss, win or lose
- ❑ The probability of success and failure is same for all the trials. However, The outcomes need not be equally likely

The probability distribution function of the Binomial distribution is: $f(k,\ n,\ p) = \binom{n}{k} p^k (1-p)^{n-k}$

where: $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

The latter expression is known as the ***binomial coefficient***, stated as "*n choose k*," or the number of possible ways to choose *k* "successes" from *n* observations.

For example, the number of ways to achieve 2 heads in a set of 4 tosses is "4 choose 2", or 4!/2!2! = 6. The possibilities are {HHTT, HTHT, HTTH, TTHH, THHT, THTH}. The binomial coefficient multiplies the probability of one of these possibilities (which is (1/2)²(1/2)² = 1/16 for a fair coin) by the number of ways the outcome may be achieved, for a total probability of 6/16.

8

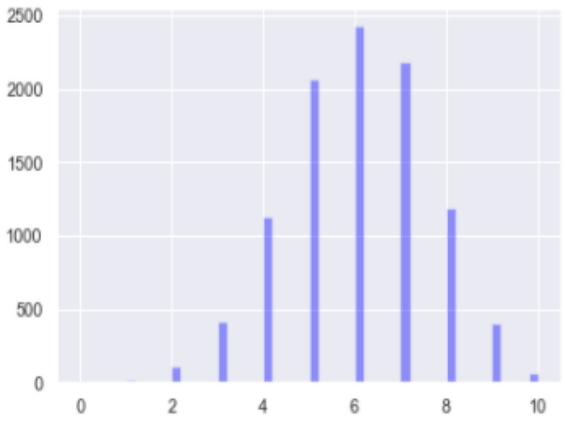
Binomial Function

```
## Import binomial distribution
from scipy.stats import binom
data_binom = binom.rvs(n=10,p=0.6,size=10000)
```

```
data_binom
```

```
array([ 6, 10,  9, ...,  5,  8,  3])
```

```
sns.distplot(data_binom,
                  kde=False,
                  color='blue')
```

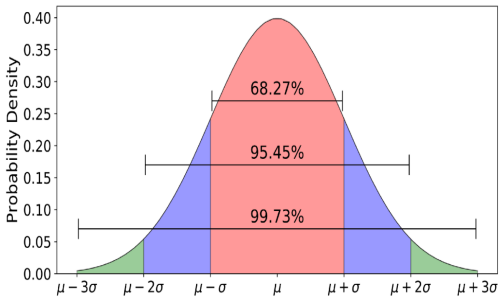


9

Normal Distribution

The normal distribution may be the most important distribution in all of statistics.

- It is a symmetric distribution where most of the observations cluster around the central peak and the probabilities for values further away from the mean taper off equally in both directions.
- It's also known as a Gaussian distribution.
- Because of it's shape, some people refer is as "bell curve."



68% of the data is within 1 standard deviation, 95% is within 2 standard deviation, 99.7% is within 3 standard deviations

10

Normal Distribution

- Popularity and importance of normal distribution comes partly from the Central Limit Theorem.
- Central Limit Theorem - The average of many samples (observations) of a random variable with finite mean and variance is itself a random variable—whose distribution converges to a normal distribution as the number of samples increases.
- Therefore, physical quantities that are expected to be the sum of many independent processes, such as measurement errors, often have distributions that are nearly normal.
- In book The normal distribution is often written as $N(\mu, \sigma^2)$ or $\mathcal{N}(\mu, \sigma^2)$.
- So when a random variable X is normally distributed with a mean μ ad standard deviation σ , we can write the following notation: $X \sim \mathcal{N}(\mu, \sigma^2)$.
- The parameter μ (mu) is the mean or expectation of the distribution (and also its median and mode), while the parameter σ (sigma) is its standard deviation. The square of sigma is the variance.

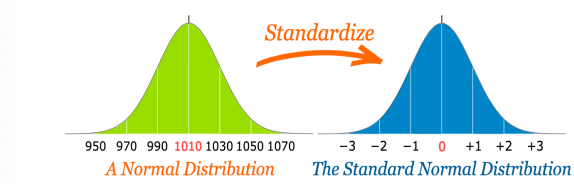
11

Standard Normal Distribution

- The **standard normal distribution** is a **normal distribution** with a mean of zero and **standard** deviation of 1

$$z = \frac{x - \mu}{\sigma}$$

- **z** is the "z-score" (Standard Score)
- **x** is the value to be standardized
- **μ** ('mu") is the mean
- **σ** ("sigma") is the standard deviation



We can take any Normal Distribution and convert it to The Standard Normal Distribution.

- In Data Science and Machine Learning, we often standardize data for better performance and to satisfy model assumptions

12

Normal Distribution

- The probability distribution function of a normal density curve with mean **μ** and standard deviation **σ** at a given point **x** is given by:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Where π (3.14) and e (2.718) are natural numbers

- The use of natural numbers π and e in the above function is consistent with the real world situations in natural and social sciences to represent real-valued random variables whose distributions are not known

13

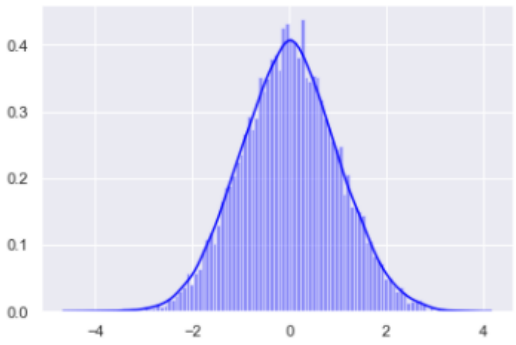
Normal Function

```
In [22]: # Import normal distribution
         from scipy.stats import norm
         # generate random numbers from N(0,1)
         data_normal = norm.rvs(size=10000,loc=0,scale
         =1)

In [23]: data_normal

Out[23]: array([-0.14564826,  0.35802503, -0.05006061,
         ..., -0.74697301,
                -0.19329948, -0.35049665])

In [24]: sns.distplot(data_normal,
                      bins=100,
                      kde=True,
                      color='blue')
```



14

Exponential Distribution (Optional)

The exponential distribution describes the time between events in a Poisson point process, i.e., a process in which events occur continuously and independently at a constant average rate.

The general formula for the probability density function of the exponential distribution is:

$$f(x) = \frac{1}{\beta} e^{-(x-\mu)/\beta} \qquad x \geq \mu;\ \beta > 0$$

where μ is the location parameter and β is the scale parameter (the scale parameter is often referred to as λ which equals 1/β). If we use λ instead of β the equation is as follows:

$$f(x;\lambda) = \lambda e^{-\lambda x}$$

15

Exponential Function (Optional)

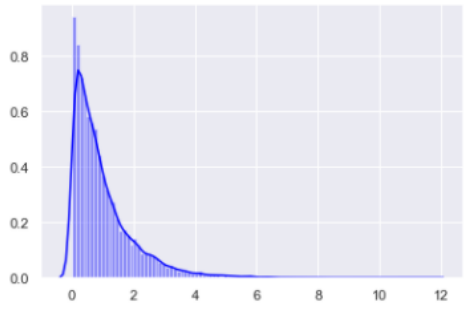
```
In [25]: # Import exponential distribution
         from scipy.stats import expon
         data_expon = expon.rvs(scale=1,loc=0,size=100
         00)

In [26]: data_expon

Out[26]: array([0.67757328, 0.62092071, 0.80363005,
         ..., 1.09499817, 0.68968595,
                2.67992783])

In [27]: sns.distplot(data_expon,
                      kde=True,
                      bins=100,
                      color='blue')
```

```
Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x7f
         d258a049e8>
```



16

Poisson Distribution (Optional)

The Poisson distribution is used to model the number of events occurring within a given time interval.

The formula for the Poisson probability mass function is

$$p(x; \lambda) = \frac{e^{-\lambda}\lambda^k}{k!} \; \lambda \to \infty. \; k = 0, 1, 2, \ldots$$

With a little math fun you can see that the normal distribution is a limiting case of the Poisson distribution with the parameter $\lambda \to \infty$.

17

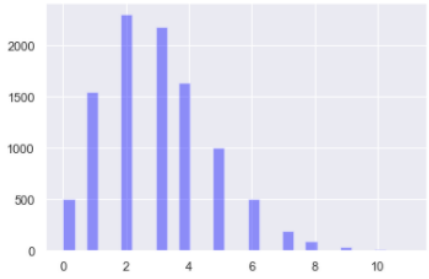
Poisson Function (Optional)

```
In [28]: # Import the poisson distribution
         from scipy.stats import poisson
         data_poisson = poisson.rvs(mu=3, size=10000)

In [29]: data_poisson
Out[29]: array([2, 2, 2, ..., 2, 1, 3])

In [30]: sns.distplot(data_poisson,
                      bins=30,
                      kde=False,
                      color='blue')
```

```
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x7f
         d260aaf8d0>
```



18

t Distribution

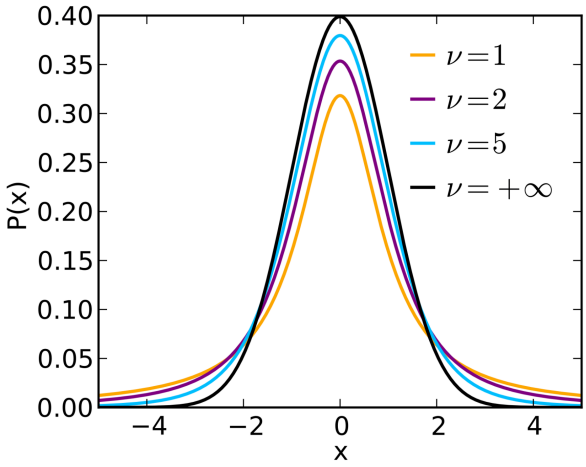
A t distribution describes samples drawn from a full population that follows a normal distribution. The larger the sample of the t distribution, the more the t distribution resembles a normal distribution.

- The t distribution or Student's t distribution is a widely used in hypothesis testing.
- It describes the standardized distances of sample means to the population mean when:
  - The population standard deviation is not known, and
  - The observations come from a normally distributed population.

19

t Distribution

- The *t*-distribution plays a role in a number of widely used statistical analyses:
  - Assessing the statistical significance of the difference between two sample means,
  - Constructing confidence intervals for the difference between two population means, and
- t-distribution becomes closer to the normal distribution as *v (degrees of freedom)* increases.



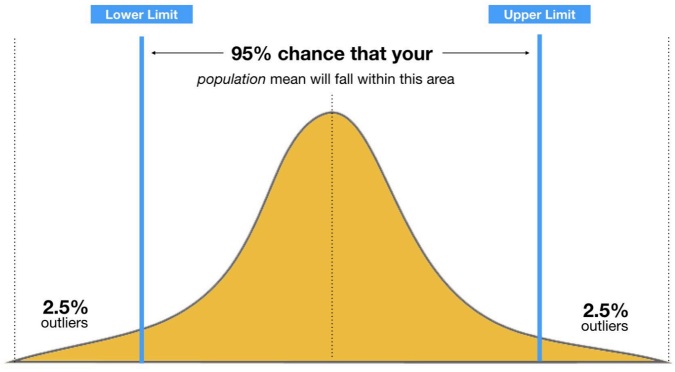
20

Confidence Intervals and Hypothesis Testing

21

Confidence Interval

A confidence interval is an interval that contains the unknown parameter (such as the population mean μ) with certain degree of confidence.



22

z-scores vs t-score

- Z-score is used for a normal distribution, and t-score is used for a t-distribution.
- Use z-score if the population variance is known. If not, use the t-score.
- Since the population variance is almost never known, we almost always use t-score to calculate the confidence interval.
- The "stats" package has a library for the t distribution.
- The "t" library functions similarly to the "norm" library, except that degrees of freedom must be specified.
- Degrees of freedom (df) calculated as n - 1. So, 21 observations would yield to 20 degrees of freedom

23

Research Question

Is the mean male body temperature in the population different from 98.6° Fahrenheit?

Assumption: The sampling distribution is approximately normal.

Steps:

1. Write the Hypothesis
2. Calculate T statistics
3. Find p Values
4. Make a decision
5. State the findings in simple language

24

Write the Hypothesis

μ

Is the mean body temperature in the population different from 98.6° Fahrenheit?

$H_0: \mu = 98.6$
$H_a: \mu \neq 98.6$

25

Calculate T-Statistic

General Form of a Test Statistic

$$test\ statistic = \frac{sample\ statistic - null\ parameter}{standard\ error}$$

$H_0: \mu = 98.6$
$H_a: \mu \neq 98.6$

*"The **standard error** of a **statistic** is the approximate **standard deviation** of a **statistical** sample population. The **standard error** is a **statistical** term that measures the accuracy with which a sample distribution represents a population by using **standard deviation**."*
-From Investopedia

Let's calculate T-Statistic mannually

```
[22]: # Male body Temperature
male_temp=body[body.sex==0]['temp']
male_temp_sample_mean=np.mean(male_temp)
male_temp_sample_mean

[22]: 98.10461538461539

[14]: from scipy.stats import sem

[15]: male_temp_se=sem(male_temp) #standard error for the male temp mean

[23]: t_statistic=((male_temp_sample_mean-98.6)/male_temp_se)
t_statistic

[23]: -5.7157574493185255
```

Now let's use the scipy stats module

```
[17]: import scipy.stats as st
true_mu = 98.6

st.ttest_1samp(body[body.sex == 0]["temp"], true_mu)

[17]: Ttest_1sampResult(statistic=-5.715757449318526, pvalue=3.08384031731503
5e-07)
```

26

p Value, Decision and Findings

- The **p value** is the evidence against a null hypothesis.
- The smaller the **p-value,** the stronger the evidence that you should reject the null hypothesis.

```
[17]: import scipy.stats as st
      true_mu = 98.6

      st.ttest_1samp(body[body.sex == 0]["temp"], true_mu)
```

```
[17]: Ttest_1sampResult(statistic=-5.715757449318526, pvalue=3.08384031731503
      5e-07)
```

Decision: This p-value is much lower than 5%. So, we can reject the null hypothesis.

Findings: This means the alternative hypothesis is correct, and mean male temperature in this sample is different from 98.6.

Thank You