## Probability Distributions:

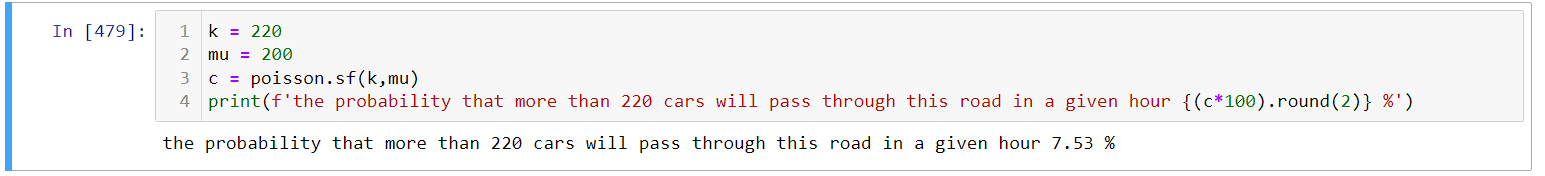
### **Poisson Distribution:**

**n probability theory and statistics, the Poisson distribution is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed...**

'''In a certain city, the number of cars passing through a particular road follows a certain distribution.

The average number of cars passing through this road in an hour is 200.

What is the probability that more than 220 cars will pass through this road in a given hour?'''



k = 220

mu = 200

c = poisson.sf(k,mu)

print(f'the probability that more than 220 cars will pass through this road in a given hour {(c\*100).round(2)} %')

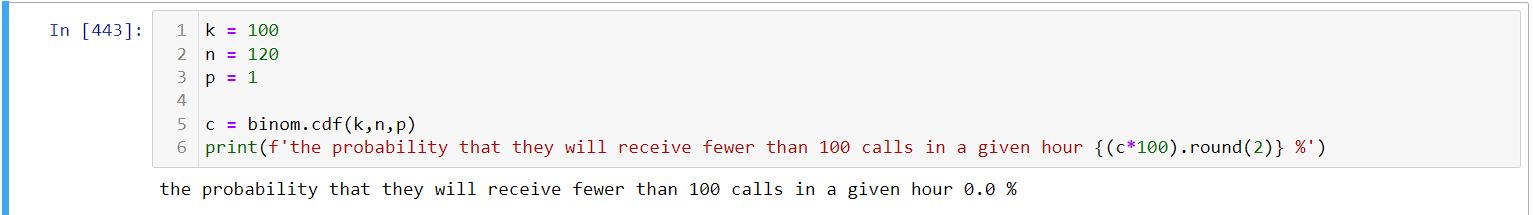
ANS: the probability that more than 220 cars will pass through this road in a given hour 7.53 %

### **Binomial Distribution:**

**Explanation**: Models the number of successes in a fixed number of independent trials, each with the same probability of success.

Real-life Example:

1. A call center receives an average of 120 calls per hour. What is the probability that they will receive fewer than 100 calls in a given hour?



k = 100

n = 120

p = 1

c = binom.cdf(k,n,p)

print(f'the probability that they will receive fewer than 100 calls in a given hour {(c\*100).round(2)} %')

ANS: **the probability that they will receive fewer than 100 calls in a given hour 0.0 %**

Explanation:

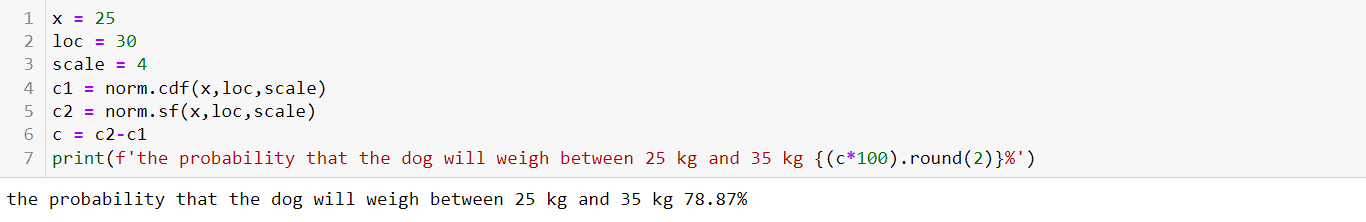
Whenever you’re provided with binom like (True or False) or (Yes or No) or (Head or Tail) means equal chance of getting both results go for binom

### **Normal Distribution:**

Explanation: Represents continuous random variables, often seen in natural phenomena like heights or test scores.

Real-life Example:

The average weight of a certain breed of dog is 30 kg with a standard deviation of 4 kg. If a person randomly selects one of these dogs, what is the probability that the dog will weigh between 25 kg and 35 kg



x = 25

loc = 30

scale = 4

c1 = norm.cdf(x,loc,scale)

c2 = norm.sf(x,loc,scale)

c = c2-c1

print(f'the probability that the dog will weigh between 25 kg and 35 kg {(c\*100).round(2)}%')

ANS: the probability that the dog will weigh between 25 kg and 35 kg 78.87%

Here x denotes number of samples and loc denotes mean and scale denotes standard deviation

# Hypothesis Testing:

Normality Test:

Explanation: Determines if a given dataset follows a normal distribution.

Real-life Example: A researcher wants to test whether the IQ scores of a sample of students follow a normal distribution.

python

Copy code

from scipy.stats import shapiro

# Generate some IQ scores (replace with actual data)

iq\_scores = np.random.normal(loc=100, scale=15, size=100)

# Perform Shapiro-Wilk test

statistic, p\_value = shapiro(iq\_scores)

# Interpret the results

alpha = 0.05

if p\_value > alpha:

print("Sample looks Gaussian (fail to reject H0)")

else:

print("Sample does not look Gaussian (reject H0)")

**Correlation Test:**

Explanation: Measures the strength and direction of the linear relationship between two variables.

Real-life Example: A researcher wants to test whether there is a significant correlation between study hours and exam scores of students.

python

Copy code

from scipy.stats import pearsonr

# Generate study hours and exam scores (replace with actual data)

study\_hours = np.random.randint(1, 10, size=100)

exam\_scores = np.random.randint(50, 100, size=100)

# Perform Pearson correlation test

corr, p\_value = pearsonr(study\_hours, exam\_scores)

# Interpret the results

alpha = 0.05

if p\_value > alpha:

print("No significant correlation (fail to reject H0)")

else:

print("Significant correlation (reject H0)")

Other Hypothesis Tests:

Independent Samples t-test:

Explanation: Compares the means of two independent samples to determine if they have different means.

Real-life Example: A researcher wants to compare the effectiveness of two different teaching methods on exam scores.

python

Copy code

from scipy.stats import ttest\_ind

# Generate data for two groups (replace with actual data)

group1\_scores = np.random.normal(loc=70, scale=10, size=100)

group2\_scores = np.random.normal(loc=75, scale=10, size=100)

# Perform independent samples t-test

statistic, p\_value = ttest\_ind(group1\_scores, group2\_scores)

# Interpret the results

alpha = 0.05

if p\_value > alpha:

print("No significant difference between the groups (fail to reject H0)")

else:

print("Significant difference between the groups (reject H0)")

Paired Samples t-test:

Explanation: Determines if there is a significant difference between two related groups.

Real-life Example: A researcher wants to test the effectiveness of a new training program by comparing employees' performance before and after the training.

python

Copy code

from scipy.stats import ttest\_rel

# Generate data for before and after scores (replace with actual data)

before\_scores = np.random.normal(loc=60, scale=10, size=100)

after\_scores = np.random.normal(loc=70, scale=10, size=100)

# Perform paired samples t-test

statistic, p\_value = ttest\_rel(before\_scores, after\_scores)

# Interpret the results

alpha = 0.05

if p\_value > alpha:

print("No significant improvement (fail to reject H0)")

else:

print("Significant improvement (reject H0)")

Analysis of Variance (ANOVA):

Explanation: Compares the means of three or more groups to determine if they are statistically different.

Real-life Example: A researcher wants to compare the effectiveness of three different training programs on exam scores.

python

Copy code

from scipy.stats import f\_oneway

# Generate data for three groups (replace with actual data)

group1\_scores = np.random.normal(loc=70, scale=10, size=100)

group2\_scores = np.random.normal(loc=75, scale=10, size=100)

group3\_scores = np.random.normal(loc=80, scale=10, size=100)

# Perform one-way ANOVA

f\_statistic, p\_value = f\_oneway(group1\_scores, group2\_scores, group3\_scores)

# Interpret the results

alpha = 0.05

if p\_value > alpha:

print("No significant difference among the groups (fail to reject H0)")

else:

print("Significant difference among the groups (reject H0)")

Chi-Square Test:

Explanation: Examines the association between two categorical variables.

Real-life Example: A researcher wants to test whether there is a significant association between gender and voting preference.

python

Copy code

from scipy.stats import chi2\_contingency

# Generate contingency table (replace with actual data)

observed = np.array([[30, 20, 10], [25, 25, 15]])

# Perform chi-square test

chi2\_statistic, p\_value, \_, \_ = chi2\_contingency(observed)

# Interpret the results

alpha = 0.05

if p\_value > alpha:

print("No significant relationship between gender and voting preference (fail to reject H0)")

else:

print("Significant relationship between gender and voting preference (reject H0)")

Mann-Whitney U Test:

Explanation: Non-parametric test to compare two independent groups when the assumptions of the t-test are not met.

Real-life Example: A researcher wants to compare the income levels between employees in two different companies.

python

Copy code

from scipy.stats import mannwhitneyu

# Generate income data for two companies (replace with actual data)

company1\_income = np.random.normal(loc=50000, scale=10000, size=100)

company2\_income = np.random.normal(loc=55000, scale=10000, size=100)

# Perform Mann-Whitney U test

u\_statistic, p\_value = mannwhitneyu(company1\_income, company2\_income)

# Interpret the results

alpha = 0.05

if p\_value > alpha:

print("No significant difference in income between the companies (fail to reject H0)")

else:

print("Significant difference in income between the companies (reject H0)")

Wilcoxon Signed-Rank Test:

Explanation: Non-parametric test to compare two related groups when the assumptions of the t-test are not met.

Real-life Example: A researcher wants to compare the job satisfaction levels before and after implementing a new policy.

python

Copy code

from scipy.stats import wilcoxon

# Generate data for before and after job satisfaction (replace with actual data)

before\_scores = np.random.randint(1, 6, size=100)

after\_scores = np.random.randint(1, 6, size=100)

# Perform Wilcoxon signed-rank test

w\_statistic, p\_value = wilcoxon(before\_scores, after\_scores)

# Interpret the results

alpha = 0.05

if p\_value > alpha:

print("No significant difference in job satisfaction (fail to reject H0)")

else:

print("Significant difference in job satisfaction (reject H0)")

I hope this consolidated overview meets your requirements. Let me know if you need further assistance!