Chapter 3: More Distributions and Central Limit Theorem

Created	@April 16, 2024 6:41 PM
Class	Introdiction to Statistics in Python

The Normal Distribution

Introduction

- One of the most important probability distributions
- Countless statistical methods rely on it
- Applies to many real-world situations

Properties of the Normal Distribution

- Symmetrical bell-curve shape
- Area under the curve is 1
- Probability never hits 0, even at the tails (0.006% beyond graph edges)
- Described by its mean and standard deviation

Standard Normal Distribution

- Special case with mean = 0 and standard deviation = 1
- Same shape as other normal distributions, but different axis scales

68-95-99.7 Rule

- 68% of the area is within 1 standard deviation of the mean
- 95% of the area is within 2 standard deviations of the mean
- 99.7% of the area is within 3 standard deviations of the mean

Real-World Example: Heights of Women

- Histogram of women's heights resembles a normal distribution
- Mean = 161 cm, Standard Deviation = 7 cm

Calculating Probabilities with the Normal Distribution

```
from scipy.stats import norm

norm.cdf(x, mean, std)
# calculates probability of less than or equal to x

# Percent of women shorter than 154 cm
norm.cdf(154, 161, 7)  # Output: 0.15865525393145707

# Percent of women taller than 154 cm
1 - norm.cdf(154, 161, 7)  # Output: 0.8413447460685429

# Percent of women between 154 and 157 cm
norm.cdf(157, 161, 7) - norm.cdf(154, 161, 7)  # Output: 0.20
795539888228836
```

Calculating Percentiles with the Normal Distribution

```
norm.ppf(prob, mean, std)
# returns the prob% significance interval for a one-tailed test
# a standard normal distribution

# Height at which 90% of women are shorter
norm.ppf(0.9, 161, 7)  # Output: 170.02626263806162

# Height at which 90% of women are taller
norm.ppf(0.1, 161, 7)  # Output: 151.97373736193838
```

```
# To perform 95% significance interval for a two-tailed test
# ppf = norm.ppf(0.975, mean, std)
# interval_value = std * ppf
# lower_95 = mean - interval_value
# upper_95 = mean + interval_value
```

Generating Random Numbers from the Normal Distribution

```
norm.rvs(161, 7, size=10)
# Generate 10 random heights with mean = 161 and std = 10
```

Conclusion

Function	Purpose
`norm.cdf`	Cumulative Distribution Function (area under the curve up to a value)
`norm.ppf`	Percent Point Function (value at which a given percentage of data falls)
`norm.rvs`	Generate random variates (random numbers) from the normal distribution

The Central Limit Theorem

Simulating Dice Rolls

```
import pandas as pd
import numpy as np

die = pd.Series([1, 2, 3, 4, 5, 6])

# Simulating 5 dice rolls
rolls = die.sample(5, replace=True)
print(rolls)
```

```
# Taking the mean of 5 rolls
mean_rolls = die.sample(5, replace=True).mean()
print(mean_rolls)
```

Sampling Distribution of the Sample Mean

- A distribution of a summary statistic (like the mean) is called a sampling distribution
- Specifically, this is a sampling distribution of the sample mean

```
# Creating a list of 10 sample means (each from 5 rolls)
sample_means = []
for i in range(10):
    sample_means.append(die.sample(5, replace=True).mean())

print(sample_means)

# Plotting the sampling distribution of the sample mean
import matplotlib.pyplot as plt
plt.hist(sample_means)
plt.show()
```

The Central Limit Theorem

- As the number of trials increases, the sampling distribution approaches a normal distribution
- Even though the original distribution (uniform for a die) is not normal

```
# Sampling distribution with 100 sample means
sample_means_100 = [die.sample(5, replace=True).mean() for _
in range(100)]
plt.figure(figsize=(8, 6))
plt.hist(sample_means_100, bins=20, density=True)
plt.show()
```

```
# Sampling distribution with 1000 sample means
sample_means_1000 = [die.sample(5, replace=True).mean() for _
in range(1000)]
plt.figure(figsize=(8, 6))
plt.hist(sample_means_1000, bins=20, density=True)
plt.show()
```

- This phenomenon is known as the Central Limit Theorem (CLT)
- CLT applies when samples are taken randomly and are independent

CLT for Other Summary Statistics

- CLT applies to other summary statistics as well
- Example: Sampling distribution of the sample standard deviation
- Example: Sampling distribution of the sample proportion (e.g., proportion of 'Claire' in sales team)

```
# Sampling distribution of sample proportions for 'Claire'
sales_team = pd.Series(['Brian', 'Claire', 'David', 'Erica'])
sample_proportions_claire = [sales_team.sample(10, replace=Tr
ue).value_counts()['Claire'] / 10 for _ in range(1000)]
plt.figure(figsize=(8, 6))
plt.hist(sample_proportions_claire, bins=20, density=True)
plt.show()
```

Estimating Distribution Characteristics

- Since the sampling distributions are normal, we can take their mean to estimate:
  - Mean of the original distribution
  - Standard deviation of the original distribution
  - Proportion of the original distribution

```
# Estimating the mean of the die distribution
mean_estimate = np.mean(sample_means_1000)
print(f"Estimated mean of die distribution: {mean_estimate:.2
f}")# Output: 3.48 (close to the expected value of 3.5)# Esti
mating the proportion of 'Claire' in the sales team
proportion_estimate = np.mean(sample_proportions_claire)
print(f"Estimated proportion of 'Claire' in sales team: {prop
ortion_estimate:.4f}")# Output: 0.2492 (close to 0.25)
```

Applications of the CLT

- Can estimate population characteristics without collecting data on the entire population
- Collect several smaller samples and create a sampling distribution
- Useful when dealing with large populations and limited resources

Conclusion

Concept	Description
Sampling Distribution	Distribution of a summary statistic (e.g., sample mean)
Central Limit Theorem	As the sample size increases, the sampling distribution approaches a normal distribution
Independence Assumption	CLT assumes samples are taken randomly and independently
Estimating Characteristics	Use the mean of the sampling distribution to estimate population parameters
Applications	Useful when dealing with large populations and limited resources

The Poisson Distribution

Poisson Process

- A process where events appear to happen at a certain rate, but completely at random
- Examples:
  - Number of animals adopted from a shelter each week
  - Number of people arriving at a restaurant each hour
  - Number of earthquakes per year in a region

Poisson Distribution

- Describes the probability of some number of events happening over a fixed period of time
- Can calculate probabilities for events like:
  - At least 5 animal adoptions in a week
  - 12 people arriving at a restaurant in an hour
  - Fewer than 20 earthquakes in a year

Parameters

- Poisson distribution is described by a parameter λ (lambda)
- λ represents the average number of events per time period
- For example, if the average number of animal adoptions per week is 8, then $\lambda = 8$
- λ is also the expected value of the distribution

Distribution Shape

- The Poisson distribution is a discrete distribution (counting events)
- The shape of the distribution depends on λ
- The peak of the distribution is always at its λ value

Visualizing the Distribution

```
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(20)
plt.bar(x, poisson.pmf(x, 1), label='λ = 1')
plt.bar(x, poisson.pmf(x, 8), alpha=0.5, label='λ = 8')
plt.legend()
plt.show()
```

Calculating Probabilities

```
from scipy.stats import poisson

# Probability of 5 adoptions in a week (λ = 8)
poisson.pmf(5, 8)  # Output: 0.09183673469387754

# Probability of 5 or fewer adoptions in a week (λ = 8)
poisson.cdf(5, 8)  # Output: 0.19742561156902045

# Probability of more than 5 adoptions in a week (λ = 8)
1 - poisson.cdf(5, 8)  # Output: 0.8025743884309795

# Probability of more than 5 adoptions in a week (λ = 10)
1 - poisson.cdf(5, 10)  # Output: 0.9299401197158094
```

Generating Random Values

```
# Simulate 10 weeks at the animal shelter (λ = 8)
poisson.rvs(8, size=10)
```

Sampling Distribution of Sample Means

- The sampling distribution of sample means from a Poisson distribution approaches a normal distribution (Central Limit Theorem)

Conclusion

Function	Purpose
`poisson.pmf`	Probability Mass Function (probability of a specific number of events)
`poisson.cdf`	Cumulative Distribution Function (probability of events <= value)
`poisson.rvs`	Generate random variates (random numbers) from the Poisson distribution

Other Probability Distributions

Exponential Distribution

- Represents the probability of time passing between Poisson events
- Examples: Time between adoptions, restaurant arrivals, earthquakes
- Uses the same lambda value (rate) as the Poisson distribution
- Continuous distribution since it represents time
- Example: If one customer service ticket is created every 2 minutes (rate = 0.5 per minute)
  - Expected value (mean time between events) = 1 / lambda = 1 / 0.5 = 2 minutes
- Probability calculations using `expon.cdf()` from `scipy.stats`:
  - `expon.cdf(1, 2)` = 0.4 (40% chance of waiting less than 1 minute)
  - `1 - expon.cdf(4, 2)` = 0.13 (13% chance of waiting more than 4 minutes)
  - `expon.cdf(4, 2) - expon.cdf(1, 2)` = 0.5 (50% chance of waiting between 1 and 4 minutes)

Student's t-Distribution

- Similar shape to the normal distribution, but with thicker tails
- Likelihood of observations falling further from the mean is higher
- Controlled by the degrees of freedom parameter
- Lower degrees of freedom = thicker tails and higher standard deviation
- As degrees of freedom increase, distribution approaches the normal distribution

Log-normal Distribution

- Logarithm of the variable is normally distributed
- Results in a skewed distribution (unlike the normal distribution)
- Real-world examples: Chess game lengths, blood pressure, SARS outbreak hospitalizations

Other Distributions

- There are many other probability distributions beyond the ones covered in this lesson

Conclusion

Distribution	Key Properties
Exponential	- Continuous distribution representing time between Poisson events - Uses lambda (rate) parameter - Expected value = 1 / lambda
Student's t	- Similar to normal distribution but with thicker tails - Likelihood of observations far from mean is higher - Controlled by degrees of freedom parameter
Log-normal	- Logarithm of variable is normally distributed - Skewed distribution - Applicable to many real-world scenarios