Essential Probability Distributions for Data Science

Contents

Introduction

This document summarizes the most important probability distributions used in data science and statistics, with:

- **Formulas** (PMF or PDF)
- **Mean** and **Variance**
- **Key Relationships**
- **Example Python Code with Plots** using NumPy, Matplotlib/Seaborn

All code snippets assume you have the following libraries:

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

sns.set_style("whitegrid")
```

1 Bernoulli Distribution

Use Case:

- Models a single trial with two outcomes (e.g., success/failure).
- Data science examples: binary classification outcomes, coin toss.

PMF

$$P(X = x) \;=\; p^x\,(1-p)^{(1-x)}, \quad x \in \{0,1\}, \quad 0 \le p \le 1$$

Mean: $\mu = p$
Variance: $\sigma^2 = p(1-p)$

Python Code with Plot

```
p = 0.3
samples_bernoulli = np.random.binomial(n=1, p=p, size=1000)

sns.histplot(samples_bernoulli, discrete=True, stat='probability')
plt.title(f"Bernoulli(p={p})")
plt.xlabel("Value")
plt.ylabel("Probability")
plt.show()
```

2 Binomial Distribution

Use Case:

- Number of successes in n independent Bernoulli trials, each with success probability p.
- Common in A/B testing, success/failure counting.

PMF

$$P(X = k) \; = \; \binom{n}{k} p^k \, (1-p)^{\,n-k}, \quad k = 0, 1, \ldots, n$$

Mean: $\mu = np$
Variance: $\sigma^2 = np(1-p)$

Relationships

- A **Bernoulli** distribution is a special case of **Binomial** when $n = 1$.
- For large n and small p (with $np = \lambda$ fixed), Binomial approximates Poisson(λ).
- For large n, Binomial approximates the **Normal** distribution with mean np and variance $np(1-p)$.

Python Code with Plot

```
n, p = 10, 0.3
samples_binomial = np.random.binomial(n=n, p=p, size=1000)

sns.histplot(samples_binomial, discrete=True, stat='probability')
plt.title(f"Binomial(n={n}, p={p})")
plt.xlabel("Number of Successes")
plt.ylabel("Probability")
plt.show()
```

3 Poisson Distribution

Use Case:

- Models the number of events in a given interval with constant average rate λ.
- Example: number of website hits per minute.

PMF

$$P(X = k) \; = \; \frac{\lambda^k \, e^{-\lambda}}{k!}, \quad k = 0, 1, 2, \ldots$$

Mean: $\mu = \lambda$
Variance: $\sigma^2 = \lambda$

Relationships

- For large λ, Poisson(λ) approximates Normal(λ, λ).
- Binomial(n, p) with large n and small p (where $np = \lambda$) approximates Poisson(λ).

Python Code with Plot

```
lam = 5
samples_poisson = np.random.poisson(lam=lam, size=1000)

sns.histplot(samples_poisson, discrete=True, stat='probability')
plt.title(f"Poisson(  ={lam})")
plt.xlabel("Number of Events")
plt.ylabel("Probability")
plt.show()
```

4 Normal (Gaussian) Distribution

Use Case:

- Continuous distribution, the classic "bell curve."
- By the Central Limit Theorem, sums/averages of large samples of i.i.d. variables tend to be Normal.

PDF

$$f(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\Bigl(-\frac{(x-\mu)^2}{2\sigma^2}\Bigr), \quad -\infty < x < \infty$$

Mean: μ
Variance: σ^2

Relationships

- Binomial(n, p) approximates Normal$(np,\, np(1-p))$ for large n.
- Poisson(λ) approximates Normal(λ, λ) for large λ.
- The t-distribution converges to Normal as $\nu \to \infty$.

Python Code with Plot

```
mu, sigma = 0, 1
samples_normal = np.random.normal(loc=mu, scale=sigma, size=1000)

sns.histplot(samples_normal, stat='density', kde=True, color='blue')
plt.title(f"Normal(  ={mu},␣  ={sigma})")
plt.xlabel("Value")
plt.ylabel("Density")
plt.show()
```

5 Uniform Distribution

Use Case:

- Constant probability across an interval $[a, b]$.
- Often used for simulation or uninformative priors in Bayesian settings.

PDF

$$f(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{otherwise} \end{cases}$$

Mean: $\mu = \dfrac{a+b}{2}$

Variance: $\sigma^2 = \dfrac{(b-a)^2}{12}$

Python Code with Plot

```
a, b = 0, 1
samples_uniform = np.random.uniform(low=a, high=b, size=1000)

sns.histplot(samples_uniform, stat='density', kde=True, color='green')
plt.title(f"Uniform(a={a}, b={b})")
plt.xlabel("Value")
plt.ylabel("Density")
plt.show()
```

6 Exponential Distribution

Use Case:

- Models the time between events in a Poisson process (rate λ).
- Has the memoryless property.

PDF

$$f(x) = \lambda\, e^{-\lambda x}, \quad x \geq 0$$

Mean: $\mu = \frac{1}{\lambda}$
Variance: $\sigma^2 = \frac{1}{\lambda^2}$

Relationships

- The Γ distribution with shape parameter $k = 1$ is exactly Exponential(λ).
- The sum of k i.i.d. Exponential(λ) variables is Gamma(k, λ).

Python Code with Plot

```
lam_exp = 2  # Rate
samples_exponential = np.random.exponential(scale=1/lam_exp, size=1000)

sns.histplot(samples_exponential, stat='density', kde=True)
plt.title(f"Exponential(  ={lam_exp})")
plt.xlabel("Value")
plt.ylabel("Density")
plt.show()
```

7 Gamma Distribution

Use Case:

- Generalization of Exponential.
- Used for waiting times, Bayesian priors (e.g., for Poisson rate parameters).

PDF (Shape-Rate Parameterization)

$$f(x) = \frac{\lambda^k}{\Gamma(k)}\, x^{k-1}\, e^{-\lambda x}, \quad x \geq 0, \quad k > 0,\ \lambda > 0$$

Mean: $\mu = \dfrac{k}{\lambda}$

Variance: $\sigma^2 = \dfrac{k}{\lambda^2}$

Relationships

- $\text{Gamma}(k = 1, \lambda) = \text{Exponential}(\lambda)$.
- $\text{Chi-square}(\nu) = \text{Gamma}(\frac{\nu}{2}, \frac{1}{2})$.

Python Code with Plot

```
k, lam_gamma = 2.0, 1.0
samples_gamma = np.random.gamma(shape=k, scale=1/lam_gamma, size=1000)

sns.histplot(samples_gamma, stat='density', kde=True)
plt.title(f"Gamma(k={k},  ={lam_gamma})")
plt.xlabel("Value")
plt.ylabel("Density")
plt.show()
```

8 Beta Distribution

Use Case:

- Models values strictly in $[0, 1]$.
- Often used as a prior over probabilities (e.g., for Bernoulli/Binomial).

PDF

$$f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}, \quad 0 \leq x \leq 1, \quad \alpha > 0,\ \beta > 0$$

where $B(\alpha, \beta)$ is the Beta function.

Mean: $\mu = \dfrac{\alpha}{\alpha + \beta}$

Variance:

$$\sigma^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

Python Code with Plot

```
alpha, beta_ = 2, 5
samples_beta = np.random.beta(alpha, beta_, size=1000)

sns.histplot(samples_beta, stat='density', kde=True)
plt.title(f"Beta( ={alpha},  ={beta_})")
plt.xlabel("Value")
plt.ylabel("Density")
plt.show()
```

9 Student's t-Distribution

Use Case:

- Used when population standard deviation is unknown, especially with small sample sizes.
- Common in hypothesis testing (t-tests) and confidence intervals.

PDF (for ν degrees of freedom)

$$f(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Mean: 0 (for $\nu > 1$)
Variance: $\frac{\nu}{\nu-2}$ (for $\nu > 2$)

Relationship

- As $\nu \to \infty$, the t-distribution approaches Normal$(0, 1)$.

Python Code with Plot

```
df = 10  # degrees of freedom
samples_t = np.random.standard_t(df, size=1000)

sns.histplot(samples_t, stat='density', kde=True)
plt.title(f"t-Distribution(df={df})")
plt.xlabel("Value")
plt.ylabel("Density")
plt.show()
```

10 Relationships and Quick Reference Table

Key Relationships

1. **Bernoulli** is a special case of **Binomial** ($n = 1$).
2. **Binomial**$(n, p) \approx$ **Poisson**$(\lambda = np)$ if n is large and p is small.
3. **Poisson**$(\lambda) \approx$ **Normal**$(\mu = \lambda, \sigma^2 = \lambda)$ if λ is large.
4. **Binomial**$(n, p) \approx$ **Normal**$(np, np(1-p))$ for large n.
5. **Gamma**$(k = 1, \lambda) =$ **Exponential**(λ).
6. **Chi-square**$(\nu) =$ **Gamma**$\left(\frac{\nu}{2}, \frac{1}{2}\right)$.
7. **t-Distribution** $\to$ Normal as $\nu \to \infty$.

Quick Reference Table

Distribution	Discrete / Continuous	Support	Parameters	Mean	Variance
Bernoulli	Discrete	$\{0, 1\}$	p	p	$p(1-p)$
Binomial	Discrete	$\{0, 1, \ldots, n\}$	n, p	np	$np(1-p)$
Poisson	Discrete	$\{0, 1, 2, \ldots\}$	λ	λ	λ
Uniform	Continuous	$[a, b]$	a, b	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
Normal	Continuous	$(-\infty, \infty)$	μ, σ^2	μ	σ^2
Exponential	Continuous	$[0, \infty)$	λ	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
Gamma	Continuous	$[0, \infty)$	k, λ	$\frac{k}{\lambda}$	$\frac{k}{\lambda^2}$
Beta	Continuous	$[0, 1]$	α, β	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$
t-Distribution	Continuous	$(-\infty, \infty)$	ν (dof)	0 (for $\nu > 1$)	$\frac{\nu}{\nu-2}$ (for $\nu > 2$)

Final Remarks

- These distributions are **central** to data science and statistics.
- Understanding their **properties** and **relationships** helps in choosing appropriate models.
- Python's `numpy` and `seaborn` libraries make it convenient to generate random samples and visualize them.

Introduction

In hypothesis testing, we use **probability distributions** to:

- Derive test statistics (e.g., z-statistic, t-statistic, chi-square statistic, etc.).
- Compute p-values, confidence intervals, and critical regions.

This document highlights several common distributions and shows how they underpin different hypothesis tests, with **Python code examples** (using `numpy` and `scipy.stats`) to illustrate practical usage.

All code snippets assume:

```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

sns.set_style("whitegrid")
```

11 Bernoulli and Binomial for Proportion Tests

11.1 Bernoulli/Binomial Overview

- **Bernoulli:** Single trial with two outcomes (0 or 1).
- **Binomial:** Number of successes in n Bernoulli trials with success probability p.

Hypothesis Testing Context:

- **One-Proportion Test (Binomial Test)**:

$$H_0 : p = p_0 \quad \text{vs.} \quad H_a : p \neq p_0 \quad \text{(or one-sided)}$$

- We often use a binomial distribution or normal approximation to test whether the true probability p differs from a hypothesized p_0.

11.2 Example: Binomial Test in Python

```python
# Suppose we observe 40 successes out of 100 trials.
# We want to test if p = 0.5.

observed_successes = 40
n = 100
p0 = 0.5

# Using scipy.stats.binom_test (deprecated in newer SciPy versions);
# for newer versions, use proportions_ztest from statsmodels.
p_value = stats.binom_test(observed_successes, n=n, p=p0, alternative='two-sided')
print("Binomial test p-value:", p_value)

# If p_value < alpha (e.g., 0.05), we reject H0 that p = 0.5.
```

Note: In newer versions of `SciPy`, `binom_test` is deprecated. You can use `statsmodels.stats.prop` for a z-based approximation.

12 Normal Distribution for z-Tests

12.1 Normal Overview

- **Normal (Gaussian)** distribution is used when sample sizes are large (by the Central Limit Theorem) or when population standard deviation is known.

Hypothesis Testing Context:

- **z-Test for Means:**

$$H_0 : \mu = \mu_0 \quad \text{vs.} \quad H_a : \mu \neq \mu_0$$

- Typically used when population variance (σ^2) is known or n is sufficiently large.

12.2 Example: z-Test for a Mean (Approximation)

In Python, pure z-tests are less common than t-tests, but we can approximate with:

```
# Synthetic data: sample from a Normal(  =0,   =1)
np.random.seed(42)
data = np.random.normal(loc=0, scale=1, size=100)

# Suppose we want to test if mean = 0 (H0:   =0).
# We'll approximate with a z-test if we assume   =1 known.

sample_mean = np.mean(data)
n = len(data)
sigma = 1.0  # known population std dev
z_stat = (sample_mean - 0) / (sigma / np.sqrt(n))

# Two-sided p-value using Normal
p_value = 2 * (1 - stats.norm.cdf(abs(z_stat)))
print("Z-statistic:", z_stat)
print("p-value:", p_value)
```

If `p_value` ¡ α (common choice 0.05), we reject H_0.

13 t-Distribution for t-Tests

13.1 Student's t Overview

- Used when the population standard deviation is unknown and estimated by the sample.
- Common in small-sample scenarios or standard parametric tests in practice.

Hypothesis Testing Context:

- **One-sample t-Test**:
$$H_0 : \mu = \mu_0$$
- **Two-sample t-Test (independent or paired)**.

13.2 Example: One-Sample t-Test in Python

```python
# Suppose we have data from an unknown distribution,
# and we want to test if the mean is 5.

data = np.array([4.9, 5.1, 5.3, 4.8, 5.2, 5.4, 5.0])
t_stat, p_val = stats.ttest_1samp(data, popmean=5)
print("One-sample t-test statistic:", t_stat)
print("p-value:", p_val)

# If p_val < alpha, reject H0: mean is 5.
```

14 Chi-square Distribution for Goodness-of-Fit or Independence

14.1 Chi-square Overview

- χ^2 distribution arises from summing squares of Normal(0,1) variables.
- Often used in **goodness-of-fit**, **independence tests** (contingency tables).

Hypothesis Testing Context:

- **Chi-square Goodness-of-Fit**:

$$H_0 : \text{The data follow a specified distribution.}$$

- **Chi-square Test of Independence** in contingency tables.

14.2 Example: Chi-square Goodness-of-Fit

```python
# Suppose we observe frequencies in 4 categories:
observed = np.array([18, 22, 15, 25])

# Suppose expected probabilities for these categories
# are [0.25, 0.25, 0.25, 0.25].
n = np.sum(observed)
expected = n * np.array([0.25, 0.25, 0.25, 0.25])

chi_stat, p_value = stats.chisquare(f_obs=observed, f_exp=expected)
print("Chi-square statistic:", chi_stat)
print("p-value:", p_value)

# If p_value < alpha, reject H0 that distribution matches [0.25, 0.25, 0.25,
    0.25].
```

15 F-Distribution for ANOVA

15.1 F-Distribution Overview

- Ratio of two scaled chi-square distributions is an F-distribution.
- Used in **ANOVA** (analysis of variance) to compare means of multiple groups.

Hypothesis Testing Context (One-Way ANOVA):

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_k \quad \text{vs.} \quad H_a : \text{at least one mean is different.}$$

15.2 Example: One-Way ANOVA

```
# Suppose we have 3 groups of data:
group1 = np.random.normal(loc=5, scale=1, size=30)
group2 = np.random.normal(loc=5.5, scale=1, size=30)
group3 = np.random.normal(loc=6, scale=1, size=30)

f_stat, p_val = stats.f_oneway(group1, group2, group3)
print("F-statistic:", f_stat)
print("p-value:", p_val)

# If p_val < alpha, reject H0 that all group means are equal.
```

16 Poisson Distribution for Rate Tests

16.1 Poisson Overview

- Models the number of events in a time/space interval with rate λ.
- Hypothesis tests often compare observed vs. expected counts at a certain rate.

Hypothesis Testing Context:

- Test whether the observed count matches a Poisson(λ_0) with a known or hypothesized rate λ_0.

16.2 Example: Test if Observed Count Matches a Poisson Rate

While not as standard as z/t-tests, we can build a likelihood-based test or use a simpler approximation:

```
# Observed count in a fixed interval
observed_count = 12
# Hypothesized rate
lambda_0 = 10

# Under H0, X ~ Poisson(lambda_0).
# Probability of observing something "as extreme or more extreme"
# can be computed as p-value.
# For a two-sided test, we might do:

p_lower = stats.poisson.cdf(observed_count, mu=lambda_0)
p_upper = 1 - stats.poisson.cdf(observed_count - 1, mu=lambda_0)
p_value_two_sided = 2 * min(p_lower, p_upper)

print("Poisson two-sided p-value:", p_value_two_sided)
```

If `p_value_two_sided` $< \alpha$, we reject $H_0 : \lambda = \lambda_0$.

17 Summary of Distributions in Hypothesis Testing

Which Distribution for Which Test?

- **Binomial** (*or Normal approx.*): Testing proportions (e.g., yes/no outcomes).
- **Normal (z-Test)**: Large-sample mean tests or known population variance.
- **t-Distribution (t-Test)**: Small-sample mean tests, unknown variance.
- **Chi-square**: Goodness-of-fit, independence tests in contingency tables.
- **F-Distribution (ANOVA)**: Comparing means of 2 or more groups.
- **Poisson**: Rate-based tests (counts over time/space).

General Steps in Hypothesis Testing

1. **Formulate** H_0 and H_a (null and alternative hypotheses).
2. **Choose** appropriate test statistic and distribution (z, t, χ^2, F, etc.).
3. **Compute** the test statistic from sample data.
4. **Obtain** the p-value or compare to a critical value.
5. **Decide** whether to reject or fail to reject H_0 based on α level.

Key Insight: Each test statistic is tied to a **reference distribution** (Normal, t, Chi-square, F, etc.) that tells us how extreme our sample results are if H_0 were true.

Conclusion

These distributions and tests form the backbone of classical hypothesis testing. Mastering their assumptions and usage is crucial for drawing valid inferences from data.