Statistics

Lecture Notes

Licence L3 — 2025–2026

Yaë Ulrich Gaba

"It is easy to lie with statistics, but it is easier to lie without them."

— Frederick Mosteller

March 25, 2026

Contents

Preface

Mathematical statistics is the science of drawing rigorous conclusions from uncertain data. Across medicine, physics, economics, and artificial intelligence, decisions rely on data analysis. This course provides the theoretical and practical foundations of the discipline.

Prerequisites. Probability theory (random variables, classical distributions, convergence), real analysis I (series, integrals), linear algebra (vector spaces, matrices).

Organization. Each chapter opens with a motivating example, develops theory rigorously, includes Python implementations, and concludes with graded exercises:

- $\star$ Computational exercises
- $\star\star$ Theoretical exercises
- $\star\star\star$ Projects

Main references.

- Casella, G. & Berger, R.L. — *Statistical Inference*, 2nd ed., Cengage.
- Wasserman, L. — *All of Statistics*, Springer.
- Saporta, G. — *Probabilités, Analyse des Données et Statistique*, Technip.
- Wackerly, Mendenhall & Scheaffer — *Mathematical Statistics with Applications.*

Chapter 1

Descriptive Statistics — Summary and Visualization

"You can't improve what you don't measure." — Lord Kelvin

1.1 Motivating Example

A company has the monthly salaries (in euros) of its 20 employees:

$$1800,\ 1950,\ 2100,\ 2100,\ 2200,\ 2300,\ 2350,\ 2400,\ 2500,\ 2600,$$

$$2650,\ 2700,\ 2800,\ 2900,\ 3100,\ 3200,\ 3500,\ 3800,\ 4500,\ 8000.$$

Natural questions: What is the "typical" salary? How spread out are the salaries? Are there outliers? To answer these, we need to *summarize* and *visualize* the data.

1.2 Population, Sample, Variables

Definition 1.1 (Population and Sample)**.** The **population** is the complete set of individuals under study. A **sample** of size n is a subset $(x_1, \ldots, x_n)$ from the population.

Definition 1.2 (Types of Variables)**.**

- **Qualitative** (categorical): values in a finite set of categories.
  - *Nominal*: no natural ordering (color, sex).
  - *Ordinal*: natural ordering (grade: fail $<$ pass $<$ good $<$ excellent).
- **Quantitative**: numerical values.
  - *Discrete*: countable set ($\mathbb{N}$, number of children).
  - *Continuous*: values in an interval of $\mathbb{R}$ (height, weight).

1.3 Measures of Central Tendency

Definition 1.3 (Arithmetic Mean)**.** For a sample $(x_1, \ldots, x_n)$, the **mean** is:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i.$$

Definition 1.4 (Median)**.** The **median** Me is the value that splits the ordered sample into two equal halves. If $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$ denotes the order statistics:

$$\mathrm{Me} = \begin{cases} x_{((n+1)/2)} & \text{if } n \text{ is odd,} \\ \frac{1}{2}\left(x_{(n/2)} + x_{(n/2+1)}\right) & \text{if } n \text{ is even.} \end{cases}$$

Definition 1.5 (Mode)**.** The **mode** is the most frequent value in the sample.

Statistical Intuition

The mean is sensitive to extreme values; the median is **robust**. In the salary example, $\bar{x} = 3007.50$ EUR is pulled upward by the 8000 EUR salary, while Me = 2625 EUR better reflects the "typical" salary.

1.4 Measures of Spread

Definition 1.6 (Variance and Standard Deviation)**.** The **sample variance** (corrected) and **standard deviation** are:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2, \qquad s = \sqrt{s^2}.$$

The uncorrected version uses n in the denominator. The factor $n-1$ (Bessel's correction) ensures s^2 is an unbiased estimator of the population variance (cf. Chapter 4).

Definition 1.7 (Range and Interquartile Range)**.**

$$\mathrm{Range} = x_{(n)} - x_{(1)}, \tag{1.1}$$

$$\mathrm{IQR} = Q_3 - Q_1, \tag{1.2}$$

where Q_1 and Q_3 are the first and third quartiles.

Definition 1.8 (Coefficient of Variation)**.** $\mathrm{CV} = s/\bar{x}$ (for $\bar{x} \neq 0$). Allows comparison of spread across variables with different units.

1.5 Shape Measures

Definition 1.9 (Skewness)**.**

$$\gamma_1 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s}\right)^3.$$

$\gamma_1 > 0$: right-skewed. $\gamma_1 < 0$: left-skewed. $\gamma_1 = 0$: symmetric.

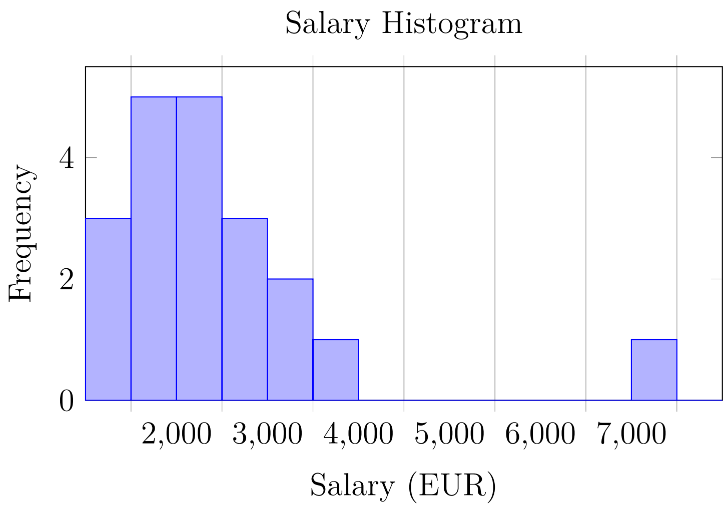
Definition 1.10 (Kurtosis)**.**

$$\gamma_2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s}\right)^4 - 3.$$

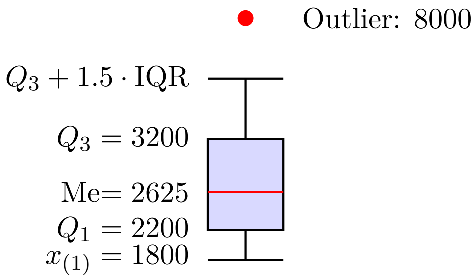
$\gamma_2 = 0$ for a normal distribution. $\gamma_2 > 0$: heavy tails (leptokurtic). $\gamma_2 < 0$: light tails (platykurtic).

1.6 Graphical Representations

1.6.1 Histogram



1.6.2 Boxplot



1.6.3 Empirical Distribution Function

Definition 1.11. The **empirical distribution function** (ECDF) is:

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{x_i \leq x}.$$

It is a step function that approximates the true CDF F (Glivenko–Cantelli theorem, cf. Chapter 2).

1.7 Bivariate Data

Definition 1.12 (Covariance and Correlation)**.** For paired data $(x_1, y_1), \ldots, (x_n, y_n)$:

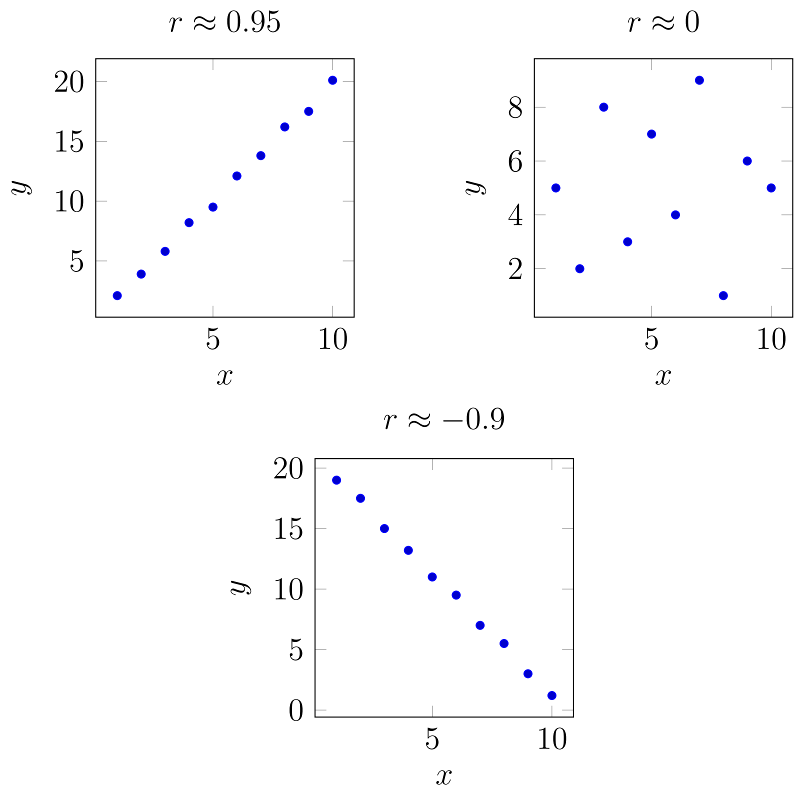
$$s_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}), \tag{1.3}$$

$$r_{xy} = \frac{s_{xy}}{s_x \cdot s_y} \in [-1, 1]. \tag{1.4}$$

r_{xy} measures the *linear association* between x and y.

Common Pitfall

Correlation does not imply causation. Two variables can be strongly correlated without a causal link (confounding variable, coincidence).



1.8 Python Implementation

Python Implementation

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats

salaries = np.array([1800, 1950, 2100, 2100, 2200, 2300, 2350, 2400,
                     2500, 2600, 2650, 2700, 2800, 2900, 3100, 3200,
                     3500, 3800, 4500, 8000])

print(f"Mean     : {np.mean(salaries):.2f}")
print(f"Median   : {np.median(salaries):.2f}")
print(f"Std dev  : {np.std(salaries, ddof=1):.2f}")
print(f"IQR      : {stats.iqr(salaries):.2f}")
print(f"Skewness : {stats.skew(salaries):.4f}")
print(f"Kurtosis : {stats.kurtosis(salaries):.4f}")

fig, axes = plt.subplots(1, 3, figsize=(14, 4))
```

```
axes[0].hist(salaries, bins=8, edgecolor='black', alpha=0.7)
axes[0].set_title("Histogram")
axes[1].boxplot(salaries, vert=True)
axes[1].set_title("Boxplot")
x_sorted = np.sort(salaries)
y_ecdf = np.arange(1, len(x_sorted)+1) / len(x_sorted)
axes[2].step(x_sorted, y_ecdf, where='post')
axes[2].set_title("ECDF")
plt.tight_layout()
plt.savefig("ch01_descriptive.pdf")
plt.show()
```

1.9 Exercises

Exercise 1.1 ($\star$ – Basic calculations)**.** Temperatures (°C) over 10 days: $12, 14, 15, 13, 16, 18, 14, 15, 17, 16$ Compute mean, median, mode, variance, standard deviation, and IQR. Plot the histogram and boxplot.

Exercise 1.2 ($\star$ – Bivariate data)**.** Compute the correlation coefficient between midterm and final exam scores for 8 students and interpret.

Exercise 1.3 ($\star\star$ – Properties of the mean)**.**

1. Show that $\sum_{i=1}^{n}(x_i - \bar{x}) = 0$.
2. Show that $\bar{x}$ minimizes $\sum_{i=1}^{n}(x_i - c)^2$ over $c \in \mathbb{R}$.
3. Deduce that $s^2 = \frac{1}{n-1}\left(\sum x_i^2 - n\bar{x}^2\right)$.

Exercise 1.4 ($\star\star$ – Empirical Chebyshev inequality)**.** Show that for any $k > 0$, the proportion of observations with $|x_i - \bar{x}| > ks$ is at most $1/k^2$.

Exercise 1.5 ($\star\star\star$ – Project: Exploratory data analysis)**.** Download a real dataset. Produce a complete report: descriptive statistics, visualizations, outlier identification, distribution shape commentary.

Key Formulas

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$\mathrm{IQR} = Q_3 - Q_1 \qquad r_{xy} = \frac{s_{xy}}{s_x s_y}$$

$$\gamma_1 = \frac{1}{n}\sum\left(\frac{x_i - \bar{x}}{s}\right)^3 \qquad \gamma_2 = \frac{1}{n}\sum\left(\frac{x_i - \bar{x}}{s}\right)^4 - 3$$

Chapter 2

Statistical Model and Sampling

"All models are wrong, some are useful." — George Box

2.1 Motivating Example

A pharmaceutical lab tests a new drug. Out of $n = 100$ patients, 68 recover. Is the drug effective? To answer, we need a *model*: assume each patient recovers independently with probability p. The 100 results form a **random sample** from Bernoulli(p).

2.2 Statistical Model

Definition 2.1 (Statistical Model)**.** A **statistical model** is a triple $(\mathcal{X}, \mathcal{A}, \mathcal{P})$ where:

- $\mathcal{X}$ is the **sample space**,
- $\mathcal{A}$ is a σ-algebra on $\mathcal{X}$,
- $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$ is a family of probability distributions indexed by $\theta \in \Theta$.

Definition 2.2 (Parametric vs. Nonparametric)**.**

- **Parametric**: $\Theta \subseteq \mathbb{R}^d$ for finite d.
- **Nonparametric**: Θ is infinite-dimensional.
- **Semiparametric**: parametric + nonparametric components.

Definition 2.3 (Identifiability)**.** The model is **identifiable** if $P_{\theta_1} = P_{\theta_2} \implies \theta_1 = \theta_2$.

2.3 Random Sample

Definition 2.4 (Random Sample)**.** A **random sample** of size n from P_θ is $(X_1, \ldots, X_n)$ where the X_i are **independent and identically distributed** (i.i.d.) with distribution P_θ. We write $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} P_\theta$.

CHAPTER 2. STATISTICAL MODEL AND SAMPLING

2.4 Statistics and Sampling Distributions

Definition 2.5 (Statistic)**.** A **statistic** is a measurable function $T = T(X_1, \ldots, X_n)$ that does *not* depend on θ.

Definition 2.6 (Sufficient Statistic)**.** T is **sufficient** for θ if the conditional distribution of $(X_1, \ldots, X_n)$ given T does not depend on θ.

Theorem 2.7 (Fisher–Neyman Factorization)**.** *$T(\mathbf{X})$ is sufficient for θ if and only if:*

$$f(\mathbf{x}; \theta) = g(T(\mathbf{x}), \theta) \cdot h(\mathbf{x}).$$

Proof. (Discrete case.) If T is sufficient, set $g(t, \theta) = P_\theta(T = t)$ and $h(\mathbf{x}) = P(\mathbf{X} = \mathbf{x} \mid T = T(\mathbf{x}))$. Conversely, if the factorization holds:

$$P_\theta(\mathbf{X} = \mathbf{x} \mid T = t) = \frac{h(\mathbf{x})}{\sum_{\mathbf{y}:T(\mathbf{y})=t} h(\mathbf{y})},$$

which does not depend on θ. □

2.5 Key Sampling Results

Theorem 2.8 (Mean and Variance of $\bar{X}$)**.** *If $X_1, \ldots, X_n \overset{i.i.d.}{\sim}$ with $\mathbb{E}[X_i] = \mu$, $\mathrm{Var}(X_i) = \sigma^2$: $\mathbb{E}[\bar{X}] = \mu$, $\mathrm{Var}(\bar{X}) = \sigma^2/n$.*

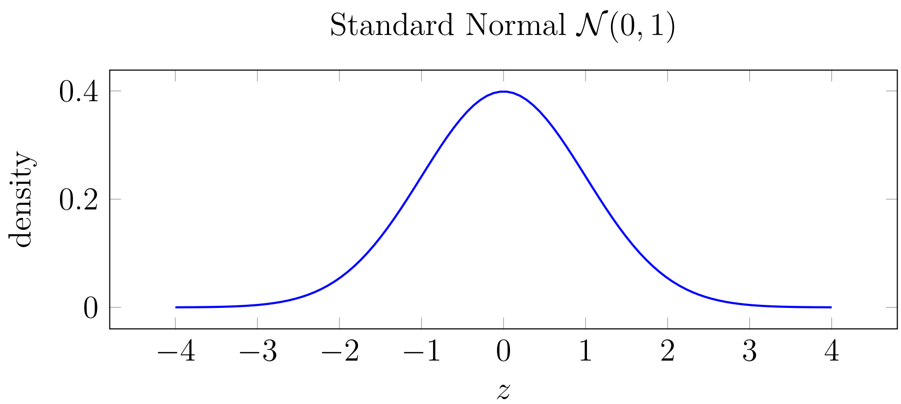
Theorem 2.9 (Law of Large Numbers)**.** *$\bar{X} \xrightarrow{\mathbb{P}} \mu$ (weak) and $\bar{X} \xrightarrow{a.s.} \mu$ (strong) as $n \to \infty$.*

Theorem 2.10 (Central Limit Theorem (CLT))**.** *If $\mathbb{E}[X_i] = \mu$ and $0 < \mathrm{Var}(X_i) = \sigma^2 < \infty$:*

$$\frac{\sqrt{n}(\bar{X} - \mu)}{\sigma} \xrightarrow{\mathcal{L}} \mathcal{N}(0, 1).$$

Statistical Intuition

The CLT explains the ubiquity of the normal distribution: whenever a quantity results from the **sum of many small independent effects**, it is approximately Gaussian.



2.6 Distribution of the Sample Variance

Theorem 2.11 (Distribution of S^2 under Normality)**.** *If* $X_1, \dots, X_n \overset{i.i.d.}{\sim} \mathcal{N}(\mu, \sigma^2)$*:*

1. $\bar{X}$ *and* S^2 *are independent (Cochran's theorem),*
2. $(n-1)S^2/\sigma^2 \sim \chi^2(n-1)$*,*
3. $(\bar{X} - \mu)/(S/\sqrt{n}) \sim t(n-1)$*.*

Theorem 2.12 (Glivenko–Cantelli)**.** $\sup_x |\hat{F}_n(x) - F(x)| \xrightarrow{a.s.} 0$.

2.7 Python Implementation

Python Implementation

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

np.random.seed(42)
n_values = [1, 2, 5, 30]
fig, axes = plt.subplots(1, 4, figsize=(16, 3.5))
for ax, n in zip(axes, n_values):
    means = [np.mean(np.random.exponential(1, n)) for _ in range(10000)]
    ax.hist(means, bins=40, density=True, alpha=0.7, edgecolor='black')
    ax.set_title(f"n = {n}")
    if n >= 5:
        x = np.linspace(min(means), max(means), 100)
        ax.plot(x, stats.norm.pdf(x, 1, 1/np.sqrt(n)), 'r-', lw=2)
plt.suptitle("CLT: means of exponential samples")
plt.tight_layout()
plt.savefig("ch02_clt.pdf")
plt.show()
```

2.8 Exercises

Exercise 2.1 ($\star$)**.** Sample of size $n = 25$ from $\mathcal{N}(10, 4)$. Compute $\mathbb{E}[\bar{X}]$, $\mathrm{Var}(\bar{X})$, and $\mathbb{P}(|\bar{X} - 10| > 0.5)$.

Exercise 2.2 ($\star$)**.** Generate 10,000 sample means of size $n = 30$ from Unif$([0, 1])$. Verify graphically that the CLT holds.

Exercise 2.3 ($\star\star$ – Sufficiency)**.** Show that $T = \sum X_i$ is sufficient for λ when $X_i \overset{\text{i.i.d.}}{\sim}$ Pois(λ).

Exercise 2.4 ($\star\star$ – CLT proof via characteristic functions)**.** Prove the CLT using the expansion $\varphi(t) = 1 - t^2/2 + o(t^2)$.

Exercise 2.5 ($\star\star\star$ – Project: Monte Carlo convergence study)**.** Study CLT convergence speed for exponential, Bernoulli, uniform, and Cauchy parent distributions.

CHAPTER 2. STATISTICAL MODEL AND SAMPLING

Key Formulas

$$\mathbb{E}[\bar{X}] = \mu \qquad \operatorname{Var}(\bar{X}) = \sigma^2/n$$

$$\frac{\sqrt{n}(\bar{X} - \mu)}{\sigma} \xrightarrow{\mathcal{L}} \mathcal{N}(0,1) \qquad \frac{(n-1)S^2}{\sigma^2} \sim \chi^2(n-1)$$

$$\frac{\bar{X} - \mu}{S/\sqrt{n}} \sim t(n-1) \qquad \sup_x |\hat{F}_n - F| \xrightarrow{\text{a.s.}} 0$$

Fisher–Neyman: $f(\mathbf{x};\theta) = g(T(\mathbf{x}),\theta) \cdot h(\mathbf{x}) \iff T$ sufficient.

Chapter 3

Point Estimation — MLE and Method of Moments

"The art of statistics is choosing the right numerical summary."

3.1 Motivating Example

Lifetimes (hours) of 10 LED bulbs: $1200, 1350, 980, 1500, 1100, 1450, 1300, 1050, 1400, 1250$. Model as $\mathrm{Exp}(\lambda)$. How to estimate λ?

3.2 General Framework

Definition 3.1 (Estimator)**.** An **estimator** of θ is a statistic $\hat{\theta} = T(X_1, \ldots, X_n)$ taking values in Θ.

3.3 Method of Moments (MoM)

Definition 3.2 (Method of Moments)**.** Equate the first $k = \dim(\theta)$ population moments to sample moments:

$$\mathbb{E}_\theta[X^j] = \frac{1}{n}\sum_{i=1}^{n} X_i^j, \quad j = 1, \ldots, k.$$

Example 3.3 (Normal distribution)**.** $\hat{\mu}_{\mathrm{MoM}} = \bar{X}$, $\hat{\sigma}^2_{\mathrm{MoM}} = \frac{1}{n}\sum(X_i - \bar{X})^2$.

Example 3.4 (Exponential distribution)**.** $\mathbb{E}[X] = 1/\lambda$, so $\hat{\lambda}_{\mathrm{MoM}} = 1/\bar{X}$.

3.4 Maximum Likelihood Estimation (MLE)

Definition 3.5 (Likelihood Function)**.**

$$L(\theta) = \prod_{i=1}^{n} f(x_i; \theta), \qquad \ell(\theta) = \ln L(\theta) = \sum_{i=1}^{n} \ln f(x_i; \theta).$$

Definition 3.6 (Maximum Likelihood Estimator)**.**

$$\hat{\theta}_{\mathrm{MLE}} = \arg\max_{\theta \in \Theta} \ell(\theta).$$

3.4.1 Fundamental Examples

Example 3.7 (Bernoulli). $\hat{p}_{\text{MLE}} = \bar{X}$.

Example 3.8 (Normal). $\hat{\mu}_{\text{MLE}} = \bar{X}$, $\hat{\sigma}^2_{\text{MLE}} = \frac{1}{n}\sum(X_i - \bar{X})^2$.

Common Pitfall

The MLE of σ^2 uses divisor n, which is **biased**. The corrected version S^2 (divisor $n-1$) is unbiased (cf. Chapter 4).

Example 3.9 (Exponential). $\hat{\lambda}_{\text{MLE}} = 1/\bar{X}$.

Example 3.10 (Poisson). $\hat{\lambda}_{\text{MLE}} = \bar{X}$.

Example 3.11 (Uniform Unif($[0,\theta]$)). $L(\theta) = \theta^{-n}\mathbf{1}_{\theta \geq x_{(n)}}$, decreasing for $\theta \geq x_{(n)}$. So $\hat{\theta}_{\text{MLE}} = X_{(n)} = \max_i X_i$. The score equation does not apply (boundary maximum).

3.5 Properties of the MLE

Theorem 3.12 (Invariance). *If $\hat{\theta}$ is the MLE of θ, then $g(\hat{\theta})$ is the MLE of $g(\theta)$.*

Theorem 3.13 (Consistency). *Under regularity conditions, $\hat{\theta}_{\text{MLE}} \xrightarrow{\mathbb{P}} \theta_0$.*

Theorem 3.14 (Asymptotic Normality). *Under regularity conditions:*

$$\sqrt{n}(\hat{\theta}_{\text{MLE}} - \theta_0) \xrightarrow{\mathcal{L}} \mathcal{N}\left(0, \frac{1}{I(\theta_0)}\right).$$

3.6 Fisher Information

Definition 3.15 (Fisher Information).

$$I(\theta) = \mathbb{E}_\theta\left[\left(\frac{\partial \ln f(X;\theta)}{\partial \theta}\right)^2\right] = -\mathbb{E}_\theta\left[\frac{\partial^2 \ln f(X;\theta)}{\partial \theta^2}\right].$$

For n observations: $I_n(\theta) = nI(\theta)$.

Example 3.16. $X \sim \mathcal{N}(\mu, \sigma^2)$ (σ^2 known): $I(\mu) = 1/\sigma^2$.

3.7 Python Implementation

Python Implementation

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

```python
data = np.array([1200, 1350, 980, 1500, 1100, 1450, 1300, 1050, 1400, 1250])

# MoM and MLE for Exp(lambda)
lambda_hat = 1 / np.mean(data)
print(f"lambda_hat = {lambda_hat:.6f}")

# MLE for Normal
mu_mle = np.mean(data)
sigma2_mle = np.var(data, ddof=0)
sigma2_unbiased = np.var(data, ddof=1)
print(f"Normal MLE: mu={mu_mle:.2f}, sigma^2={sigma2_mle:.2f}")
print(f"Unbiased:   sigma^2={sigma2_unbiased:.2f}")

# Log-likelihood plot
lambdas = np.linspace(0.0005, 0.002, 200)
log_lik = [np.sum(stats.expon.logpdf(data, scale=1/lam)) for lam in lambdas]
plt.figure(figsize=(8, 5))
plt.plot(lambdas, log_lik, 'b-', lw=2)
plt.axvline(lambda_hat, color='r', ls='--', label=f'MLE = {lambda_hat:.5f}')
plt.xlabel(r'$\lambda$'); plt.ylabel(r'$\ell(\lambda)$')
plt.title('Exponential log-likelihood'); plt.legend()
plt.savefig("ch03_loglik.pdf"); plt.show()

# Fisher information
I_fisher = 1 / lambda_hat**2
var_asymp = 1 / (len(data) * I_fisher)
print(f"Asymptotic SE: {np.sqrt(var_asymp):.6f}")
```

3.8 Exercises

Exercise 3.1 ($\star$)**.** Compute the MLE for (a) Geometric(p), (b) Unif($[a, b]$) with both a, b unknown.

Exercise 3.2 ($\star$)**.** Find MoM estimators for Beta(α, β).

Exercise 3.3 ($\star\star$ – Invariance)**.** Under $\mathcal{N}(\mu, \sigma^2)$, find the MLE of $\mathbb{P}(X > c)$.

Exercise 3.4 ($\star\star$ – Fisher information)**.** Compute $I(\theta)$ for Bernoulli, Poisson, and the Fisher information matrix for $\mathcal{N}(\mu, \sigma^2)$.

Exercise 3.5 ($\star\star\star$ – Project: MoM vs MLE comparison)**.** For $\Gamma(\alpha, \beta)$: simulate and compare bias and MSE of MoM vs MLE for $n = 20, 50, 200$.

CHAPTER 3. POINT ESTIMATION — MLE AND METHOD OF MOMENTS

Key Formulas

$$\begin{aligned}
L(\theta) &= \prod f(x_i;\theta), \quad \ell(\theta) = \sum \ln f(x_i;\theta) \\
\hat{\theta}_{\text{MLE}} &= \arg\max_{\theta} \ell(\theta) \\
I(\theta) &= -\mathbb{E}\left[\frac{\partial^2 \ln f}{\partial \theta^2}\right] \\
\sqrt{n}(\hat{\theta}_{\text{MLE}} - \theta) &\xrightarrow{\mathcal{L}} \mathcal{N}(0, 1/I(\theta))
\end{aligned}$$

Chapter 4

Estimator Properties — Bias, Consistency, Efficiency

"A good estimator should aim true and not scatter too much."

4.1 Motivating Example

Two shooters aim at a target. The first hits the center on average but is scattered. The second clusters tightly but consistently to the right. Who is "better"? This is the bias-variance tradeoff.

4.2 Bias

Definition 4.1 (Bias)**.** $\mathrm{Bias}(\hat{\theta}) = \mathbb{E}_\theta[\hat{\theta}] - \theta$. The estimator is **unbiased** if $\mathrm{Bias}(\hat{\theta}) = 0$ for all θ.

Proposition 4.2. *$S^2 = \frac{1}{n-1}\sum(X_i - \bar{X})^2$ is unbiased for σ^2.*

Proof. $\mathbb{E}[\sum(X_i - \bar{X})^2] = n(\sigma^2 + \mu^2) - n(\sigma^2/n + \mu^2) = (n-1)\sigma^2$. □

4.3 Mean Squared Error

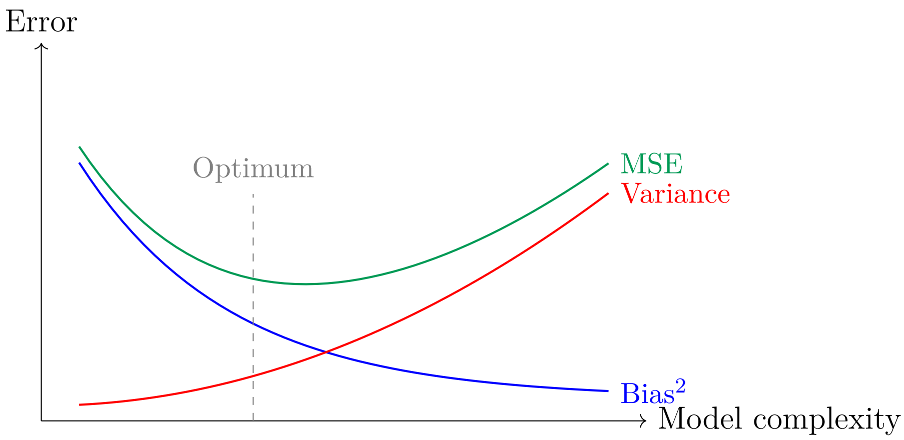
Definition 4.3 (MSE)**.** $\mathrm{MSE}(\hat{\theta}) = \mathbb{E}_\theta[(\hat{\theta} - \theta)^2]$.

Theorem 4.4 (Bias-Variance Decomposition)**.** $\mathrm{MSE}(\hat{\theta}) = \mathrm{Var}(\hat{\theta}) + [\mathrm{Bias}(\hat{\theta})]^2$.

Proof. Let $b = \mathbb{E}[\hat{\theta}] - \theta$. Then $\mathrm{MSE} = \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}] + b)^2] = \mathrm{Var}(\hat{\theta}) + b^2$. □

Statistical Intuition

MSE combines two error sources. A biased estimator can have lower MSE than an unbiased one if its variance is sufficiently reduced. This is the **bias-variance tradeoff**, fundamental in statistical learning.



4.4 Consistency

Definition 4.5 (Consistent Estimator). $\hat{\theta}_n$ is **consistent** if $\hat{\theta}_n \xrightarrow{\mathbb{P}} \theta$.

Proposition 4.6. *If* $\mathrm{Bias}(\hat{\theta}_n) \to 0$ *and* $\mathrm{Var}(\hat{\theta}_n) \to 0$*, then* $\hat{\theta}_n$ *is consistent.*

4.5 Efficiency and Cramér–Rao Bound

Theorem 4.7 (Cramér–Rao Inequality). *Under regularity conditions, for any unbiased estimator* $\hat{\theta}$ *of* $g(\theta)$*:*

$$\mathrm{Var}_\theta(\hat{\theta}) \geq \frac{[g'(\theta)]^2}{nI(\theta)}.$$

Proof. By Cauchy–Schwarz. Let $V = \sum \frac{\partial}{\partial\theta} \ln f(X_i;\theta)$ (total score). Then $\mathbb{E}[V] = 0$, $\mathrm{Var}(V) = nI(\theta)$, and $\mathrm{Cov}(\hat{\theta}, V) = g'(\theta)$. So $[g'(\theta)]^2 \leq \mathrm{Var}(\hat{\theta}) \cdot nI(\theta)$. □

Definition 4.8 (Efficient Estimator). $\hat{\theta}$ is **efficient** if it attains the Cramér–Rao bound.

Example 4.9. For $X_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, \sigma^2)$ (σ^2 known): $\mathrm{Var}(\bar{X}) = \sigma^2/n = 1/(nI(\mu))$. So $\bar{X}$ is efficient.

4.6 UMVUE and Rao–Blackwell

Theorem 4.10 (Rao–Blackwell). *If* $\hat{\theta}$ *is unbiased for* $g(\theta)$ *and* T *is sufficient, then* $\hat{\theta}^* = \mathbb{E}[\hat{\theta} \mid T]$ *is unbiased with* $\mathrm{Var}(\hat{\theta}^*) \leq \mathrm{Var}(\hat{\theta})$.

Theorem 4.11 (Lehmann–Scheffé). *If* T *is sufficient and complete, then any unbiased estimator that is a function of* T *is UMVUE.*

4.7 Python Implementation

Python Implementation

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

np.random.seed(42)
sigma2_true = 4; n = 10; n_sim = 50000

biased = np.array([np.var(np.random.normal(0, 2, n), ddof=0) for _ in
↪  range(n_sim)])
unbiased = np.array([np.var(np.random.normal(0, 2, n), ddof=1) for _ in
↪  range(n_sim)])

print(f"MLE (n):  E={np.mean(biased):.4f},
↪  MSE={np.mean((biased-sigma2_true)**2):.4f}")
print(f"S^2 (n-1): E={np.mean(unbiased):.4f},
↪  MSE={np.mean((unbiased-sigma2_true)**2):.4f}")
print(f"Cramer-Rao bound for mu: {sigma2_true/n:.4f}")

fig, ax = plt.subplots(figsize=(8, 5))
ax.hist(biased, bins=50, alpha=0.5, density=True, label='MLE (n)')
ax.hist(unbiased, bins=50, alpha=0.5, density=True, label='$S^2$ (n-1)')
ax.axvline(sigma2_true, color='red', lw=2,
↪  label=f'$\\sigma^2={sigma2_true}$')
ax.legend(); ax.set_title(f'Estimators of $\\sigma^2$ (n={n})')
plt.savefig("ch04_bias_variance.pdf"); plt.show()
```

4.8 Exercises

Exercise 4.1 ($\star$)**.** Is $\hat{\lambda} = 1/\bar{X}$ unbiased for λ when $X_i \overset{\text{i.i.d.}}{\sim} \text{Exp}(\lambda)$?

Exercise 4.2 ($\star\star$ – Cramér–Rao)**.** Show that $\bar{X}$ attains the CR bound for p in Bernoulli(p).

Exercise 4.3 ($\star\star$ – Rao–Blackwell)**.** For $X_i \overset{\text{i.i.d.}}{\sim} \text{Pois}(\lambda)$, improve $\hat{\theta} = \mathbf{1}_{X_1=0}$ as an estimator of $e^{-\lambda}$ using $T = \sum X_i$.

Exercise 4.4 ($\star\star$ – Bias-variance tradeoff)**.** For $\hat{\theta}_c = c \cdot S^2$ estimating σ^2, find the c^* minimizing MSE.

Exercise 4.5 ($\star\star\star$ – Project)**.** Verify numerically that $\bar{X}$ is efficient for Poisson(λ). Compare MSE of mean, median, and sample variance as estimators of λ.

Key Formulas

$$\begin{aligned}
\mathrm{Bias}(\hat{\theta}) &= \mathbb{E}[\hat{\theta}] - \theta \\
\mathrm{MSE}(\hat{\theta}) &= \mathrm{Var}(\hat{\theta}) + [\mathrm{Bias}(\hat{\theta})]^2 \\
\text{CR bound}: \mathrm{Var}(\hat{\theta}) &\geq \frac{[g'(\theta)]^2}{nI(\theta)} \\
\text{Rao–Blackwell}: \hat{\theta}^* &= \mathbb{E}[\hat{\theta} \mid T],\ \mathrm{Var}(\hat{\theta}^*) \leq \mathrm{Var}(\hat{\theta})
\end{aligned}$$

Chapter 5

Confidence Intervals

"An estimate without a measure of uncertainty is an incomplete estimate."

5.1 Motivating Example

A poll of $n = 1000$ people finds 520 in favor of a reform: $\hat{p} = 0.52$. How precise is this? A confidence interval answers: "with 95% confidence, $p \in [0.489, 0.551]$."

5.2 Definition and Interpretation

Definition 5.1 (Confidence Interval). A **confidence interval** (CI) of level $1 - \alpha$ for θ is $[L(\mathbf{X}), U(\mathbf{X})]$ such that $\mathbb{P}_\theta(L \leq \theta \leq U) \geq 1 - \alpha$ for all θ.

Common Pitfall

The correct interpretation is **frequentist**: if we repeat the experiment many times, $(1 - \alpha) \times 100\%$ of constructed CIs will contain θ.
Common error: "There is a 95% probability that θ is in the CI." No! θ is fixed; the interval is random. After observation, the CI either contains θ or it doesn't.

Definition 5.2 (Pivotal Quantity). A **pivotal quantity** is $Q(X_1, \ldots, X_n, \theta)$ whose distribution does not depend on θ.

5.3 CI for the Mean

Theorem 5.3 (Known variance). *Under $\mathcal{N}(\mu, \sigma^2)$ with σ^2 known:*

$$\bar{X} \pm z_{\alpha/2} \frac{\sigma}{\sqrt{n}}.$$

Theorem 5.4 (Unknown variance). *Under $\mathcal{N}(\mu, \sigma^2)$ with σ^2 unknown:*

$$\bar{X} \pm t_{\alpha/2, n-1} \frac{S}{\sqrt{n}}.$$

CHAPTER 5. CONFIDENCE INTERVALS

5.4 CI for the Variance

Theorem 5.5. *Under normality:*

$$\left[\frac{(n-1)S^2}{\chi^2_{\alpha/2,n-1}}, \frac{(n-1)S^2}{\chi^2_{1-\alpha/2,n-1}}\right].$$

5.5 CI for a Proportion

Theorem 5.6 (Wald CI)**.** *For large n:*

$$\hat{p} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}.$$

Common Pitfall

The Wald CI can perform poorly for p near 0 or 1, or small n. The **Wilson interval** is preferable:

$$\tilde{p} = \frac{\hat{p}n + z^2/2}{n+z^2}, \quad \tilde{p} \pm z_{\alpha/2}\sqrt{\frac{\tilde{p}(1-\tilde{p})}{n+z^2}}.$$

5.6 CI for Two Means

With common variance σ^2 unknown:

$$(\bar{X}-\bar{Y}) \pm t_{\alpha/2,\nu} \cdot S_p\sqrt{1/n_1 + 1/n_2}, \quad S_p^2 = \frac{(n_1-1)S_1^2 + (n_2-1)S_2^2}{n_1+n_2-2}.$$

5.7 Sample Size Determination

For margin of error E: $n \geq (z_{\alpha/2}\sigma/E)^2$. For proportions: $n \geq z^2_{\alpha/2}/(4E^2)$.

5.8 Python Implementation

Python Implementation

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

data = np.array([12.1, 11.8, 12.5, 12.3, 11.9, 12.7, 12.0, 12.4, 11.7,
    12.2])
n = len(data); xbar = np.mean(data); s = np.std(data, ddof=1)
alpha = 0.05; t_crit = stats.t.ppf(1-alpha/2, df=n-1)
ci = (xbar - t_crit*s/np.sqrt(n), xbar + t_crit*s/np.sqrt(n))
print(f"95% CI for mu: [{ci[0]:.4f}, {ci[1]:.4f}]")

# Proportion CI (Wald and Wilson)
```

```
n_poll, x_fav = 1000, 520
p_hat = x_fav/n_poll; z = stats.norm.ppf(0.975)
margin = z*np.sqrt(p_hat*(1-p_hat)/n_poll)
print(f"Wald CI:   [{p_hat-margin:.4f}, {p_hat+margin:.4f}]")
p_t = (x_fav + z**2/2)/(n_poll + z**2)
m_w = z*np.sqrt(p_t*(1-p_t)/(n_poll+z**2))
print(f"Wilson CI: [{p_t-m_w:.4f}, {p_t+m_w:.4f}]")

# Coverage visualization
mu_true, sigma_true, n_ci = 12.0, 0.3, 10
fig, ax = plt.subplots(figsize=(10, 6))
for i in range(50):
    sample = np.random.normal(mu_true, sigma_true, n_ci)
    m, se = np.mean(sample), np.std(sample, ddof=1)/np.sqrt(n_ci)
    t_c = stats.t.ppf(0.975, df=n_ci-1)
    lo, hi = m-t_c*se, m+t_c*se
    color = 'blue' if lo <= mu_true <= hi else 'red'
    ax.plot([lo, hi], [i, i], color=color, lw=1.5)
ax.axvline(mu_true, color='green', lw=2, ls='--')
ax.set_title('50 CIs at 95%: red ones miss $\\mu$')
plt.savefig("ch05_coverage.pdf"); plt.show()
```

5.9 Exercises

Exercise 5.1 ($\star$). 15 tea bags: $\bar{x} = 2.03$g, $s = 0.12$g. Build a 99% CI for the mean weight.

Exercise 5.2 ($\star$). 23 defective out of 500 items. Build 95% CIs (Wald and Wilson) for the defect rate.

Exercise 5.3 ($\star\star$). Find n to estimate mean height within 1 cm at 95% confidence ($\sigma \approx 8$ cm).

Exercise 5.4 ($\star\star\star$ – Project). Compare Wald and Wilson coverage rates for $p = 0.05$ and $n = 20, 50, 100, 500$ via Monte Carlo.

Key Formulas

$$\text{Mean } (\sigma \text{ known}) : \bar{X} \pm z_{\alpha/2}\sigma/\sqrt{n}$$
$$\text{Mean } (\sigma \text{ unknown}) : \bar{X} \pm t_{\alpha/2,n-1}S/\sqrt{n}$$
$$\text{Proportion} : \hat{p} \pm z_{\alpha/2}\sqrt{\hat{p}(1-\hat{p})/n}$$
$$\text{Variance} : \left[\frac{(n-1)S^2}{\chi^2_{\alpha/2}}, \frac{(n-1)S^2}{\chi^2_{1-\alpha/2}}\right]$$

CHAPTER 5. CONFIDENCE INTERVALS

Chapter 6

Hypothesis Testing — General Framework

"Absence of evidence is not evidence of absence." — Carl Sagan

6.1 Motivating Example

A manufacturer claims screws have mean strength 50 kN. A client samples $n = 25$: $\bar{x} = 48.2$ kN, $s = 4.5$ kN. Is the difference significant or due to sampling variability?

6.2 Hypotheses

Definition 6.1 (Null and Alternative Hypotheses)**.** H_0: **null hypothesis** (status quo). H_1: **alternative hypothesis**.

Type	H_0	H_1
Two-sided	$\theta = \theta_0$	$\theta \neq \theta_0$
One-sided left	$\theta \geq \theta_0$	$\theta < \theta_0$
One-sided right	$\theta \leq \theta_0$	$\theta > \theta_0$

6.3 Type I and Type II Errors

Definition 6.2.

	H_0 true	H_0 false
Reject H_0	**Type I error** (α)	Correct
Fail to reject	Correct	**Type II error** (β)

α: significance level. $1 - \beta$: **power**.

6.4 Test Structure

Definition 6.3 (p-value)**.** The probability, under H_0, of observing a test statistic at least as extreme as the one observed. Reject H_0 iff $p \leq \alpha$.

Common Pitfall

The p-value is **NOT** the probability that H_0 is true. Other pitfalls:

- $p > 0.05$ does not mean H_0 is true—only that there is insufficient evidence against it.
- Statistical significance ($p < 0.05$) does not imply practical significance.

6.5 Power Function

Definition 6.4 (Power Function)**.** $\pi(\theta) = \mathbb{P}_\theta(\text{reject } H_0)$. For $\theta \in \Theta_0$: $\pi(\theta) \leq \alpha$. For $\theta \in \Theta_1$: $\pi(\theta) = 1 - \beta(\theta)$.

6.6 Neyman–Pearson Lemma

Theorem 6.5 (Neyman–Pearson)**.** *For simple $H_0 : \theta = \theta_0$ vs $H_1 : \theta = \theta_1$, the most powerful test rejects when:*

$$\Lambda(\mathbf{x}) = \frac{L(\theta_1; \mathbf{x})}{L(\theta_0; \mathbf{x})} > k_\alpha.$$

6.7 Generalized Likelihood Ratio Test

Definition 6.6.

$$\lambda(\mathbf{x}) = \frac{\sup_{\Theta_0} L(\theta; \mathbf{x})}{\sup_{\Theta} L(\theta; \mathbf{x})}.$$

Reject H_0 for small λ.

Theorem 6.7 (Wilks' Theorem)**.** *Under H_0: $-2\ln\lambda \xrightarrow{\mathcal{L}} \chi^2(r)$ where $r = \dim(\Theta) - \dim(\Theta_0)$.*

6.8 Python Implementation

Python Implementation

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

data = np.array([48.2, 47.5, 49.1, 46.8, 50.3, 48.7, 47.9, 49.5,
                 46.2, 48.8, 49.0, 47.3, 48.1, 50.1, 47.0,
                 49.4, 48.6, 46.5, 49.8, 47.2, 48.3, 49.7,
                 47.8, 48.5, 46.9])

t_stat, p_val = stats.ttest_1samp(data, 50)
print(f"t = {t_stat:.4f}, p = {p_val:.6f}")
```

```python
# Power function
from scipy.stats import norm
def power_z(mu, mu0, sigma, n, alpha=0.05):
    z_a = norm.ppf(1-alpha/2)
    d = (mu-mu0)/(sigma/np.sqrt(n))
    return 1 - norm.cdf(z_a-d) + norm.cdf(-z_a-d)

mu_vals = np.linspace(44, 56, 200)
plt.plot(mu_vals, [power_z(m, 50, 4.5, 25) for m in mu_vals], 'b-', lw=2)
plt.axhline(0.05, color='red', ls='--')
plt.xlabel(r'$\mu$'); plt.ylabel('Power')
plt.title('Power function'); plt.savefig("ch06_power.pdf"); plt.show()
```

6.9 Exercises

Exercise 6.1 ($\star$)**.** A teacher claims average grade is 12. With $n = 20$, $\bar{x} = 11.2$, $s = 2.5$, test $H_0 : \mu = 12$ vs $H_1 : \mu < 12$ at 5%.

Exercise 6.2 ($\star\star$)**.** Apply Neyman–Pearson for $H_0 : \mu = 0$ vs $H_1 : \mu = 1$ with $X_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, 1)$.

Exercise 6.3 ($\star\star$)**.** For $H_0 : \mu = 100$, $\sigma = 15$, $\alpha = 0.05$: compute power at $\mu_1 = 105$ with $n = 25$. Find n for 90% power.

Exercise 6.4 ($\star\star\star$ – Project)**.** Simulate Type I error rates and power for the t-test under normal, exponential, and uniform distributions.

Key Formulas

$$\alpha = \mathbb{P}(\text{reject } H_0 \mid H_0) \qquad \beta = \mathbb{P}(\text{fail to reject} \mid H_1)$$
$$\text{NP} : \text{reject if } L(\theta_1)/L(\theta_0) > k_\alpha \qquad \lambda = \sup_{\Theta_0} L / \sup_{\Theta} L$$

Wilks: $-2 \ln \lambda \xrightarrow{\mathcal{L}} \chi^2(r)$.

Chapter 7

Classical Tests: z, t, χ^2, F

"Theory without practice is sterile; practice without theory is blind."

7.1 Motivating Example

Two machines A and B produce parts. We want to compare precision (variance) and mean length. Machine A: $n_1 = 20$, $\bar{x}_1 = 50.2$, $s_1 = 1.5$. Machine B: $n_2 = 25$, $\bar{x}_2 = 49.5$, $s_2 = 2.1$.

7.2 z-Test (Known Variance)

For $H_0 : \mu = \mu_0$ with σ known: $Z = (\bar{X} - \mu_0)/(\sigma/\sqrt{n}) \sim \mathcal{N}(0, 1)$.

Proportions: $Z = (\hat{p} - p_0)/\sqrt{p_0(1 - p_0)/n}$.

Two proportions: $Z = (\hat{p}_1 - \hat{p}_2)/\sqrt{\hat{p}_{\text{pool}}(1 - \hat{p}_{\text{pool}})(1/n_1 + 1/n_2)}$.

7.3 Student's t-Test

One sample: $T = (\bar{X} - \mu_0)/(S/\sqrt{n}) \sim t(n - 1)$.

Two independent samples (equal variances):

$$T = \frac{\bar{X} - \bar{Y}}{S_p\sqrt{1/n_1 + 1/n_2}} \sim t(n_1 + n_2 - 2), \quad S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}.$$

Welch's test (unequal variances):

$$T = \frac{\bar{X} - \bar{Y}}{\sqrt{S_1^2/n_1 + S_2^2/n_2}} \sim t(\nu), \quad \nu = \frac{(S_1^2/n_1 + S_2^2/n_2)^2}{\frac{(S_1^2/n_1)^2}{n_1 - 1} + \frac{(S_2^2/n_2)^2}{n_2 - 1}}.$$

Paired test: On differences $D_i = X_i - Y_i$: $T = \bar{D}/(S_D/\sqrt{n}) \sim t(n - 1)$.

7.4 χ^2 Tests

Variance test: $\chi^2 = (n - 1)S^2/\sigma_0^2 \sim \chi^2(n - 1)$.

Goodness-of-fit: $\chi^2 = \sum(O_i - E_i)^2/E_i \sim \chi^2(k - 1 - r)$.

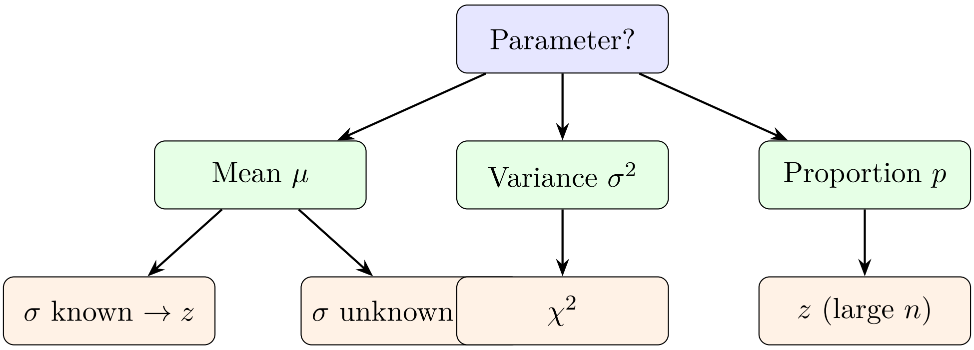
Independence test: For $r \times c$ contingency table: $\chi^2 = \sum\sum(O_{ij} - E_{ij})^2/E_{ij} \sim \chi^2((r - 1)(c - 1))$.

CHAPTER 7. CLASSICAL TESTS: z, t, χ^2, F

7.5 F-Test

For $H_0 : \sigma_1^2 = \sigma_2^2$: $F = S_1^2/S_2^2 \sim F(n_1 - 1, n_2 - 1)$.

7.6 Test Selection Guide



7.7 Python Implementation

Python Implementation

```python
import numpy as np
from scipy import stats

# One-sample t-test
data = np.array([50.2, 49.8, 51.1, 48.9, 50.5, 49.3, 50.8, 49.1,
                 50.0, 49.7, 50.3, 49.5, 50.6, 49.2, 50.4])
t_stat, p_val = stats.ttest_1samp(data, 50)
print(f"One-sample t: t={t_stat:.4f}, p={p_val:.4f}")

# Two-sample t-test (Welch)
A = np.random.normal(50, 1.5, 20)
B = np.random.normal(49.5, 2.1, 25)
t_w, p_w = stats.ttest_ind(A, B, equal_var=False)
print(f"Welch: t={t_w:.4f}, p={p_w:.4f}")

# F-test for equal variances
f_stat = np.var(A, ddof=1) / np.var(B, ddof=1)
p_f = 2*min(stats.f.cdf(f_stat, 19, 24), 1-stats.f.cdf(f_stat, 19, 24))
print(f"F-test: F={f_stat:.4f}, p={p_f:.4f}")

# Chi-squared goodness-of-fit
obs = np.array([8, 12, 10, 11, 9, 10])
chi2, p_chi = stats.chisquare(obs)
print(f"Chi2 GoF: chi2={chi2:.4f}, p={p_chi:.4f}")

# Chi-squared independence
table = np.array([[30, 20, 10], [25, 30, 15], [15, 20, 35]])
chi2_ind, p_ind, dof, _ = stats.chi2_contingency(table)
```

```
print(f"Chi2 independence: chi2={chi2_ind:.4f}, p={p_ind:.4f}")
```

7.8 Exercises

Exercise 7.1 ($\star$)**.** Reaction times after coffee for 12 subjects. Normal mean is 220ms. Does coffee reduce reaction time?

Exercise 7.2 ($\star$)**.** A die rolled 120 times. Test whether it is fair at 5%.

Exercise 7.3 ($\star\star$)**.** Compare two treatments: first test equality of variances (F-test), then choose the appropriate t-test.

Exercise 7.4 ($\star\star\star$ – Project)**.** Study the robustness of the t-test to non-normality via simulation.

Key Formulas

$$z = \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}} \qquad t = \frac{\bar{X} - \mu_0}{S/\sqrt{n}}$$

$$\chi^2 = \frac{(n-1)S^2}{\sigma_0^2} \qquad F = S_1^2/S_2^2$$

$$\chi^2_{\text{GoF}} = \sum \frac{(O_i - E_i)^2}{E_i} \qquad S_p^2 = \frac{(n_1-1)S_1^2 + (n_2-1)S_2^2}{n_1+n_2-2}$$

CHAPTER 7. CLASSICAL TESTS: z, t, χ^2, F

Chapter 8

Simple Linear Regression

"Regression is the most useful and most misused tool in statistics."

8.1 Motivating Example

Data on living area (m^2) and sale price (kEUR) for 15 apartments. Goal: model Price = f(Area).

8.2 The Model

Definition 8.1 (Simple Linear Regression Model)**.** $Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$, $i = 1, \ldots, n$, with $\mathbb{E}[\varepsilon_i] = 0$, $\mathrm{Var}(\varepsilon_i) = \sigma^2$ (homoscedasticity), $\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0$, and optionally $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

8.3 OLS Estimation

Theorem 8.2 (OLS Estimators)**.**

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} = \frac{\sum(x_i - \bar{x})(Y_i - \bar{Y})}{\sum(x_i - \bar{x})^2}, \qquad \hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x}.$$

Proof. Minimize $Q = \sum(Y_i - \beta_0 - \beta_1 x_i)^2$. Setting $\partial Q/\partial\beta_0 = 0$ gives $\beta_0 = \bar{Y} - \beta_1 \bar{x}$. Substituting and solving $\partial Q/\partial\beta_1 = 0$ yields the result. □

Theorem 8.3 (Gauss–Markov)**.** *Under assumptions (1)–(3), OLS estimators are BLUE (Best Linear Unbiased Estimators).*

8.4 Variance Decomposition

Theorem 8.4. $\underbrace{\sum(Y_i - \bar{Y})^2}_{SST} = \underbrace{\sum(\hat{Y}_i - \bar{Y})^2}_{SSR} + \underbrace{\sum(Y_i - \hat{Y}_i)^2}_{SSE}.$

Definition 8.5 (R^2)**.** $R^2 = \mathrm{SSR}/\mathrm{SST} = 1 - \mathrm{SSE}/\mathrm{SST} = r_{xy}^2$ in simple regression.

CHAPTER 8. SIMPLE LINEAR REGRESSION

8.5 Inference under Normality

$T_1 = \hat{\beta}_1/\mathrm{SE}(\hat{\beta}_1) \sim t(n-2)$. Test $H_0 : \beta_1 = 0$.
$F = \mathrm{SSR}/1 \;/\; \mathrm{SSE}/(n-2) \sim F(1, n-2)$. In simple regression, $F = T_1^2$.

8.5.1 Prediction

CI for $\mathbb{E}[Y|x_0]$: $\hat{Y}_0 \pm t_{\alpha/2,n-2}\hat{\sigma}\sqrt{1/n + (x_0 - \bar{x})^2/S_{xx}}$.
PI for Y_{new}: add 1 under the square root.

8.6 Residual Diagnostics

Check: (1) residuals vs fitted (linearity, homoscedasticity), (2) QQ-plot (normality), (3) residuals vs order (autocorrelation).

8.7 Python Implementation

Python Implementation

```python
import numpy as np
from scipy import stats
import statsmodels.api as sm
import matplotlib.pyplot as plt

area = np.array([25,30,35,40,45,50,55,60,65,70,75,80,90,100,120])
price =
↪ np.array([95,110,125,140,148,160,175,185,195,210,220,240,260,290,350])

slope, intercept, r, p, se = stats.linregress(area, price)
print(f"beta_1={slope:.4f} (p={p:.6f}), R^2={r**2:.4f}")

X = sm.add_constant(area)
model = sm.OLS(price, X).fit()
print(model.summary())

fig, axes = plt.subplots(2, 2, figsize=(12, 10))
axes[0,0].scatter(area, price); axes[0,0].plot(area, model.fittedvalues,
↪ 'r-')
axes[0,0].set_title('Regression line')
axes[0,1].scatter(model.fittedvalues, model.resid)
axes[0,1].axhline(0, color='r', ls='--'); axes[0,1].set_title('Residuals
↪ vs Fitted')
stats.probplot(model.resid, plot=axes[1,0]);
↪ axes[1,0].set_title('QQ-plot')
axes[1,1].hist(model.resid, bins=6, edgecolor='black')
axes[1,1].set_title('Residual distribution')
plt.tight_layout(); plt.savefig("ch08_regression.pdf"); plt.show()
```

8.8 Exercises

Exercise 8.1 ($\star$)**.** Compute $\hat{\beta}_0$, $\hat{\beta}_1$, R^2 by hand for $x = (1, 2, 3, 4, 5)$, $y = (2.1, 3.8, 6.2, 7.9, 10.3)$.

Exercise 8.2 ($\star\star$)**.** Prove Gauss–Markov for simple regression: among $\hat{\beta}_1 = \sum c_i Y_i$ linear unbiased, OLS has minimum variance.

Exercise 8.3 ($\star\star\star$ – Project)**.** Collect real data, fit a simple regression, test $\beta_1 = 0$, analyze residuals, construct prediction intervals.

Key Formulas

$$\hat{\beta}_1 = S_{xy}/S_{xx}, \quad \hat{\beta}_0 = \bar{Y} - \hat{\beta}_1\bar{x}$$
$$R^2 = 1 - \text{SSE}/\text{SST}$$
$$T = \hat{\beta}_1/(\hat{\sigma}/\sqrt{S_{xx}}) \sim t(n-2)$$

CHAPTER 8. SIMPLE LINEAR REGRESSION

Chapter 9

Multiple Linear Regression and ANOVA

"In science, reality is rarely one-dimensional."

9.1 Motivating Example

Predict apartment price from area, number of rooms, floor, and distance to city center.

9.2 Matrix Model

Definition 9.1 (General Linear Model)**.** $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$.

9.3 OLS Estimation

Theorem 9.2. $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$, *with* $\mathrm{Cov}(\hat{\boldsymbol{\beta}}) = \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}$.

Definition 9.3 (Hat Matrix)**.** $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top$: $\hat{\mathbf{Y}} = \mathbf{H}\mathbf{Y}$, $\mathbf{e} = (\mathbf{I} - \mathbf{H})\mathbf{Y}$.

9.4 Inference

Individual: $T_j = \hat{\beta}_j / \mathrm{SE}(\hat{\beta}_j) \sim t(n-p)$.
Global F-test: $F = \frac{R^2/(p-1)}{(1-R^2)/(n-p)} \sim F(p-1, n-p)$.

Definition 9.4 (Adjusted R^2)**.** $R^2_{\mathrm{adj}} = 1 - \frac{n-1}{n-p}(1 - R^2)$.

Nested model comparison: $F = \frac{(\mathrm{SSE}_{\mathrm{red}} - \mathrm{SSE}_{\mathrm{full}})/(p-q)}{\mathrm{SSE}_{\mathrm{full}}/(n-p)} \sim F(p-q, n-p)$.

9.5 One-Way ANOVA

Definition 9.5. $Y_{ij} = \mu + \alpha_j + \varepsilon_{ij}$, $H_0 : \alpha_1 = \cdots = \alpha_k = 0$.

Theorem 9.6 (ANOVA Table).

Source	***SS***	***df***	F
Factor	$SSB = \sum n_j(\bar{Y}_j - \bar{Y})^2$	$k-1$	$F = MSB/MSW$
Residual	$SSW = \sum\sum(Y_{ij} - \bar{Y}_j)^2$	$n-k$	
Total	SST	$n-1$	

Under H_0: $F \sim F(k-1, n-k)$.

Common Pitfall

After rejecting H_0, use multiple comparison corrections (Bonferroni, Tukey HSD) to identify which groups differ.

9.6 Python Implementation

Python Implementation

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
from statsmodels.formula.api import ols
from scipy import stats

# Multiple regression
np.random.seed(42)
n = 50
df = pd.DataFrame({
    'area': np.random.uniform(25, 120, n),
    'rooms': np.random.randint(1, 6, n),
    'floor': np.random.randint(0, 10, n)
})
df['price'] = 30 + 2.5*df['area'] + 15*df['rooms'] + 3*df['floor'] +
↪  np.random.normal(0, 15, n)
X = sm.add_constant(df[['area', 'rooms', 'floor']])
model = sm.OLS(df['price'], X).fit()
print(model.summary())

# One-way ANOVA
A = [20.1, 21.5, 19.8, 22.0, 20.5, 21.2]
B = [23.4, 24.1, 22.8, 23.9, 24.5, 23.2]
C = [19.5, 20.2, 18.9, 20.8, 19.1, 20.5]
f_stat, p_val = stats.f_oneway(A, B, C)
print(f"ANOVA: F={f_stat:.4f}, p={p_val:.6f}")

# Tukey HSD
from statsmodels.stats.multicomp import pairwise_tukeyhsd
df_a = pd.DataFrame({'y': A+B+C, 'g': ['A']*6+['B']*6+['C']*6})
print(pairwise_tukeyhsd(df_a['y'], df_a['g'], alpha=0.05))
```

9.7 Exercises

Exercise 9.1 ($\star$)**.** Fit a multiple regression for car price on year, mileage, and horsepower. Interpret coefficients.

Exercise 9.2 ($\star\star$)**.** Show that **H** is idempotent and symmetric.

Exercise 9.3 ($\star\star\star$ – Project)**.** Compare models using AIC, BIC, R^2_{adj}, and cross-validation.

Key Formulas

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{Y}$$

$$F_{\text{global}} = \frac{R^2/(p-1)}{(1-R^2)/(n-p)}$$

$$R^2_{\text{adj}} = 1 - \frac{n-1}{n-p}(1-R^2)$$

$$F_{\text{ANOVA}} = \text{MSB}/\text{MSW} \sim F(k-1, n-k)$$

CHAPTER 9. MULTIPLE LINEAR REGRESSION AND ANOVA

Chapter 10

Introduction to Bayesian Statistics

"Probability is the language of uncertainty." — Laplace

10.1 Motivating Example

A screening test has 95% sensitivity and 90% specificity. A patient tests positive. With 1% prevalence, Bayes' theorem gives $\approx 8.7\%$ probability of being truly ill—far from the naive 95%.

10.2 The Bayesian Paradigm

Definition 10.1 (Bayesian Approach)**.** θ is a **random variable** with:

- **Prior** $\pi(\theta)$: beliefs before data.
- **Likelihood** $L(\theta; \mathbf{x})$.
- **Posterior**: updated beliefs.

Theorem 10.2 (Bayes' Theorem)**.**

$$\pi(\theta \mid \mathbf{x}) \propto L(\theta; \mathbf{x}) \cdot \pi(\theta).$$

$\boxed{\textit{Posterior} \propto \textit{Likelihood} \times \textit{Prior}}$

10.3 Conjugate Priors

Likelihood	**Conjugate Prior**	**Posterior**
Bernoulli(p)	Beta(α, β)	Beta$(\alpha + s, \beta + n - s)$
Pois(λ)	$\Gamma(\alpha, \beta)$	$\Gamma(\alpha + \sum x_i, \beta + n)$
$\mathcal{N}(\mu, \sigma^2)$	$\mathcal{N}(\mu_0, \tau^2)$	$\mathcal{N}(\mu_n, \tau_n^2)$

Definition 10.3 (Noninformative Priors)**. Uniform**: $\pi(\theta) \propto 1$. **Jeffreys**: $\pi(\theta) \propto \sqrt{I(\theta)}$.

CHAPTER 10. INTRODUCTION TO BAYESIAN STATISTICS

10.4 Key Examples

Example 10.4 (Beta-Binomial)**.** Prior $p \sim \text{Beta}(\alpha, \beta)$, data s successes in n trials. Posterior mean:

$$\mathbb{E}[p \mid \mathbf{x}] = \frac{\alpha + s}{\alpha + \beta + n} = w \cdot \frac{s}{n} + (1 - w) \cdot \frac{\alpha}{\alpha + \beta}.$$

A weighted average of the MLE and the prior mean.

Example 10.5 (Normal-Normal)**.** Prior $\mu \sim \mathcal{N}(\mu_0, \tau^2)$, data $\mathcal{N}(\mu, \sigma^2)$ known. Posterior: $\mu_n = (\mu_0/\tau^2 + n\bar{x}/\sigma^2)/(1/\tau^2 + n/\sigma^2)$. Posterior precision = prior precision + data precision.

10.5 Bayesian Estimators

Definition 10.6.

- **Posterior mean**: minimizes quadratic loss.
- **MAP** (mode): $\arg\max_\theta \pi(\theta \mid \mathbf{x})$.
- **Credible interval**: $\mathbb{P}(\theta \in [a, b] \mid \mathbf{x}) = 1 - \alpha$.

Common Pitfall

A credible interval has a direct interpretation: "the probability that $\theta \in [a, b]$ is $1 - \alpha$." Unlike a frequentist CI, this depends on the prior choice.

10.6 Python Implementation

Python Implementation

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

alpha_pr, beta_pr = 2, 2; n, s = 100, 68
alpha_po, beta_po = alpha_pr + s, beta_pr + n - s

p = np.linspace(0, 1, 500)
fig, ax = plt.subplots(figsize=(10, 6))
ax.plot(p, stats.beta.pdf(p, alpha_pr, beta_pr), 'b--', lw=2,
↪  label='Prior')
lik = stats.binom.pmf(s, n, p)
ax.plot(p, lik/lik.max()*stats.beta.pdf(p, alpha_po, beta_po).max(),
↪  'g:', lw=2, label='Likelihood')
ax.plot(p, stats.beta.pdf(p, alpha_po, beta_po), 'r-', lw=2,
↪  label='Posterior')
ax.legend(); ax.set_title('Beta-Binomial Model')
plt.savefig("ch10_bayesian.pdf"); plt.show()

print(f"Posterior mean: {alpha_po/(alpha_po+beta_po):.4f}")
print(f"MAP: {(alpha_po-1)/(alpha_po+beta_po-2):.4f}")
```

```
print(f"95% credible interval: {stats.beta.interval(0.95, alpha_po,
↪  beta_po)}")
```

10.7 Exercises

Exercise 10.1 ($\star$)**.** Basketball: 7/10 shots made, Beta$(1, 1)$ prior. Find posterior, mean, and 90% credible interval.

Exercise 10.2 ($\star\star$)**.** Derive the Normal-Normal posterior by completing the square.

Exercise 10.3 ($\star\star$)**.** Compute Jeffreys prior for Bernoulli, Poisson, and Normal (known variance).

Exercise 10.4 ($\star\star\star$ – Project)**.** Compare frequentist (MLE + CI) and Bayesian (posterior mean + credible interval) for varying priors and sample sizes. Show convergence as $n \to \infty$.

Key Formulas

$$\pi(\theta \mid \mathbf{x}) \propto L(\theta; \mathbf{x}) \cdot \pi(\theta)$$
$$\text{Beta-Binomial} : \text{Beta}(\alpha, \beta) \to \text{Beta}(\alpha + s, \beta + n - s)$$
$$\text{Normal-Normal} : \mu_n = \frac{\mu_0/\tau^2 + n\bar{x}/\sigma^2}{1/\tau^2 + n/\sigma^2}$$

CHAPTER 10. INTRODUCTION TO BAYESIAN STATISTICS

Chapter 11

Non-Parametric Methods

"Do not make unnecessary assumptions." — William of Ockham

11.1 Motivating Example

A psychologist compares well-being scores (ordinal, 1–10 scale) between meditation and control groups. Data are clearly non-normal. Non-parametric methods provide valid alternatives.

11.2 Why Non-Parametric?

- Ordinal or rank data.
- Small samples where normality cannot be verified.
- Heavily skewed distributions or outliers.
- Unknown distribution: no parametric model desired.

Common Pitfall

Non-parametric tests are generally **less powerful** than parametric counterparts when parametric assumptions hold, but they are more **robust**.

11.3 Sign Test

For $H_0 : \text{median} = m_0$: count $S^+ = \#\{i : X_i > m_0\}$. Under H_0: $S^+ \sim \text{Bin}(n, 1/2)$.

11.4 Wilcoxon Signed-Rank Test

Compute $D_i = X_i - m_0$, rank $|D_i|$, and $W^+ = \sum_{\{D_i>0\}} R_i$.

CHAPTER 11. NON-PARAMETRIC METHODS

11.5 Mann–Whitney / Wilcoxon Rank-Sum Test

Combine and rank both samples. $U_1 = \sum R_i - n_1(n_1+1)/2$. For large samples:

$$Z = \frac{U - n_1 n_2/2}{\sqrt{n_1 n_2(n_1+n_2+1)/12}} \xrightarrow{\mathcal{L}} \mathcal{N}(0,1).$$

11.6 Kruskal–Wallis Test

Extension to k groups: $H = \frac{12}{n(n+1)} \sum R_j^2/n_j - 3(n+1) \sim \chi^2(k-1)$.

11.7 Kolmogorov–Smirnov Test

One-sample: $D_n = \sup_x |\hat{F}_n(x) - F_0(x)|$. Two-sample: $D_{n_1,n_2} = \sup_x |\hat{F}_{n_1}(x) - \hat{G}_{n_2}(x)|$.

11.8 Kernel Density Estimation

Definition 11.1. $\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^n K\left(\frac{x-X_i}{h}\right)$, where K is a kernel and h the bandwidth.

Statistical Intuition

Small h: noisy (low bias, high variance). Large h: smooth (high bias, low variance). The bias-variance tradeoff in density estimation.

11.9 Spearman Correlation

$r_s = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$ where $d_i = \text{rank}(x_i) - \text{rank}(y_i)$. Measures monotonic (not necessarily linear) association.

11.10 Python Implementation

Python Implementation

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

# Wilcoxon signed-rank
before = np.array([5.2, 4.8, 6.1, 5.5, 4.9, 5.8, 6.3, 5.1, 4.7, 5.4])
after = np.array([4.8, 4.2, 5.5, 5.0, 4.3, 5.1, 5.7, 4.5, 4.1, 4.9])
stat, p = stats.wilcoxon(before, after, alternative='greater')
print(f"Wilcoxon signed-rank: W={stat:.1f}, p={p:.4f}")

# Mann-Whitney
A = np.array([23, 25, 28, 30, 27, 24, 26, 29, 31, 22])
B = np.array([18, 20, 22, 19, 21, 17, 23, 20, 19, 18])
```

```
u, p_mw = stats.mannwhitneyu(A, B, alternative='two-sided')
print(f"Mann-Whitney: U={u:.1f}, p={p_mw:.4f}")

# Kruskal-Wallis
h, p_kw = stats.kruskal([6.4,7.1,6.8,7.3,6.9], [8.2,7.9,8.5,8.1,8.4],
↪  [5.5,6.2,5.8,6.0,5.7])
print(f"Kruskal-Wallis: H={h:.4f}, p={p_kw:.4f}")

# KS test
data = np.random.exponential(2, 50)
d, p_ks = stats.kstest(data, 'norm', args=(np.mean(data), np.std(data)))
print(f"KS (normality): D={d:.4f}, p={p_ks:.4f}")

# KDE
data_kde = np.concatenate([np.random.normal(-2, 0.8, 100),
↪  np.random.normal(2, 1.2, 150)])
x_grid = np.linspace(-6, 7, 300)
fig, ax = plt.subplots(figsize=(10, 5))
ax.hist(data_kde, bins=30, density=True, alpha=0.3)
for bw in [0.2, 0.5, 1.0]:
    kde = stats.gaussian_kde(data_kde, bw_method=bw)
    ax.plot(x_grid, kde(x_grid), lw=2, label=f'h={bw}')
ax.legend(); ax.set_title("Kernel Density Estimation")
plt.savefig("ch11_kde.pdf"); plt.show()

# Spearman
x = np.arange(1, 11); y = x**2
r_s, p_s = stats.spearmanr(x, y)
print(f"Spearman: r_s={r_s:.4f} (Pearson: {stats.pearsonr(x,y)[0]:.4f})")
```

11.11 Exercises

Exercise 11.1 (⋆)**.** Two algorithms tested on 8 datasets. Is algorithm A significantly faster? (Mann–Whitney)

Exercise 11.2 (⋆⋆)**.** Compare Type I error rates of t-test and Wilcoxon for log-normal data via simulation.

Exercise 11.3 (⋆⋆)**.** Implement a Gaussian KDE. Study bandwidth effect on MISE for a Gaussian mixture.

Exercise 11.4 (⋆⋆⋆ – Project)**.** Collect non-normal real data. Apply all non-parametric methods from this chapter. Compare with parametric results.

CHAPTER 11. NON-PARAMETRIC METHODS

Key Formulas

$$U = \sum R_i - n_1(n_1+1)/2 \qquad H = \frac{12}{n(n+1)} \sum R_j^2/n_j - 3(n+1)$$

$$D_n = \sup_x |\hat{F}_n(x) - F_0(x)| \qquad r_s = 1 - 6 \sum d_i^2/[n(n^2-1)]$$

$$\hat{f}_h(x) = \frac{1}{nh} \sum K\left(\frac{x - X_i}{h}\right)$$

Formula Sheet

.1 Common Probability Distributions

Distribution	Parameters	Mean	Variance
$\mathrm{Bin}(n,p)$	$n \in \mathbb{N}^*,\ p \in [0,1]$	np	$np(1-p)$
$\mathrm{Pois}(\lambda)$	$\lambda > 0$	λ	λ
$\mathrm{Unif}([a,b])$	$a < b$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
$\mathrm{Exp}(\lambda)$	$\lambda > 0$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
$\mathcal{N}(\mu,\sigma^2)$	$\mu \in \mathbb{R},\ \sigma > 0$	μ	σ^2
$\chi^2(k)$	$k \in \mathbb{N}^*$	k	$2k$
$t(k)$	$k \in \mathbb{N}^*$	$0\ (k>1)$	$\frac{k}{k-2}\ (k>2)$
$F(d_1,d_2)$	$d_1, d_2 \in \mathbb{N}^*$	$\frac{d_2}{d_2-2}$	—
$\Gamma(\alpha,\beta)$	$\alpha,\beta > 0$	$\frac{\alpha}{\beta}$	$\frac{\alpha}{\beta^2}$

.2 Estimation Formulas

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{1}$$

$$\hat{\theta}_{\mathrm{MLE}} = \arg\max_{\theta} \sum_{i=1}^{n} \ln f(x_i;\theta) \qquad \hat{\theta}_{\mathrm{MoM}} : \mathbb{E}_\theta[X^k] = \frac{1}{n}\sum_{i=1}^{n} X_i^k \tag{2}$$

.3 Confidence Intervals

$$\text{Mean } (\sigma \text{ known}) : \bar{X} \pm z_{\alpha/2}\frac{\sigma}{\sqrt{n}} \tag{3}$$

$$\text{Mean } (\sigma \text{ unknown}) : \bar{X} \pm t_{\alpha/2,n-1}\frac{S}{\sqrt{n}} \tag{4}$$

$$\text{Proportion} : \hat{p} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \tag{5}$$

$$\text{Variance} : \left[\frac{(n-1)S^2}{\chi^2_{\alpha/2,n-1}},\ \frac{(n-1)S^2}{\chi^2_{1-\alpha/2,n-1}}\right] \tag{6}$$

.4 Test Statistics

$$Z = \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}} \sim \mathcal{N}(0, 1) \tag{7}$$

$$T = \frac{\bar{X} - \mu_0}{S/\sqrt{n}} \sim t(n-1) \tag{8}$$

$$\chi^2 = \frac{(n-1)S^2}{\sigma_0^2} \sim \chi^2(n-1) \tag{9}$$

$$F = \frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2} \sim F(n_1 - 1, n_2 - 1) \tag{10}$$

Statistical Tables

Statistical tables (normal, t, χ^2, F) are available as digital appendices. In practice, use Python:

```
from scipy import stats
stats.norm.ppf(0.975)         # z_{0.025} = 1.96
stats.t.ppf(0.975, df=10)     # t_{0.025,10}
stats.chi2.ppf(0.95, df=5)    # chi2_{0.05,5}
stats.f.ppf(0.95, 3, 20)      # F_{0.05,3,20}
```

APPENDIX . STATISTICAL TABLES

Bibliography

[1] Casella, G. & Berger, R.L. (2002). *Statistical Inference*. 2nd edition, Cengage Learning.

[2] Wasserman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*. Springer.

[3] Saporta, G. (2011). *Probabilités, Analyse des Données et Statistique*. Technip, 3rd edition.

[4] Wackerly, D., Mendenhall, W. & Scheaffer, R. (2008). *Mathematical Statistics with Applications*. 7th edition, Cengage.