

SEN608 DATA ANALYTICS AND STATISTICS

Week 3: Data and Sampling Distributions

Assoc. Prof. Dr. Onur Polat

Hacettepe University • Graduate School of Informatics

M.Sc. (Thesis) in Systems Engineering • Fall 2026, Lecture 3 / 14

Source: Bruce et al. (2026), Chapter 2 — Data and Sampling Distributions

Recap and Today's Agenda

Last Week (Week 2)

- Location: mean, median, trimmed mean
- Variability: std, IQR, MAD
- Distributions: histogram, boxplot, KDE
- Correlation and scatterplots
- FacetGrid and AI-assisted exploration

This Week (Week 3)

- Random sampling and sample bias
- Selection bias and data snooping
- Sampling distribution and standard error
- Central Limit Theorem (CLT)
- The bootstrap
- Confidence intervals
- Normal, t, binomial, Poisson distributions

After This Week You Will Be Able To...

#	Learning Outcome
1	Distinguish random, stratified, and biased sampling
2	Explain the difference between data distribution and sampling distribution
3	State the CLT and explain why it matters
4	Compute bootstrap standard errors and confidence intervals
5	Generate and visualize Normal, t, binomial, and Poisson distributions in scipy
6	Assess normality using QQ-plots



PART I

Random Sampling and Bias

Population, Sample, and Random Selection

- Population: the entire set of possible observations — usually inaccessible.
- Sample: a subset drawn from the population — all analysis rests on this.
- Random sampling: every individual has equal probability of selection → representative sample.
- With replacement vs. without replacement.
- Even in the big-data era, sampling is needed: data quality and computational cost.

Textbook: “The proliferation of data of varying quality and relevance reinforces the need for sampling.” (Bruce et al., p. 55)

The Literary Digest Disaster (1936)

- Literary Digest polled 10 million people and predicted Landon would win.
- George Gallup polled 50,000 and correctly predicted Roosevelt.
- Problem: magazine subscribers and telephone owners did not represent the electorate.
- Gallup's success: random, representative sample selection.
- Lesson: 10 million biased observations are worse than 50,000 well-chosen ones.

Stratified Sampling

- Divide the population into homogeneous subgroups (strata).
- Draw a random sample from each stratum separately.
- Advantage: guarantees each subgroup is adequately represented.
- Example: political polls stratified by demographic groups.
- SE connection: A/B tests stratified by user segment ensure each segment has enough observations.

Types of Bias

Bias Type	Definition	Example
Sample bias	Sample misrepresents population	Literary Digest poll
Self-selection	Participants choose themselves	Yelp / Google reviews
Data snooping	Searching data until something 'interesting' appears	p-hacking, data dredging
Vast search effect	Many models/questions → spurious result	Picking best of 100 features
Regression to the mean	Extreme values revert toward center on next measurement	Star performer's next year

Guarding Against Selection Bias

- Specify hypothesis → design experiment → collect data → test: this order prevents bias.
- The reverse: scan data → find pattern → form hypothesis post hoc → p-hacking.
- Holdout set: validate on data the model never saw.
- Target shuffling: shuffle the response and refit — a permutation test for your model.
- Pre-registration: commit to your analysis plan before data collection.

"If you don't know what you're looking for, look hard enough and you'll find it." — adapted from Yogi Berra (Bruce et al.)



PART II

Sampling Distribution and the Central Limit Theorem

Data Distribution vs. Sampling Distribution

Data Distribution

- Distribution of individual observations
- Can be skewed, multimodal, heavy-tailed
- Does not change with sample size
- Directly observed via EDA

Sampling Distribution

- Distribution of a statistic (e.g. $\bar{x}$)
computed across many samples
- Tends to be regular and bell-shaped
- Narrows and normalizes as n grows
- The CLT is its theoretical foundation

The CLT — The Most Important Theorem in Statistics

Regardless of the source population's distribution, the distribution of means from sufficiently large samples will approximate the Normal distribution.

- “Sufficiently large” is conventionally $n \geq 30$ (rough rule of thumb).
- The more skewed the source, the larger n needs to be for normalization.
- The CLT underpins t-tests, ANOVA, and most classical parametric methods.
- It applies to other statistics (median, proportion, std) but convergence rates vary.
- Thanks to the CLT, we can use parametric tests confidently in most situations.

CLT in Action — Python Simulation

```
# CLT simulation with loans_income (Textbook Figure 2-6, pp. 67-68)
loans_income = pd.read_csv(f'{DATA_URL}/loans_income.csv').squeeze()

results = []
results.append(pd.DataFrame({'income': np.random.choice(loans_income, 1000),
                           'type': 'Raw Data'}))
results.append(pd.DataFrame({
    'income': [np.random.choice(loans_income, 5).mean() for _ in range(1000)],
    'type': 'Mean of 5'}))
results.append(pd.DataFrame({
    'income': [np.random.choice(loans_income, 20).mean() for _ in range(1000)],
    'type': 'Mean of 20'}))

all_data = pd.concat(results)
g = sns.FacetGrid(all_data, col='type', col_wrap=1, height=2, aspect=3)
g.map(plt.hist, 'income', range=[0, 200_000], bins=40, edgecolor='white')
plt.show()
```

Standard Error (SE) — Measuring Estimation Uncertainty

$$SE(\bar{x}) = s / \sqrt{n}$$

- Standard error measures how much a sample statistic varies from sample to sample.
- Standard deviation $\neq$ standard error: SD measures the data, SE measures the estimate.
- SE decreases with n : inversely proportional to $\sqrt{n}$.
- To halve the SE, you need to quadruple the sample size.
- The bootstrap is the modern, general-purpose method to estimate SE.



PART III

The Bootstrap

The Bootstrap Idea

- Draw repeatedly with replacement from your observed sample.
- Compute the statistic of interest (mean, median, std) from each draw.
- The distribution of these statistics → estimate of the sampling distribution.
- No distributional assumptions required — all information comes from the data.
- Popularized by Bradley Efron (Stanford, 1979).

Conceptually: replicate your sample thousands of times to create a hypothetical population, then draw samples from it.

Bootstrap Algorithm

Step	Action
1	Draw n observations with replacement from the original sample → bootstrap sample
2	Compute the statistic (mean, median, std, ...)
3	Repeat steps 1–2 R times ($R = 1,000$ or $10,000$)
4	Compute the standard deviation of the R statistics → SE estimate
5	Optionally construct a percentile confidence interval

Bootstrap in Python — Median SE and CI

```
# Bootstrap: median SE and 95% CI (Textbook pp. 70-72)
R = 5_000
medians = np.array([
    np.median(np.random.choice(loans_income, len(loans_income), replace=True))
    for _ in range(R)
])

se_boot = medians.std()
ci_low, ci_high = np.percentile(medians, [2.5, 97.5])

print(f'Original median: ${np.median(loans_income):,.0f}')
print(f'Bootstrap SE:    ${se_boot:,.0f}')
print(f'95% CI:          [{ci_low:,.0f}, {ci_high:,.0f}']')
```

Bootstrap vs. Permutation: Bootstrap samples with replacement (estimates sampling distribution); permutation shuffles without replacement (tests null hypothesis). They answer different questions.



PART IV

Confidence Intervals

From Point Estimate to Interval Estimate

- Point estimate: a single number ($\bar{x} = \$72,000$). Creates a false sense of precision.
- Confidence interval: a plausible range ($[\$68,000; \$76,000]$). Shows uncertainty.
- 95% CI: 95% of intervals constructed this way will contain the true parameter.
- Width depends on sample size and variability.
- Narrow CI $\rightarrow$ precise estimate; wide CI $\rightarrow$ uncertain estimate.

Subtle point: “95% probability the true value is in this interval” is technically wrong — the correct statement is about the procedure, not any single interval. In practice most people use the first interpretation.

Bootstrap CI vs. t-Distribution CI

Percentile Method (Bootstrap)

CI = $[P_{2.5}, P_{97.5}]$ of bootstrap statistics

- Simple and intuitive
- No distributional assumptions
- May be biased for very small samples
- Our default method in this course

t-Distribution Formula

CI = $\bar{x} \pm t_{n-1}(\alpha/2) \times s/\sqrt{n}$

- Classical formula
- Assumes normality (or relies on CLT for large n)
- Fast to compute
- Covered formally in Week 4



PART V

Statistical Distributions

The Normal (Gaussian) Distribution

- Bell-shaped, symmetric, fully described by μ and σ .
- 68% of data within $\mu \pm 1\sigma$, 95% within $\mu \pm 2\sigma$, 99.7% within $\mu \pm 3\sigma$.
- Raw data is usually NOT Normal — but sampling distributions often are (thanks to CLT).
- z-score: $(x - \mu) / \sigma \rightarrow$ converts to standard Normal for comparison.
- Converting to z-scores does NOT make data Normal — it only standardizes the scale.

QQ-Plot — Assessing Normality

- QQ-plot compares sample quantiles against theoretical Normal quantiles.
- Points on the line → approximately Normal.
- S-shaped departure → heavy tails (leptokurtic).
- J-shaped departure → skewness.
- Financial returns typically show S-shaped departure — fat tails!

```
# QQ-plot: Netflix returns (Textbook Figure 2-12)
nflx = sp500_px['NFLX']
nflx_ret = np.diff(np.log(nflx[nflx > 0].values))

fig, ax = plt.subplots(figsize=(4, 4))
stats.probplot(nflx_ret, plot=ax)
ax.set_title('NFLX Returns QQ-Plot')
plt.show()
```

“Most Data Is Not Normal”

- In long-tailed distributions, extreme values occur far more often than Normal predicts.
- Financial returns, income distributions, internet traffic → typical long-tailed data.
- Modeling these with Normal → severely underestimates the probability of extreme events.
- Fitting distributions to data is as much art as science — domain knowledge required.
- QQ-plot is the fastest visual tool for detecting heavy tails.

Student's t-Distribution — For Small Samples

- W. S. Gosset (Guinness Brewery, 1908) published under the name “Student.”
- A heavier-tailed version of the Normal, controlled by degrees of freedom (df).
- As df increases $\rightarrow$ t approaches Normal (virtually identical at $df > 30$).
- Confidence interval: $\bar{x} \pm t_{n-1}(\alpha/2) \times s/\sqrt{n}$
- scipy: `stats.t.rvs(df=10, size=1000)`

Binomial — Yes/No Outcomes

- Distribution of the number of successes in n trials, each with probability p .
- Bernoulli trial: a single trial with two outcomes (click/no-click, pass/fail).
- Mean = np , Variance = $np(1-p)$.
- Approximates Normal for large n .
- SE example: “If click-to-sale rate is 0.02, what is $P(0 \text{ conversions in } 200 \text{ visits})$?”

```
# P(X=2) and P(X<=2) for n=5, p=0.10
print(f'P(X=2) = {stats.binom.pmf(2, n=5, p=0.1):.4f}')
```

```
print(f'P(X<=2) = {stats.binom.cdf(2, n=5, p=0.1):.4f}')
```

Poisson and Exponential — Events and Waiting Times

- Events occurring at a constant rate per unit time (or space) → Poisson.
- Time between events → Exponential distribution.
- Parameter: λ (mean events per unit time).
- Examples: calls/minute, defects/m², typos/page.
- Changing event rate over time → Weibull distribution (device lifetime modeling).

```
# Poisson simulation: call center with lambda=2 calls/min
events = stats.poisson.rvs(mu=2, size=100, random_state=42)
print(f'100-minute simulation: mean={events.mean():.2f} calls/min')

# Exponential: time between calls
waiting = stats.expon.rvs(scale=1/2, size=100, random_state=42)
print(f'Mean waiting time: {waiting.mean():.2f} minutes')
```

Key Distributions — Summary Table

Distribution	Models What?	Params	scipy
Normal	Symmetric, bell-shaped continuous	μ, σ	stats.norm
t	Small n, heavier tails	df	stats.t
Binomial	Successes in n trials	n, p	stats.binom
Poisson	Events per unit time	λ	stats.poisson
Exponential	Time between events	λ (or $1/\lambda$)	stats.expon
Weibull	Lifetime with changing hazard	k, λ	stats.weibull_min
Chi-square	Departure from expectation	df	stats.chi2
F	Variance ratio (ANOVA)	df ₁ , df ₂	stats.f



PART VI

AI-Assisted Exploration and Summary

Textbook 3rd Edition AI Exercises (Ch. 2)

From the textbook's AI Exercises section (p. 94):

1. Send LendingClub income data to AI; ask it to generate bootstrap samples.
2. Ask AI to calculate confidence intervals from the bootstrap.
3. Ask AI to derive a CI based on t-statistic calculations.
4. Ask AI to produce stratified samples by “home ownership” status.

Caution: Always verify AI outputs — AI can hallucinate statistics.

Week 3 — Key Takeaways

Conceptual

- ✓ Random sampling prevents bias
- ✓ Large n does not fix biased sampling
- ✓ Data distribution $\neq$ sampling distribution
- ✓ CLT: means normalize with large n
- ✓ Bootstrap needs no distributional assumptions
- ✓ CI shows the estimate's uncertainty
- ✓ Most raw data is not Normal

Practical

- ✓ CLT simulation with income data
- ✓ Bootstrap SE and CI computation
- ✓ QQ-plot normality assessment
- ✓ scipy: Normal, t, binomial, Poisson generation and visualization
- ✓ Stratified sampling
- ✓ Heavy-tail detection via QQ-plot

Assignment 3 — Sampling Distributions and Bootstrap

- T1. Plot `loans_income` histogram + QQ-plot. Assess normality.
- T2. CLT simulation: $n=5,10,20,50 \rightarrow 1,000$ means each, `FacetGrid` + SE table.
- T3. Bootstrap median ($R=5,000$): compute SE, build 95% percentile CI, plot.
- T4. Compare bootstrap CI vs. t-distribution CI for the mean. Which is wider? Why?
- T5. Bootstrap 95% CI for state `Murder.Rate`.
- T6. Bootstrap CIs for smoker vs non-smoker `tip_ratio` means in tips.
Do the CIs overlap? What does that mean?

Due: Before Week 4 class • Moodle $\rightarrow$ SEN608 $\rightarrow$ Week 3 $\rightarrow$ Assignment

Week 4 — Statistical Experiments and Significance Testing

Preparation

- Read Bruce et al., Chapter 3: Hypothesis testing and permutation sections
- Submit Assignment 3
- Next week: H_0/H_1 , p-values, permutation tests, t-tests, Cohen's d, CIs

SE connection: Bootstrap CIs from this week set the stage for hypothesis testing next week — if two CIs don't overlap, that's informal evidence of a real difference.

Contact: onurpolat@hacettepe.edu.tr