Statistics Fundamentals for Data Science

What is Statistics?

Statistics is the science of collecting, analyzing, interpreting, and presenting data. It provides methods to:

- Summarize data (descriptive statistics)
- Draw conclusions from data (inferential statistics)
- Understand uncertainty and variability
- Make decisions based on evidence rather than guesswork

Two Branches of Statistics

Descriptive Statistics: Summarizing and describing data using numbers and visualizations.

- Computing averages, spreads, and shapes of distributions
- Creating charts, graphs, and summaries
- Making patterns visible

Inferential Statistics: Drawing conclusions about larger populations based on sample data.

- Estimating population parameters from samples
- Testing hypotheses (is this difference real or random?)
- Predicting future outcomes
- Quantifying uncertainty in predictions

Statistics vs Probability

Probability: Asks "Given the rules, what are the odds?" (forward reasoning)

- Example: If a coin is fair, what's the probability of 10 heads?

Statistics: Asks "Given the data, what are the rules?" (backward reasoning)

- Example: I flipped a coin 100 times and got 60 heads. Is it fair?

Both are essential: probability provides the theoretical foundation; statistics applies it to real data.

Why Statistics is Critical in Data Science

1. Understanding Data Before Modeling

Before building ML models, you must understand your data:

- What is typical? (central tendency)
- How much variation exists? (variability)
- Are there outliers or unusual patterns? (shape measures)
- Are distributions symmetric or skewed?

Without understanding data, you risk:

- Building models on corrupted or biased data
- Making wrong conclusions from noisy measurements
- Overfitting to noise instead of signal

Example: If a dataset has extreme outliers, using mean-based algorithms (like linear regression) gives misleading results. Understanding variance and skewness helps you choose robust methods.

2. Feature Engineering and Selection

Statistics helps decide which features to use:

- **Variance**: Features with zero variance (constant values) are useless
- **Correlation**: Highly correlated features provide redundant information
- **Skewness**: Skewed distributions might need transformation
- **Outliers**: Detected using statistical methods, then handled appropriately

3. Model Evaluation

Statistics provides tools to evaluate models fairly:

- **Confidence intervals**: Not just point estimates, but ranges of plausible values
- **P-values and significance tests**: Determining if results are real or random noise
- **Cross-validation with statistics**: Understanding variability in performance across splits

4. Detecting Bias and Fairness Issues

Statistical analysis reveals if models discriminate against groups:

- Comparing performance metrics across demographic groups
- Detecting systematic errors in predictions
- Quantifying disparities

5. Making Decisions Under Uncertainty

Real-world data is always noisy and incomplete. Statistics quantifies uncertainty:

- "Our model achieves 85% accuracy, with 95% confidence interval [82%, 88%]"
- This range captures real-world variability

6. Hypothesis Testing in A/B Testing

Companies use statistics to validate decisions:

- "Does changing the website layout increase clicks?" (hypothesis test)
- "What's the true effect size?" (confidence interval)
- "How many samples do we need?" (power analysis)

7. Handling Missing Data and Outliers

Statistical methods inform how to treat problematic data:

- Imputation strategies (replace missing values with statistically sound estimates)
- Outlier detection (statistical definitions of "unusual")
- Robustness (which statistics are affected by outliers?)

8. Time Series and Forecasting

Predicting the future requires understanding:

- Trends (systematic changes over time)
- Seasonality (repeating patterns)
- Variance (how much things fluctuate)
- Autocorrelation (past values predict future)

All grounded in statistics.

Measures of Central Tendency: Mean, Median, Mode

Central tendency describes the "center" or typical value of a dataset. Three main measures exist, each with different properties.

Mean (Average)

The **mean** is the sum of all values divided by the count:

```
mean = (x₁ + x₂ + ... + xₙ) / n
```

Or using mathematical notation:

```
x̄ = Σxᵢ / n
```

Example:

```
Data: [2, 4, 6, 8, 10]
mean = (2 + 4 + 6 + 8 + 10) / 5 = 30 / 5 = 6
```

Properties:

- Uses all data points (most information)
- Mathematically convenient (appears in formulas for variance, regression, etc.)
- Sensitive to outliers (one extreme value pulls the mean away from typical values)

When to use:

- Normal, symmetric distributions without extreme outliers
- When you want to preserve all information
- For mathematical operations in ML

Example sensitivity to outliers:

```
Data 1: [1, 2, 3, 4, 5]        mean = 3
Data 2: [1, 2, 3, 4, 1000]     mean = 202
```

One extreme value (1000) completely distorts the mean.

Median

The **median** is the middle value when data is sorted.

- If odd number of values: the middle one
- If even number of values: the average of the two middle ones

Example (odd count):

```
Data: [2, 4, 6, 8, 10]
Sorted: [2, 4, 6, 8, 10]
median = 6 (middle value)
```

Example (even count):

```
Data: [2, 4, 6, 8]
Sorted: [2, 4, 6, 8]
median = (4 + 6) / 2 = 5 (average of two middle values)
```

Properties:

- Robust to outliers (not affected by extreme values)
- Doesn't use all data (location only, not values)
- Less mathematically convenient (harder to work with in formulas)

When to use:

- Data with outliers or extreme values (income, house prices, website traffic)
- Skewed distributions
- When you want the "typical" value that's resistant to noise

Example robustness to outliers:

```
Data 1: [1, 2, 3, 4, 5]        median = 3
Data 2: [1, 2, 3, 4, 1000]     median = 3 (unchanged!)
```

Adding an extreme value doesn't change the median.

Mode

The **mode** is the most frequently occurring value.

Example:

```
Data: [1, 2, 2, 3, 3, 3, 4]
mode = 3 (appears 3 times, more than any other)
```

Properties:

- Works with categorical data (colors, categories)

- Can be multiple modes (multimodal)
- Ignores actual values (only cares about frequency)

When to use:

- Categorical data (eye color, product category)
- Discrete data with clear "popular" values
- Understanding frequency distributions

Example with categories:

```
Data: ["blue", "red", "red", "green", "red", "blue"]
mode = "red" (appears 3 times)
```

Comparison and Selection

Measure	Use Case	Robust to Outliers	Mathematical	Information Loss
Mean	Symmetric, no outliers	No	Yes	None (uses all)
Median	Skewed, outliers	Yes	Limited	Loses values
Mode	Categorical data	Yes	Limited	Very high

Real-world example: Company salaries

```
Salaries: [$40k, $45k, $50k, $55k, $10M]
Mean: $438k (misleading!)
Median: $50k (realistic typical salary)
Mode: None (or analyze by category)
```

The CEO's $10M salary distorts the mean. The median better represents a typical employee.

Measures of Variability: Range, Variance, Standard Deviation

Variability (or dispersion) measures how spread out data is. Do values cluster tightly or scatter widely?

Range

The **range** is the difference between maximum and minimum values:

```
range = max(data) - min(data)
```

Example:

```
Data: [2, 5, 8, 12, 15]
range = 15 - 2 = 13
```

Properties:

- Simple and intuitive
- Extremely sensitive to outliers (only uses two extreme values)
- Ignores all middle values
- Limited usefulness in statistics

When to use:

- Quick assessment of spread
- Understanding bounds of data
- Not ideal for serious statistical analysis

Example of outlier sensitivity:

```
Data 1: [5, 6, 7, 8, 9]          range = 4
Data 2: [5, 6, 7, 8, 1000]       range = 995 (one outlier destroys it)
```

Variance

Variance measures how far values are from the mean on average. It captures the "typical squared deviation."

Mathematical definition:

```
Variance = Σ(xᵢ - mean)² / n
```

(Population variance: divide by n)

```
Variance = Σ(xᵢ - mean)² / (n-1)
```

(Sample variance: divide by n-1, which is unbiased)

Step-by-step example:

```
Data: [2, 4, 6, 8, 10]
mean = 6

Deviations from mean: [2-6, 4-6, 6-6, 8-6, 10-6] = [-4, -2, 0, 2, 4]
Squared deviations: [16, 4, 0, 4, 16]
Variance = (16 + 4 + 0 + 4 + 16) / 5 = 40 / 5 = 8
```

Properties:

- Captures spread as squared units (hard to interpret)
- Uses all data points
- Sensitive to outliers (squared deviations amplify extreme values)
- Mathematically convenient (appears in many formulas)

Why square the deviations?

- Squaring makes all deviations positive (prevents cancellation)
- Amplifies large deviations (outliers matter more)
- Makes math convenient (appears in optimization)

Standard Deviation

Standard deviation is the square root of variance:

```
Std Dev = √Variance
```

This brings variance back to original units (easier to interpret).

Example:

```
Variance = 8
Std Dev = √8 ≈ 2.83

This means: on average, values deviate from the mean by about 2.83 units
```

Properties:

- Intuitive (same units as original data)
- Related to normal distribution (68-95-99.7 rule)
- Standard measure in statistics and ML

The 68-95-99.7 Rule (for normal distributions):

- **68%** of data falls within ±1 standard deviation from mean
- **95%** falls within ±2 standard deviations
- **99.7%** falls within ±3 standard deviations

Example:

```
Test scores have mean = 100, std dev = 15
- 68% of scores fall in [85, 115] (within 1σ)
- 95% fall in [70, 130] (within 2σ)
- 99.7% fall in [55, 145] (within 3σ)
```

Coefficient of Variation

When comparing variability across datasets with different scales, use the **coefficient of variation**:

```
CV = (Std Dev / mean) × 100%
```

This is a unitless measure, useful for comparison.

Example:

```
Dataset A: mean = 100, std dev = 10   → CV = 10%
Dataset B: mean = 1000, std dev = 50  → CV = 5%

Dataset A has more relative variation (10% vs 5%)
```

Variability in ML

High variance:

- Model is sensitive to training data changes
- Overfitting (fitting noise)
- Solution: regularization, more data

Low variance:

- Model is stable across datasets
- Underfitting (missing signal)

- Solution: more complex model

Understanding variability is essential for the **bias-variance tradeoff** in ML.

Measures of Shape: Skewness and Kurtosis

These measures describe the **shape** of the distribution: is it symmetric? Are there tails?

Skewness: Asymmetry of Distribution

Skewness measures whether a distribution is symmetric or tilted to one side.

Mathematical definition:

```
Skewness = Σ(xᵢ - mean)³ / (n × std_dev³)
```

(Normalize by std dev to make it unitless)

Interpretation:

- **Skewness ≈ 0**: Symmetric distribution (balanced on both sides)
- **Skewness > 0**: Right-skewed (positively skewed) — long tail on the right
- **Skewness < 0**: Left-skewed (negatively skewed) — long tail on the left

Right-Skewed Distribution (Skewness > 0)

The tail stretches to the right. Most data clusters on the left.

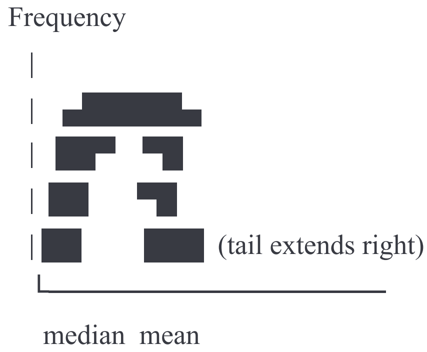
Example: Income distribution

```
Many people earn $30k-$60k (left cluster)
Few earn $500k-$10M (right tail)
```

Relationship to mean and median:

- **mean > median** (mean pulled toward the long tail)
- The extreme high values pull the mean rightward

Visual:



Left-Skewed Distribution (Skewness < 0)

The tail stretches to the left. Most data clusters on the right.

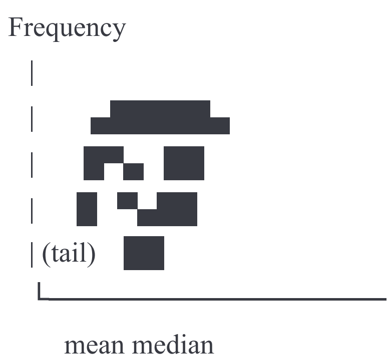
Example: Test scores (when exam is easy)

```
Many people score 80-100 (right cluster)
Few score below 30 (left tail)
```

Relationship to mean and median:

- **mean < median** (mean pulled toward the long tail)

Visual:



Practical Importance of Skewness

Why it matters:

- Mean ≠ median suggests skewness (potential misleading summaries)
- Skewed data might violate assumptions of statistical tests
- Some algorithms prefer symmetric distributions

Handling skewed data:

- **Log transformation**: Converts right-skewed to more symmetric
  - Example: income (log scale) → approximately normal
- **Box-Cox transformation**: Generalization of log transformation
- **Use median/percentiles**: More robust than mean for skewed data

Example: Right-skewed website traffic

```
Original: [10, 15, 20, 100, 200, 500]  (highly skewed)
Log scale: [1, 1.2, 1.3, 2, 2.3, 2.7]  (more symmetric)
```

Kurtosis: Heaviness of Tails

Kurtosis measures whether a distribution has heavy tails (outliers) or light tails.

Mathematical definition:

$$\text{Kurtosis} = \Sigma(x_i - \text{mean})^4 / (n \times \text{std\_dev}^4) - 3$$

(Subtract 3 to center at 0 for normal distribution)

Interpretation:

- **Kurtosis ≈ 0**: Normal distribution (mesokurtic)
- **Kurtosis > 0**: Heavy tails (leptokurtic) — more outliers than normal
- **Kurtosis < 0**: Light tails (platykurtic) — fewer outliers than normal

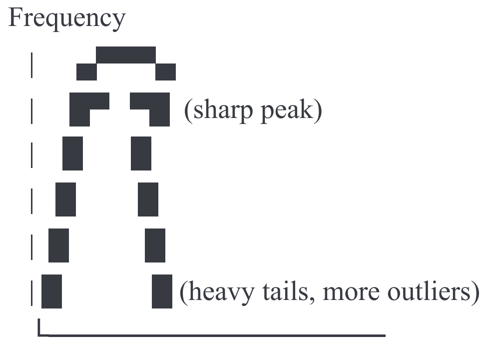
Leptokurtic: Heavy Tails (Kurtosis > 0)

More extreme values (outliers) than normal distribution.

Example: Stock returns

```
Most returns are moderate: ±2-3%
But occasionally: ±10-20% (crash or boom)
More extreme than you'd expect
```

Visual:



Platykurtic: Light Tails (Kurtosis < 0)

Fewer extreme values (outliers) than normal distribution. Distribution is flatter.

Example: Uniform distribution (all values equally likely)

```
No clustering around mean
Values spread uniformly
Very few extreme outliers
```

Visual:

Practical Importance of Kurtosis

Why it matters:

- Heavy-tailed data (high kurtosis) requires careful handling
- Statistical tests assume specific kurtosis (often normal)
- Risk management cares about tail risk (outliers)

Implications:

- **Heavy tails**: Standard deviation underestimates risk (outliers more common than expected)
- **Light tails**: Distribution is more stable and predictable

Example: Financial risk

```
Normal distribution kurtosis: 0
Stock returns kurtosis: 3-10 (much heavier tails)

Consequence: portfolio losses are more extreme than models predict
This is why "Black Swan" events surprise risk managers
```

Relationship Between Skewness and Kurtosis

- **Skewness** = asymmetry (left vs right)
- **Kurtosis** = tail heaviness (outliers)

A distribution can be:

- Symmetric but heavy-tailed (symmetric with outliers)
- Skewed with heavy tails (asymmetric with outliers)
- Skewed with light tails (asymmetric, no outliers)

Summary: Using These Measures in Data Science

Workflow: Exploratory Data Analysis (EDA)

1. **Compute central tendency** (mean, median, mode)
   - Understand typical values
   - Detect if mean ≠ median (suggests skewness)
2. **Compute variability** (std dev, variance, range)
   - Understand data spread
   - Detect near-zero variance (useless features)
3. **Check for skewness**
   - Identify if data needs transformation
   - Decide between mean or median
4. **Check for kurtosis**
   - Identify heavy tails and outliers
   - Decide if robust methods are needed
5. **Visualize**
   - Histograms reveal shape visually

- Box plots show central tendency and spread
- Q-Q plots check normality

Real-World Example: Property Values Dataset

```
Properties: [100k, 150k, 200k, 250k, 300k, 5M]

Mean: $747k      (heavily influenced by one expensive property)
Median: $225k    (typical property price)
Std Dev: $2.1M   (huge spread)
Skewness: 2.3    (heavily right-skewed)
Kurtosis: 4.1    (heavy tails — one property creates outlier)

Interpretation:
- Median is more representative than mean
- Extreme outlier (5M property) distorts mean and std dev
- Use log transformation or robust methods
- Consider separate analysis: luxury vs standard properties
```

Takeaway

These five measures (mean, median, mode, variance, std dev, skewness, kurtosis) form the foundation of statistical thinking in data science. They answer:

- **What's typical?** (Central tendency)
- **How much does it vary?** (Variability)
- **Is it balanced or lopsided?** (Skewness)
- **Are outliers common?** (Kurtosis)

Answer these questions before modeling, and you'll build better models on cleaner data with realistic expectations.