Advanced Python Programming for Data Science

CH – 04

Applied Statistics and Exploratory Analysis

Baikuntha Acharya (baikunth2a@gmail.com)

Senior Lecturer, Sagarmatha Engineering College, Sanepa, Lalitpur

Statistical Measures

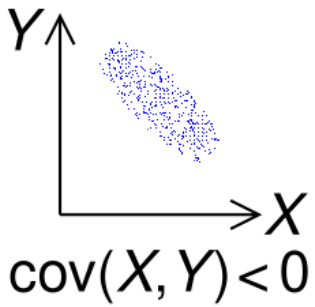
- ✓ Statistical measures help summarize, describe, and understand patterns in data.
  - In data science, they are used to:
    - Identify relationships between variables
    - Understand data distribution and shape
    - Support exploratory data analysis (EDA) and modeling decisions
  - In this section, we focus on four key measures:
    - **Correlation** – strength and direction of relationship
    - **Covariance** – joint variability of two variables
    - **Skewness** – asymmetry of data distribution
    - **Kurtosis** – peakedness and tail behavior of data
- ✓ These measures form the foundation for exploratory analysis and statistical modeling in Python.

Dataset:

```
np.random.seed(42)
x = np.random.lognormal(0, 0.5, 100)
y = 1.5 * x + np.random.normal(0, 0.2, 100)

sns.jointplot(x=x, y=y, kind='reg')
plt.show()
```

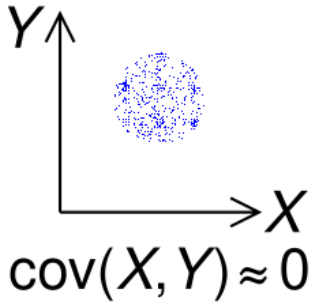
Covariance

- ✓ Variance Vs Covariance
  - **Variance** measures how a single variable deviates from its mean, whereas **covariance** measures how two variables vary in tandem from their means

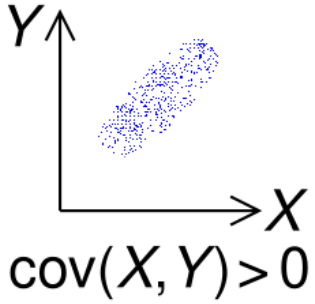


- ✓ **Covariance** (*The Joint Variability*) measures the tendency of two variables to **vary together**.
  - Consider two variables **X** and **Y**
  - Their deviations from the mean are:

$$dx_i = x_i - \bar{x}$$
$$dy_i = y_i - \bar{y}$$



- ✓ If **X and Y vary together**, their deviations usually have the **same sign**.
  - Same sign → positive contribution
  - Opposite sign → negative contribution
  - Adding these products summarizes how variables move together.



Covariance (Contd..)

✓ Covariance is defined as the **average of the product of deviations**:

$$\mathrm{Cov}(X,Y) = \frac{1}{n}\sum dx_i\, dy_i$$

- n = number of observations
- Both variables must have the **same length**

$$cov(X,Y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

✓ **Interpretation:**

- **Positive covariance** → variables move in the same direction
- **Negative covariance** → variables move in opposite directions
- **Zero covariance** → no linear co-variation

✓ **Limitation:**

- It is scale-dependent. If you change units (*e.g., meters to kilometers*), the covariance value changes, making it hard to interpret "strength."

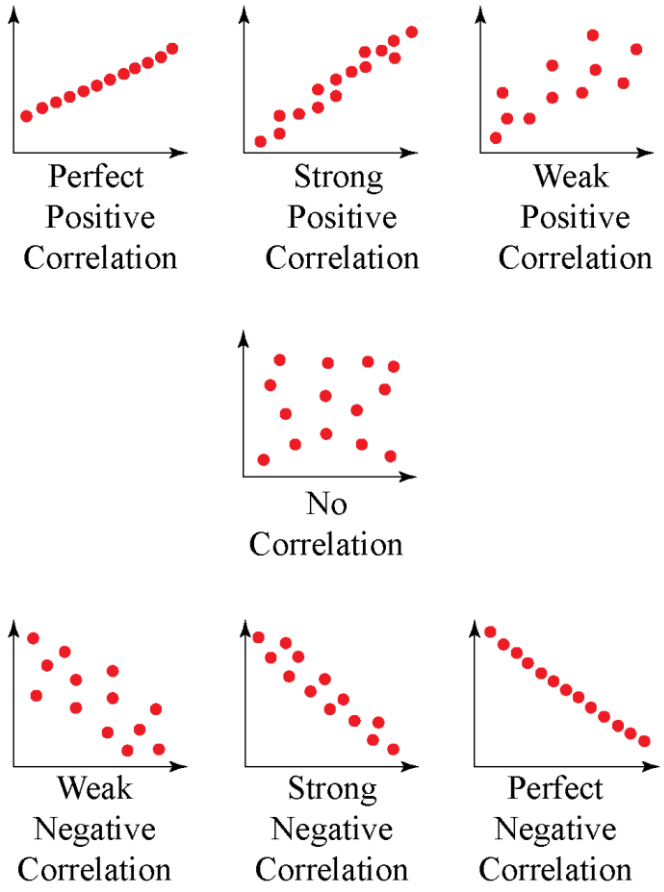
```
np.cov(x, y)[0, 1]
```

```
def Cov(xs, ys, meanx=None, meany=None):
    xs = np.asarray(xs)
    ys = np.asarray(ys)

    if meanx is None:
        meanx = np.mean(xs)
    if meany is None:
        meany = np.mean(ys)

    cov = np.dot(xs-meanx, ys-meany) / len(xs)
    return cov
```

Correlation

- ✓ Correlation is a statistical measure used to **quantify the strength of the relationship** between two variables.
  - Correlation is the dimensionless version of covariance.
  - **Range (Fixed Bounds):** -1.0 to +1.0
  - **Interpretation:** +1 (perfect linear), 0 (no linear relationship), -1 (perfect inverse linear)..
- ✓ Types:
  - df.corr() defaults to Pearson

Method	Best Used For...	Sensitivity
Pearson	Linear relationships between continuous variables.	High (Sensitive to outliers)
Spearman	Rank-based (monotonic) relationships; non-linear.	Low (Robust to outliers)
Kendall's Tau	Small datasets; ordinal data.	Very Low (Very robust)



Pearson's Correlation

- ✓ **Pearson's correlation (ρ)** measures the **strength and direction of a linear relationship** between two numerical variables.
  - Raw covariance is difficult to interpret because it depends on units
  - Pearson's correlation solves this by using **standardized values (z-scores)**
- ✓ Pearson's correlation is defined as:

$$\rho = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

  - Where:
  - $\mathrm{Cov}(X, Y)$ =covariance between X and Y
  - σ_X, σ_Y =standard deviations of X and Y
  - **Interpretation**
    - $\rho \in [-1, +1]$
    - **+1** → Perfect positive linear relationship
    - **−1** → Perfect negative linear relationship
    - **0** → No linear relationship

```
def Corr(xs, ys):
    xs = np.asarray(xs)
    ys = np.asarray(ys)

    meanx, varx = MeanVar(xs)
    meany, vary = MeanVar(ys)

    corr = Cov(xs, ys, meanx, meany) / math.sqrt(varx * vary)
    return corr
```

Pearson's Correlation

✓ Limitations of Pearson's Correlation

- Measures **only linear relationships**
- Sensitive to:
  - **Outliers**
  - **Skewed distributions**
- A correlation near **0 does not always mean no relationship**
- **Always visualize data using a scatter plot before relying on correlation values**
- Figure below shows scatter plots and correlation coefficients for several carefully constructed datasets.

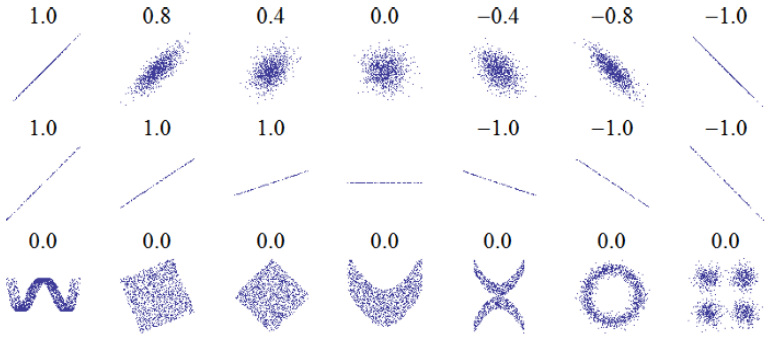


Figure: Examples of datasets with a range of correlations.

Spearman's Rank Correlation

- ✓ **Spearman's correlation (ρ_s)** measures the **monotonic relationship** between two variables using **ranks instead of raw values**.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

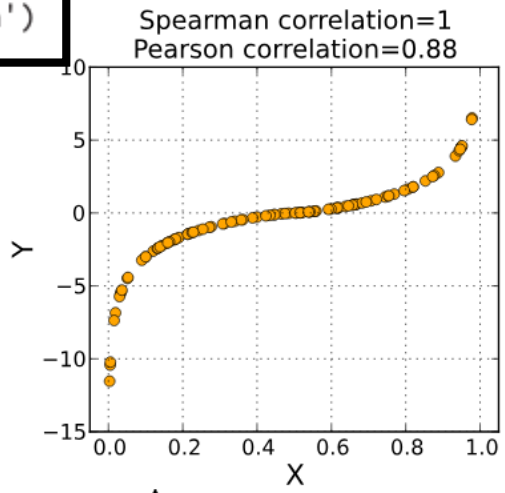
ρ = Spearman's rank correlation coefficient

d_i = difference between the two ranks of each observation

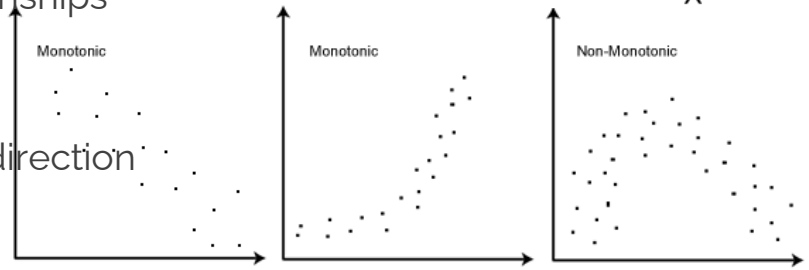
n = number of observations

```
def SpearmanCorr(xs, ys):
    xranks = pandas.Series(xs)
    yranks = pandas.Series(ys)
    return xs.corr(ys, method='spearman')
```

- ✓ **Key Characteristics**
  - Converts values to **ranks**
  - Computes Pearson's correlation on ranked data
  - More **robust to outliers and skewed data**
  - Captures **nonlinear but monotonic relationships**



- ✓ **When to Use Spearman**
  - Ordinal or ranked data or non-linear relationships
  - Data is skewed or contains outliers
  - Relationship is nonlinear but consistent in direction

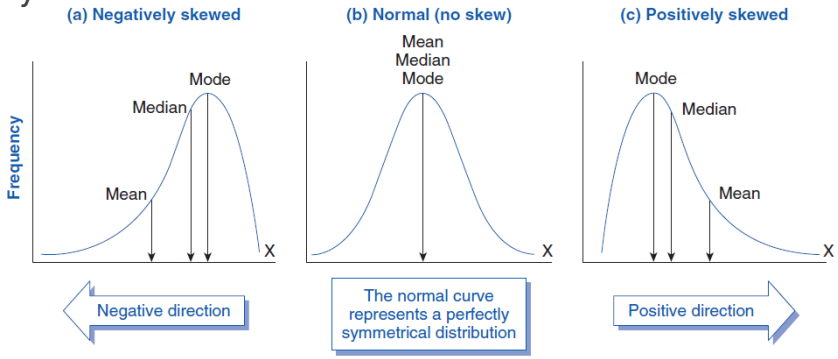


Python Implementation – Using Libraries

Feature	Library	Function	Primary Arguments
Pearson Correlation	**NumPy**	*np.corrcoef(x, y)*	**rowvar**: If True (default), rows are variables; if False, columns are variables. **dtype**: Specifies the data type of the resulting array for precision control.
	Pandas	*df.corr()*	**method**: Logic used ('pearson', 'spearman', 'kendall'). **min_periods**: Minimum valid observations required to return a non-NaN result.
	SciPy	*stats.pearsonr(x, y)*	**alternative**: Defines the hypothesis direction ('two-sided', 'less', 'greater').
Spearman Rank	**SciPy**	*stats.spearmanr(x, y)*	**nan_policy**: Defines handling for NaNs ('propagate', 'raise', 'omit'). **axis**: Determines if the calculation is along rows (1) or columns (0).
	Pandas	*x.corr(y, method="spearman")*	**numeric_only:** Control whether non-numeric columns are ignored.
Covariance	**NumPy**	*np.cov(x, y)*	**ddof**: Delta Degrees of Freedom; 1 for sample (default) or 0 for population. **fweights**: 1D array of integer weights representing observation frequency.
	Pandas	*df.cov()*	**min_periods**: Minimum observations needed per pair of columns. **ddof**: Degrees of freedom (defaults to 1 for sample covariance).

Skewness

- ✓ Skewness is a property that describes **the shape of a distribution**.
  - If the distribution is symmetric around its central tendency, it is unskewed.
- ✓ If the values extend farther to the right, it is "right skewed" and if the values extend left, it is "left skewed."
  - Skewness describes the shape (asymmetry) of a distribution
    - → It does NOT mean the data is biased
    - → It says nothing about sampling bias



- ✓ Direction of skewness
  - **Right (Positive) Skew**: Long tail on the right; ***Mean > Median > Mode***. Common in wealth distribution. The majority of data points are clustered on the left.
  - **Left (Negative) Skew**: Long tail on the left; ***Mean < Median < Mode***. Common in human longevity data. The majority of data points are clustered on the right.
  - Zero Skew: The distribution is symmetrical (Normal). **Mean = Median = Mode**.

Skewness (Contd..)

✓ Skewness is formally defined as the **third standardized moment**.

- While the first moment is the mean (location) and the second is variance (spread), the third moment captures the balance of the distribution.
- **μ_3 (Third Central Moment):** Cubing the deviations preserves their sign, allowing us to see which side of the mean has "heavier" or more distant values.
- **σ^3 (Standardization):** Dividing by σ^3 makes the measure dimensionless, meaning it is independent of the scale of the data.
- It allows us to quantify the "lean" of our data

$$\gamma_1 = E\left[\left(\frac{X - \mu}{\sigma}\right)^3\right] = \frac{\mu_3}{\sigma^3}$$

```
def CentralMoment(xs, k):
    # Calculate the average of (x - mean)^k
    return np.mean((xs - np.mean(xs))**k)
```

```
def StandardizedMoment(xs, k):
    var = CentralMoment(xs, 2)
    std = math.sqrt(var)
    return CentralMoment(xs, k) / std**k

def Skewness(xs):
    return StandardizedMoment(xs, 3)
```

✓ **Limitations**

- Highly sensitive to outliers
- Difficult to interpret magnitude alone

Skewness (Contd..)

- ✓ Pearson's median skewness coefficient is a measure of skewness based on the difference between the sample mean and median:

$$g_p = \frac{3(\bar{x} - m)}{S}$$

  - Where,
    - $\bar{X}$ is the sample mean,
    - m is the median, and
    - S is the standard deviation.
  - Interpretation:
    - $g_p < 0 \rightarrow$ left-skewed distribution
    - $g_p > 0 \rightarrow$ right-skewed distribution

```
def pearson_median_skewness(xs):
    xs = np.array(xs)
    mean = np.mean(xs)
    median = np.median(xs)
    std = np.std(xs, ddof=1) # Using sample

    return 3 * (mean - median) / std
```

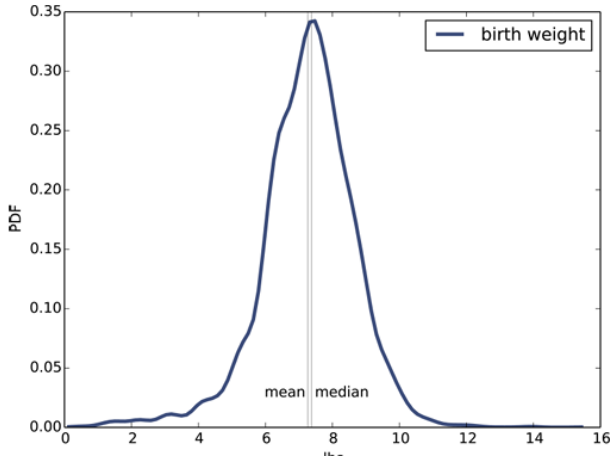
  - Uses mean and median, not higher-order moments
  - More robust than sample skewness because it is less affected by extreme values

Skewness (Contd..)

✓ Practical Interpretation with Real Data

- **Birth weight data (NSFG)**
  - Distribution shows a **longer left tail**
  - Mean < Median → indicates **left skew**
  - Both skewness measures are **negative**
  - Pearson's median skewness confirms left skewness

- **Adult weight data (BRFSS)**
  - Distribution shows a **longer right tail**
  - Mean > Median → indicates **right skew**
  - Both skewness measures are **positive**
  - Pearson's median skewness confirms right skewness



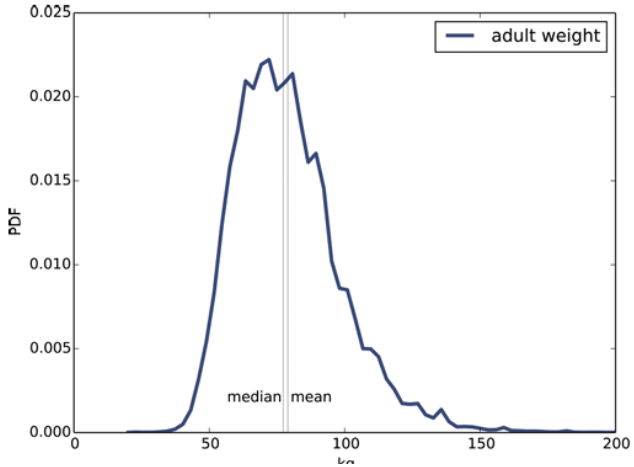
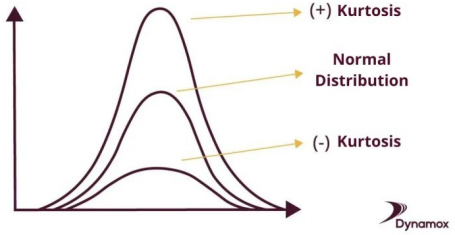


Figure: Estimated PDF of **(a)** birthweight data from the NSFG, **(b)** adult weight data from the BRFSS

Kurtosis

- ✓ **Kurtosis** is a statistical measure that describes the "***tailedness***" of a distribution.
  - It indicates how heavily the tails differ from a normal distribution.
  - For univariate data Y_1, Y_2, ..., Y_N, the formula for kurtosis is:

$$\text{kurtosis} = \frac{\sum_{i=1}^{N}(Y_i - \bar{Y})^4/N}{s^4}$$



  - Note that in computing the kurtosis, the standard deviation is computed using N in the denominator rather than N - 1.

Excess Kurtosis

- ✓ Standard normal distribution has kurtosis = **3**

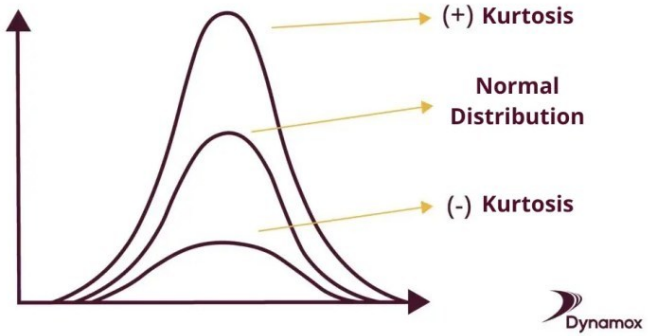
$$\textbf{Excess } \textit{Kurtosis} = \textit{Kurtosis} - 3$$

  - It is the measure of a distribution's kurtosis relative to that of a normal distribution,

Kurtosis (Contd..)

✓ Interpretation

- Excess Kurtosis > 0 → **Heavy-tailed** distribution
- Excess Kurtosis < 0 → **Light-tailed** distribution
- Excess Kurtosis = 0 → Normal distribution



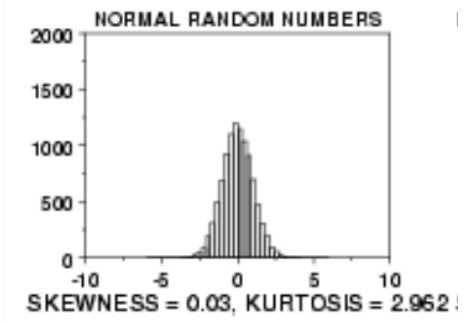

```
def StandardizedMoment(xs, k):
    variance = CentralMoment(xs, 2)
    standard_deviation = variance ** 0.5
    return CentralMoment(xs, k) / (standard_deviation ** k)

def Kurtosis(xs):
    return StandardizedMoment(xs, 4) - 3
```

Skewness and Kurtosis in Different Distributions

Skewness and Kurtosis in Different Distributions - *(Based on histograms of 10,000 randomly generated values)*

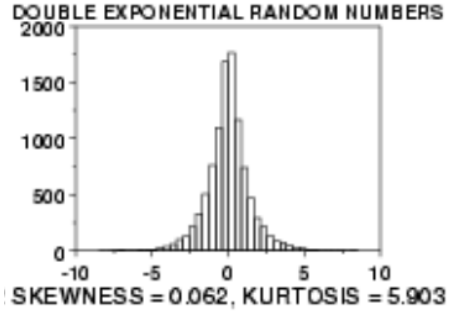
✓ Normal Distribution

- Symmetric with well-behaved tails
- **Skewness ≈ 0.03** → nearly zero (symmetric)
- **Kurtosis ≈ 2.96** → close to theoretical value (3)
- Confirms properties of a normal distribution



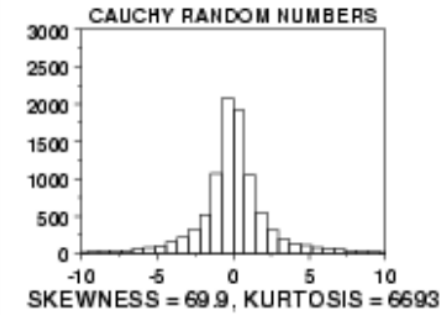
✓ Double Exponential Distribution

- Symmetric but more peaked than normal
- **Skewness ≈ 0.06** → symmetric
- **Kurtosis ≈ 5.90** → heavier tails than normal
- Indicates higher probability of extreme values



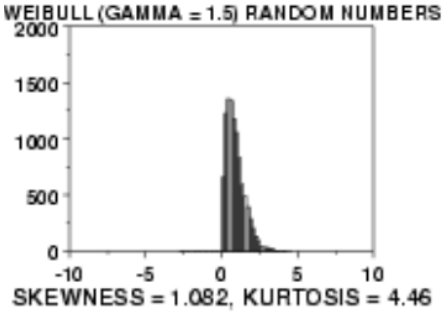
Skewness and Kurtosis in Different Distributions

✓ Cauchy Distribution



- Symmetric with extremely heavy tails
- **Skewness ≈ 69.99**
- **Kurtosis ≈ 6,693**
- Extreme values strongly distort skewness and kurtosis
- Demonstrates sensitivity of higher moments to outliers

✓ Weibull Distribution (Shape = 1.5)



- Positively skewed distribution
- **Skewness ≈ 1.08** → right-skewed
- **Kurtosis ≈ 4.46** → moderately heavy tails
- Shape depends on distribution parameter

Skewness and Kurtosis in Different Distributions

```
# Normal Distribution
data_norm = np.random.standard_normal(10000)
plt.hist(data_norm, bins=50)
plt.title(f"Skew: {Skewness(data_norm):.2f} | Kurt: {Kurtosis(data_norm):.2f}")
plt.show()

# Double Exponential
data_exp = np.random.laplace(size=10000)

# Cauchy
data_cauchy = np.random.standard_cauchy(10000)

# Weibull (Shape = 1.5)
data_weibull = np.random.weibull(1.5, 10000)
```

Python Implementation – Using Libraries

2. Shape Measures (Skewness & Kurtosis)

These functions describe the asymmetry and tail-weight of your data.

Feature	Library	Function	Key Parameters
Skewness	**Pandas**	`df.skew()`	`axis` : 0 for index, 1 for columns; `skipna=True`.
	SciPy	`stats.skew(a)`	`bias=True` : If False, calculations are corrected for statistical bias.
Kurtosis	**Pandas**	`df.kurt()`	Returns **Excess Kurtosis** (Normal = 0) by default.
	SciPy	`stats.kurtosis(a)`	`fisher=True` : If True, Fisher's definition is used (Normal = 0).

Probability Review

- ✓ **Probability** is the mathematical formalization of uncertainty, mapping the likelihood of an event to a value in the interval [0, 1]
  - **Probability** quantifies uncertainty associated with events
  - Used extensively in data science to:
    - Build models
    - Evaluate models
    - Make decisions under uncertainty

```
# Simulating 10,000 coin flips
flips = np.random.choice(['Heads', 'Tails'], size=10000)
p_heads = np.sum(flips == 'Heads') / 10000
print(f"Empirical Probability of Heads: {p_heads}")
```

- ✓ Sample Space (Ω) and Events:
  - An **experiment** has a set of all possible outcomes called the **sample space**
  - Sample Space: The set of all possible outcomes of a random experiment.
  - **Event:** Outcome of an experiment or a subset of the sample space.
  - Axiom of Probability: $P(\Omega) = 1$; for mutually exclusive events, $P(A \cup B) = P(A) + P(B)$

Probability Review (Contd..)

✓ Independence

- Two events are **independent** if knowing one does **not** affect the other
  - Example: In coin flip, first flip = *heads* gives no information about second flip
- **Mathematical Definition of Independence**:

$$P(E, F) = P(E)\,P(F)$$

✓ Dependance (Conditional Probability)

- Events are **dependent** if information about one affects the probability of the other.
- **Conditional Probability:** is the likelihood of an event E occurring, given that another event F has already occurred.

$$P(E|F) = \frac{P(E \cap F)}{P(F)}$$

- **Probability that event E occurs, given that F has already occurred**

$$P(E, F) = P(E|F)\,P(F)$$

Probability Review (Contd..)

✓ Bayes's Theorem:

- This is a formula used to update the probability of a hypothesis (H) as we gain new evidence (E).
- Mathematical Definition: $$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)}$$
  - **Prior ($P(H)$):** Initial probability before seeing evidence.
  - **Likelihood ($P(E \mid H)$):** Probability of the evidence if the hypothesis is true.
  - **Posterior ($P(H \mid E)$):** Updated probability after seeing evidence.
- Expanded form:

$$P(\mathrm{H} \mid E) = \frac{P(\mathrm{E} \mid H)P(H)}{P(\mathrm{E} \mid H)P(H) + P(\mathrm{E} \mid \neg H)P(\neg H)}$$

```
p_rain = 0.20
p_cloudy = 0.40
p_cloudy_given_rain = 0.85 # If it rains, it's almost always cloudy

# Bayes' Theorem: P(Rain | Cloudy)
p_rain_given_cloudy = (p_cloudy_given_rain * p_rain) / p_cloudy
print(f"Probability of rain given it is cloudy: {p_rain_given_cloudy:.2f}")
```

Probability Review (Contd..)

✓ Bayes's Theorem: **Medical Test Example**

- Disease affects **1 in 10,000 people**
- Test accuracy: **99%**
- Define events:
  - **D**: Person has disease
  - **T**: Test result is positive
- **Apply Bayes's Theorem**

- **Known Probabilities**
  - $P(D) = 0.0001$
  - $P(\neg D) = 0.9999$
  - $P(T \mid D) = 0.99$
  - $P(T \mid \neg D) = 0.01$

$$P(D \mid T) = \frac{0.99 \times 0.0001}{(0.99 \times 0.0001) + (0.01 \times 0.9999)} \approx 0.0098$$

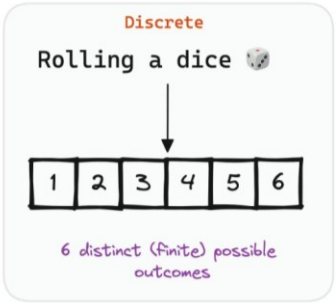
Result: Less than 1% of people who test positive actually have the disease

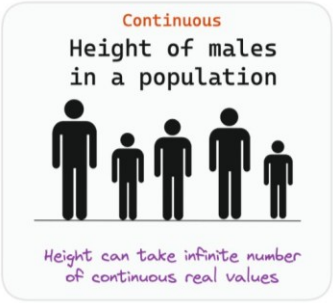
```
P_D = 0.0001          # Probability of having disease
P_notD = 1 - P_D
P_T_given_D = 0.99  # Probability test is positive if disease
P_T_given_notD = 0.01  # Probability test is positive if no disease

# Bayes' Theorem
P_D_given_T = (P_T_given_D * P_D) / ((P_T_given_D * P_D) + (P_T_given_notD * P_notD))
```

Probability Review (Contd..)

✓ Random Variables

- A **random variable** assigns numerical values to outcomes of a random process
- Each possible value has an associated **probability**
- **Examples**
  - **Coin flip**
    - Heads → 1, Tails → 0
    - $P(X = 0) = 0.5,\ P(X = 1) = 0.5$





✓ Discrete random variable:

- X is a discrete random variable if the sample space is a **finite countable** set.
  - E.g: Coin flip => $S = \{ HHH, HHT, HTH, HTT, THH, THT, TTH, TTT \}$

✓ **Continuous distributions**

- values over a range (real numbers), uncountable (*e.g., height, weight, time*)
- **Probability of a single value is zero**; we calculate the probability over an interval

Probability Review (Contd..)

✓ Random Variables : **Expected Value**

- The **expected value** is the weighted average of all possible values

$$E[X] = \sum x \cdot P(X = x)$$

- **Examples**
  - **Coin flip:**

$$E[X] = 0 \times 0.5 + 1 \times 0.5 = 0.5$$

  - **range(10):**

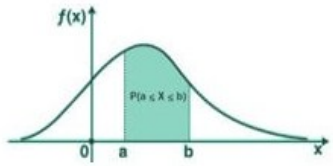
$$E[X] = 0 \times 0.1 + 1 \times 0.1 + 2 \times 0.1 + \ \ldots\ldots + 9 \times 0.1 = 4.5$$

```
values = [0, 1]          # possible outcomes
probs = [0.3, 0.7]       # their probabilities

expected_value = sum(x * p for x, p in zip(values, probs))
```

✓ **Probability Density Function (PDF)**

- Used for continuous random variables
- The probability of an exact point is 0 and probability of an interval = **area under the curve:** $P(a \leq X \leq b) = \int_a^b f\,(x)\,dx$ $\quad F(x) = \mathrm{P}(a \leq x \leq b) = \int_a^b f(x)d(x)$
- **Python:** `scipy.stats.[dist].pdf(x)`

Here,
$f(x)$ is the PDF and $F(x)$ is the CDF



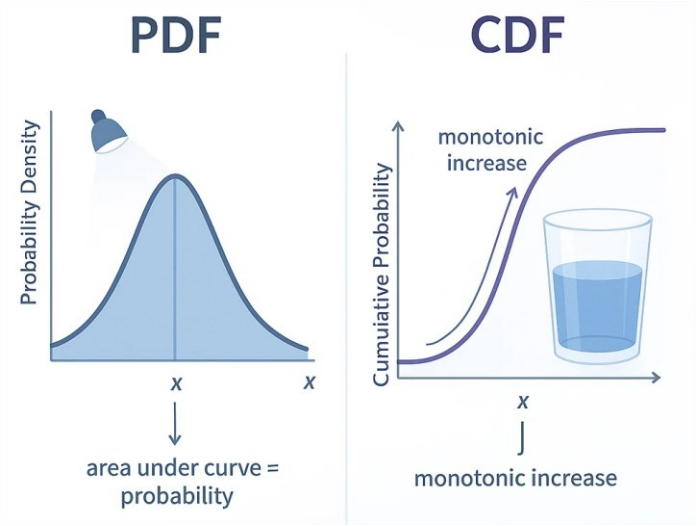
Probability Review (Contd..)

✓ Probability Mass Function (PMF)

- **Definition:** Specific to Discrete variables. It returns the probability that X is exactly equal to a value k
- Python: `sscipy.stats.[dist].pmf(k)`

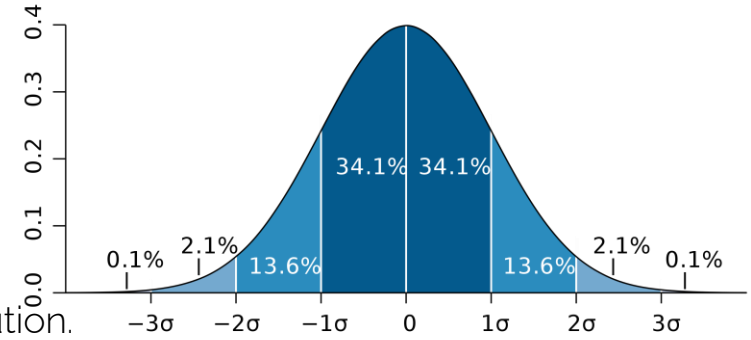
✓ Cumulative Distribution Function (CDF)

- It gives the probability that a random variable is less than or equal to a certain value. $\boldsymbol{P(X \leq x)}$
- **Python:** `sscipy.stats.[dist].cdf(x)`
- Applicable to **both:** pdf and pmf



Probability Review (Contd..)

Normal Distribution



✓ The Normal Distribution

- It is the classic **bell-shaped** distribution.
- It is completely defined by two parameters:
  - Mean (μ): determines the center of the distribution.
  - Standard deviation (**σ**): controls how wide or spread out the curve is.
- Symmetric around the mean: left and right sides are mirror images.
- Many real-world phenomena (heights, test scores, measurement errors) approximately follow a normal distribution.

✓ **Probability Density Function (PDF)**

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- The PDF gives the **density** at value x
- Higher σ → wider curve
- Lower σ → narrower curve

In the `scipy.stats` library, the **`norm.cdf()`** function is used to determine the **Cumulative Distribution Function** of a Normal (Gaussian) distribution.

Probability Review (Contd..)

✓ The Normal Distribution - **Python Implementation**

```
import math
SQRT_TWO_PI = math.sqrt(2 * math.pi)

def normal_pdf(x: float, mu: float = 0, sigma: float = 1) -> float:
    return (math.exp(-(x-mu) ** 2 / 2 / sigma ** 2) / (SQRT_TWO_PI * sigma))
```

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- In Figure 6-2, we plot some of these PDFs to see what they look like :

```
import matplotlib.pyplot as plt
xs = [x / 10.0 for x in range(-50, 50)]
plt.plot(xs,[normal_pdf(x,sigma=1) for x in xs],'-',label='mu=0,sigma=1')
plt.plot(xs,[normal_pdf(x,sigma=2) for x in xs],'--',label='mu=0,sigma=2')
plt.plot(xs,[normal_pdf(x,sigma=0.5) for x in xs],':',label='mu=0,sigma=0.5')
plt.plot(xs,[normal_pdf(x,mu=-1)   for x in xs],'-.',label='mu=-1,sigma=1')
plt.legend()
plt.title("Various Normal pdfs")
plt.show()
```

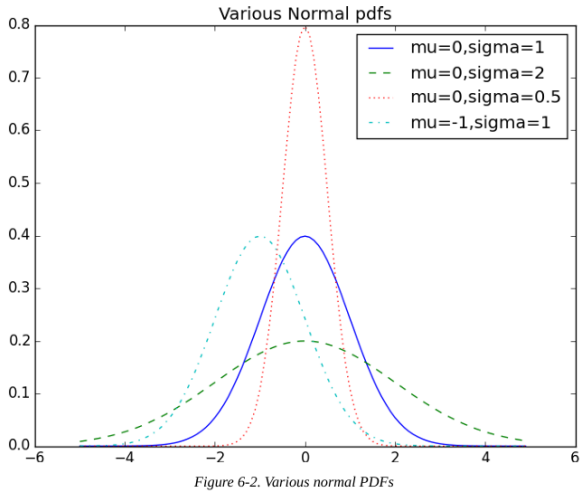


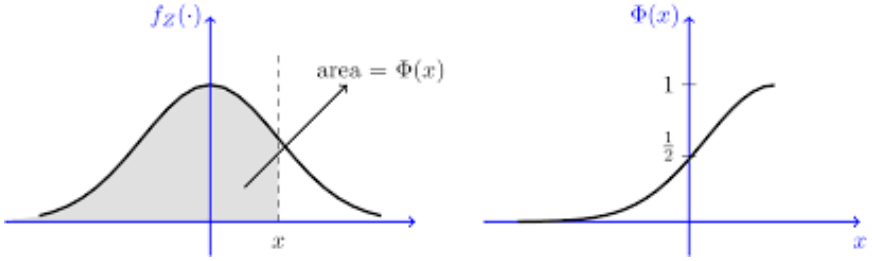
Figure 6-2. Various normal PDFs

- Same μ, larger σ → curve becomes wider
- Smaller σ → curve becomes taller and narrower
- Changing μ shifts the curve left or right
- Standard normal distribution ($\mu = 0, \sigma = 1$)
  - About **68%** of values lie within **1 standard deviation** of the mean, **95%** within **2 standard deviations**, and **99.7%** within **3 standard deviations** (Empirical Rule).

Normal Distribution (Cont..)

- ✓ **Standard Normal Distribution & CDF**
- ✓ Standard Normal Distribution
  - When **μ = 0** and **σ = 1**
  - Called **Standard Normal Distribution**
  - $X = \sigma Z + \mu \quad \Rightarrow \quad Z = \frac{X-\mu}{\sigma}$

Note: PDF ≠ CDF



- ✓ **Normal CDF**
  - CDF gives: $P(X \leq x)$
  - Using Library:

```
from scipy.stats import norm

norm.cdf(x)         # Standard normal: mean=0, std=1
norm.cdf(x, loc=μ, scale=σ)  # Normal with mean μ and std σ
```

  - Using Python's **error function (erf):**

$$\text{cdf}(x) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right]$$

```
def normal_cdf(x, mu=0, sigma=1):
    return 0.5 * (1 + math.erf((x - mu) / (sigma * math.sqrt(2))))
```

Probability Review (Contd..)

Normal Distribution (Cont..)

- ✓ Calculate the probability that sales will be between 90 and 120, given an average (μ) of 100 and a standard deviation (σ) of 15.

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

# Parameters
mu, sigma = 100, 15
low, high = 90, 120
dist = norm(mu, sigma)

# 1. Calculation: P(90 < Sales < 120)
# Logic: CDF(Upper) - CDF(Lower)
prob = dist.cdf(high) - dist.cdf(low)

# 2. Visualization
x = np.linspace(50, 150, 100)
pdf = dist.pdf(x)
plt.plot(x, pdf, 'k')
plt.fill_between(x, pdf, where=((x >= low) & (x <= high)), color='green', alpha=0.3)
plt.title(f"P({low} < Sales < {high}) = {prob:.4f}")
plt.show()
```

Alternatively, you can calculate z-score and use CDF of standard normal distribution

Normal Distribution - Example

- ✓ Let's say the height of students in a class is modeled as a **continuous random variable** with a normal distribution. The height is measured in centimeters and ranges from 150 to 190 cm. What is the probability of a randomly selected student having a height between **165 cm** and **175 cm?**
  - Mean height = 170 cm and SD = 5 cm
- ✓ Standardize the range using Z-scores:

$$Z_1 = \frac{165 - 170}{5} = -1 \qquad Z_2 = \frac{175 - 170}{5} = 1$$

$$P(165 \leq X \leq 175) = P(Z \leq 1) - P(Z \leq -1)$$
$$= 0.8413 - 0.1587 = 0.6826$$

```
from scipy.stats import norm

z1 = (165 - 170) / 5
z2 = (175 - 170) / 5

prob = norm.cdf(z2) - norm.cdf(z1)
```

```
from scipy.stats import norm

prob = norm.cdf(175, loc=170, scale=5) - norm.cdf(165, loc=170, scale=5)
```

Probability Review (Contd..)

Bernoulli Distribution

- ✓ **Bernoulli Distribution** is a discrete distribution for a single trial with two outcomes: success (1) and failure (0).
- ✓ It is defined by one parameter p, the probability of success; failure occurs with probability $1 - p$.
  - *Example:* One coin toss where heads = 1 and tails = 0.
- ✓ Its **PMF** gives the probability of each possible outcome:

$f(x) = p^x(1-p)^{(1-x)}$, x=0,1 or $P(X = x) = \begin{cases} p, & \text{if } x = 1 \\ 1 - p, & \text{if } x = 0 \end{cases}$

Where:

- p Probability of success ($0 < p < 1$).
- $1-p$ Probability of failure.
- x Outcome of a single trial
  - x=1 Indicates success.
  - x=0 Indicates failure.

Mean
$\mu = p$ (average probability of success).
Variance
$\sigma^2 = p(1-p)$ (measure of variability in outcomes).

Probability Review (Contd..)

Bernoulli Distribution (Cont..)

- ✓ A company manufactures electronic devices, and the probability of a device passing quality control is 70% *(i.e., defect-free, $p = 0.7$)*.
  - a) The probability of a device failing quality control is 30% *($1-p = 0.30$)*.
  - b) Average probability of success?

1. **For defect-free ($X = 1$):**

$$P(X = 1) = p^1(1-p)^{1-1} = (0.7)^1(0.3)^0 = 0.7$$

2. **For defective ($X = 0$):**

$$P(X = 0) = p^0(1-p)^{1-0} = (0.7)^0(0.3)^1 = 0.3$$

```python
from scipy.stats import bernoulli

p = 0.7                              # Probability of passing quality control

# a) Probability of a device failing quality control
fail_prob = bernoulli.pmf(0, p)

# b) Average probability of success (expected value)
avg_success = bernoulli.mean(p)
```

Binomial Distribution

✓ **Binomial Distribution** is a discrete distribution that counts the number of successes in a ***fixed number of independent Bernoulli trials***, each with success probability p.

- The Binomial distribution is an extension of the Bernoulli distribution to multiple trials.
- Or, Bernoulli distribution is a special case of Binomial distribution.
- *Example:* Toss a coin 10 times and count heads.

✓ **Key Properties:**

- Fixed number of trials n
- Success probability p constant for each trial
- Random variable k = number of successes

✓ **PMF:** Probability of exactly k successes in n trials.

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

- X is the number of successes,
- n is the number of trials,
- k is the number of successes,
- p is the probability of success on a single trial,
- $\binom{n}{k}$ is the binomial coefficient, calculated as $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

Mean (μ):

The expected number of successes in n trials is:

$$\mu = np$$

Variance (σ^2):

The variance of the number of successes in n trials is:

$$\sigma^2 = np(1-p)$$

Binomial Distribution - Example

✓ A company manufactures electronic devices, and the probability of a device passing quality control is 85% (i.e., defect-free, p=0.85).

a) Probability of exactly 3 defect-free devices out of 5?

$$P(X = 3) = \binom{5}{3}(0.85)^3(0.15)^2$$

- Since we have multiple trials (5 trials), this becomes a **Binomial Distribution** problem.
  - $n = 5$ (number of trials),
  - $k = 3$ (desired number of successes),
  - $p = 0.85$ (probability of success),
  - $1 - p = 0.15$ (probability of failure),

b) Probability of exactly 4 failed devices out of 5

- Now, we are interested in exactly 4 failures, so k=4:

$$P(X = 4) = \binom{5}{4}(0.15)^4(0.85)^{5-4}$$

```
from scipy.stats import binom

# a) Probability of exactly 3 defect-free devices out of 5
prob_3_pass = binom.pmf(k=3, n=5, p=0.85)

# b) Probability of exactly 4 failed devices out of 5
prob_4_fail = binom.pmf(k=4, n=5, p=0.15)
```

```
prob_3_pass = math.comb(5, 3) * 0.85**3 * 0.15**2
prob_4_fail = math.comb(5, 4) * 0.15**4 * 0.85**1
```

Probability Review (Contd..)

Poisson Distribution

- ✓ It is a **discrete probability distribution** that models the number of events occurring within a fixed interval of time or space, given the average rate of occurrence.
- ✓ It is often used for rare events that happen independently over time or space.
- ✓ Key Properties:
  - Model rare events happening in a fixed interval.
  - Defined by a single parameter λ, which is the average number of events in the interval.
  - The events are **independent** and occur with a **constant mean rate**.

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

$P(X = k)$ is the probability of exactly k events occurring,

λ is the average rate (mean number of events in a fixed interval),

e is Euler's number (≈ 2.718),

k is the number of events,

Probability Review (Contd..)

Poisson Distribution - Numerical

✓ Let's say that a call center receives **4 calls per hour** on average (λ = 4).

- We want to find the probability that the call center will receive exactly **3 calls** in an hour.

$\lambda = 4$ (average number of calls),

$k = 3$ (number of calls),

e is Euler's number (approximately 2.71828).

Plug the values into the formula:

$$P(X = 3) = \frac{4^3 e^{-4}}{3!} = \frac{64 \times e^{-4}}{6} \approx \frac{64 \times 0.0183}{6} \approx 0.197$$

```
prob_3_calls = (4**3 * math.exp(-4)) / math.factorial(3)
```

```
from scipy.stats import poisson

prob_3_calls = poisson.pmf(k=3, mu=4)
```

Probability Review (Contd..)

✓ Commonly Used Libraries for Probability

- **SciPy**, **NumPy**, and **Matplotlib**.

✓ SciPy (scipy.stats)

- This is the primary library for probability theory.
- To calculate exact probabilities (PDF/PMF), cumulative probabilities (CDF), and inverse lookups (PPF).

✓ NumPy (numpy.random)

- To generate large arrays of random data for frequentist probability experiments (e.g., simulating 1,000,000 dice rolls).

```
np.random.rand(d0=5)                                    # Uniform [0, 1)
np.random.randn(d0=5)                                   # Standard normal (μ=0, σ=1)
np.random.randint(low=1, high=10, size=5)               # Random integers [1, 10)
np.random.choice(a=[10, 20, 30], size=2, replace=True) # Random selection from array
np.random.normal(loc=0, scale=1, size=5)                # Normal distribution (μ=0, σ=1)
np.random.uniform(low=5, high=15, size=5)               # Uniform distribution [low, high)
```

Probability Review (Contd..)

✓ SciPy

1. Universal Distribution Methods

Every distribution object in `scipy.stats` (e.g., `norm`, `binom`, `poisson`) uses these standard methods:

Function	Academic Terminology	Purpose
`.pmf(k)`	**Probability Mass Function**	Calculates the probability of an exact outcome $P(X = k)$ for **discrete** data.
`.pdf(x)`	**Probability Density Function**	Calculates the relative likelihood (density) for **continuous** data.
`.cdf(x)`	**Cumulative Distribution Function**	Calculates the probability that X is less than or equal to x ($P(X \leq x)$).
`.ppf(q)`	**Percent Point Function**	The inverse of the CDF. Given a probability q, it returns the corresponding value x.
`.rvs(size)`	**Random Variates**	Generates random samples following the specified distribution.

Sampling

- ✓ Sampling is the process of selecting a subset of data (sample) from a larger set called the population
  - Let the **population** be denoted by $\mathcal{P} = \{x_1, x_2, \ldots, x_N\}$,where $N$is the population size. A **sample** is a subset $\mathcal{S} = \{x_{i_1}, x_{i_2}, \ldots, x_{i_n}\}$,where $n \ll N$.
  - In statistics, a population is a large, defined (sometimes theoretical) collection of data
  - In practice, we usually cannot observe the entire population
  - Therefore, estimation of population parameters are drawn using sample data
- ✓ **Population**
  - Represents all possible observations
  - Has an unknown underlying distribution
- ✓ **Sample**
  - A subset drawn from the population
  - Has an empirical (observed) distribution
  - A **sampling procedure** connects population → sample

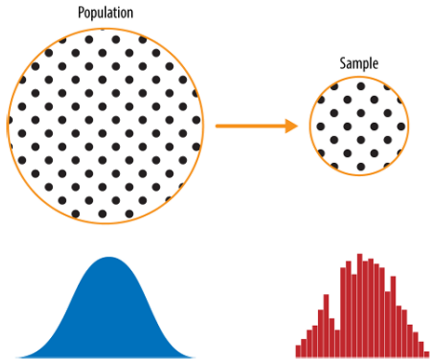


Figure: Population versus sample

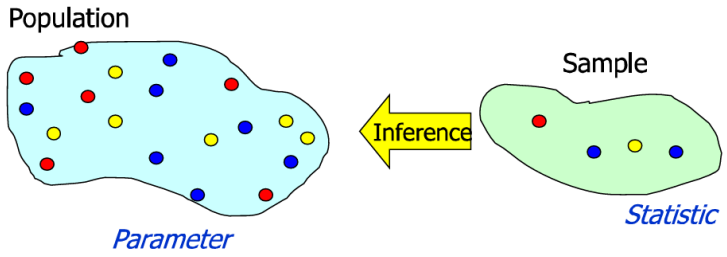
Sampling (Cont..)

- ✓ **Purpose of Sampling**
  - Estimate population parameters
  - Reduce cost and time
  - Enable statistical inference
- ✓ **Sampling Methods (Overview)**
  - **Probability sampling**: Random, unbiased, generalizable
  - **Non-probability sampling**: Convenient, faster, less reliable
- ✓ **Sampling Error:** Difference between a statistic and its corresponding parameter due to random selection.
- ✓ **Bias:** occurs when the sample is not representative of the population due to less sample size, poor sampling frame or non-random selection.
- ✓ **Sample Size (n):** Larger n: lower variability, higher accuracy
  - Smaller n: higher uncertainty, lower cost

Sampling (Cont..)

Parameter vs Statistic

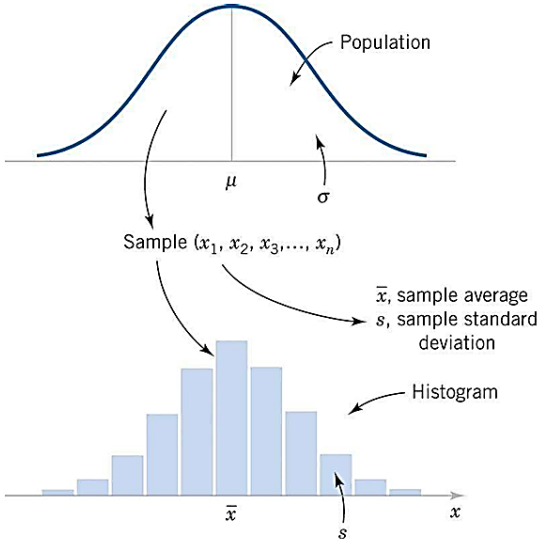
- ✓ **Population ($\mathcal{P}$, N):** Entire set of elements under study => *Represented by* ***parameters***
- ✓ **Sample ($\mathcal{S}$, n):** Selected subset where $n \ll Nn \ll N$ => *Represented by* ***statistic***



Aspect	Parameter (Population)	Statistic (Sample)
Definition	Numerical measure of population	Numerical measure of sample
Known/Unknown	Usually unknown	Computable
Mean	μ	$\bar{x}$
Variance	σ^2	s^2
Proportion	p	$\hat{p}$
Role	Target of inference	Estimator of parameter

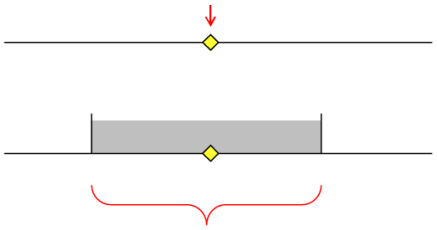
Point & Interval Estimate

✓ From sampling statistic, we can say (with some __ % certainty) that the population parameter of interest is between some lower and upper bounds.



1) Point Estimate →

2) Interval Estimate →

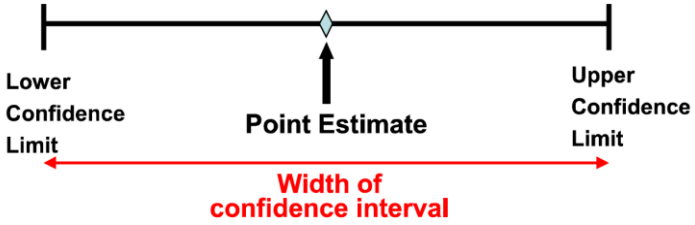


- **Point estimate:**

A point estimate is a single number,

- **Interval estimate**

Provides additional information about variability



For example, suppose we want to estimate the mean summer income of a class of business students. For n=25 students, $\bar{x}$ is calculated to be 400 $/week.

point estimate

interval estimate

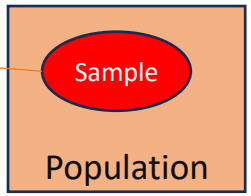
An alternative statement is:

The mean income is ***between*** 380 and 420 $/week.

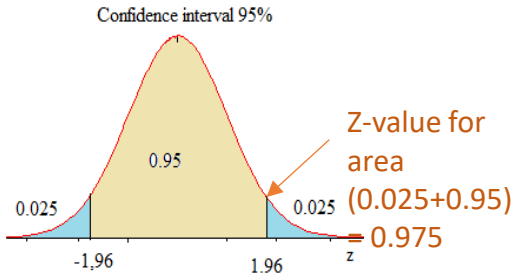
Sampling (Cont..)

Parameter Estimation - Example

- ✓ A company surveys **200 employees** to find the average time spent coding daily. The sample mean ($\bar{x}$) is **6 hours**, with a standard deviation (s) of **1 hour**.



  - Estimate the **population mean coding time** for all employees with **95% confidence**.



$$CI = \bar{x} \pm Z \cdot \frac{s}{\sqrt{n}}$$

$$CI = 6 \pm 1.96 \cdot \frac{1}{\sqrt{200}}$$

$$CI = 6 \pm 1.96 \cdot 0.0707 = 6 \pm 0.138$$

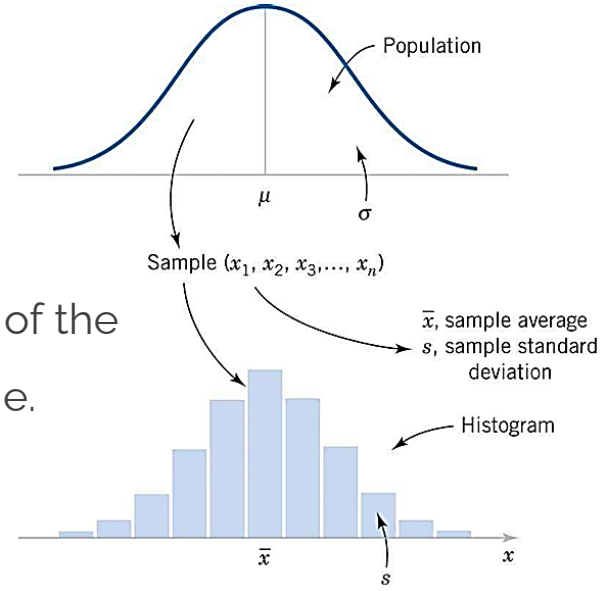
$$CI = [5.862, 6.138]$$

```
# Standard error
sd_err = 1 / math.sqrt(200)

# 95% confidence interval using z
ci = norm.interval(0.95, loc=6, scale=sd_err)
```

```
# Standard error
sd_err = 1 / math.sqrt(200)

# Upper-tail Z value for 95% confidence
Z_upper = 0.95 + (1 - 0.95)/2

# 95% confidence interval
ci = (
    6 - norm.ppf(Z_upper) * sd_err,
    6 + norm.ppf(Z_upper) * sd_err
)
```

- ✓ **Interpretation**: With 95% confidence, the average coding time for all employees is between **5.86 and 6.14 hours per day**.

Sampling Distributions



- ✓ Suppose we wish to estimate the mean $\boldsymbol{\mu}$ of a population.
  - Imagine however that we take sample after sample, all of the same size $\boldsymbol{n}$, and compute the sample mean $\overline{\boldsymbol{x}}$ each time.
  - The sample mean $\overline{\boldsymbol{x}}$ is a random variable:
    - it varies from sample to sample in a way that cannot be predicted with certainty.
  - We will write $\overline{\boldsymbol{X}}$ when the sample mean is thought of as a random variable, and write $\overline{\boldsymbol{x}}$ for the values that it takes.
  - The random variable $\overline{\boldsymbol{X}}$ has a mean, denoted $\boldsymbol{\mu_{\overline{X}}}$, and a standard deviation, denoted $\boldsymbol{\sigma_{\overline{X}}}$
  - Sample mean and sample SD with relation to population is:

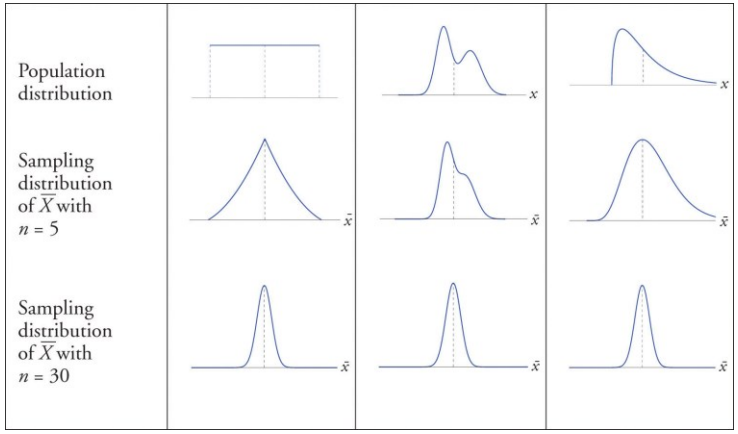
$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}}$$

$$\mu_{\bar{X}} = \mu$$

Central Limit Theorem

✓ It states that sampling distribution (*the distribution of point estimates*) will **approach a normal distribution** as we increase the number of samples taken.

- For samples of size $\boldsymbol{30\ or\ more}$, the sample mean is approximately normally distributed, with mean $\boldsymbol{\mu_{\overline{X}} = \mu}$ and standard deviation $\boldsymbol{\sigma_{\overline{X}} = \frac{\sigma}{\sqrt{n}}}$, where $\boldsymbol{n}$ is the sample size.
- The larger the sample size, the better the approximation.



✓ **Mathematical Idea**

- Let $x_1, x_2, \ldots, x_n$ be i.i.d. random variables with:
  - Mean = μ
  - Standard deviation = σ
- For large n:
  - The average $\frac{x_1+\cdots+x_n}{n}$ is approximately Normal with:
    - Mean ($\bar{X}$)= μ
    - Standard deviation (s) = $\frac{\sigma}{\sqrt{n}}$

Sampling (Cont..)

Central Limit Theorem

- ✓ Roll a fair 6-sided die repeatedly; take 30 rolls at a time and repeat this **1000 times** to show that the distribution of sample means follows a normal distribution.

```
population = np.arange(1,7)  # Die values 1-6

plt.hist(population)
plt.title("Original Die Roll Distribution")
plt.show()

sample_size, num_samples = 30, 1000  # 30 rolls per sample, repeated 1000 times

sample_means = []
for _ in range(num_samples):
    sample = np.random.choice(population, sample_size, replace=True)
    sample_means.append(np.mean(sample))

plt.hist(sample_means, bins=30, density=True)
x = np.linspace(min(sample_means), max(sample_means), 100)
plt.plot(x, norm.pdf(x, np.mean(sample_means), np.std(sample_means)))
plt.title("Distribution of Sample Means ~ Normal by CLT")
plt.show()

# Compare population and sample statistics
print("Population mean:", np.mean(population), "SD:", np.std(population, ddof=0))
print("Sample means mean:", np.mean(sample_means), "SD:", np.std(sample_means, ddof=1))
```

Central Limit Theorem (Cont..)

Numerical - Example

Let $\overline{X}$ be the mean of a random $sample\ size = 50$ drawn from a population with $mean = 112$ and $standard\ deviation = 40$.

- Find the probability that $\overline{X}$ assumes a value between $110\ \ and\ \ 114$
- Recall:

$$\mu_{\bar{X}} = \mu \qquad \sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}}$$

Since the sample size is at least 30, the Central Limit Theorem applies:

$$\begin{aligned} P(110 < \overline{X} < 114) &= P\left(\frac{110 - \mu_{\overline{X}}}{\sigma_{\overline{X}}} < Z < \frac{114 - \mu_{\overline{X}}}{\sigma_{\overline{X}}}\right) \\ &= P\left(\frac{110 - 112}{5.65685} < Z < \frac{114 - 112}{5.65685}\right) \\ &= P(-0.35 < Z < 0.35) \\ &= 0.6368 - 0.3632 \\ &= 0.2736 \end{aligned}$$

```
mu, sigma, n = 112, 40, 50
se = sigma / math.sqrt(n)

z1, z2 = (110 - mu)/se, (114 - mu)/se
prob = norm.cdf(z2) - norm.cdf(z1)
```

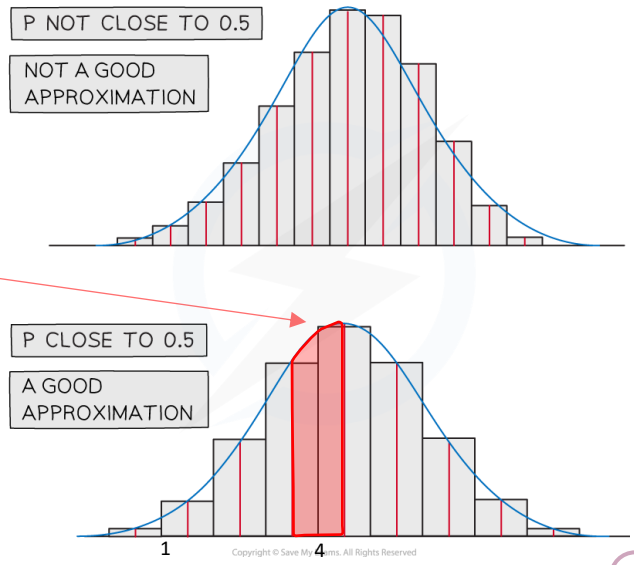
be careful to use $\sigma_{\overline{X}}$ and not σ when we standardize.

Normal Approximation

- ✓ It is a statistical method used to approximate the probability distribution of a discrete random variable by a continuous normal distribution.
- ✓ This is useful when the number of trials (n) is large, and the discrete distribution becomes close to a normal distribution by the **Central Limit Theorem (CLT)**.

Eg.: Number of heads in 100 tosses, 1000 times

- ✓ Point Correction: $P(X = k) \approx P(k - 0.5 \leq Z \leq k + 0.5)$
  - Discrete ($X = 4$) → Continuous ($3.5 \leq X_N < 4.5$)
- ✓ **Conditions for Normal Approximation:**
  - The sample size (n) should be sufficiently large.
  - The success probability (p) of the event should not be too close to 0 or 1 ***(i.e., 0 < p < 1). Ideally, 0.5***



Normal Approximation (Cont..)

Conditions for Normal Approximation

✓ Binomial Distribution

- If X is a binomial random variable with parameters n and p, then:

$$X \sim \text{Binomial}(n, p)$$

For large n, we approximate X by a normal distribution:

$$X \sim \mathcal{N}(\mu = np, \sigma = \sqrt{np(1-p)})$$

- This approximation is valid when:

$$n \cdot p \geq 10 \quad \text{and} \quad n \cdot (1-p) \geq 10$$

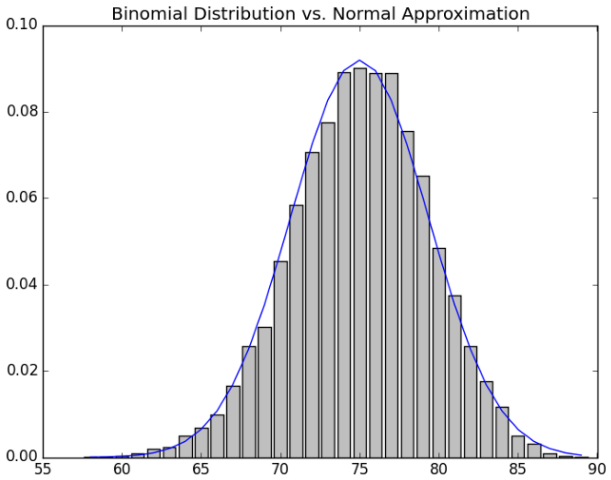


Figure 6-4. The output from binomial histogram

✓ Poisson's Distributions

- Condition: $\lambda \geq 10$.
- Approximation: $X \sim \mathcal{N}(\mu = \lambda, \sigma = \sqrt{\lambda})$.

Normal Approximation

Numerical Example - Binomial

✓ The random variable $X \sim B\,(1250, 0.4)$. Use a suitable approximating distribution to approximate $P\,(485 \leq X \leq 530)$.

Step 1: Find $X_N \sim N(\mu, \sigma^2)$

$\mu = np = (1250)(0.4) = 500$

$\sigma^2 = np\,(1-p) = (1250)(0.4)(0.6) = 300$

$\sigma = \sqrt{300}$

Step 2: Apply continuity corrections

$485 \leqslant X \leqslant 530$

Include 485 — Include 530

$484.5 \leqslant X_N \leqslant 530.5$

Step 3: Calculate the probability

$P(484.5 \leqslant X_N \leqslant 530.5) = 0.775451...$

Use Normal CD
Remember $\sigma = \sqrt{300}$
not 300

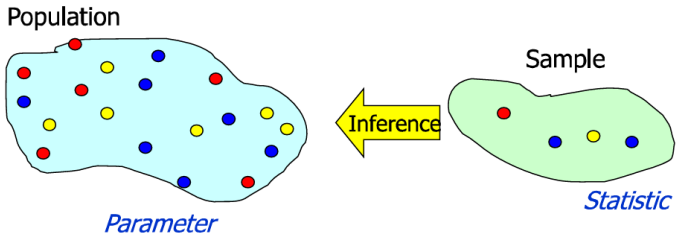
0.775 (3.s.f.)


```
n, p = 1250, 0.4
mu = n * p
sigma = math.sqrt(n * p * (1 - p))

# Apply continuity correction step by step
upper = (530 + 0.5 - mu) / sigma
lower = (485 - 0.5 - mu) / sigma

prob = norm.cdf(upper) - norm.cdf(lower)
```

Normal Approximation of Binomial | Edexcel
A Level Maths: Statistics Revision Notes 2018

Hypothesis Testing

- ✓ Hypothesis testing is a statistical method used to determine:
  - whether there is **enough evidence in a sample** to infer a condition about the population.
- ✓ It involves formulating two competing hypotheses and making decisions based on sample data.



- ✓ Key Components:
  - **Null Hypothesis (H_0):** Represents the default or no-effect assumption (e.g., "*There is no difference between sample and population means*").
  - **Alternative Hypothesis (H_a):** Represents the claim or effect to be tested (e.g., "*There is a difference between sample and population means*").
  - **Significance Level (α):** Probability of rejecting H_0 when it is true (commonly set at 0.05 or 5%).
  - **Test Statistic:** A value computed from sample data used to make a decision, such as Z-score or t-statistic.
  - **P-value:** Probability of observing the sample data if H_0 is true. A smaller p-value indicates stronger evidence against H_0 .

Hypothesis Testing Procedures (Cont..)

- ✓ Formulate Hypotheses:
  - H_0 : Population mean $\mu = \mu_0$ (no difference in sample and population).
  - H_a : Population mean $\mu \neq \mu_0$ (there is difference in sample and population).
- ✓ Select Test and Significance Level:
  - Choose a statistical test based on the data type (e.g., Z-test, t-test, ANOVA).
- ✓ Set the significance level (α)
  - Defines rejection region, selected by researcher at the beginning (0.01. 0.05, 0.10)
- ✓ Two types of test
  - One tail & Two tail test
- ✓ Calculate Test Statistic (e.g., t or Z).

$$Z = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

$$T = \frac{\bar{X} - \mu_0}{\frac{s}{\sqrt{n}}}$$

- ✓ Decision Rule:
  - Compare p-value or test statistic with critical values.
  - Reject H_0 if p-value < α, otherwise fail to reject H_0 .

Hypothesis Testing Procedures (Cont..)

Tests about mean of a normal population

There are two tests used to test the mean of a normal population

✓ **t-test** is performed when:

- Sample size $n < 30$ (small sample).
- Population standard deviation (σ) is unknown.
- The sample is drawn from a normal distribution.

$$T = \frac{\bar{X} - \mu_0}{\frac{s}{\sqrt{n}}}$$

- Where $\overline{X}$: Sample mean, μ_0: Hypothesized mean, s: Sample standard deviation, n: Sample size.

✓ **Z-test** is performed when:

- Sample size $n > 30$ (large sample).
- Population standard deviation (σ) is known.

$$Z = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

- The sample is drawn from a normal distribution or approximated by the Central Limit Theorem.
- Where $\overline{X}$: Sample mean, μ: Hypothesized mean, σ: Population standard deviation, n: Sample size.

Hypothesis Tests About Mean of a Normal Population

t-Test

- ✓ The t-test compares the sample mean ($\overline{X}$) to the population mean (μ_0), standardizing the difference using the sample standard deviation (s) and the sample size (n). The formula is:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

- ✓ Formulate hypotheses:
  - H_0: $\mu = \mu$, $\quad H_a$: $\mu \neq \mu_0$.
- ✓ Calculate the test statistic (t).
- ✓ Use t-tables to find the critical t-value for the given significance level (α) and degrees of freedom $df \text{ or } v = (n-1)$.
- ✓ Compare $|t|$ to the critical value:
  - Reject H_0 if $|t|$ exceeds the critical value or $p\_value < significance\ level\ (\alpha)$.
  - Otherwise, fail to reject H_0.

Hypothesis Testing (Cont..)

Types of Errors

- ✓ A Type I error occurs when we **reject a true null hypothesis**.
  - That is, a Type I error occurs when the jury convicts an innocent person.
- ✓ A Type II error occurs when we **don't reject a false null hypothesis**.
  - That occurs when a guilty defendant is acquitted.

Possible Hypothesis Test Outcomes

	Actual Situation	
Decision	H_0 True	H_0 False
Do Not Reject H_0	**No error** $(1 - \alpha)$	**Type II Error** (β)
Reject H_0	**Type I Error** (α)	**No Error** $(1 - \beta)$

Key:
Outcome
(Probability)

Hypothesis Tests About Mean of a Normal Population

t-Test Example

✓ The average cost of a hotel room in New York is said to be $168 per night. A random sample of 25 hotels resulted in $\overline{X} = \$172.50$ and

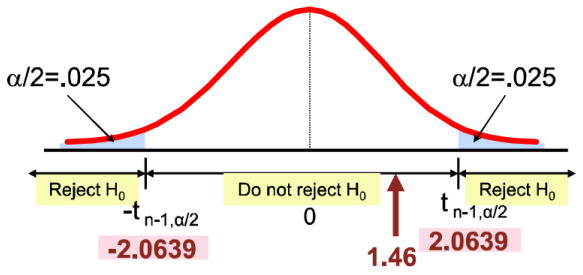
- $s = \$15.40$.
- Test at the $\alpha = 0.05$ level. (Assume the population distribution is normal)

H_0: μ = 168
H_1: μ ≠ 168

α = 0.05

- n = 25
- σ is unknown, so use a t statistic
- Critical Value:

  $t_{24} = \pm 2.0639$



$$t_{n-1} = \frac{\overline{X} - \mu}{\frac{S}{\sqrt{n}}} = \frac{172.50 - 168}{\frac{15.40}{\sqrt{25}}} = 1.46$$

Do not reject H_0: not sufficient evidence that true mean cost is different than $168

```python
from scipy.stats import t

# Given data
X_bar, mu0, s, n = 172.5, 168, 15.4, 25
alpha = 0.05

# Test statistic
t_stat = (X_bar - mu0) / (s / math.sqrt(n))

# Degrees of freedom
df = n - 1

# Two-tailed p-value
p_value = 2 * (1 - t.cdf(abs(t_stat), df))

# Conclusion
if p_value < alpha:
    conclusion = "Reject H0: Evidence sugges
else:
    conclusion = "Fail to reject H0: Not eno
```

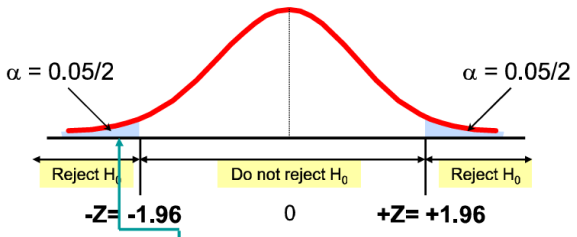
Hypothesis Tests About Mean of a Normal Population

Z-Test - Example

- ✓ Suppose the sample results are $n = 100$, $\overline{X} = 2.84$ ($\sigma = 0.8$ is assumed known)
- ✓ So, the test statistic is z-score.
- ✓ Formulate hypotheses:
  - H_0: $\mu = 3$, H_a: $\mu \neq 3$.

$$Z = \frac{\overline{X} - \mu}{\frac{s}{\sqrt{n}}} = \frac{2.84 - 3}{\frac{0.8}{\sqrt{100}}} = \frac{-.16}{.08} = -2.0$$

Reject H_0 if $Z < -1.96$ or $Z > 1.96$; otherwise do not reject H_0



Here, Z = -2.0 < -1.96, so the test statistic is in the rejection region

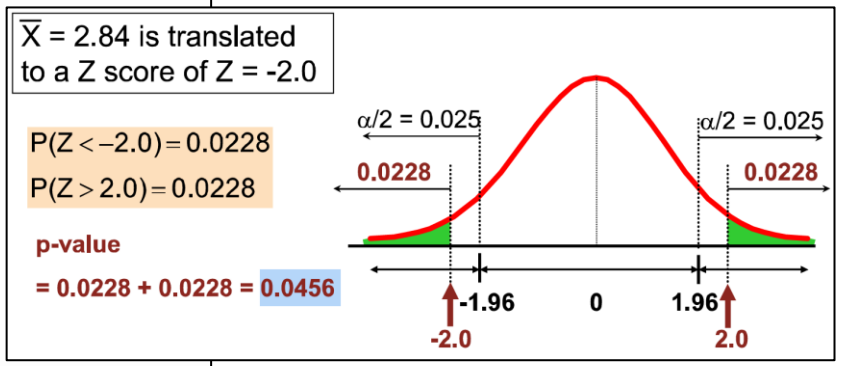
Since Z = -2.0 < -1.96, we reject the null hypothesis and conclude that there is sufficient evidence that the mean number of TVs in US homes is not equal to 3

```python
from scipy.stats import norm
import math

# Given data
X_bar, mu0, sigma, n = 2.84, 3.0, 0.8, 100
alpha = 0.05

# Z-test statistic
z_stat = (X_bar - mu0) / (sigma / math.sqrt(n))

# Two-tailed p-value
p_value = 2 * (1 - norm.cdf(abs(z_stat)))

# Print conclusion directly
if p_value < alpha:
    print("Reject H0\nTrue mean differs from 3")
else:
    print("Fail to reject H0\nMean ~ 3")
```

Z-Test - Example

- ✓ Problem: Test the claim that the true mean number of TV sets in US homes is different from 3 units. When we take random sample from 100 homes, we get 2.84 sets on average with standard deviation 0.8.
- ✓ ($s = 0.8$, $\alpha = 0.05$ and $n = 100$ for this test)
  - **p-value:** How likely is it to see a sample mean of 2.84 (or something further from the mean, in either direction) if the true mean is $\mu = 3.0$?

```
from scipy.stats import norm

# Given data
X_bar, mu0, s, n = 2.84, 3.0, 0.8, 100
alpha = 0.05

# Z-test statistic
z_stat = (X_bar - mu0) / (s / math.sqrt(n))

# Two-tailed p-value
p_value = 2 * (1 - norm.cdf(abs(z_stat)))

# Conclusion
conclusion = "Reject H0" if p_value < alpha else "Fail to reject H0"

z_stat, p_value, conclusion
```



Differences Between two Populations Means (Cont..)

Z-Test

- ✓ Two groups of students, Group A and Group B, took a data science test. Their scores are as follows:
  - Group A: $\overline{X}_1 = 75, s_1 = 10, n_1 = 50$
  - Group B: $\overline{X}_2 = 80, s_2 = 12, n_2 = 60$
- ✓ Assume:
  - The test significance level is $\alpha = 0.05$.
- ✓ Test the null hypothesis:
  - H_0: $\mu_1 = \mu_2$ (no difference in means)
  - H_a: $\mu_1 \neq \mu_2$
- ✓ For a two-tailed test at:
  - α=0.05, the critical Z-value is ±1.96
- ✓ $Z = -2.38$ falls outside the range $(-1.96, +1.96)$
- ✓ **Decision:** Reject H_0 :
  - There is sufficient evidence to conclude that the means of the two groups are different.

$$Z = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

$$Z = \frac{75 - 80}{\sqrt{\frac{10^2}{50} + \frac{12^2}{60}}}$$

$$Z = \frac{-5}{2.1} \approx -2.38$$

```
# Given data
X1, s1, n1 = 75, 10, 50
X2, s2, n2 = 80, 12, 60
alpha = 0.05

# Standard error
se = math.sqrt(s1**2/n1 + s2**2/n2)

# Z-test statistic
z_stat = (X1 - X2) / se

# Two-tailed p-value
p_value = 2 * (1 - norm.cdf(abs(z_stat)))

# Simple conclusion
if p_value < alpha:
    print("Reject H0\nMeans of Group A and B differ")
else:
    print("Fail to reject H0\nNo significant differen
```

Regression & Trend Analysis using Stats Models

- ✓ TO BE COMPLETED!

Regression - Simple Linear Regression

- ✓ Simple Linear Regression (SLR) models the linear relationship between a dependent variable (Y) and a single independent variable (X).
  - The regression equation is: $Y = \beta_0 + \beta_1 X + \epsilon$

    ϵ: Error term accounting for variability not explained by X
- ✓ It considers a single regressor or predictor **X** and a dependent response variable **Y.**
- ✓ Suppose that we have n pairs of observations, $(x_1, y_1), \ldots . (x_n, y_n)$
- ✓ The **method of least squares** is used to estimate the parameters, β_0 and β_1 by minimizing the sum of the squares of the vertical deviations in the Figure.

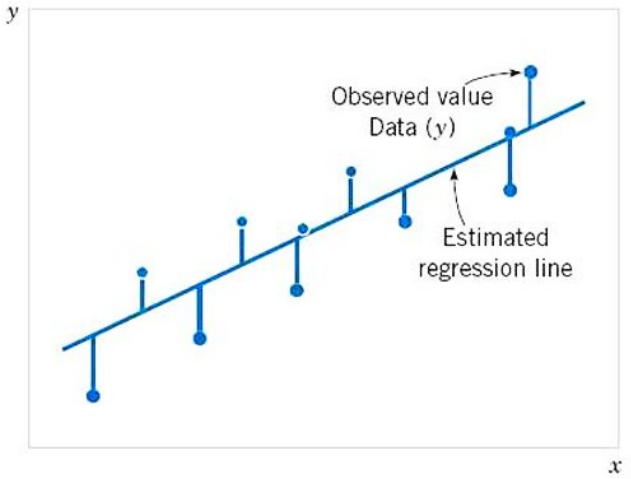


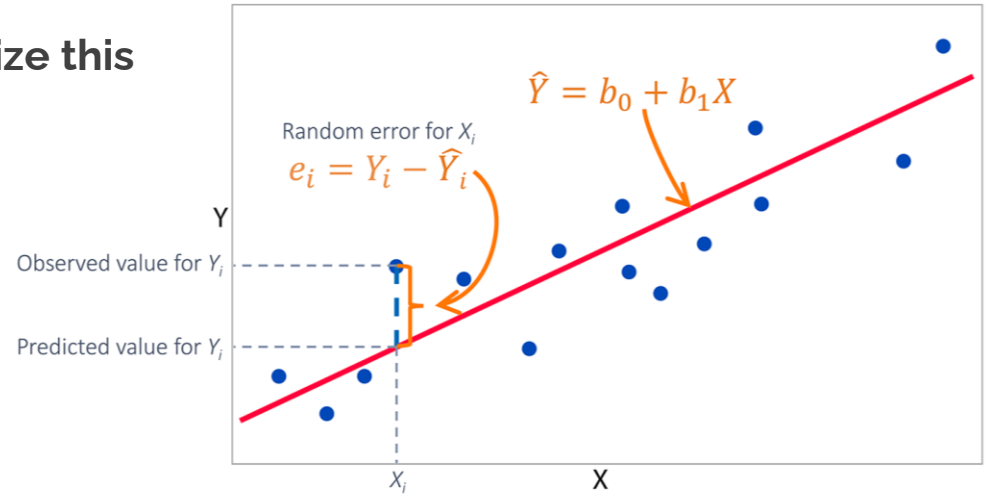
Figure 11-3 Deviations of the data from the estimated regression model.

Regression - Least Squares Estimation (LSE)

- ✓ LSE is a **geometric approach** that finds the best-fit line by minimizing the sum of squared residuals (errors).
  - Imagine drawing a straight line through a scatterplot. The goal is to **minimize the total vertical distances** between points and the line.
  - **Objective Function:** minimum sum of squared residual

$$SSE = \sum (y_i - \hat{y_i})^2 = \sum (y_i - \beta_0 - \beta_1 x_i)^2$$

  - This calculates how far each predicted value is from the actual data point.
  - **Goal:** Find β_0 and β_1 that **minimize this sum**.
  - **How?**
    - Take partial derivative and find minimum value of the function.

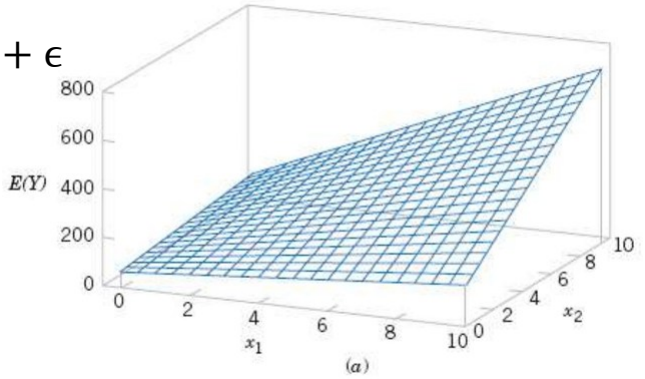


Regression - Multiple Linear Regression (MLR)

- ✓ Multiple Linear Regression (MLR) extends simple linear regression by modeling the relationship between a dependent variable or response Y and **multiple independent (predictor or regressor) variables**.
- ✓ In MLR, multiple predictors allows a more comprehensive understanding of the factors influencing the dependent variable.
- ✓ General Form of MLR Model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon$$

- Y = dependent variable (response variable)
- $X_1, X_2, \ldots, X_k$ = independent variables (predictors)
- β_0 = intercept
- $\beta_1, \beta_2, \ldots, \beta_k$ = regression coefficients
- ϵ = error term (random noise)



- This model describes a hyperplane in the k-dimensional space of the regressor variables $\{x_j\}$
- The parameter β_j represents the expected change in response Y per unit change in x_j when all the remaining regressors X_i $(i \neq j)$ are held constant.

Matrix Notation

- ✓ Matrix notation provides a compact and computationally efficient way to represent MLR.
- ✓ The model can be rewritten as:

$$y = X\beta + \epsilon$$

- where: Response Vector: $y = [y_1, y_2, \ldots, yn]^T \ (n \times 1)$
- In the matrix form:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1k} \\ 1 & x_{21} & x_{22} & \ldots & x_{2k} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nk} \end{bmatrix}$$

- $n \times (k+1)$ including an intercept column of ones.
- Coefficient Vector: $\beta = [\beta_0, \beta_1, \ldots, \beta_k]^T_{(k+1) \times 1}$
- Error Vector: $\epsilon = [\epsilon_1, \epsilon_2, \ldots, \epsilon_n]^T \mathrm{n} \times 1$

Matrix Notation (Cont..)

- ✓ The least squares estimation method minimizes the sum of squared residuals:

$$L = \sum_{i=1}^{n} (y_i - \hat{y_i})^2 = \sum_{i=1}^{n} (y_i - X_i\beta)^2$$

- ✓ The optimal regression coefficients are obtained by solving:

$$\frac{\partial L}{\partial \beta} = 0$$

  - Which results in the **Normal Equations**:

$$X^T X\beta = X^T y$$

- ✓ Solving for β:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

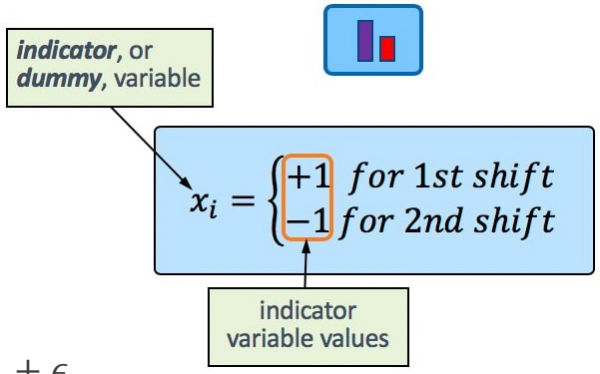
  - $X^T X$ is the covariance matrix of the predictors.
  - $X^T y$ represents the correlation between predictors and the response.

Handling Categorical Variables in Regression

- ✓ Categorical variables represent discrete groups or categories (e.g., gender, region, type).
  - **Problem:** Regression models require numerical input, so categorical variables must be converted.
  - **Solution: Use dummy (indicator) variables** to convert categorical data into binary numerical format.

- ✓ **What Are Dummy Variables?**
  - Binary variables (0 or 1) that indicate the presence of a categorical attribute.
  - Example: For "Color" with categories $\{\boldsymbol{Red, Blue, Green}\}$ or staffs with 1st and 2nd shift, the dummy variables could be:

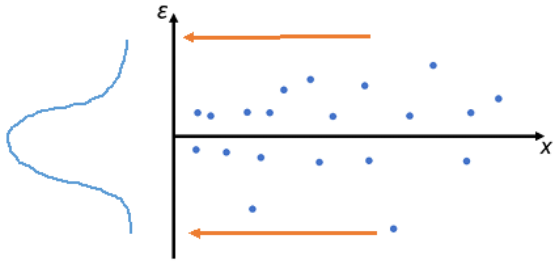
Color	D_Red	D_Blue	D_Green
Red	1	0	0
Blue	0	1	0
Green	0	0	1



$$Y = \beta_0 + \beta_1 D_{D\_Red} + \beta_2 D_{D\_Blue} + \beta_2 D_{D\_Green} + \epsilon$$

Regression - Model Performance Metrics

- ✓ Residual Analysis
  - Residuals → differences between observed values (Y_i) and predicted values ($\widehat{Y}_i$).
  - Randomly scattered residuals (with no pattern) suggest a good model fit.
  - Residual distribution follows normal distribution centered around 0.



- ✓ Mean Squared Error (MSE)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2$$

- ✓ Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{MSE}$$

- ✓ Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\, Y_i - \widehat{Y}_i \,\right|$$

Regression - Model Performance Metrics (Cont..)

✓ Coefficient of Determination (R^2)

- R^2, also known as the **coefficient of determination**, measures how well the independent variables explain the variance in the dependent variable in a regression model. It ranges from **0 to 1**, where:
  - $R^2 = 1$: The model perfectly explains all the variability in the dependent variable.
  - $R^2 = 0$: The model does not explain any of the variability.
  - A **higher** R^2 value indicates a better fit (*model with Pearson correlation coefficient (r) $= 0.7$ is almost double better than model with $r = 0.5$, **How?***)
- **Formula**

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

  - $SS_{\text{res}} = \sum(Y_i - \hat{Y}_i)^2 \rightarrow$ **Residual Sum of Squares (SSR):** Measures the total error between actual and predicted values.
  - $SS_{\text{tot}} = \sum(Y_i - \bar{Y})^2 \rightarrow$ **Total Sum of Squares (SST):** Measures the total variability in Y around its mean $\bar{Y}$
- R^2 represents the **proportion of variance in Y that is explained by the independent variables**.

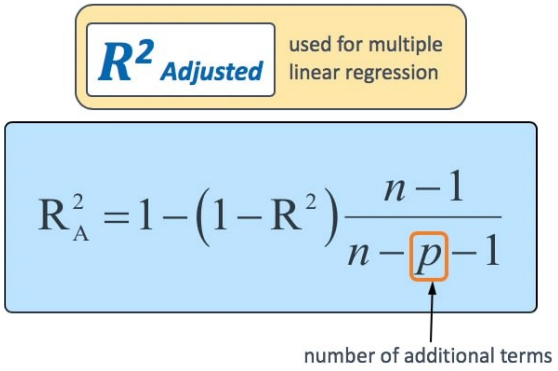
Regression - Model Performance Metrics (Cont..)

✓ Limitations of (R^2) :

- **It does not account for model complexity**: Adding more predictors will **always** increase R^2, even if the new variables do not improve model quality.

✓ **Adjusted R^2**

- It **accounts for useless predictors** by penalizing models that add unnecessary variables for unnecessary complexity.



- n = Number of data points p = Number of predictors
- If a new predictor does not improve the model enough to justify its inclusion, **Adjusted R^2 may decrease**, even though R^2 increases.

Regression & Trend Analysis using `statsmodels`

- ✓ **Statsmodels** is a Python library for **statistical modeling and hypothesis testing**, providing tools to estimate, interpret, and validate statistical models.
- ✓ **Key Features**
  - **Regression Models**
    - **OLS (Ordinary Least Squares)** – standard linear regression.
    - **GLM (Generalized Linear Models)** – logistic, Poisson, etc.
    - **Logit/Probit** – binary outcomes.
    - **MixedLM** – hierarchical or grouped data.
  - **Model Evaluation & Diagnostics**
    - **Residual analysis** – check assumptions of linearity, normality, and homoscedasticity.
    - **Influence measures** – detect outliers and leverage points.
    - **Multicollinearity checks** – Variance Inflation Factor (VIF).
  - **Statistical Tests**
    - **t-tests** – significance of individual coefficients.
    - **F-tests** – joint significance of multiple coefficients.
    - **Durbin-Watson** – autocorrelation in residuals.

Regression & Trend Analysis using `statsmodels`

✓ Key Features of statsmodels (Cont..):

- **Time Series Analysis**
  - **ARMA/ARIMA models** – for modeling autocorrelated data.
  - **Exponential smoothing** – simple & Holt-Winters.
  - **Seasonal decomposition** – trend, seasonal, residual analysis.
- **Data Handling & Formula Interface**
  - Uses **patsy formulas** (Y ~ X1 + X2) to specify models.
  - Supports **DataFrames**, making it compatible with pandas.
- **Visualization & Interpretation**
  - Residual plots, influence plots, autocorrelation plots.
  - Summary tables with coefficients, standard errors, R^2, p-values.

✓ Why Use Statsmodels?

- Provides **detailed statistical output**, unlike scikit-learn which focuses on predictions.
- Useful for **inferential statistics**, hypothesis testing, and reporting results in research papers.
- Supports **both classical and modern statistical methods**.

Regression & Trend Analysis using `statsmodels`

✓ Statsmodels

- Fit statistical models: OLS, GLM, Logit
- Time series analysis: ARIMA, Exponential Smoothing
- Hypothesis testing: t-tests, F-tests, confidence intervals
- Rich diagnostics: residuals, influence, goodness-of-fit

✓ Patsy (dmatrices)

- Converts R-like formulas (Y ~ X1 + X2) into design matrices
- Automatically adds intercept terms
- Encodes categorical variables automatically (dummy / one-hot encoding)
- Supports interactions and polynomial terms
- Works seamlessly with pandas DataFrames

1️⃣ Installation

Bash

```
pip install statsmodels
```

- Also install `pandas` and `patsy` if not already:

Bash

```
pip install pandas patsy
```

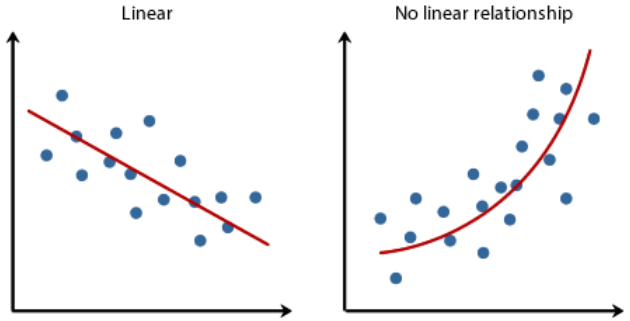
2️⃣ Imports

Python

```
import statsmodels.api as sm      # Core statsmodels functions
from patsy import dmatrices       # Formula interface for model design
import pandas as pd               # Data handling
```

Regression - Non-linear Regression

- ✓ Linear regression assumes:
  - Relationship between the dependent variable and independent variables is linear.
- ✓ Non-linear regression is an extension of linear for capturing non-linear relationships among dependent and independent variables.
  - **Need for Extension:**
    - Many real-world relationships between variables are not strictly linear.
  - **How to Extend:**
    - Use polynomial or other non-linear transformations (e.g., logarithms, exponentials)
- ✓ Examples
  - Exponential Regression: $y = \alpha e^{\beta x}$
  - Logarithmic Regression: $y = \alpha + \beta \ln(x)$
  - Power Regression: $y = \alpha x^{\beta}$
  - Generalized Additive Models (GAMs): $y = f1(x1) + f2(x2) + \cdots \ldots . fn(xn) + \epsilon$



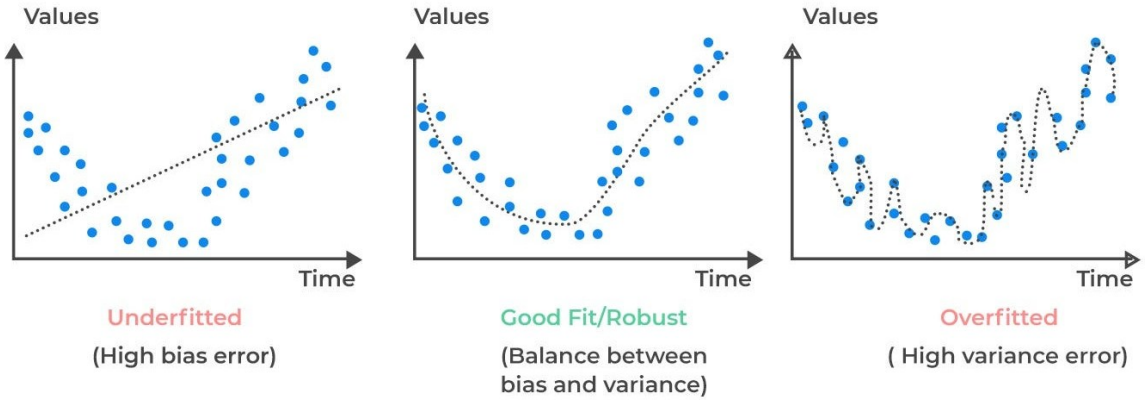
Regression - Non-linear Regression (Cont..)

Polynomial Regression

- ✓ Polynomial regression models extend the linear model by including additional predictors that are powers of the original predictor variable.

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \ldots + \beta_k X^k + \epsilon$$

- ✓ **Key Point:**
  - While higher degree polynomials can capture more complex patterns, they may also lead to overfitting.



Avoiding Overfitting Techniques (Cont..)

2. **Use AIC or BIC Information** to choose optimum degree:

i) AIC (Akaike Information Criterion): AIC is used for model selection by penalizing complexity while rewarding goodness of fit.

$$\text{AIC} = 2\text{k} + \text{n}\, ln\left(\frac{\text{RSS}}{\text{n}}\right)$$

- n: Number of data points.
- k: Number of model parameters (degree + 1 for polynomial regression).
- RSS (Residual Sum of Squares): Measures the model's error.

- **Penalty:** Term $2k$ increases with more parameters, discouraging complex models.
  - First term ($2k$) → Penalizes complexity.
  - Second term → Measures the error. If RSS is low, the model fits well.
- How to Use AIC/BIC to Select Polynomial Degree?
  - Fit polynomial regression models for different degrees ($d = 1,2,3, \ldots$)
  - Select the model with the lowest AIC/BIC (*covered later in this slide*) value.

Avoiding Overfitting Techniques (Cont..)

✓ 2. Use AIC or BIC Information to choose optimum degree (Cont..):

ii) BIC (Bayesian Information Criterion): BIC is similar to AIC but includes a stronger penalty for complexity:

$$\text{AIC} = \text{k}\, ln(n) + \text{n}\, ln\left(\frac{\text{RSS}}{\text{n}}\right)$$

- The key difference is that the penalty term is $\boldsymbol{k\, ln(n)}$ instead of **2k**.
- This penalty **grows faster** than AIC, making BIC **stricter** for selecting simpler models when **n** is large.

- **Intuition:**
  - If we have **a lot of data (n is large)** → BIC prefers simpler models because complexity is penalized more.
  - If we have **less data** → BIC and AIC behave more similarly.
- A lower AIC/BIC score means a better balance between fit and complexity.

Exploratory Data Analysis

- ✓ Exploratory Data Analysis (EDA) is a process of examining datasets to summarize their main characteristics, identify patterns, detect anomalies, and validate assumptions using visual and statistical techniques.
- ✓ It forms a critical step before applying advanced analytical methods and modeling, ensuring data readiness and reliability.
- ✓ Key Features of EDA

1. **Summarization of Data:** Provides an overview of central tendency, dispersion, and distribution.
2. **Pattern Identification:** Unveils trends and relationships within the data.
3. **Anomaly Detection:** Highlights outliers and inconsistencies.
4. **Hypothesis Formation:** Facilitates initial understanding to frame research questions.
5. **Data Cleaning Insight:** Identifies missing values, duplicates, and erroneous entries for rectification.

Exploratory Data Analysis (Cont..)

✓ Importance of Exploratory Data Analysis (EDA)

- **Ensures data quality** by identifying and addressing missing, duplicate, or erroneous data.
- **Reveals key insights and patterns** that inform decision-making and highlight data trends.
- **Prepares data** for advanced analytics, including predictive and prescriptive models, by transforming and structuring it.
- **Validates assumptions** about data distributions and relationships to ensure reliable model development.
- **Detects outliers and anomalies** that could skew results or indicate underlying data issues.

✓ **Steps in EDA:** It is a systematic process to explore and understand data.

1. **Data Inspection:** Understand data structure, size, and types of variables.
2. **Data Cleaning:** Handle missing, inconsistent, or erroneous data entries.
3. **Data Transformation:** Normalize, scale, or encode data for further analysis.
4. **Visualization:** Create plots to identify patterns, trends, and correlations.
5. **Statistical Summaries:** Calculate measures like mean, median, variance, and correlation coefficients.

Descriptive Vs Inferential Statistics

Aspect	Descriptive Statistics	Inferential Statistics
Purpose	Summarizes and organizes data to describe its main features.	Makes inferences, predictions, or generalizations about a population using sample data.
Scope	Focuses on analyzing and presenting the characteristics of the specific dataset under study.	Extends findings from the sample to represent the larger population.
Objective	Provides insights into the dataset without generalizing beyond it.	Generalizes findings, tests hypotheses, and makes predictions about a population.
Population Representation	Represents only the features within the dataset being analyzed.	Represents features and characteristics of the entire population.
Data Analysis	Summarizes and visualizes data using measures like mean, median, variance, and charts.	Analyzes sample data to test hypotheses and make population-level predictions.
Statistical Techniques	Includes measures like central tendency (mean, median) and dispersion (variance, standard deviation).	Involves techniques like hypothesis testing, regression analysis, and confidence intervals.
Examples	Calculating the mean score of students in a specific class or visualizing data with bar charts.	Estimating the average score of all students in a school based on a sample of students.
Goal	Provides a clear understanding of the dataset's characteristics.	Enables decision-making and conclusions for an entire population based on sample data.

Exploratory Data Analysis (Cont..)

Descriptive Statistics for EDA

✓ Basic Data Summaries and Visualizations

- Basic summaries and visualizations are the cornerstone of EDA, offering quick insights.
- Summaries:
  - Descriptive Statistics: Mean, median, standard deviation, and count.
  - Aggregates: Sums, averages, counts, or frequencies.
- Visualizations:
  - Box Plots: Summarize data distribution and highlight outliers.
  - Scatter Plots: Visualize relationships between two continuous variables.
  - Bar Charts: Summarize and compare categorical data.
  - And other plots…
- **Applications:** Visualizing product sales trends, comparing performance metrics.

Exploratory Data Analysis (Cont..)

Descriptive Statistics for EDA (Cont..)

✓ Measures of Central Tendency

- **Mean:** The arithmetic average, representing overall data behavior but sensitive to outliers.
- **Median:** The middle value in sorted data, useful for skewed distributions.
- **Mode:** The most frequently occurring value, often applied to categorical data.
- **Applications:**
  - Identifying typical values in sales, temperatures, or performance metrics.

✓ Measures of Dispersion

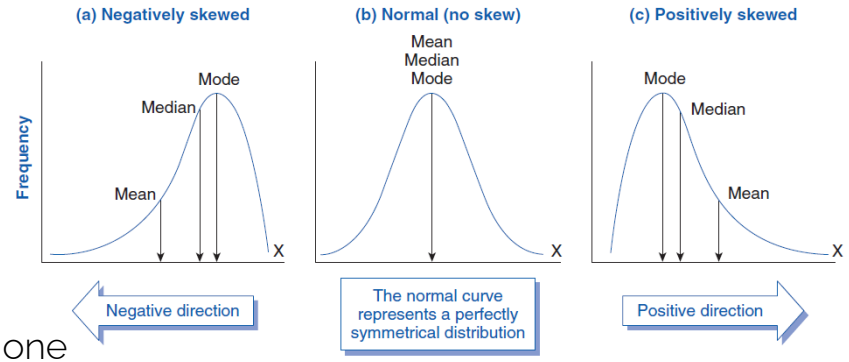
- **Range:** Difference between the highest and lowest values.
- **Variance:** Measure of how far data points are from the mean.
- **Standard Deviation:** Square root of variance, representing data spread.
- **Interquartile Range (IQR):** Middle 50% of data, used to detect outliers.
- **Applications:**
  - Understanding variability in stock prices, exam scores, or production rates.

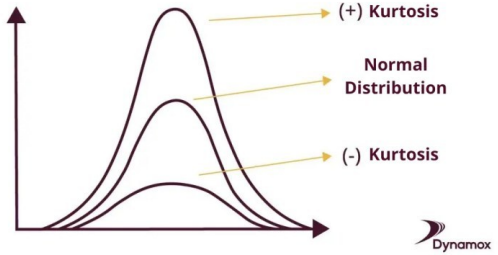
Exploratory Data Analysis (Cont..)

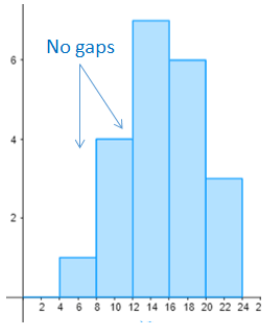
Descriptive Statistics for EDA (Cont..)

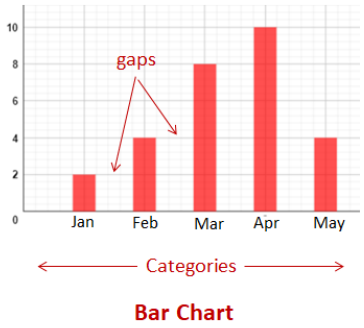
✓ Data distribution and histograms reveal the shape, spread, and symmetry of datasets.

- Data Distribution:
  - Symmetric (normal) distribution:
  - Mean ≈ median ≈ mode.
  - Skewed distribution: Presence of outliers on one side (positive or negative skew).
  - Metrics: Skewness (asymmetry), kurtosis (tailedness).
- Histograms Vs Bar Charts:
  - Histogram: Display frequency distributions of bins.
  - Bar charts: Frequency of categories.
  - Applications: Examining income distributions, product sales ranges, or customer age groups.









Descriptive Statistics for EDA (Cont..)

✓ Identifying Outliers and Missing Values

- Outliers and missing values are key issues in datasets, potentially skewing results if not addressed.
- **Outliers:** Extreme values that deviate significantly from the norm.
  - Detection Techniques:
    - IQR Method: $Q3 + 1.5 \times IQR,\ Q1 - 1.5 \times IQR$.
    - Z-scores: Flag values beyond 3 standard deviations from the mean.
  - Applications: Identifying fraudulent transactions, equipment malfunctions.
- **Missing Values:** Null or incomplete entries in a dataset.
  - Handling Techniques:
    - Imputation: Replace with mean, median, or use advanced methods like k-Nearest Neighbors.
    - Removal: Drop rows/columns with excessive missing data.
  - Applications: Cleaning customer records, survey datasets.

Descriptive and Inferential Statistics

Descriptive statistics

- ✓ It involves summarizing and describing the main features of a dataset. These methods organize, visualize, and interpret data meaningfully without making inferences beyond the dataset.
  - Describes what is happening in the dataset.
  - No prediction or generalization is made.
- ✓ Statistical Approaches
  - **Central Tendency:** Mean, Median, Mode
  - **Dispersion:** Range, Variance, Standard Deviation
  - **Frequency Distribution:** Histograms, Frequency Tables
  - **Visualization:** Bar Charts, Pie Charts, Box Plots

You should be able to calculate these basic statistical calculations on your own.

- ✓ Example:
  - A delivery company tracks the delivery times (in hours) for 100 packages to assess efficiency. A random sample of 10 delivery times is: 3.5, 4.0, 3.7, 3.8, 4.2, 3.6, 3.9, 4.1, 3.4, 3.7
  - Task: Calculate the **mean delivery time** to summarize the central tendency.

Descriptive Statistics

Mean

If the n observations in a random sample are denoted by $x_1, x_2, ..., x_n$, the sample mean is

$$\bar{x} = \frac{x_1 + x_2 + ... + x_n}{n} = \frac{\sum_{i=1}^{n} x_i}{n}$$

Variance Defined

If the n observations in a sample are denoted by $x_1, x_2, ..., x_n$, the sample variance is

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

Standard Deviation Defined

- The standard deviation is the square root of the variance.
- σ is the population standard deviation symbol.
- s is the sample standard deviation symbol.

Interquartile Range

The interquartile range (IQR) is defined as:

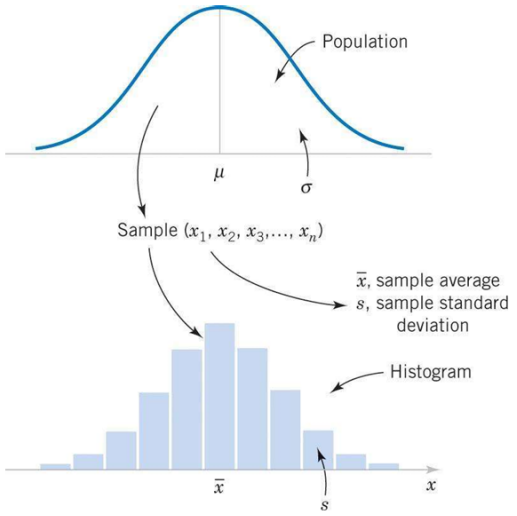
IQR = $q_1 - q_3$.

Quartiles

The three quartiles partition the data into four equally sized counts or segments.

- 25% of the data is less than q_1.
- 50% of the data is less than q_2, the median.
- 75% of the data is less than q_3.

Populations & Samples



Descriptive Statistics (Cont..)

Frequency Distributions

A frequency distribution is a compact summary of data, expressed as a table, graph, or function.

The data is gathered into bins or cells, defined by class intervals.

Table 6-4 Frequency Distribution of Table 6-2 Data

Class	Frequency	Relative Frequency	Cumulative Relative Frequency
$70 \le x < 90$	2	0.0250	0.0250
$90 \le x < 110$	3	0.0375	0.0625
$110 \le x < 130$	6	0.0750	0.1375
$130 \le x < 150$	14	0.1750	0.3125
$150 \le x < 170$	22	0.2750	0.5875
$170 \le x < 190$	17	0.2125	0.8000
$190 \le x < 210$	10	0.1250	0.9250
$210 \le x < 230$	4	0.0500	0.9750
$230 \le x < 250$	2	0.0250	1.0000
	80	1.0000	

Stem-and-Leaf Diagrams

Dot diagrams (dotplots) are useful for small data sets. Stem & leaf diagrams are better for large sets.

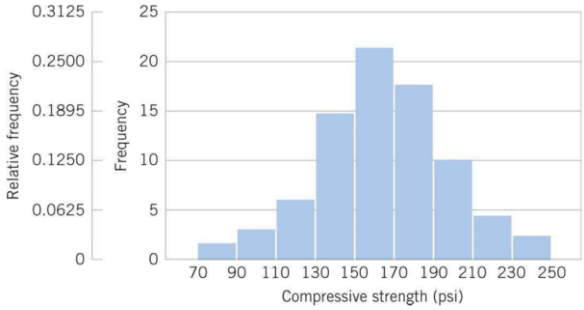
Table 6-2 Compressive Strength (psi) of Aluminum-Lithium Specimens

105	221	183	186	121	181	180	143
97	154	153	174	120	168	167	141
245	228	174	199	181	158	176	110
163	131	154	115	160	208	158	133
207	180	190	193	194	133	156	123
134	178	76	167	184	135	229	146
218	157	101	171	165	172	158	169
199	151	142	163	145	171	148	158
160	175	149	87	160	237	150	135
196	201	200	176	150	170	118	149

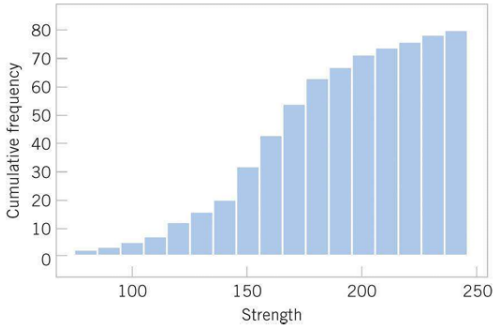
Stem	Leaf	Frequency
7	6	1
8	7	1
9	7	1
10	5 1	2
11	5 8 0	3
12	1 0 3	3
13	4 1 3 5 3 5	6
14	2 9 5 8 3 1 6 9	8
15	4 7 1 3 4 0 8 8 6 8 0 8	12
16	3 0 7 3 0 5 0 8 7 9	10
17	8 5 4 4 1 6 2 1 0 6	10
18	0 3 6 1 4 1 0	7
19	9 6 0 9 3 4	6
20	7 1 0 8	4
21	8	1
22	1 8 9	3
23	7	1
24	5	1

Histograms

A histogram is a visual display of a frequency distribution, similar to a bar chart or a stem-and-leaf diagram.

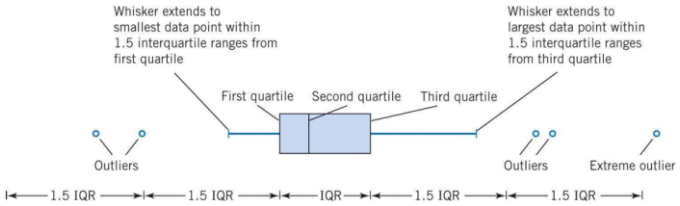


Cumulative Frequency Plot



Box Plot or Box-and-Whisker Chart

- A box plot is a graphical display showing center, spread, shape, and outliers (SOCS).
- It displays the 5-number summary: *min*, q_1, *median*, q_3, and *max*.



Time Sequence Plots

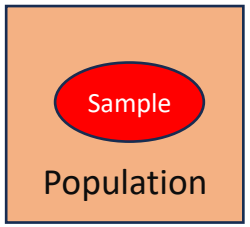
A time series plot shows the data value, or statistic, on the vertical axis with time on the horizontal axis.

A time series plot reveals trends, cycles or other time-oriented behavior that could not be otherwise seen in the data.

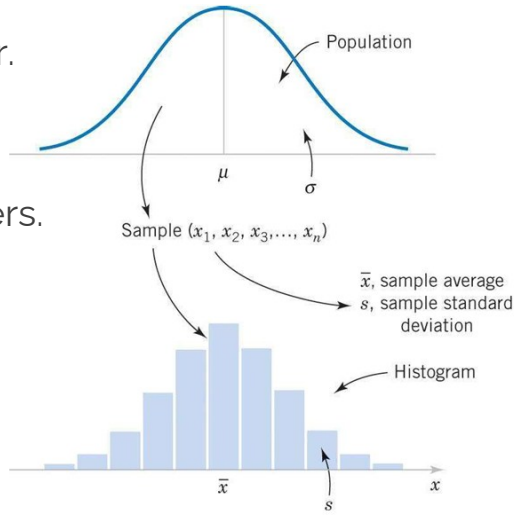
Descriptive and Inferential Statistics (Cont..)

Inferential statistics

- ✓ It extend findings from a **sample** to the **population** by making predictions, generalizations, or testing hypotheses.
  - Goes beyond data description to predict or generalize.
  - Relies on probability theory.

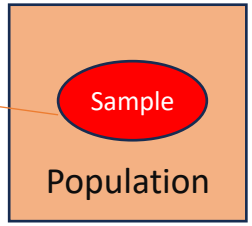


- ✓ Inferential Statistical Approaches:
  - **Hypothesis Testing:** Validating claims about a population parameter.
  - **Regression Analysis:** Examining relationships between variables.
  - **Confidence Intervals:** Estimating the range for population parameters.
  - **Prediction**: Forecasting outcomes based on sample data.

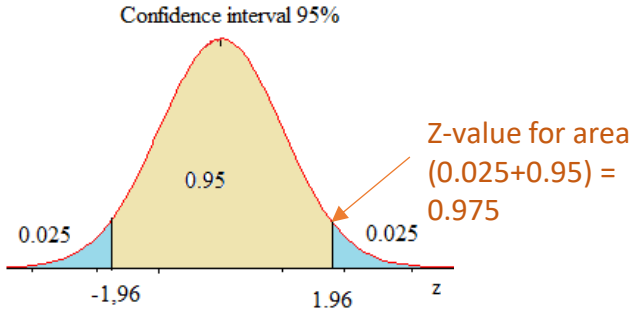


Inferential statistics - Example

- ✓ A company surveys **200 employees** to find the average time spent coding daily. The sample mean ($\bar{x}$) is **6 hours**, with a standard deviation (s) of **1 hour**.
  - Estimate the **population mean coding time** for all employees with **95% confidence**.



- ✓ Given Data:
  - Sample size (***n***) = 200
  - Sample mean ($\bar{x}$) = 6
  - Sample standard deviation (***s***) = 1
  - Z-value for 95% CI = **1.96** (from the **Z-table**)



$$CI = \bar{x} \pm Z \cdot \frac{s}{\sqrt{n}}$$

$$CI = 6 \pm 1.96 \cdot \frac{1}{\sqrt{200}}$$

$$CI = 6 \pm 1.96 \cdot 0.0707 = 6 \pm 0.138$$

$$CI = [5.862, 6.138]$$

Standard error (SE): Measures **variability of the sample mean** if you repeated sampling many times.

- ✓ **Interpretation**: With 95% confidence, the average coding time for all employees is between **5.86 and 6.14 hours per day**.

* Detailed theory about Confidence interval (CI) will be covered later.

Automation of EDA Workflows using Python

- ✓ TO BE COMPLETED!