

DATA SCIENCE

Complete Beginner Course

हिंदी + English | Detailed Notes | Examples | Practice Exercises

Step-by-step learning: Data basics ☐ Python ☐ NumPy ☐ Pandas ☐ Statistics ☐ Charts ☐ Machine Learning ☐ Projects.

Self-study workbook • No prior Data Science experience required

Version 1.0 • October 2026

How to Use This Course | कोर्स कैसे पढ़ें

यह course बिल्कुल beginner के लिए है। हर chapter में concept, आसान explanation, code examples और practice tasks हैं। रोज़ 30–60 मिनट पढ़ें, code स्वयं type करें और outputs को समझें।

- Practice environment: Google Colab, Jupyter Notebook या local Python.
- पहले सवाल तय करें, फिर data देखें; सीधे complex model से शुरू न करें.
- हर project में source, cleaning steps, charts, conclusions और limitations लिखें.
- Data privacy का ध्यान रखें; वास्तविक students/customers की पहचान वाली जानकारी publish न करें.

Learning outcomes

- Data Science workflow समझना.
- Python fundamentals और core data structures सीखना.
- NumPy/Pandas से data process और clean करना.
- Statistics, probability और visualization के basics समझना.
- Machine Learning models को train/evaluate करने का परचिय.
- एक छोटा end-to-end portfolio project बनाना.

Table of Contents | वषिय-सूची

Chapter	Topic
1	Introduction to Data Science
2	Data Types, Lifecycle & Ethics
3	Python Setup and First Program
4	Python Fundamentals
5	Lists, Tuples, Sets and Dictionaries
6	Conditions, Loops and Functions
7	NumPy Basics
8	Pandas: Series and DataFrames
9	Data Cleaning and Preparation
10	Exploratory Data Analysis
11	Data Visualization
12	Statistics for Data Science
13	Probability and Distributions
14	Machine Learning Fundamentals
15	Model Evaluation and Pitfalls
16	SQL for Data Analysis
17	End-to-End Mini Projects
18	Practice Questions and Answer Hints
19	Glossary and 8-Week Study Plan

1. Introduction to Data Science

Data Science is the process of collecting, preparing, analyzing and interpreting data to answer questions, discover patterns or support decisions. आसान भाषा में: data से useful information निकालना।

Where is it used?

Area	Example
Education	Understand attendance and exam–result patterns
Healthcare	Analyze trends in medical research data
Retail	Estimate demand and compare product sales
Transport	Study travel time and traffic patterns
Banking	Detect unusual transactions and assess risk
Media	Recommend content based on user activity

Related fields

Field	Main focus
Data Analysis	Describe what happened and investigate why
Data Science	Analysis, experimentation and predictive modelling
Machine Learning	Algorithms that learn patterns from examples
Data Engineering	Reliable pipelines and data storage
Business Intelligence	Dashboards, reports and business metrics

Typical workflow

Question ☐ Collect ☐ Clean ☐ Explore ☐ Model if needed ☐ Evaluate ☐ Communicate ☐ Monitor

- Start with a clear question and success criteria.
- Check accuracy, relevance and permissions of the data.
- Use a simple baseline before a complex model.
- Association does not automatically prove causation.

Practice

- Write three questions a shop could answer using sales records.
- List the columns needed for each question.
- Explain the difference between Data Science and Data Analysis.

2. Data Types, Lifecycle & Ethics

Data type	Meaning	Example
Continuous numeric	Measured quantity	Height, temperature
Discrete numeric	Countable quantity	Number of orders
Nominal category	Labels without order	City, colour
Ordinal category	Categories with order	Low, medium, high
Date/time	Time-based values	Order date
Text	Free-form words	Customer review

Structured vs unstructured

Structured data fits a predictable table. Unstructured data includes free-form text, images, audio and video. JSON is an example of semi-structured data.

Data lifecycle

- Define question and success criteria.
- Collect data lawfully and record its source.
- Store securely and document column meanings.
- Clean and validate data.
- Analyze, visualize and model where useful.
- Communicate results, retain reproducible steps and monitor changes.

Ethics and privacy

Collect only data needed for a legitimate purpose. Limit access, protect files and follow applicable law and policy. Removing names alone may not prevent re-identification.

- Bias: some groups may be under-represented.
- Consent: people should understand appropriate uses of their data.
- Security: restrict access to sensitive records.
- Transparency: state assumptions, limitations and uncertainty.
- Fairness: examine whether errors differ across groups.

Practice

- Classify age, city, rating 1–5 and review text.
- Why can a large dataset still be biased?
- Name two steps that protect private data.

3. Python Setup and First Program

Python is popular in Data Science because it is readable and has libraries such as NumPy, Pandas, Matplotlib and scikit-learn. Google Colab lets you run notebooks in a browser.

First program

```
print("Hello, Data Science!")
print(2 + 3)
```

Output: Hello, Data Science! and 5.

Variables and data types

```
student_name = "Asha" # str: text
marks = 87             # int: whole number
attendance = 0.92       # float: decimal
is_present = True       # bool: True/False
print(type(marks))
```

A variable is a name referring to a value. Python is case-sensitive; marks and Marks are different.

Input and conversion

```
name = input("Enter your name: ")
age = int(input("Enter your age: "))
print(f"Hello {name}; age {age}")
```

`input()` returns text. Convert with `int()` or `float()` when calculations require numeric values.

Practice

- Print your name, city and learning goal.
- Store five subject marks and calculate their total.
- Ask for rectangle length and width and calculate area.

4. Python Fundamentals

Operators

Operator	Meaning	Example
+ - * /	Arithmetic	10 + 3 = 13; 10 / 4 = 2.5
// % **	Floor division, remainder, power	10 // 3 = 3; 10 % 3 = 1; 2**3 = 8
= = != > <	Comparison	5 == 5 is True
and / or / not	Logical	age >= 18 and has_id

Strings

```
course = "Data Science"  
print(course.lower())  
print(course.upper())  
print(len(course))  
print(course[0])
```

String indexing begins at 0. Strings are immutable: you cannot replace one character in place.

Errors

- `SyntaxError`: invalid code structure.
- `NameError`: undefined or misspelled variable.
- `TypeError`: incompatible operation or type.
- `ValueError`: conversion cannot use the supplied value.
- `IndexError`: index is outside the sequence.

Practice

- Convert 153 minutes into hours and remaining minutes.
- Calculate the average of 72, 84, 91 and 67.
- Create an f-string that reports your name and favourite subject.

5. Lists, Tuples, Sets and Dictionaries

Lists

```
marks = [72, 88, 91, 67]
print(marks[0])
marks.append(95)
print(sum(marks) / len(marks))
```

Lists are ordered, mutable and may contain duplicates.

Tuples, sets and dictionaries

```
coordinates = (25.3, 82.9) # tuple
unique_numbers = {1, 2, 2, 3} # set
student = {"name": "Asha", "marks": 88}
print(student["name"])
print(student.get("city", "Not provided"))
```

Structure	Use	Property
List	Collection that changes	Ordered, duplicates allowed
Tuple	Fixed group of values	Ordered sequence
Set	Unique values	No duplicates
Dictionary	Look up values by keys	Key–value pairs

Practice

- Find min, max and average of seven temperatures.
- Make a set from [1, 2, 2, 3, 3, 3].
- Create a book dictionary with title, author and year.

6. Conditions, Loops and Functions

Conditions

```
marks = 76
if marks >= 90:
    grade = "A"
elif marks >= 75:
    grade = "B"
else:
    grade = "C"
print(grade)
```

Indentation defines Python code blocks. Keep it consistent.

Loops

```
for number in range(1, 6):
    print(number)

count = 3
while count > 0:
    print(count)
    count -= 1
```

`range(1, 6)` generates 1 through 5. A while loop must eventually reach a false condition to avoid an infinite loop.

Functions

```
def average(values):
    if len(values) == 0:
        return None
    return sum(values) / len(values)

print(average([70, 80, 90]))
```

Practice

- Write a function that returns the larger of two numbers.
- Print even numbers from 2 to 30.
- Calculate the sum from 1 to 100.
- Count marks that are at least 40.

7. NumPy Basics

NumPy is a library for efficient numerical computation. Its main structure is the `ndarray`, which supports vectorized operations.

```
import numpy as np
marks = np.array([70, 80, 90, 100])
print(marks.mean())
print(marks.max())
print(marks + 5)
```

The expression `marks + 5` adds five to each element.

Shape and dimensions

```
a = np.array([1, 2, 3])
b = np.array([[1, 2, 3], [4, 5, 6]])
print(a.shape) # (3,)
print(b.shape) # (2, 3)
print(b.ndim)  # 2
```

Statistics and filtering

```
x = np.array([12, 25, 8, 31, 19])
print(np.mean(x), np.median(x), np.std(x))
print(x[x >= 20])
```

Standard deviation measures spread. NumPy defaults to population standard deviation (`ddof=0`); a sample estimate commonly uses `ddof=1`.

Practice

- Create an array of ten numbers and find mean and median.
- Filter values above 50 from `[22, 67, 41, 88, 53]`.
- Create a 2×3 array and inspect its shape.

8. Pandas: Series and DataFrames

A Pandas Series is one labelled column; a DataFrame is a labelled table.

```
import pandas as pd
data = {"student": ["Asha", "Ravi", "Mina", "Kabir"],
        "marks": [88, 72, 95, 61],
        "attendance": [0.96, 0.82, 0.98, 0.75]}
df = pd.DataFrame(data)
print(df.head())
print(df["marks"].mean())
```

Inspect, filter, sort

```
print(df.shape)
print(df.info())
print(df.describe())
high_scores = df[df["marks"] >= 80]
print(high_scores)
print(df.sort_values("marks", ascending=False))
```

Group and save

```
df["passed"] = df["marks"] >= 40
print(df.groupby("passed")["marks"].mean())
df.to_csv("students.csv", index=False)
# Load later:
df2 = pd.read_csv("students.csv")
```

`groupby()` aggregates rows by category. After loading a CSV, inspect columns, data types, date formats and missing values.

Practice

- Build a table of five products with price and quantity.
- Calculate value = price × quantity.
- Filter quantities below 10, sort by price and save as CSV.

9. Data Cleaning and Preparation

Real-world data can contain missing values, duplicate rows, inconsistent spelling, impossible values, incorrect types and outliers.

Inspect and clean

```
print(df.isna().sum())
print(df.duplicated().sum())
df = df.drop_duplicates()
```

Missing values

```
# Remove rows only when justified
df_small = df.dropna(subset=["marks"])

# Or fill numeric missing values with median
median_marks = df["marks"].median()
df["marks"] = df["marks"].fillna(median_marks)
```

There is no universal missing-value solution. Understand why values are absent, document the method and test whether conclusions change.

Types, text and outliers

```
df["date"] = pd.to_datetime(df["date"], errors="coerce")
df["city"] = df["city"].astype("string").str.strip().str.title()
```

An outlier may be a genuine rare event or an error; investigate before deleting it. Validate units and ranges, and record each transformation.

Data leakage

Do not use information unavailable at prediction time. For example, using final exam marks as a feature when predicting pass/fail before the exam leaks the answer.

Practice

- Find missing values in each column.
- Decide how to handle missing ages and explain why.
- Give an example where deleting an outlier would be harmful.

10. Exploratory Data Analysis (EDA)

EDA is the early investigation of data using summaries, tables and charts. It helps reveal patterns, data-quality problems and new questions.

Questions to ask

- How many rows and columns exist? What does each row represent?
- Which columns are numeric, categorical, dates or text?
- Are there missing values, duplicates or implausible values?
- What are typical values and how widely do they vary?
- Do groups differ, or do numeric variables appear related?

Useful commands

```
df.head()  
df.shape  
df.info()  
df.describe(include="all")  
df.isna().sum()  
df["category"].value_counts()  
df.groupby("category")["sales"].mean()
```

Correlation is not causation. Two variables may move together because of a third variable, reverse direction or coincidence.

Practice

- Write five questions about a public dataset before charting.
- Find the three most common values in a category column.
- Compare average sales by category and state one limitation.

11. Data Visualization

Chart	Good for	Caution
Bar chart	Compare categories	Use clear category labels
Line chart	Trend over time	Keep time order correct
Histogram	Numeric distribution	Bin width affects shape
Scatter plot	Two numeric variables	Association is not causation
Box plot	Median, spread, potential outliers	Outlier marks are not automatically errors

Matplotlib example

```
import matplotlib.pyplot as plt
months = ["Jan", "Feb", "Mar", "Apr"]
sales = [120, 150, 135, 190]
plt.plot(months, sales, marker="o")
plt.title("Monthly Sales")
plt.xlabel("Month")
plt.ylabel("Sales")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()
```

Histogram

```
plt.hist([12,14,15,15,18,19,20,22,25], bins=5)
plt.title("Distribution of Values")
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.show()
```

- Use a descriptive title and labelled axes with units.
- Avoid misleading scales and unnecessary 3D effects.
- Use accessible, consistent colours.
- Write one sentence describing the chart's main finding.

Practice

- Make a bar chart of subject marks.
- Plot monthly expenses as a line chart.
- Make a histogram and describe its shape.

12. Statistics for Data Science

Population and sample

A population is the complete group of interest; a sample is the subset observed. Statistical inference uses sample data to learn about a population while accounting for uncertainty.

Measure	Meaning	Useful when
Mean	Sum divided by count	Values are reasonably balanced
Median	Middle value after sorting	Data contain extreme values
Mode	Most frequent value	Categories or repeated values
Range	Maximum minus minimum	Quick overall spread
IQR	Q3 minus Q1; middle 50% spread	Robust spread summary
Standard deviation	Typical spread around mean	Quantifying numeric variation

Worked example

Values: 2, 4, 4, 6, 9. Mean = $25/5 = 5$. Median = 4. Range = $9 - 2 = 7$. The value 9 pulls the mean above the median.

Uncertainty

Random samples vary. A larger sample can improve precision but does not automatically fix biased selection or bad measurement. A 95% confidence interval comes from a method that captures the true parameter in 95% of repeated samples under its assumptions. A p-value is not the probability that the null hypothesis is true.

Practice

- Find mean, median and range of 3, 5, 5, 7, 20.
- Which centre measure is more robust to extremes?
- Why may volunteer survey results be unrepresentative?

13. Probability and Distributions

Probability measures likelihood from 0 (impossible) to 1 (certain). For equally likely outcomes: probability = favourable outcomes / total outcomes.

$P(\text{rolling a 6 on a fair die}) = 1/6$
 $P(\text{rolling an even number}) = 3/6 = 1/2$

Distribution	Idea	Example
Bernoulli	One trial, success/failure	Pass/fail
Binomial	Success count across fixed independent trials	Heads in 10 flips
Normal	Symmetric bell-shaped distribution	Some measurement errors
Uniform	Equal density across a range	Ideal random number in interval

Independent events do not change each other's probabilities. Conditional probability asks about an event given that another event occurred.

Practice

- Probability of drawing a heart from a 52-card deck?
- How many possible sequences in three coin flips?
- Explain probability versus certainty.

14. Machine Learning Fundamentals

Machine Learning (ML) algorithms learn patterns from examples to make predictions or decisions. Good problem definition, quality data and appropriate evaluation matter more than algorithm complexity.

Type	Goal	Example
Supervised	Learn from labelled examples	Predict price or pass/fail
Unsupervised	Find structure without target labels	Group customers
Reinforcement	Learn actions through rewards	Agent learning a game

Regression predicts a numeric value; classification predicts a category; clustering groups similar observations without predefined labels.

Workflow

- Define the target and success metric.
- Split data into training and test sets.
- Fit preprocessing on training data only.
- Train a simple baseline.
- Evaluate on held-out data.
- Inspect errors, fairness and performance on new data.

Linear regression example

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
import pandas as pd

data = pd.DataFrame({"hours": [1, 2, 3, 4, 5, 6, 7, 8],
                     "score": [35, 40, 50, 55, 65, 70, 78, 85]})
X, y = data[["hours"]], data["score"]
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42)
model = LinearRegression().fit(X_train, y_train)
pred = model.predict(X_test)
print("MAE:", mean_absolute_error(y_test, pred))
```

यह छोटा artificial dataset केवल syntax सीखने के लिए है; इससे यह साबित नहीं होता कि marks केवल study hours पर निर्भर करते हैं.

Underfitting and overfitting

- Underfitting: model misses useful patterns.
- Overfitting: model learns training noise and performs poorly on new data.
- Generalization: performance on unseen data from the intended setting.

Practice

- Give one regression and one classification problem.
- Why should test data not be used repeatedly for tuning?
- Suggest a simple baseline for daily sales prediction.

15. Model Evaluation and Common Pitfalls

Metric	Meaning	Note
MAE	Average absolute prediction error	Same units as target
MSE	Average squared error	Penalizes large errors
RMSE	Square root of MSE	Same units as target
R ²	Fit relative to a mean baseline in common settings	Can be negative on test data

Classification metric	Question answered
Accuracy	What fraction of all predictions are correct?
Precision	Of predicted positives, how many are truly positive?
Recall	Of actual positives, how many were found?
F1-score	Harmonic mean of precision and recall
Confusion matrix	How many true/false positives and negatives?

For imbalanced classes, accuracy can mislead: if 99% of cases are negative, predicting negative every time yields 99% accuracy but finds no positives.

Pitfalls

- Data leakage: future or target information enters training.
- Overfitting: tuning against test data compromises the final evaluation.
- Wrong metric: chosen score does not match the cost of mistakes.
- Distribution shift: future data differs from training data.
- One metric can hide uncertainty or poor subgroup performance.

Practice

- Why might recall matter for screening? Why can precision matter too?
- 90 correct predictions out of 100: calculate accuracy.
- Give an example of leakage in sales forecasting.

16. SQL for Data Analysis

SQL (Structured Query Language) queries relational databases. Common commands include SELECT, WHERE, GROUP BY, ORDER BY and JOIN.

order_id	city	product	amount
1	Varanasi	Pen	20
2	Kanpur	Book	120
3	Varanasi	Book	100
4	Lucknow	Pen	25

Filter and sort

```
SELECT city, product, amount
FROM sales
WHERE amount >= 50
ORDER BY amount DESC;
```

Aggregate

```
SELECT city, SUM(amount) AS total_sales
FROM sales
GROUP BY city
ORDER BY total_sales DESC;
```

JOIN combines rows using related keys. Learn primary keys, foreign keys, INNER JOIN and LEFT JOIN. Joining on a wrong or non-unique key can multiply rows.

Practice

- Show all orders from Varanasi.
- Calculate total amount by product.
- Sort orders from highest to lowest amount.
- Explain why a wrong JOIN key can duplicate data.

17. End-to-End Mini Projects

Project A: Student performance

Question: What patterns appear in marks and attendance? Use synthetic or properly authorized data.

- Columns: anonymous student __id, subject, marks, attendance __percent.
- Validate ranges and handle missing marks.
- Calculate subject-wise mean and median.
- Visualize mark distributions and attendance versus marks.
- Write three findings and two limitations.

Project B: Retail sales

- Columns: date, product, category, quantity, unit __price, city.
- Create $\text{revenue} = \text{quantity} \times \text{unit_price}$.
- Check missing dates, duplicates and impossible quantities.
- Group revenue by month, product and city.
- Create monthly line and category bar charts.
- Recommend one action supported by results.

Project C: Movie ratings

- Explore rating distribution and number of ratings per movie.
- Compare genres only if labels are consistent.
- Separate popularity (rating count) from average rating.
- Discuss selection bias: raters may not represent all viewers.

Report template

Section	Include
Question	What do you want to learn?
Data source	Origin, row meaning, permissions
Preparation	Missing values, duplicates, assumptions
Methods	Statistics, charts or model and why
Results	Key numbers and visual evidence
Limitations	Bias, uncertainty, missing data
Conclusion	Supported next step or recommendation

- Notebook runs from top to bottom.
- Charts have titles and labelled axes.
- Findings are supported by data.
- Sensitive information is excluded.

- Limitations are clear.

18. Practice Questions and Answer Hints

Quick quiz

1. Define Data Science and name three workflow steps.
2. Is city nominal or ordinal data?
3. How is a list different from a set?
4. What does `df.head()` show?
5. Why investigate missing values before filling them?
6. Which chart commonly shows a time trend?
7. How does median differ from mean?
8. Regression vs classification?
9. What is overfitting?
10. Why is correlation not causation?
11. What does SQL GROUP BY do?
12. Name two ways to prevent leakage.

Hands-on tasks

- Write a function returning mean and median.
- Create a DataFrame with 10 rows and filter/group it.
- Load a CSV, inspect missingness, remove duplicates and save a cleaned copy.
- Create a bar chart and histogram.
- Write an EDA report with three findings and two limitations.

Answer hints

- 1. Use data, programming, statistics and context to answer questions.
- 2. Nominal: cities have no inherent order.
- 3. Lists are ordered and allow duplicates; sets store unique values.
- 4. First five rows by default.
- 5. Missingness can be systematic and handling affects results.
- 6. Line chart.
- 7. Mean is sensitive to extremes; median is the middle value.
- 8. Numeric outcome versus category.
- 9. Fitting training-specific noise that harms new-data performance.
- 10. Confounding, reverse direction or coincidence may explain association.
- 11. Group rows so aggregate functions summarize each group.
- 12. Split correctly; fit preprocessing on training data; avoid future/target information.

Self-assessment

Skill	Not yet	Practising	Confident
Python basics	☐	☐	☐
Pandas operations	☐	☐	☐
Data cleaning	☐	☐	☐
Charts and interpretation	☐	☐	☐
Statistics	☐	☐	☐
ML evaluation	☐	☐	☐
End-to-end project	☐	☐	☐

19. Glossary and 8–Week Study Plan

Term	Simple meaning
Algorithm	Steps or rules for solving a task.
Bias	Systematic distortion or unfairness in data or decisions.
Feature	Input variable used by a model.
Target/label	Outcome a supervised model learns to predict.
DataFrame	Labelled two–dimensional table in Pandas.
EDA	Exploratory Data Analysis.
Model	Learned mathematical / computational representation.
Overfitting	Learning noise that harms new–data performance.
Training data	Data used to fit model parameters.
Test data	Held–out data for final evaluation.
Visualization	Graphical representation of information.
SQL	Language for querying relational databases.

Suggested 8–week plan

Week	Focus	Deliverable
1	Data Science overview + Python	Run first programs
2	Python fundamentals	10 small exercises
3	NumPy + Pandas	Explore one CSV
4	Cleaning + EDA	Cleaned dataset and summary
5	Statistics + charts	Charts and findings
6	SQL	10 practice queries
7	Machine Learning	One baseline model
8	Portfolio project	Notebook and report

Useful official learning resources

- Python: docs.python.org/3/tutorial/
- Pandas: pandas.pydata.org/docs/
- NumPy: numpy.org/doc/
- Matplotlib: matplotlib.org/stable/tutorials/
- scikit–learn: [scikit–learn.org/stable/user__guide.html](https://scikit-learn.org/stable/user_guide.html)

Final reminder: tools change, but clear questions, careful preparation, sound reasoning and honest communication remain the foundation of good Data Science.