

Noteskarts

Smart Notes. Bright Future.

COMPUTER APPLICATIONS IN PHARMACY

Unit 4 — Data Handling with Pandas

Short Notes | Diagrams | Tables | Quick Concepts

— Smart Notes, Bright Future —

For D.Pharm & B.Pharm Students

www.noteskarts.com

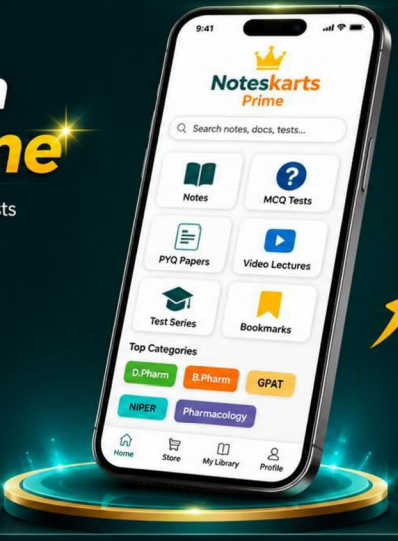
Get Noteskarts Prime — Full Access



Study Smarter with Noteskarts Prime

Handwritten & digital notes, previous-year papers, MCQ tests and full access for **D.Pharm, B.Pharm, GPAT** & Pharmacy Competitive Exam—all in one app.

- Exam-focused notes** for every subject in simple language
- Previous-year papers** with solutions + Topic-wise MCQ tests
- Learn anytime, anywhere** on the Prime app





SCAN ME

Scan to get the app

GET IT ON Google Play



FOCUSSED LEARNING
Better Prepared



SMART PRACTICE
Higher Scores



TRUSTED BY
Thousands of Students



ONE APP
Endless Success

noteskarts.com

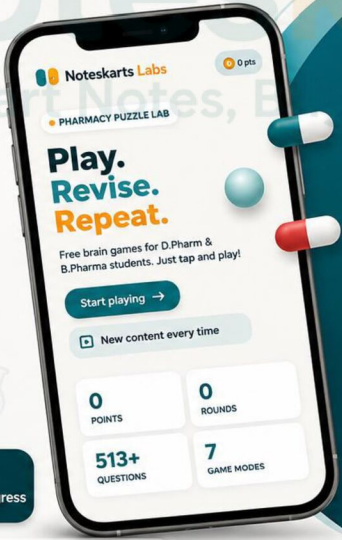
Scan the QR code above with your phone camera to download Noteskarts Prime and unlock complete detailed notes, PYQ papers, MCQ tests & video lectures for D.Pharm, B.Pharm, GPAT & Pharmacy Competitive Exams.

Free Bonus — Noteskarts Puzzle Lab



Play. Revise. Repeat.

Free brain games for D.Pharm & B.Pharm students. No sign-up — just **tap and play!**





100% FREE
No Sign-up Needed!

Scan & Play Now!

YOUR NEXT QUESTION IS JUST A SCAN AWAY!



7+
Game Modes



500+
Questions



Instant
Learning



Track
Your Progress

SEVEN EXCITING GAME MODES



Rapid MCQ



Capsule Match



Term Scramble



Drug Speller



Abbreviations



True or False



Odd One Out

Play 5 minutes. Remember for the exam.
Active recall beats re-reading. Come back daily — you'll keep getting questions you haven't seen yet.



Made for Pharmacy Students

Trusted by Thousands of Students

Perfect for GPAT, DPEE & University Exams

noteskarts.com

Revise this exact topic through fun MCQ games — Rapid MCQ, Term Scramble, True or False & more. 100% free, no sign-up needed. Scan and play!

Introduction to the Pandas Library

Pandas is a powerful, open-source, third-party Python library used for data manipulation and analysis. It is built on top of NumPy and provides fast, flexible tools to work with structured (tabular) data - exactly the kind of data seen in PK study sheets, ADR reports, and hospital records.

Why Pandas for Pharmacy Data?

- Reads data directly from CSV, Excel, and other file formats
- Handles missing/incomplete clinical data gracefully
- Allows quick filtering — e.g. 'all patients with Cmax > 50'
- Supports grouping & statistics — e.g. average dose per drug category
- Works seamlessly with NumPy and Matplotlib for further analysis/plots

Installing & Importing Pandas

```
pip install pandas
```

```
import pandas as pd    # 'pd' is the standard short alias
```

□ Two Core Pandas Structures

Pandas is built around two main data structures: the Series (1-dimensional) and the DataFrame (2-dimensional, table-like). Both are covered in detail next.

Pandas Series Structure

A Series is a one-dimensional, labelled array - similar to a single column of a spreadsheet. Every value has an associated index label (by default 0, 1, 2, ...).

```
import pandas as pd

bp_readings = pd.Series([120, 128, 135, 118])
print(bp_readings)
# 0    120
# 1    128
# 2    135
# 3    118
# dtype: int64
```

Custom Index & Access

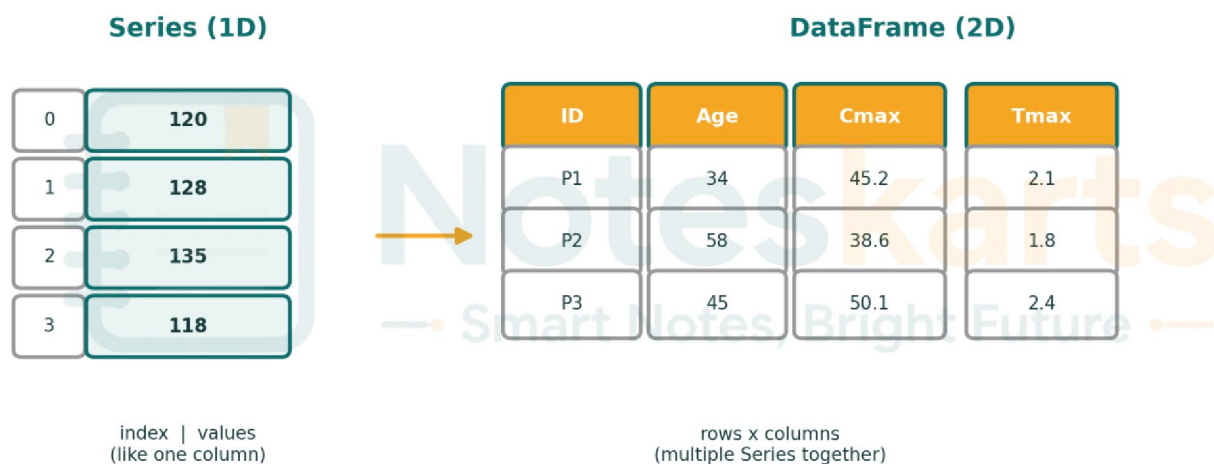
```
doses = pd.Series([500,
250, 1000],
index=['Paracetamol',
'Cetirizine',
'Metformin'])
```

```
print(doses['Metformin'])
# 1000
print(doses.mean())
# average of all values
```

Series Attribute/Method	Purpose
<code>series.values</code>	Returns just the values as an array
<code>series.index</code>	Returns the index labels
<code>series.mean()</code> / <code>.sum()</code>	Basic statistics on the values

Pandas DataFrame Structure

A DataFrame is a two-dimensional, table-like structure with labelled rows and columns - essentially a collection of Series sharing the same index. It is the primary structure used for real datasets.



```
data = {
    'Patient_ID':
    ['P1', 'P2',
    'P3'],
    'Age': [34,
    58, 45],
    'Cmax': [45.2,
    38.6, 50.1],
}
df =
pd.DataFrame(data)
print(df)
```

Term	Meaning
Column	A single field/attribute (e.g. Age) — itself a Series
Row / Record	One observation (e.g. one patient)
Index	Row labels (default: 0, 1, 2, ...)



Noteskarts
— Smart Notes, Bright Future —

Reading CSV and Excel Files

Pandas can directly load PK study datasets and ADR reports that are usually shared as .csv or .xlsx files, turning them into a DataFrame ready for analysis.

Reading a CSV File

```
pk_data = pd.read_csv('pk_study.csv')
print(pk_data)
```

Reading an Excel File

```
adr_data =
pd.read_excel('adr_reports.xlsx',
sheet_name='Sheet1')
print(adr_data)
```

Function	File Type	Common Parameters
pd.read_csv()	.csv	filepath, sep=',', header=0
pd.read_excel()	.xlsx / .xls	filepath, sheet_name
df.to_csv()	Save as .csv	filepath, index=False
df.to_excel()	Save as .xlsx	filepath, index=False

❏ Common Mistake

pd.read_excel() needs the 'openpyxl' library installed (pip install openpyxl) to read modern .xlsx files - a FileNotFoundError or ImportError often means this is missing.

Inspecting Datasets

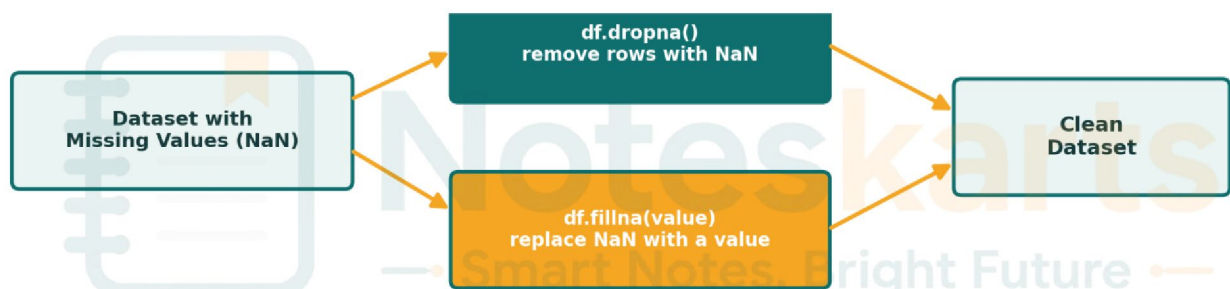
Before analysing any dataset, it's essential to inspect its structure, size, and content. Pandas provides quick built-in functions for this.

Function	Purpose
df.head(n)	Shows the first n rows (default 5)
df.tail(n)	Shows the last n rows (default 5)
df.info()	Column names, data types, non-null counts
df.describe()	Summary statistics (mean, std, min, max, etc.) for numeric columns
df.shape	Tuple of (rows, columns)

Function	Purpose
<code>df.columns</code>	List of all column names
<pre>print(pk_data.head(3)) # first 3 rows print(pk_data.info()) # column types & missing counts print(pk_data.describe()) # mean, std, min, 25%, 50%, 75%, max print(pk_data.shape) # e.g. (120, 6) -> 120 rows, 6 columns</pre>	

Data Cleaning & Handling Missing Values

Real-world PK/ADR datasets often have missing, inconsistent, or duplicate entries. Pandas represents a missing value as NaN (Not a Number).



Function	Purpose
<code>df.isnull()</code>	Returns True/False for each cell — is it missing?
<code>df.isnull().sum()</code>	Count of missing values per column
<code>df.dropna()</code>	Removes rows (or columns) containing NaN
<code>df.fillna(value)</code>	Replaces NaN with a given value (e.g. 0 or mean)
<code>df.drop_duplicates()</code>	Removes duplicate rows

Function	Purpose
<code>df.astype(type)</code>	Converts a column to a different data type
<pre>print(pk_data.isnull().sum()) # missing values per column pk_data['Cmax'] = pk_data['Cmax'].fillna(pk_data['Cmax'].mean()) pk_data_clean = pk_data.dropna(subset=['Patient_ID']) pk_data_clean = pk_data_clean.drop_duplicates()</pre>	

Filtering and Selecting Data

Selecting Columns

```
print(pk_data['Cmax'])          # single column -> Series
print(pk_data[['Age', 'Cmax']]) # multiple columns -> DataFrame
```

Filtering Rows by Condition

```
high_cmax = pk_data[pk_data['Cmax'] > 50]
elderly = pk_data[pk_data['Age'] >= 60]

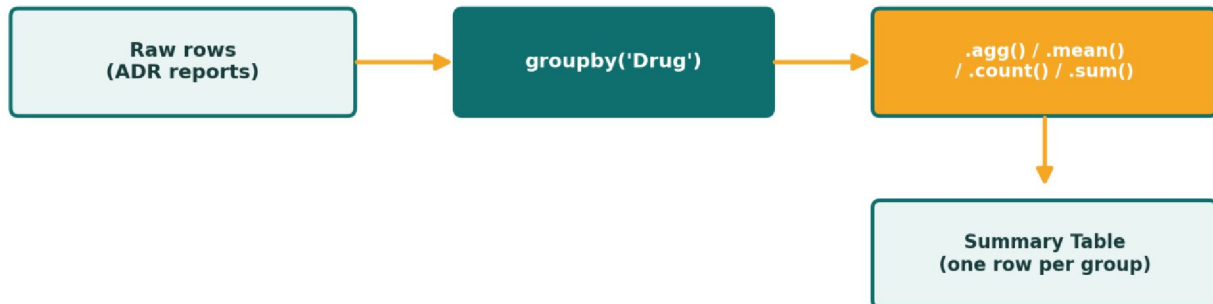
# Multiple conditions: & (and), | (or) – each condition in ( )
elderly_high = pk_data[(pk_data['Age'] >= 60) & (pk_data['Cmax'] > 40)]
```

loc[] and iloc[]

Method	Selects By	Example
<code>df.loc[]</code>	Label / condition	<code>df.loc[0:2, 'Age']</code>
<code>df.iloc[]</code>	Integer position	<code>df.iloc[0:3, 1]</code>

Grouping Data & Aggregation Functions

`groupby()` splits data into groups based on a column's values, then an aggregation function (mean, sum, count...) is applied to summarise each group - very useful for ADR analysis by drug or age group.



```

adr_summary =
adr_data.groupby('Drug_Name')['Reaction_Severity'].count()
print(adr_summary)

avg_cmax_by_group =
pk_data.groupby('Age_Group')['Cmax'].mean()
print(avg_cmax_by_group)
  
```

Aggregation Function	Purpose
<code>.count()</code>	Number of entries in each group
<code>.mean()</code> / <code>.median()</code>	Average / middle value per group
<code>.sum()</code>	Total per group
<code>.agg(['mean', 'max'])</code>	Apply multiple functions at once

9. Mini Case Study - ADR Report Analysis

Bringing every step together on a small Adverse Drug Reaction (ADR) dataset: import, inspect, clean, filter, and summarise.

```
import pandas as pd

adr = pd.read_csv('adr_reports.csv')
print(adr.head())
print(adr.info())

adr['Age'] = adr['Age'].fillna(adr['Age'].mean())    # clean missing ages
adr = adr.drop_duplicates()

serious = adr[adr['Reaction_Severity'] == 'Serious'] # filter
print('Serious ADR cases:', len(serious))

summary = adr.groupby('Drug_Name')['Reaction_Severity'].count() # group +
aggregate
print(summary.sort_values(ascending=False).head())
```

Sample Output:

```
Serious ADR cases: 14
Drug_Name
Metformin      22
Amoxicillin     17
Ibuprofen       11
```

□ Concepts Used

read_csv(), head()/info() for inspection, fillna() and drop_duplicates() for cleaning, condition-based filtering, and groupby().count() for aggregation — the full pandas workflow in one example.

Important Points for Examination

- Pandas is a third-party library built on NumPy for structured/tabular data analysis; import as pd.
- Series = 1D labelled array (like one column); DataFrame = 2D table of rows & columns (like a full spreadsheet).
- pd.read_csv() and pd.read_excel() load PK/ADR data directly into a DataFrame; reading Excel needs 'openpyxl' installed.
- head()/tail() preview rows; info() shows column types & missing counts; describe() gives summary statistics; shape gives (rows, columns).
- Missing values appear as NaN; handle with dropna() (remove) or fillna() (replace).
- drop_duplicates() removes repeated rows; astype() converts a column's data type.

- Filter rows with a condition: `df[df['col'] > x]`; combine conditions with `&` (and) / `|` (or), each wrapped in parentheses.
- `loc[]` selects by label/condition; `iloc[]` selects by integer position.
- `groupby('column')` splits data into groups; follow with an aggregation like `.mean()`, `.count()`, `.sum()`, or `.agg()` to summarise each group.
- A typical pandas workflow: Read -> Inspect -> Clean -> Filter/Select -> Group & Aggregate.



Stay Connected with Noteskarts

Join our community for daily updates, free notes, and exam alerts



WhatsApp Channel



Telegram Group



YouTube Channel

Thank You for Studying with Noteskarts!

www.noteskarts.com

Smart Notes. Bright Future.