

Data Analytics (315326)

Model Question Paper — Set C · Model Answers

Q.1

a) Data analytics; two tools.

2 M

Data Analytics is the science of examining raw data — collecting, cleaning, transforming and modelling it — to find meaningful patterns and conclusions that help in decision-making.

Tools (any two): Microsoft Excel, Python (pandas, NumPy, Matplotlib), R, Power BI, Tableau, SQL.

(Definition 1 M, two tools 1 M)

b) Skewness; relation in positive skew.

2 M

Skewness is the lack of symmetry of a distribution about its central value.

In a **positively skewed** distribution (long tail on the right): **Mean > Median > Mode**. Empirical relation: $\text{Mode} \approx 3 \text{ Median} - 2 \text{ Mean}$.

(Definition 1 M, relation 1 M)

c) Properties of normal distribution.

2 M

1. Bell-shaped curve, symmetric about the mean.
2. Mean = Median = Mode.
3. Total area under the curve = 1; the curve never touches the x-axis.
4. About 68.27%, 95.45% and 99.73% of values lie within $\mu \pm 1\sigma$, 2σ and 3σ .
5. Skewness = 0; mesokurtic ($\beta_2 = 3$).

(Any four, ½ M each)

d) Calculated field.

2 M

A **calculated field** is a new field created inside a PivotTable using a formula on other fields of the source data. It does not change the source data. Created by *PivotTable Analyze → Fields, Items & Sets → Calculated Field*.

Example: Name: *Profit*, Formula: = Sales – Cost; or *Tax* = Sales * 0.18.

(Definition 1 M, example 1 M)

e) Chart elements.

2 M

Chart title, axis titles, data labels, legend, gridlines, axes, data table, error bars, trendline (added from *Chart Elements (+)* or *Chart Design → Add Chart Element*).

(Any four, ½ M each)

f) plt.savefig() and dpi.

2 M

Syntax: plt.savefig(fname, dpi=None, format=None, bbox_inches=None, transparent=False)

e.g. `plt.savefig('sales.png', dpi=300, bbox_inches='tight')`

dpi (dots per inch) sets the **resolution** of a raster image (PNG/JPG). Higher dpi gives a sharper, larger image file (e.g. 300 dpi for printing); default is 100 (from rcParams).

(Syntax 1 M, dpi 1 M)

g) Null and alternative hypothesis.

2 M

Null hypothesis (H_0): a statement of 'no difference' or 'no effect / no association' which is tested, e.g. $H_0: \mu = 60$ (the average marks are 60).

Alternative hypothesis (H_1): the statement accepted when H_0 is rejected; it shows a difference or effect, e.g. $H_1: \mu \neq 60$.

(1 M each)

Q.2

a) Data cleaning and its steps.

4 M

Data cleaning (data cleansing / scrubbing) is the process of detecting and correcting (or removing) incorrect, incomplete, duplicate, inconsistent or irrelevant data so that the dataset is accurate and ready for analysis.

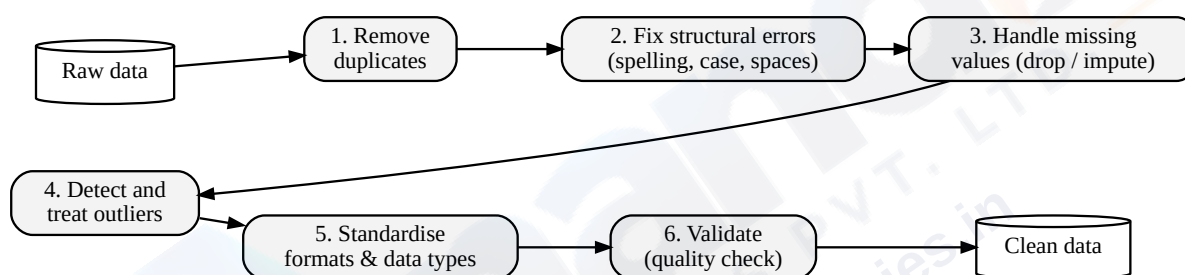


Fig.: Steps in data cleaning

1. **Remove duplicate / irrelevant records** (Excel: Data → Remove Duplicates).
2. **Fix structural errors** — spelling mistakes, extra spaces, wrong case (TRIM, UPPER, LOWER, PROPER; Find & Replace).
3. **Handle missing values** — delete the record or impute (mean / median / mode).
4. **Detect and treat outliers** — IQR method / box plot; correct or remove if they are errors.
5. **Standardise** formats — dates, units, data types (text → number).
6. **Validate** — check that data follows rules and is consistent; document the cleaning.

(Definition 1 M, diagram 1 M, steps 2 M)

b) Box plot and five-number summary.

4 M

Ascending order: 32, 36, 38, 41, 45, 47, 50, 55; $n = 8$

Minimum = 32

$Q_1 = 1(n + 1)/4$ th value = 2.25th value = 2th value + $0.25 \times (3\text{th} - 2\text{th}) = 36 + 0.25 \times (38 - 36) = 36.50$

Q_2 (Median) = $2(n + 1)/4$ th value = 4.50th value = 4th value + $0.50 \times (5\text{th} - 4\text{th}) = 41 + 0.50 \times (45 - 41) = 43$

$Q_3 = 3(n + 1)/4$ th value = 6.75th value = 6th value + $0.75 \times (7\text{th} - 6\text{th}) = 47 + 0.75 \times (50 - 47) =$

49.25

Maximum = 55

$IQR = 49.25 - 36.50 = 12.75$; fences 17.38 and 68.38 $\rightarrow$ no outlier.

Minimum	Q1	Median	Q3	Maximum
32	36.50	43	49.25	55

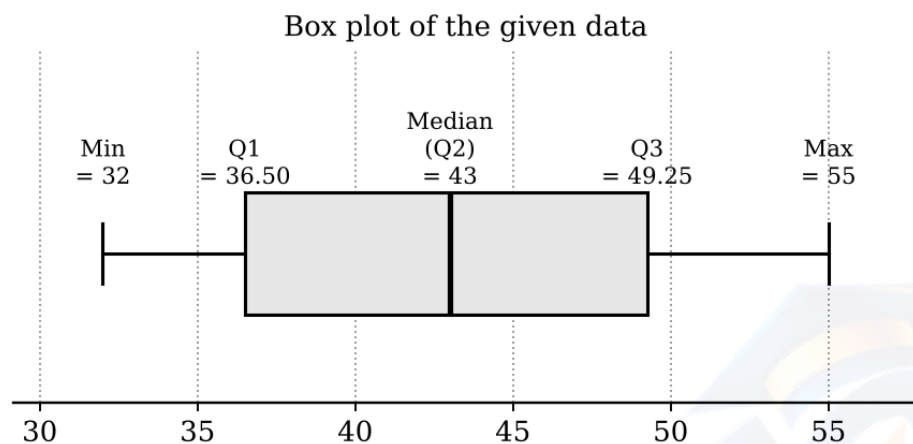


Fig.: Box-and-whisker plot

(Five-number summary 2 M, box plot 2 M)

c) Discovery and Data Preparation phases.

4 M

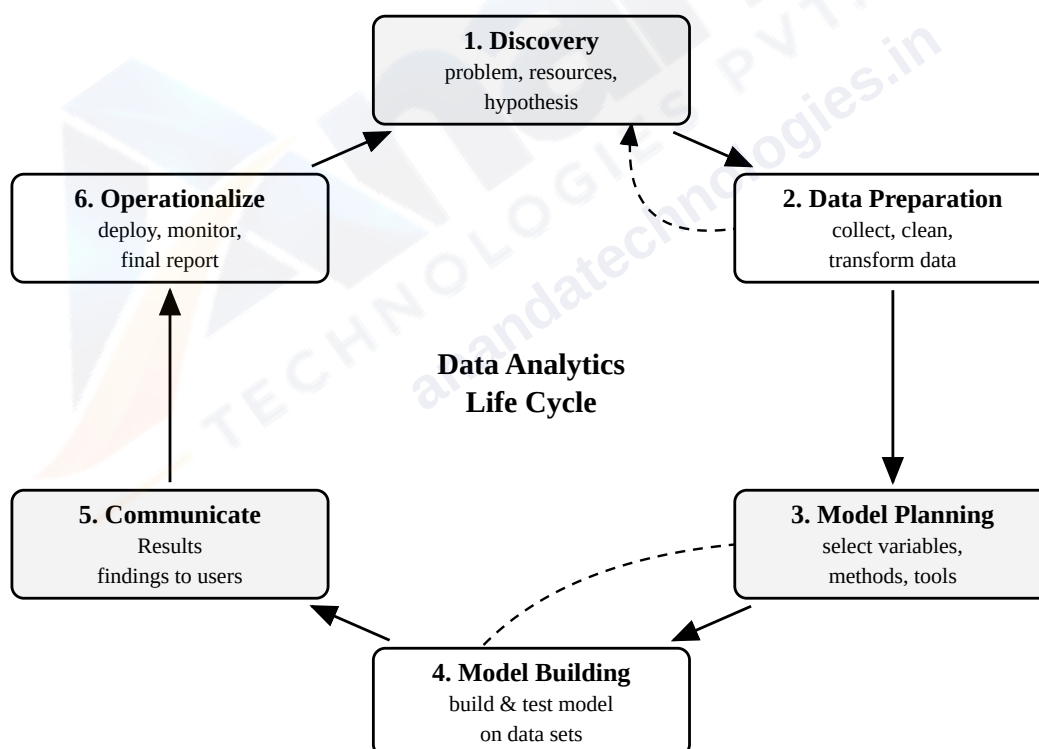


Fig.: Data Analytics life cycle (six phases; dashed arrows show going back to an earlier phase)

Phase 1 — Discovery:

- Learn the business domain and history (similar past projects).

- Identify resources — people, technology, time and data available.
- Frame the problem: state the business problem as an analytics problem with clear objectives and success criteria.
- Identify key stakeholders and form **initial hypotheses** (IH) to be tested; identify possible data sources.

Phase 2 — Data preparation:

- Prepare an **analytic sandbox** (separate workspace) so that analysis does not disturb production data.
- Perform **ETL / ELT** — extract, transform and load the data into the sandbox.
- Understand the data (data conditioning): clean, handle missing values, remove duplicates, normalise and transform.
- Survey and **visualise** data to check quality; tools: Excel, Python (pandas), OpenRefine, Hadoop. Typically takes the most time of the project.

(Diagram 1 M, Discovery 1½ M, Data preparation 1½ M)

d) Pie chart in Excel — create, move to own sheet, add data labels.

4 M

Create:

1. Enter categories in one column and values in the next (e.g. A1:B5).
2. Select the range → **Insert** → **Charts** → **Insert Pie or Doughnut Chart** → **2-D Pie**. The pie appears as an embedded chart.

Move to its own sheet:

3. Select the chart → **Chart Design** → **Move Chart** → choose **New sheet**, give a name (e.g. 'Budget Pie') → **OK**. (Shortcut: select data and press **F11** to create a chart directly on a new chart sheet.)

Add data labels:

4. Click the chart → **Chart Elements (+)** → tick **Data Labels** → choose position (Center, Inside End, Outside End, Best Fit, Data Callout).
5. For percentages: right-click a label → **Format Data Labels** → tick *Category Name*, *Value* and/or *Percentage*.

Monthly Budget (with data labels)

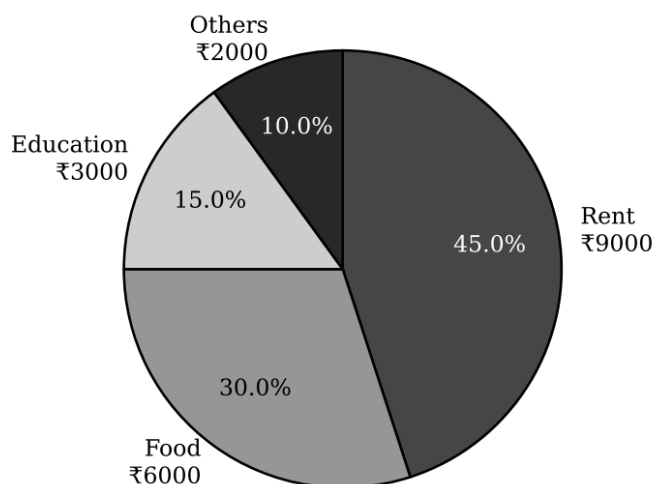


Fig.: Pie chart with data labels (category, value, percentage)

(Create 1 M, move 1½ M, data labels 1½ M)

Q.3

a) ANOVA; one-way ANOVA steps.

4 M

ANOVA (Analysis of Variance), developed by R. A. Fisher, is a statistical test used to check whether the **means of three or more groups** are significantly different. It compares the variation *between* groups with the variation *within* groups using the F-ratio.

One-way ANOVA — one factor (independent variable) with k groups, e.g. marks of students taught by three methods.

1. State $H_0: \mu_1 = \mu_2 = \dots = \mu_k$ (all means equal); H_1 : at least one mean differs.
2. Find group means and grand mean. Compute **SSB** (between groups) = $\sum n_i(\bar{x}_i - \bar{x})^2$ and **SSW** (within groups) = $\sum \sum (x - \bar{x}_i)^2$.
3. Mean squares: $MSB = SSB/(k - 1)$, $MSW = SSW/(N - k)$.
4. $F = MSB / MSW$.
5. Compare with F-table value at $(k - 1, N - k)$ d.f. and chosen α : if $F_{cal} > F_{table}$, reject H_0 .

Illustration (computed): marks with three teaching methods A [70, 72, 68, 74], B [78, 80, 76, 82], C [65, 67, 69, 63]:

Source	SS	d.f.	MS	F
Between groups	344	2	172	25.80
Within groups	60	9	6.67	
Total	404	11		

$F = 25.80 > F_{0.05}(2, 9) = 4.26$, so H_0 is rejected — the teaching methods give different mean marks.

(Meaning 1 M, steps 3 M)

b) Grouping items in a PivotTable; Pivot Chart.

4 M

1. Create the PivotTable: **Insert** → **PivotTable** → **New Worksheet**; drag **Date** → **Rows**, **Item** → **Columns**, **Qty** → **Values**.
2. **Group dates by month**: right-click a date in Row Labels → **Group...** → select **Months** (and **Years** if needed) → **OK**.
3. **Group items into categories**: in the column labels select *Pen* and *Pencil* (Ctrl + click) → right-click → **Group**. Excel makes 'Group1'; rename it **Writing**. Similarly group Notebook and Register as **Paper**. Collapse the groups (– button) to see only group totals.
4. **Pivot Chart**: click inside the PivotTable → **PivotTable Analyze** → **PivotChart** → **Clustered Column** → **OK**. Add title and axis titles.
5. To undo grouping: right-click the group → **Ungroup**.

Result (Sum of Qty, computed):

Month / Group	Paper	Writing	Grand Total
Apr	60	200	260
May	110	150	260
Jun	55	90	145
Grand Total	225	440	665

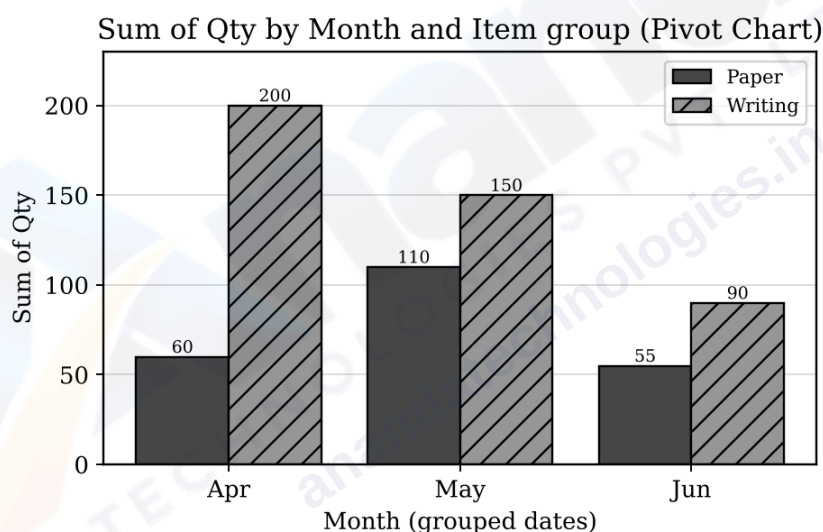


Fig.: Pivot Chart of grouped data

(Grouping 2 M, Pivot Chart 1 M, result 1 M)

c) Figure size and axis customisation in Matplotlib.

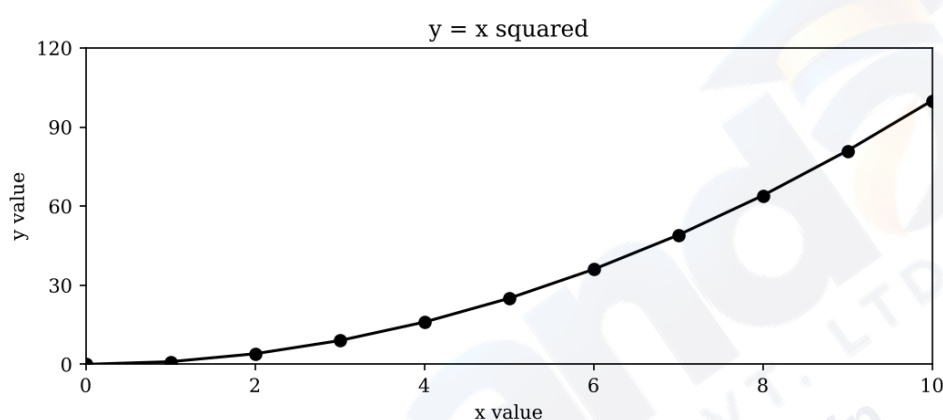
4 M

- `plt.figure(figsize=(w, h))` — sets the figure width and height in inches; the ratio `w : h` is the aspect ratio. (`ax.set_aspect('equal')` makes one unit equal on both axes.)
- `plt.xlim(min, max), plt.ylim(min, max)` — set axis limits (range shown).
- `plt.xticks(positions, labels), plt.yticks(...)` — set tick positions and their text.
- `plt.xlabel(), plt.ylabel(), plt.title()` — axis labels and title.

```
# Changing figure size and customising axes
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(0, 11)
y = x ** 2

plt.figure(figsize=(8, 3))          # 8 in wide, 3 in high (aspect 8:3)
plt.plot(x, y, color='black', marker='o')
plt.xlim(0, 10)                    # x-axis limits
plt.ylim(0, 120)                   # y-axis limits
plt.xticks(range(0, 11, 2))        # x ticks: 0, 2, 4, ... 10
plt.yticks([0, 30, 60, 90, 120], ['0', '30', '60', '90', '120'])
plt.title('y = x squared')
plt.xlabel('x value')
plt.ylabel('y value')
plt.show()
```



Output window (plt.show())

(Explanation 2 M, program 2 M)

d) Mean, median and mode with example.

4 M

- **Mean (arithmetic mean):** sum of all observations divided by their number, $\bar{x} = \Sigma x / n$. Uses all values but is affected by extreme values.
- **Median:** the middle value when the data are arranged in order (average of the two middle values if n is even). Not affected by extreme values.
- **Mode:** the value that occurs most frequently. Useful for categorical data (e.g. most sold shoe size).

Example: Runs scored in 7 matches: 12, 15, 11, 15, 18, 15, 14

Mean $\bar{x} = \Sigma x / n = (12 + 15 + 11 + 15 + 18 + 15 + 14) / 7 = 100 / 7 = \mathbf{14.29}$

Arranged in ascending order: 11, 12, 14, 15, 15, 15, 18

n = 7 (odd), median = ((n+1)/2)th = 4th value = **15**

Mode: 15 occurs 3 times → **Mode = 15**

(Each measure 1 M, example 1 M)

Q.4

a) Correlation vs regression.

4 M

Correlation	Regression
Measures the degree and direction of relationship between two variables.	Gives the average functional relationship to estimate/predict one variable from another.
Both variables are treated alike; no dependent / independent variable.	One variable is dependent (y) and the other independent (x).
$r_{xy} = r_{yx}$ (symmetric).	$b_{yx} \neq b_{xy}$ in general; two different regression lines.
Coefficient r lies between -1 and +1; unit-free.	Regression coefficients can have any value and carry units.
Does not show cause and effect; not used for prediction.	Used for prediction / forecasting ($y = a + bx$).

(Any four points, 1 M each)

b) Central Limit Theorem; importance.

4 M

Statement: For a population with mean μ and finite standard deviation σ , the distribution of sample means $\bar{x}$ of random samples of size n approaches a normal distribution $N(\mu, \sigma^2/n)$ as n increases ($n \geq 30$ is usually enough), whatever the shape of the population.

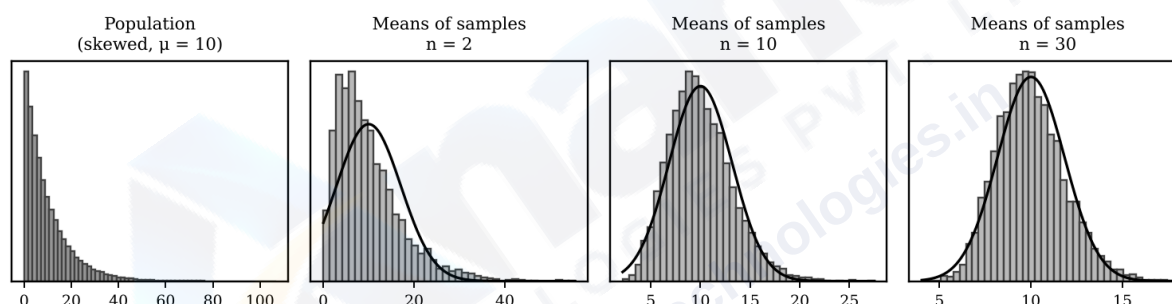


Fig.: Sample means from a skewed population become normal as n increases

Importance:

- Allows normal-distribution based tools (z-test, confidence intervals) to be used for sample means even when the population is not normal.
- Basis of estimating population mean from a sample in surveys, quality control and A/B testing.
- Explains why the standard error $\sigma/\sqrt{n}$ decreases with larger samples — bigger samples give more precise estimates.
- Justifies the use of sampling instead of studying the whole population (saves time and cost).

(Statement 2 M, importance 2 M)

c) Suitable chart for crop data.

4 M

The data compares **three crops** over **three years**, so a **multiple bar / clustered column chart** is suitable — for every year the three crop bars stand side by side, which makes both comparisons easy.

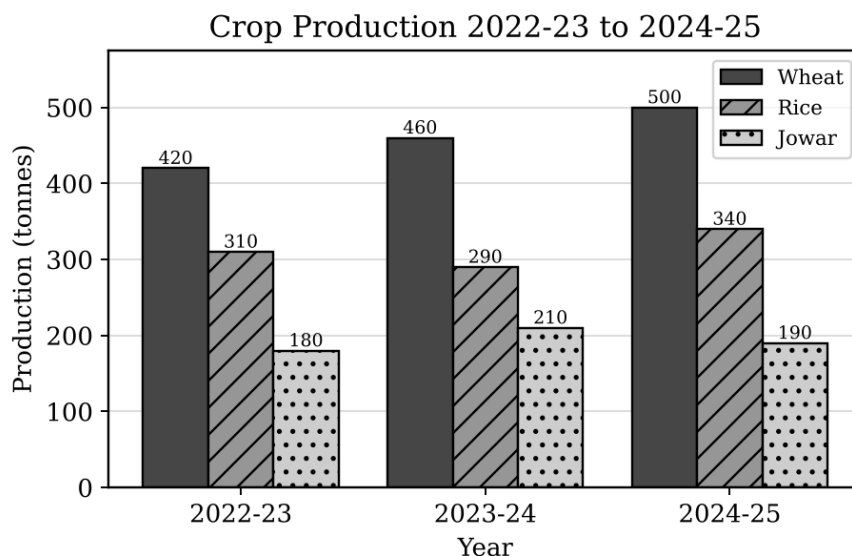


Fig.: Clustered column (multiple bar) chart

Excel: select A1:D4 → Insert → Column → **Clustered Column**; use *Switch Row/Column* to put years on the x-axis; add title, axis titles, legend and data labels.

(Correct chart 3 M, naming / labels 1 M)

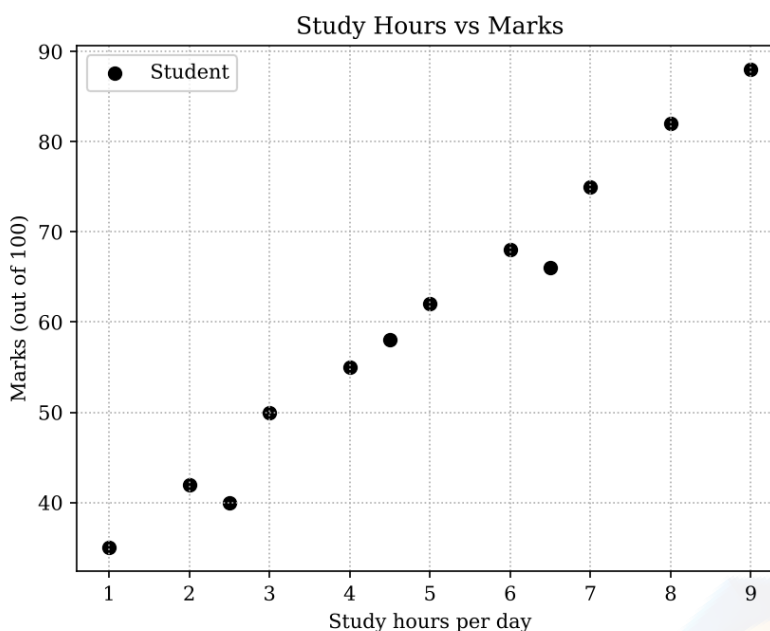
d) Python program — scatter plot.

4 M

```
# Scatter plot: study hours vs marks
import matplotlib.pyplot as plt

hours = [1, 2, 2.5, 3, 4, 4.5, 5, 6, 6.5, 7, 8, 9]
marks = [35, 42, 40, 50, 55, 58, 62, 68, 66, 75, 82, 88]

plt.scatter(hours, marks, color='black', marker='o', s=40, label='Student')
plt.title('Study Hours vs Marks')
plt.xlabel('Study hours per day')
plt.ylabel('Marks (out of 100)')
plt.legend()
plt.grid(True, linestyle=':')
plt.show()
```



Output window (plt.show())

(Program 3 M, title/labels/legend 1 M)

e) Slicers to filter and format a PivotTable.

4 M

A **slicer** is a floating panel of buttons for one field; clicking buttons filters the PivotTable visually.

1. Click inside the PivotTable → **PivotTable Analyze** → **Insert Slicer** (or Insert → Slicer).
2. Tick the fields (e.g. Region, Product) → OK. One slicer box appears for each field.
3. **Filter:** click a button to show only that item; **Ctrl + click** (or Multi-Select button) for several items; **Clear Filter** (funnel with ×) to show all.
4. **Format:** select the slicer → **Slicer** tab → choose a **Slicer Style**, set *Columns* (buttons in 2–3 columns), button height/width, slicer caption and size.
5. **Report Connections** (Slicer tab) — connect one slicer to many PivotTables/Pivot Charts so that a dashboard filters together. A **Timeline** works similarly for date fields.

(Insert 1 M, filtering 1½ M, formatting 1½ M)

Q.5

a) Range, variance, SD and coefficient of variation.

6 M

Range = Largest – Smallest = 40 – 22 = **18**; coefficient of range = $(40 - 22)/(40 + 22) = 0.2903$

Mean $\bar{x} = 240/8 = 30$

x	$x - \bar{x}$	$(x - \bar{x})^2$
24	-6	36
30	0	0
28	-2	4
36	6	36
22	-8	64

40	10	100
30	0	0
30	0	0
$\Sigma x = 240$		$\Sigma(x - \bar{x})^2 = 240$

Variance $\sigma^2 = \Sigma(x - \bar{x})^2/n = 240/8 = 30$

Standard deviation $\sigma = \sqrt{30} = 5.4772$

Coefficient of variation C.V. = $(\sigma / \bar{x}) \times 100 = (5.4772 / 30) \times 100 = 18.26\%$

C.V. shows the relative variability; a smaller C.V. means more consistent sales.

(Range 1 M, variance 2 M, SD 1 M, C.V. 2 M)

b) Types of skewness and kurtosis.

6 M

Skewness — lack of symmetry. Measured by Karl Pearson's coefficient $Sk = (\text{Mean} - \text{Mode})/\sigma$ or $3(\text{Mean} - \text{Median})/\sigma$.

- Symmetric (zero skew):** both halves are mirror images; Mean = Median = Mode; $Sk = 0$. e.g. heights of adults.
- Positive (right) skew:** longer tail to the right; Mean > Median > Mode; $Sk > 0$. e.g. income of people.
- Negative (left) skew:** longer tail to the left; Mean < Median < Mode; $Sk < 0$. e.g. marks in an easy test.

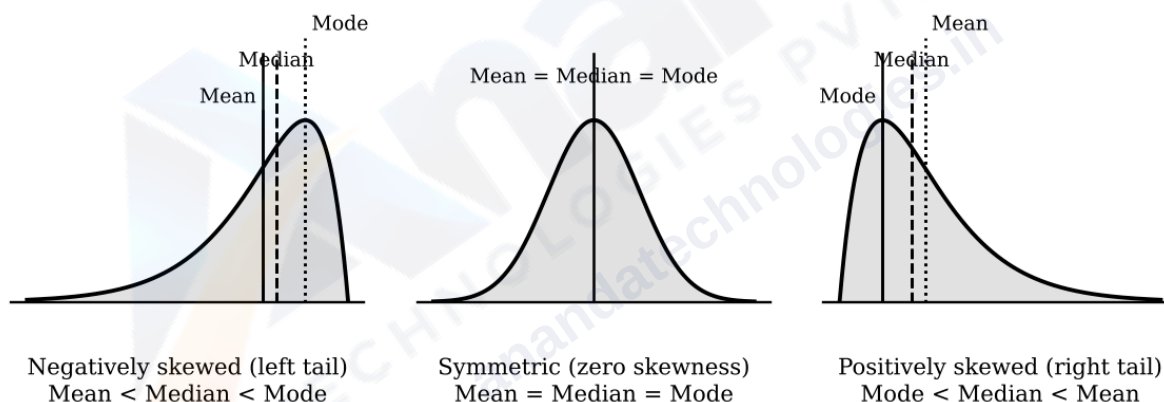


Fig.: Negative, zero and positive skewness

Kurtosis — degree of peakedness / tail heaviness compared with normal curve; $\beta_2 = \mu_4/\mu_2^2$.

- Leptokurtic** ($\beta_2 > 3$): sharp peak, heavy tails.
- Mesokurtic** ($\beta_2 = 3$): normal curve.
- Platykurtic** ($\beta_2 < 3$): flat top, light tails.

Types of kurtosis (all curves have the same variance)

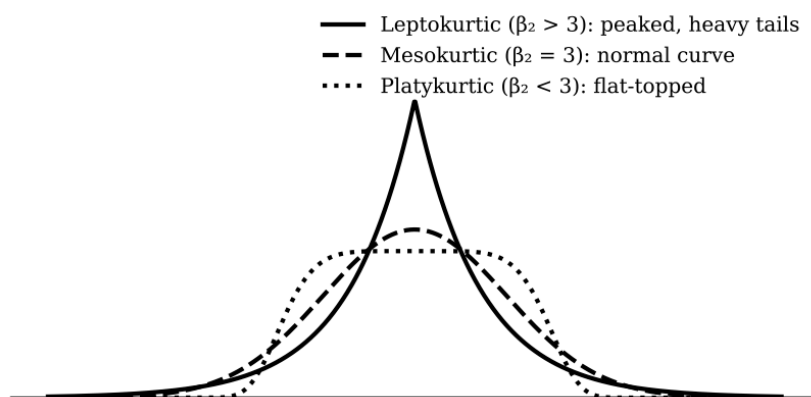


Fig.: Types of kurtosis

(Skewness types with diagram 3 M, kurtosis types with diagram 3 M)

c) Python program — pandas summary and bar chart.

6 M

```
# Summarise sales data with pandas and draw a bar chart
import pandas as pd
import matplotlib.pyplot as plt

data = {'Region': ['North', 'South', 'East', 'West', 'North', 'South', 'East', 'West'],
        'Quarter': ['Q1', 'Q1', 'Q1', 'Q1', 'Q2', 'Q2', 'Q2', 'Q2'],
        'Sales': [250, 310, 180, 220, 280, 290, 210, 260]}
df = pd.DataFrame(data)

summary = df.pivot_table(values='Sales', index='Region', columns='Quarter',
                           aggfunc='sum')
print(summary)
print('\nTotal sales by region:')
print(df.groupby('Region')['Sales'].sum())

summary.plot(kind='bar', color=['gray', 'lightgray'], edgecolor='black')
plt.title('Quarter-wise Sales by Region')
plt.xlabel('Region')
plt.ylabel('Sales (Rs. thousand)')
plt.xticks(rotation=0)
plt.legend(title='Quarter')
plt.tight_layout()
plt.savefig('region_sales.png')
plt.show()
```

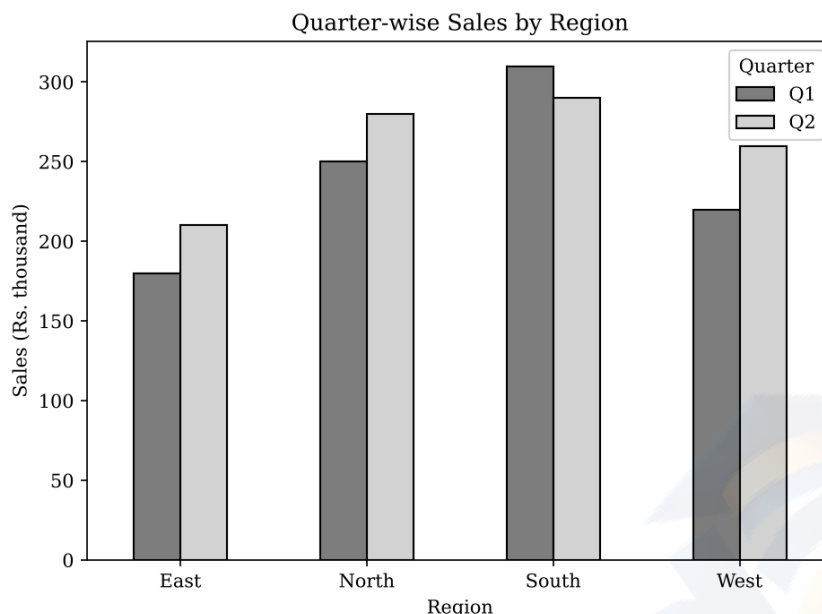
Output:

Quarter	Q1	Q2
Region		
East	180	210
North	250	280
South	310	290
West	220	260

Total sales by region:

Region	Sales
East	390

```
North    530
South    600
West     480
Name: Sales, dtype: int64
```



Output: saved file region_sales.png

`pd.DataFrame()` creates the table; `pivot_table()` summarises Sales by Region (rows) and Quarter (columns) like an Excel PivotTable; `groupby().sum()` gives region totals; `DataFrame.plot(kind='bar')` draws grouped bars using Matplotlib; `savefig()` saves the PNG.

(DataFrame and summary 2 M, chart 3 M, save 1 M)

Q.6

a) Excel dashboard features.

6 M

i) Tables and data grids

- Source data is converted to an **Excel Table** (Ctrl + T / Format as Table): it gets a name, banded rows, filter buttons and expands automatically when rows are added, so linked charts and formulas update.
- Data grids on the dashboard show summarised figures (top products, region totals) using PivotTables or formulas (SUMIFS, AVERAGEIFS, XLOOKUP), with **conditional formatting** (colour scales, icon sets, data bars) and **sparklines**.

ii) Dynamic filters and controls

- Slicers** and **timelines** connected to PivotTables / Pivot Charts filter all visuals with one click.
- Form controls** (drop-down list / combo box, option buttons, check boxes, scroll bar from Developer tab) and **Data Validation** lists change a cell value that formulas use, so charts change dynamically.

iii) Trend analysis and forecasting

- Line charts and **trendlines** (Chart Elements → Trendline: linear, exponential, moving average) show the direction of data over time; trendline can be extended forward to forecast.
- Forecast Sheet** (Data → Forecast Sheet) and functions FORECAST.LINEAR, TREND, FORECAST.ETS predict future values with confidence bounds.

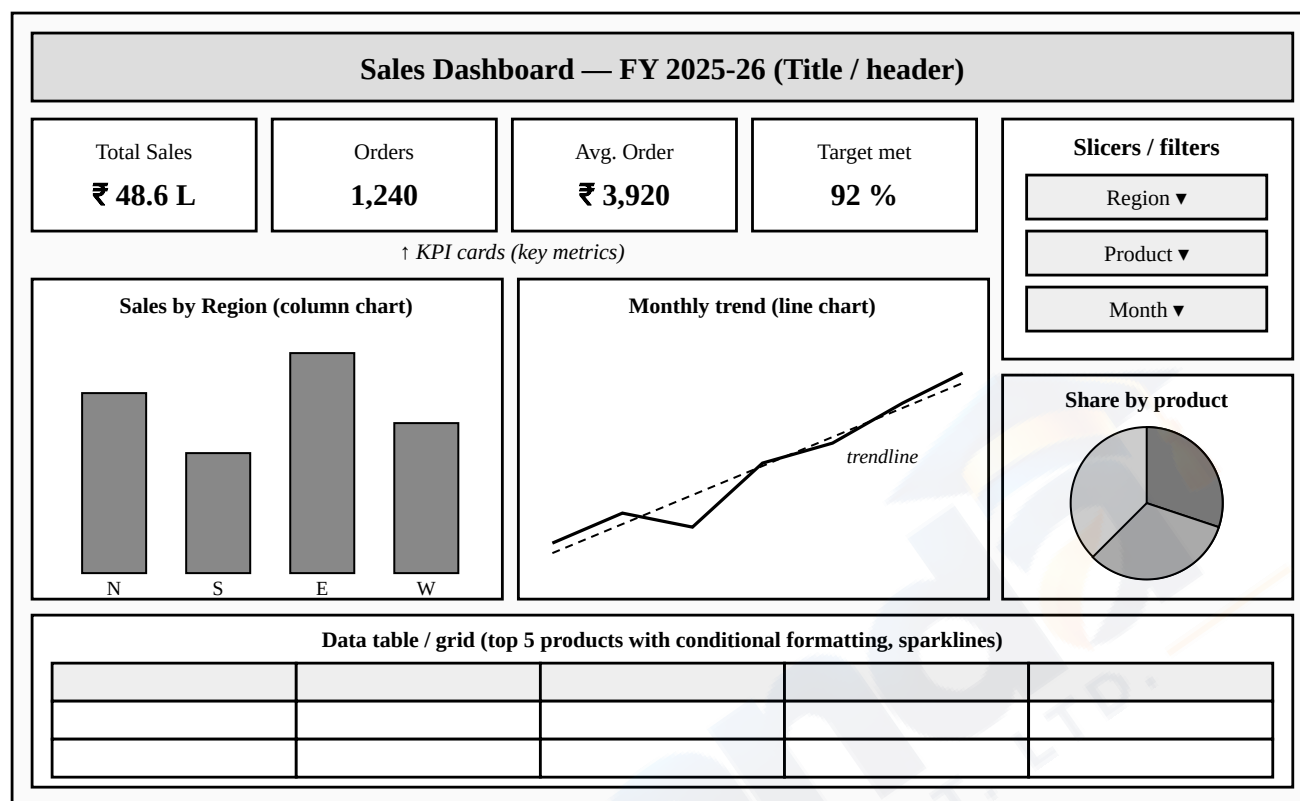


Fig.: Typical elements of an Excel dashboard (layout sketch)

(Each feature 2 M)

b) Normal distribution numerical.

6 M

$X \sim N(\mu = 60, \sigma = 10)$, $N = 1000$. Standard normal variate $z = (x - \mu)/\sigma$.

i) More than 75: $z = (75 - 60)/10 = 1.5$. $P(X > 75) = P(z > 1.5) = 0.5 - (\text{area } 0 \text{ to } 1.5) = 0.5 - 0.4332 = \mathbf{0.0668}$

Number of students = $1000 \times 0.0668 = 66.81 \approx \mathbf{67}$

ii) Between 50 and 70: $z_1 = (50 - 60)/10 = -1$, $z_2 = (70 - 60)/10 = 1$. $P(-1 < z < 1) = 2 \times (\text{area } 0 \text{ to } 1) = 2 \times 0.3413 = \mathbf{0.6827}$

Number of students = $1000 \times 0.6827 = 682.69 \approx \mathbf{683}$

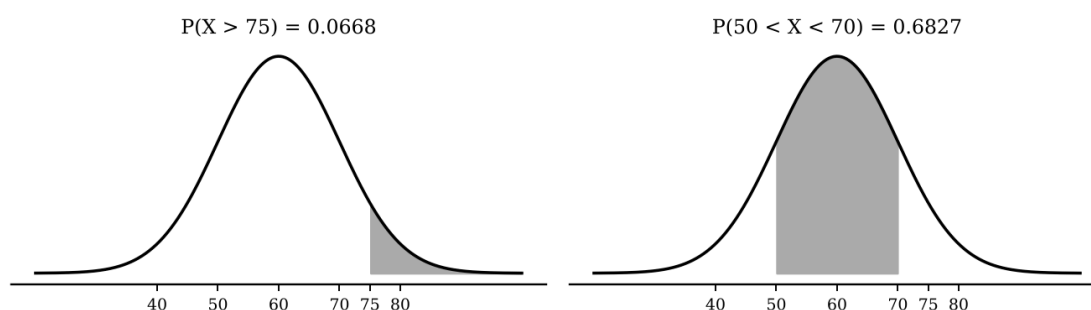


Fig.: Required areas under the normal curve

(Areas from the standard normal table. z-values 2 M, probabilities 2 M, number of students 2 M)

c) Python program — subplots, saved at 300 dpi.

6 M

```
# Line chart and bar chart of the same data in one figure, saved at 300 dpi
import matplotlib.pyplot as plt

years = ['2021', '2022', '2023', '2024', '2025']
admissions = [180, 210, 195, 240, 265]

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 4))

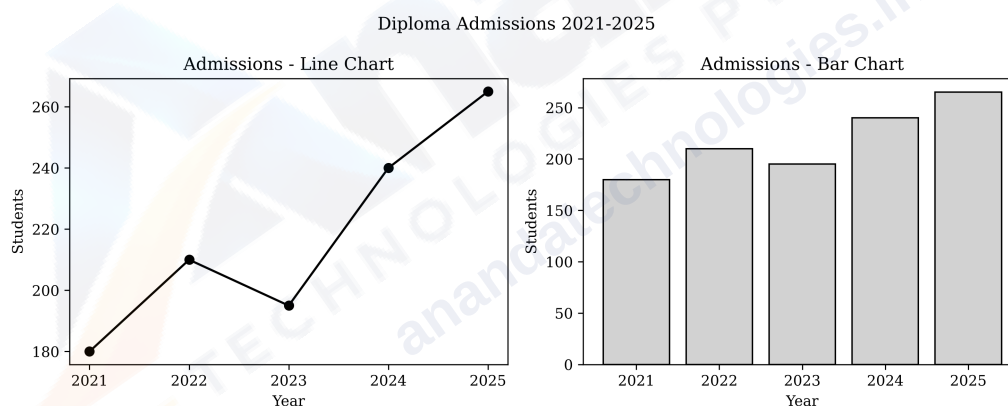
ax1.plot(years, admissions, marker='o', color='black')
ax1.set_title('Admissions - Line Chart')
ax1.set_xlabel('Year')
ax1.set_ylabel('Students')

ax2.bar(years, admissions, color='lightgray', edgecolor='black')
ax2.set_title('Admissions - Bar Chart')
ax2.set_xlabel('Year')
ax2.set_ylabel('Students')

fig.suptitle('Diploma Admissions 2021-2025')
plt.tight_layout()
plt.savefig('admissions.png', dpi=300)          # high resolution
plt.savefig('admissions.pdf')                  # vector format
print('Saved admissions.png (300 dpi) and admissions.pdf')
plt.show()
```

Output:

Saved admissions.png (300 dpi) and admissions.pdf



Output: saved file admissions.png

Files created: admissions.pdf

Functions used:

- `plt.subplots(1, 2, figsize=(10, 4))` — creates one figure with 1 row × 2 columns of axes (ax1, ax2).
- `ax.plot()` / `ax.bar()` — draw line and bar charts on each axes.
- `ax.set_title()`, `ax.set_xlabel()`, `ax.set_ylabel()` — title and labels of each sub-plot; `fig.suptitle()` — common title.
- `plt.tight_layout()` — adjusts spacing so labels do not overlap.

- `plt.savefig('admissions.png', dpi=300)` — saves a high-resolution PNG; the PDF copy is vector (sharp at any zoom).

(Program 4 M, explanation 2 M)

Note: This is a **model question paper prepared by Ananda Technologies** only for practice and exam preparation, based on the pattern of previous MSBTE question papers and the K-scheme syllabus. It is **not an official MSBTE paper**, and it does **not** mean that these questions will come in the exam. Answers follow standard textbook points; marks shown are indicative. Free for students — anandatechnologies.in

