

Data Analytics (315326)

Model Question Paper — Set B · Model Answers

Q.1

a) Importance of data analytics.

2 M

- **Better decision-making** — decisions are based on facts and evidence instead of guesswork.
- **Understanding customers** — buying patterns and preferences help in targeted marketing.
- **Cost reduction and efficiency** — finds waste and bottlenecks in processes.
- **Forecasting and risk management** — predicts demand, detects fraud and risks early.
- **Competitive advantage** — new products and services can be designed from trends in data.

(Any four, ½ M each)

b) Kurtosis and its types.

2 M

Kurtosis is the measure of peakedness or flatness of a frequency curve (and heaviness of its tails) compared with the normal curve. Measured by $\beta_2 = \mu_4/\mu_2^2$.

Types: Leptokurtic ($\beta_2 > 3$, more peaked), Mesokurtic ($\beta_2 = 3$, normal), Platykurtic ($\beta_2 < 3$, flat).

(Definition 1 M, types 1 M)

c) Levels of measurement.

2 M

Scale	Meaning	Example
Nominal	Names / categories, no order	Gender, city, blood group
Ordinal	Categories with order, gaps not equal	Grade A/B/C, rating 1–5 stars
Interval	Ordered, equal gaps, no true zero	Temperature in °C, calendar year
Ratio	Equal gaps with true zero	Height, weight, salary, marks

(½ M each)

d) Pivot Chart.

2 M

A **Pivot Chart** is a chart (column, bar, line, pie etc.) that is created from a PivotTable and is linked to it. It shows the summarised PivotTable data graphically; when fields are filtered, rearranged or refreshed in the PivotTable, the Pivot Chart changes automatically, and vice versa. It has field buttons to filter data directly on the chart.

(Definition 2 M)

e) Purpose of exploding a pie slice.

2 M

A slice is exploded (pulled away from the centre of the pie) to **emphasise** it — to draw the reader's attention to one important category such as the largest share, smallest share or the item under discussion. The values are not changed; only the presentation becomes clearer. (Excel: Format Data Point → Point Explosion; Matplotlib: `explode` parameter of `pie()`.)

(Purpose 2 M)

f) plt.legend().

2 M

Use: `plt.legend()` displays a **legend** — a box that identifies each data series by its colour/marker and the text given in the `label` argument of `plot()`, `bar()`, `scatter()` etc. It is needed when more than one series is shown.

Syntax: `plt.legend(loc='best', title=None, fontsize=None, ncol=1)`
e.g. `plt.plot(x, y, label='2025'); plt.legend(loc='upper left')`

(Use 1 M, syntax 1 M)

g) Outlier; method to detect.

2 M

An **outlier** is an observation that lies at an abnormally large distance from the other values of the data set (very high or very low), e.g. 95 in 12, 15, 14, 18, 95.

Detection — IQR method: a value below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$ is an outlier (seen as a separate point on a box plot). Other: z-score method ($|z| > 3$).

(Definition 1 M, method 1 M)

Q.2

a) Data Analytics life cycle.

4 M

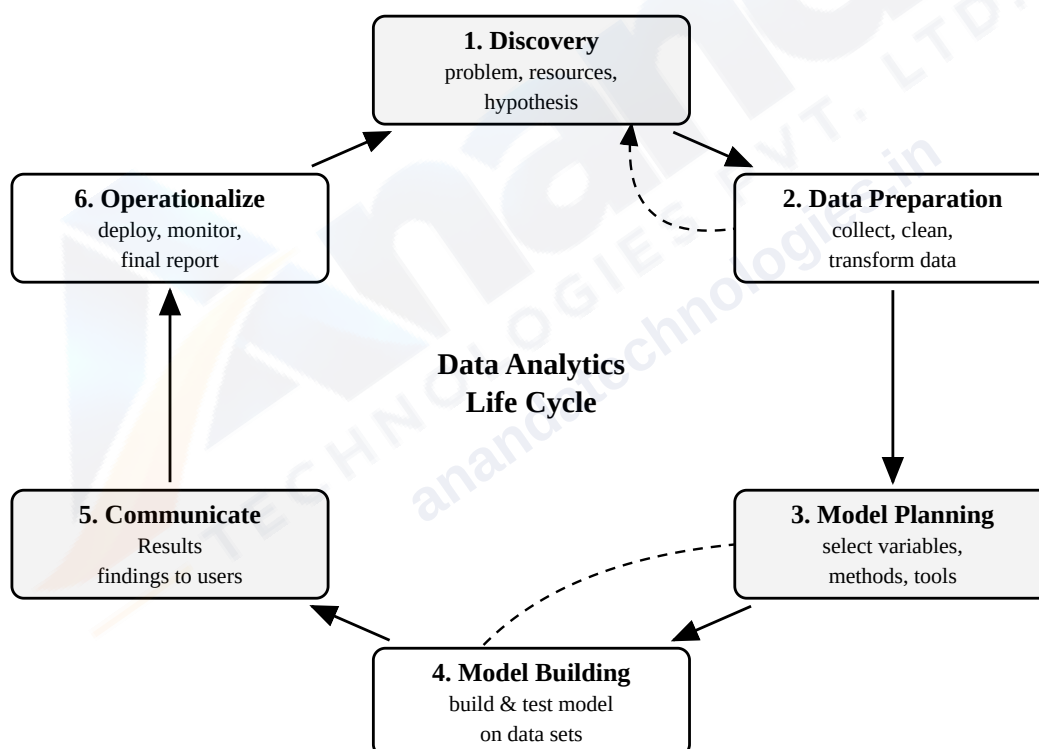


Fig.: Data Analytics life cycle (six phases; dashed arrows show going back to an earlier phase)

1. **Discovery** — learn the business domain, define the problem and objectives, identify data sources and resources, frame initial hypotheses.
2. **Data preparation** — collect data into an analytic sandbox, perform ETL, clean (missing values, duplicates, errors), transform and explore the data.

3. **Model planning** — study relationships between variables and choose suitable analytical techniques (regression, clustering, decision tree) and tools (Python, R, Excel).
4. **Model building** — create training and test data sets, build and run the model and measure its accuracy; refine if needed.
5. **Communicate results** — decide whether the project succeeded against its criteria; present insights with visuals to stakeholders.
6. **Operationalize** — deploy the model on a small scale first (pilot), then in production; give final reports, code and technical documents; monitor.

(Diagram 2 M, explanation 2 M)

b) Central Limit Theorem with example.

4 M

Statement: When random samples of a sufficiently large size n (usually $n \geq 30$) are taken from a population having mean μ and standard deviation σ , the distribution of the sample means is approximately **normal**, with mean $\mu_{\bar{x}} = \mu$ and standard error $\sigma_{\bar{x}} = \sigma/\sqrt{n}$ — even if the population itself is not normal.

Example: The monthly mobile data used by students has mean $\mu = 50$ GB and $\sigma = 12$ GB, and the distribution is skewed. If many samples of $n = 36$ students are taken, the sample means will follow approximately $N(50, (12/\sqrt{36})^2) = N(50, 2^2)$. Hence about 95% of sample means will lie between $50 \pm 1.96 \times 2$, i.e. 46.08 GB to 53.92 GB.

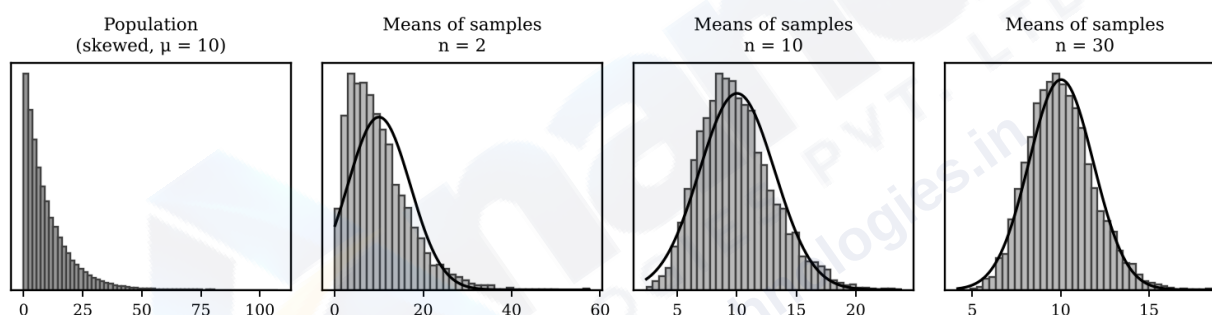


Fig.: Simulation — a skewed population; distribution of sample means becomes normal as n increases

Importance: allows the use of normal-distribution methods (confidence intervals, z-tests) on sample means for any population.

(Statement 2 M, example / diagram 2 M)

c) Sorting and filtering a PivotTable.

4 M

A) Sorting (descending by total amount):

1. Right-click any value in the *Grand Total* column of the PivotTable.
2. Choose **Sort** → **Sort Largest to Smallest**. (Or click the drop-down arrow of *Row Labels* → **More Sort Options** → Descending (Z to A) by *Sum of Amount* → OK.)
3. To sort labels alphabetically, use the *Row Labels* arrow → Sort A to Z / Z to A.

B) Filtering:

1. Click the drop-down arrow of *Row Labels* (or *Column Labels*).
2. Uncheck (*Select All*) and tick only the required items (e.g. Pen drive, Keyboard) → OK; **or** use **Label Filters** / **Value Filters** (e.g. Sum of Amount greater than 25000) / **Top 10**.

3. A field placed in the **Filters** area gives a report filter above the table; slicers can also be used.

4. To remove: click the filter icon → *Clear Filter From...*

Example (data of Q.6 b) after sorting products by Grand Total, largest first:

Sum of Amount / Product	Nagpur	Nashik	Pune	Grand Total
Keyboard	9,500	11,000	15,000	35,500
Pen drive	14,000	9,000	12,000	35,000
Mouse	7,000	6,500	12,000	25,500
Grand Total	30,500	26,500	39,000	96,000

(Sorting 2 M, filtering 2 M)

d) Chart from non-adjacent cells.

4 M

Non-adjacent cells are ranges that are not next to each other. To chart only some columns, select them with the **Ctrl** key.

Example data (A1:D5) — we want a column chart of Month (column A) and Profit (column D) only:

Month	Sales	Expenses	Profit
Jan	50	38	12
Feb	62	45	17
Mar	58	47	11
Apr	70	49	21

1. Select the first range A1:A5 (Month) by dragging.
2. Hold down the **Ctrl** key and drag to select D1:D5 (Profit). Both ranges are now highlighted; the selected ranges must be of the same size (same rows).
3. Click **Insert** → **Charts** → **Column** → **Clustered Column**.
4. Excel draws the chart with Month as categories and Profit as values; add chart title and axis titles.

(Alternatively, create the chart from any range and use *Chart Design* → *Select Data* to set the series and category ranges.)

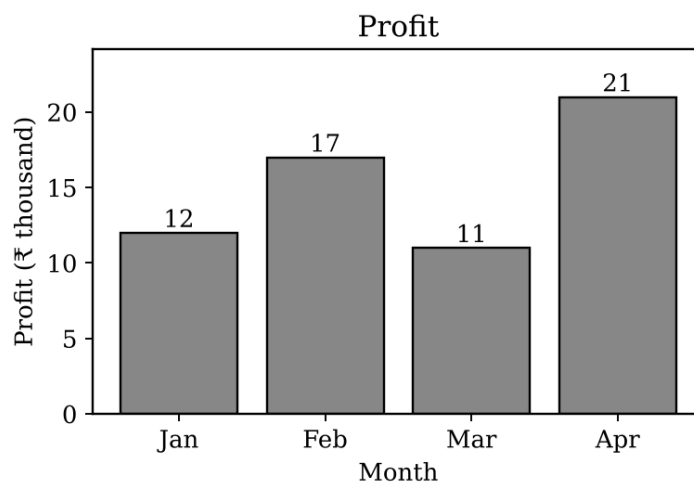


Fig.: Resulting chart of non-adjacent columns Month and Profit

(Explanation 1 M, steps 2 M, example 1 M)

Q.3

a) Predictive vs prescriptive analytics.

4 M

Predictive analytics	Prescriptive analytics
Answers 'What is likely to happen?'	Answers 'What should we do?' / 'How can we make it happen?'
Forecasts future outcomes from past data.	Recommends the best action among possible choices.
Techniques: regression, time-series forecasting, classification, machine learning.	Techniques: optimisation, simulation, decision analysis, rules engines.
Output is a probability / forecast value.	Output is a decision or action plan.
Less complex than prescriptive.	Most complex and highest-value type of analytics.
Example: predicting next month demand.	Example: deciding how much stock to order and when.

(Any four points, 1 M each)

b) Data imputation; handling missing values.

4 M

Imputation means replacing missing (blank / NaN) values in a dataset with substituted values estimated from the available data, so that the data can be analysed without losing records.

- Deletion:** remove the rows (listwise deletion) or columns having missing values — simple, but data is lost.
- Mean / median / mode imputation:** replace missing numeric values by the column mean (or median when outliers exist) and categorical values by the mode.
- Forward / backward fill:** use the previous (or next) value — useful in time-series data.
- Regression / KNN imputation:** predict the missing value from other variables or from the most similar records.

5. **Constant value:** fill with a fixed value such as 0 or 'Unknown'.

Example with pandas:

```
# Handling missing values (imputation) with pandas
import pandas as pd
import numpy as np

df = pd.DataFrame({'Name': ['Amit', 'Bina', 'Chetan', 'Deepa', 'Esha', 'Farhan'],
                  'Marks': [78, np.nan, 65, 82, np.nan, 70],
                  'City': ['Pune', 'Nashik', None, 'Pune', 'Pune', 'Nagpur']})
print('Original data:\n', df)
print('\nMissing values per column:\n', df.isnull().sum())

print('\n1) Drop rows with missing values:\n', df.dropna())

df2 = df.copy()
df2['Marks'] = df2['Marks'].fillna(df2['Marks'].mean()) # mean imputation
df2['City'] = df2['City'].fillna(df2['City'].mode()[0]) # mode imputation
print('\n2) Mean (Marks) and mode (City) imputation:\n', df2)

print('\n3) Forward fill:\n', df[['Marks']].ffill().T)
```

Output:

Original data:

	Name	Marks	City
0	Amit	78.0	Pune
1	Bina	NaN	Nashik
2	Chetan	65.0	NaN
3	Deepa	82.0	Pune
4	Esha	NaN	Pune
5	Farhan	70.0	Nagpur

Missing values per column:

```
Name      0
Marks     2
City      1
dtype: int64
```

1) Drop rows with missing values:

	Name	Marks	City
0	Amit	78.0	Pune
3	Deepa	82.0	Pune
5	Farhan	70.0	Nagpur

2) Mean (Marks) and mode (City) imputation:

	Name	Marks	City
0	Amit	78.00	Pune
1	Bina	73.75	Nashik
2	Chetan	65.00	Pune
3	Deepa	82.00	Pune
4	Esha	73.75	Pune
5	Farhan	70.00	Nagpur

3) Forward fill:

	0	1	2	3	4	5
Marks	78.0	78.0	65.0	82.0	82.0	70.0

(Definition 1 M, any three techniques 3 M)

c) Matplotlib; role; four plots.

4 M

Matplotlib is a free, open-source 2-D plotting library of Python, created by John D. Hunter (2003). Its **pyplot** module (`import matplotlib.pyplot as plt`) gives MATLAB-like functions to create figures.

Role in data visualization:

- Converts data (lists, NumPy arrays, pandas DataFrames) into charts that show trends, comparisons and distributions.
- Gives full control over figure size, colours, titles, labels, legends, axes and ticks.
- Saves plots in many formats (PNG, JPG, PDF, SVG) at the required resolution for reports.
- Base for other libraries such as Seaborn and pandas plotting; works in Jupyter notebooks and scripts.

Plots: line plot – `plot()`, bar chart – `bar()`, histogram – `hist()`, scatter plot – `scatter()`, pie chart – `pie()`, box plot – `boxplot()`.

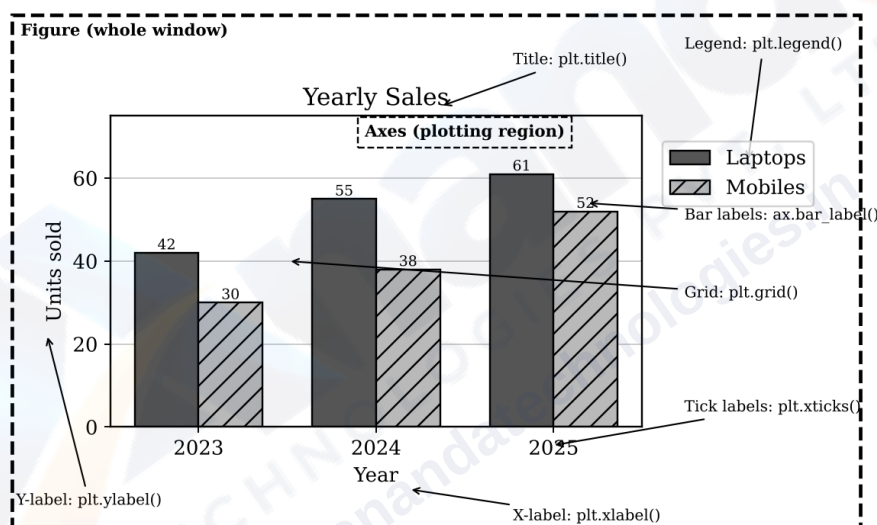


Fig.: Parts of a Matplotlib figure

(Description 1 M, role 2 M, four plots 1 M)

d) Coefficient of quartile deviation and of mean deviation.

4 M

Data in ascending order: 10, 14, 18, 20, 24, 30, 31; $n = 7$

i) Quartile deviation:

$$Q1 = 1(n + 1)/4 \text{ th value} = 1 \times 8/4 = 2\text{th value} = 14$$

$$Q3 = 3(n + 1)/4 \text{ th value} = 3 \times 8/4 = 6\text{th value} = 30$$

$$Q.D. = (Q3 - Q1)/2 = (30 - 14)/2 = 8$$

$$\text{Coefficient of Q.D.} = (Q3 - Q1)/(Q3 + Q1) = (30 - 14)/(30 + 14) = 16/44 = 0.3636$$

ii) Mean deviation about mean: $\bar{x} = 147/7 = 21$

x	10	14	18	20	24	30	31	Total
---	----	----	----	----	----	----	----	-------

$ x - \bar{x} $	11	7	3	1	3	9	10	44
-----------------	----	---	---	---	---	---	----	----

$$\text{M.D.} = \Sigma|x - \bar{x}| / n = 44/7 = 6.2857$$

$$\text{Coefficient of M.D.} = \text{M.D.} / \bar{x} = 6.2857/21 = \mathbf{0.2993}$$

(Coefficient of Q.D. 2 M, coefficient of M.D. 2 M)

Q.4

a) Spearman's rank correlation.

4 M

Rank 1 is given to the highest value in each series (no ties).

x	y	Rank R_1 (x)	Rank R_2 (y)	d = $R_1 - R_2$	d ²
65	70	5	4	1	1
72	68	3	5	-2	4
58	60	6	6	0	0
80	85	1	1	0	0
69	72	4	3	1	1
75	78	2	2	0	0
				$\Sigma d = 0$	$\Sigma d^2 = 6$

$$\rho = 1 - 6\Sigma d^2 / (n(n^2 - 1)) = 1 - (6 \times 6) / (6(6^2 - 1)) = 1 - 36/210 = \mathbf{0.8286}$$

Interpretation: $\rho = 0.83$ shows a high positive rank correlation between marks in Maths and Science.

(Ranks 2 M, formula and value 2 M)

b) Binomial probability.

4 M

Each student either passes (success) or fails, trials are independent and p is constant, so X = number of students passing follows a **binomial distribution**: $P(X = r) = {}^nC_r p^r q^{n-r}$, with $n = 5$, $p = 0.6$, $q = 1 - p = 0.4$.

$$\text{i) } P(X = 3) = {}^5C_3 (0.6)^3 (0.4)^2 = 10 \times 0.216 \times 0.16 = \mathbf{0.3456}$$

$$\text{ii) } P(X \geq 4) = P(4) + P(5) = {}^5C_4 (0.6)^4 (0.4) + {}^5C_5 (0.6)^5 = 5 \times 0.1296 \times 0.4 + 0.07776 = 0.2592 + 0.07776 = \mathbf{0.3370}$$

(Formula 1 M, (i) 1½ M, (ii) 1½ M)

c) Role of a dashboard; components.

4 M

Role:

- **Consolidation:** brings data and charts from many sheets/PivotTables into one screen, giving a single view of performance.
- **At-a-glance monitoring:** KPIs (sales, profit, targets) are visible immediately; conditional formatting highlights problems.
- **Interactivity:** slicers, timelines and drop-down controls let users filter all connected charts together.

- **Trend analysis and forecasting** through line charts, trendlines and sparklines; helps quick decisions.
- Auto-updates when data is refreshed, saving time in preparing reports.

Components: title, KPI cards, charts (Pivot Charts, column, line, pie), tables / data grids, slicers and timelines (dynamic filters), sparklines, conditional formatting.

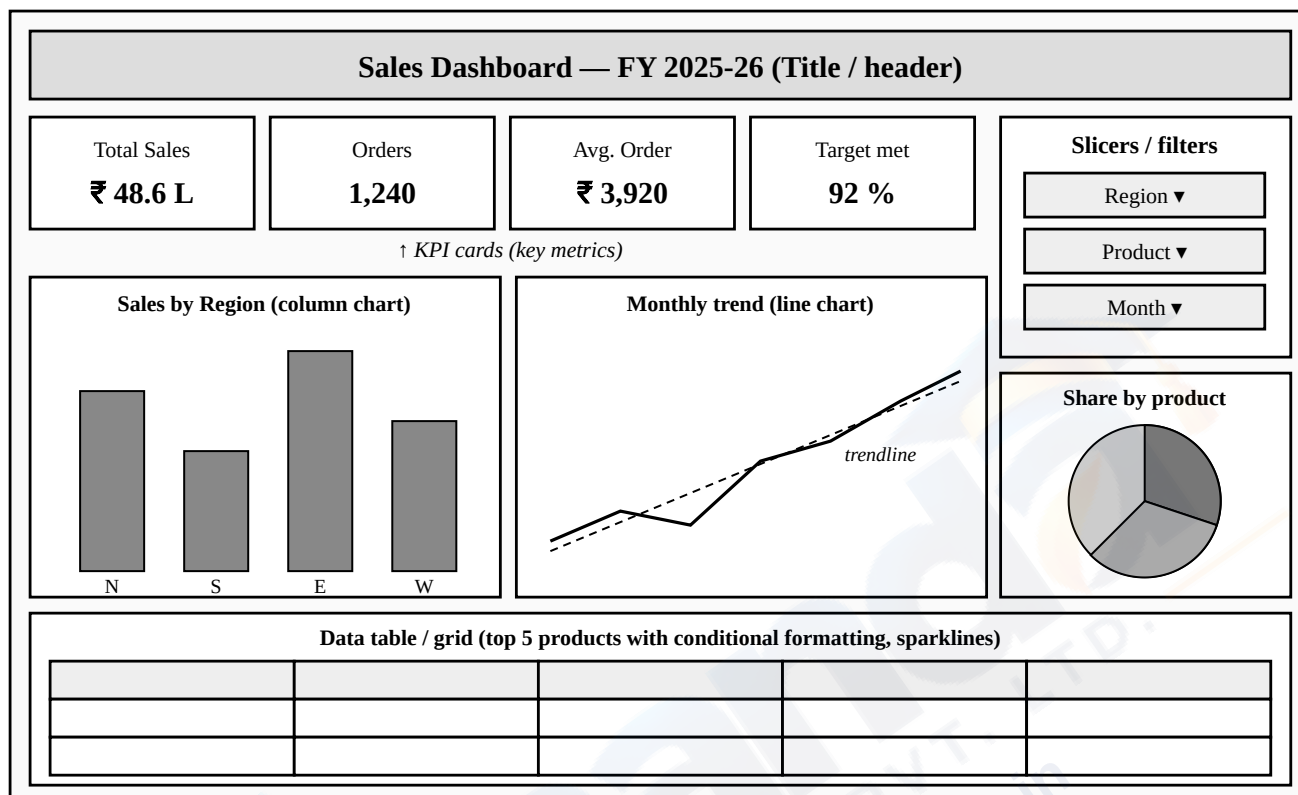


Fig.: Typical elements of an Excel dashboard (layout sketch)

(Role 2 M, components 2 M)

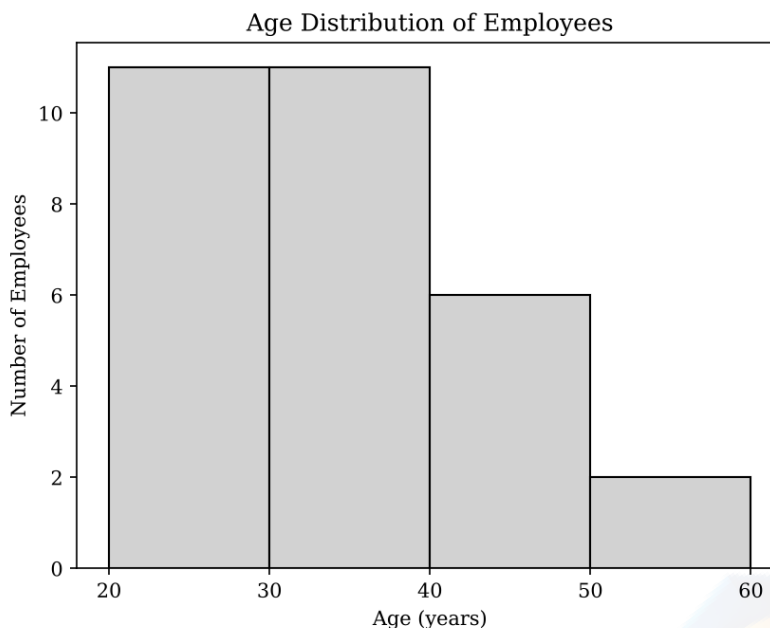
d) Python program — histogram of ages.

4 M

```
# Histogram of ages of 30 employees
import matplotlib.pyplot as plt

ages = [22, 25, 27, 29, 31, 24, 35, 38, 41, 45,
        23, 26, 28, 33, 36, 39, 42, 47, 52, 58,
        30, 32, 34, 37, 29, 26, 44, 49, 31, 27]

plt.hist(ages, bins=[20, 30, 40, 50, 60], color='lightgray', edgecolor='black')
plt.title('Age Distribution of Employees')
plt.xlabel('Age (years)')
plt.ylabel('Number of Employees')
plt.xticks([20, 30, 40, 50, 60])
plt.show()
```



Output window (plt.show())

`bins=[20, 30, 40, 50, 60]` gives the class intervals; each bar's height is the number of employees in that interval (last bin includes 60).

(Program 3 M, title and labels 1 M)

e) Moving an embedded chart to a chart sheet; sizing.

4 M

A) Move to a separate chart sheet:

1. Click the embedded chart to select it.
2. Click **Chart Design** → **Move Chart** (or right-click the chart border → *Move Chart...*).
3. In the Move Chart dialog select **New sheet**, type a name (e.g. 'Sales Chart') and click **OK**. The chart now fills its own chart sheet. (Choose *Object in* to move it back to a worksheet.)

B) Sizing an embedded chart:

1. Click the chart; eight **sizing handles** appear on its border (4 corners, 4 sides).
2. Point to a handle — the pointer becomes a double-headed arrow — and drag to make it larger or smaller. A corner handle changes height and width together; hold **Shift** to keep the proportion, **Alt** to snap to cell borders.
3. For exact size: **Format** → **Size group** → enter Height and Width (or Format Chart Area → Size & Properties).

(Moving 2 M, sizing 2 M)

Q.5

a) IQR method for outliers; box plot.

6 M

Ascending order: 18, 21, 22, 24, 25, 26, 27, 28, 30, 33, 45; $n = 11$

$Q1 = 1(n + 1)/4$ th value = $1 \times 12/4 = 3$ th value = **22**

Median $Q2 = 2(n + 1)/4$ th value = $2 \times 12/4 = 6$ th value = **26**

$Q3 = 3(n + 1)/4$ th value = $3 \times 12/4 = 9$ th value = **30**

$IQR = Q3 - Q1 = 30 - 22 = 8$

Lower fence = $Q1 - 1.5 \times IQR = 22 - 1.5 \times 8 = 10$

Upper fence = $Q3 + 1.5 \times IQR = 30 + 1.5 \times 8 = 42$

Values outside 10 to 42: **45** → outlier = 45.

ii) Box plot: box from $Q1 = 22$ to $Q3 = 30$, median line at 26, lower whisker to 18, upper whisker to 33 (largest value inside the fence), outlier 45 plotted as a separate point.

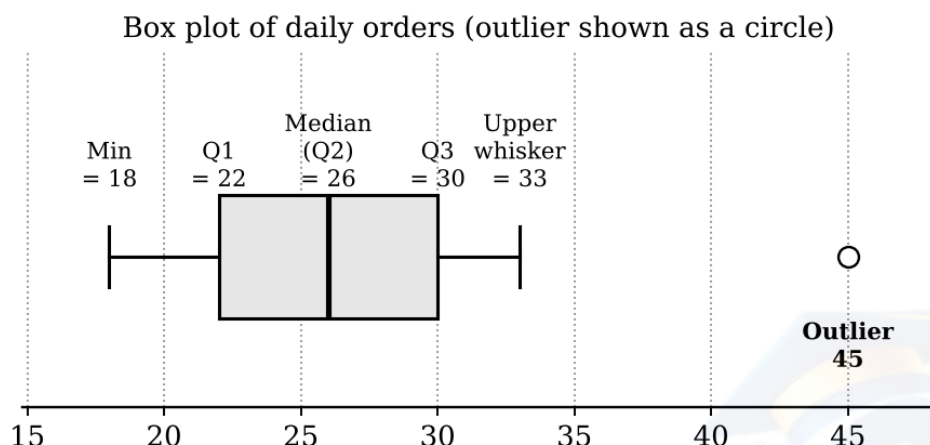


Fig.: Box plot with outlier

(Quartiles 2 M, IQR and fences 1 M, outlier 1 M, box plot 2 M)

b) Confidence interval; sampling variation; numerical.

6 M

Sampling variation: Different samples drawn from the same population give different values of a statistic (e.g. sample mean). This natural difference from sample to sample, caused only by chance of selection, is called sampling variation. Its size is measured by the **standard error** (SE of mean = $\sigma/\sqrt{n}$); it decreases as the sample size increases.

Confidence interval (CI): A range of values, calculated from a sample, which is expected to contain the unknown population parameter with a stated probability (confidence level such as 95%). For the mean with known σ : $CI = \bar{x} \pm z_{\alpha/2} \times \sigma/\sqrt{n}$ ($z = 1.96$ for 95%, 2.576 for 99%).

Numerical: $n = 36$, $\bar{x} = 64$, $\sigma = 12$. $SE = \sigma/\sqrt{n} = 12/\sqrt{36} = 12/6 = 2$

95% CI = $64 \pm 1.96 \times 2 = 64 \pm 3.92 = (60.08, 67.92)$

We are 95% confident that the population mean marks lie between 60.08 and 67.92.

(Sampling variation 1½ M, CI 1½ M, numerical 3 M)

c) Python program — pie chart, explode largest slice, save as SVG.

6 M

```
# Pie chart of monthly expenses; explode the largest slice and save as SVG
import matplotlib.pyplot as plt

heads = ['Rent', 'Food', 'Travel', 'Education', 'Savings']
amount = [8000, 6000, 2000, 3000, 1000]

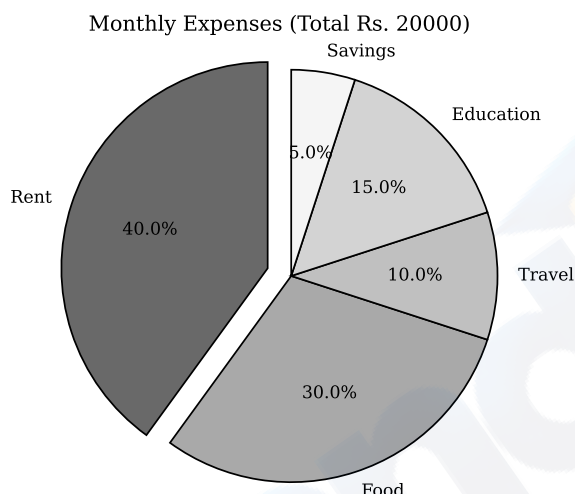
big = amount.index(max(amount)) # index of largest slice
explode = [0.12 if i == big else 0 for i in range(len(amount))]

plt.pie(amount, labels=heads, explode=explode, autopct='%1.1f%%',
        startangle=90, shadow=False,
        colors=['dimgray', 'darkgray', 'silver', 'lightgray', 'whitesmoke'],
```

```
wedgeprops={'edgecolor': 'black'})
plt.title('Monthly Expenses (Total Rs. %d)' % sum(amount))
plt.axis('equal') # keep pie circular
plt.savefig('expenses.svg')
print('Largest head :', heads[big])
print('Saved file : expenses.svg')
plt.show()
```

Output:

Largest head : Rent
Saved file : expenses.svg



Output: saved file expenses.svg

explode list gives the distance of each slice from centre (0.12 for the largest);
autopct='%1.1f%%' prints percentages; plt.axis('equal') keeps the pie round;
savefig('expenses.svg') saves in SVG (vector) format.

(Pie with labels/percent 3 M, explode 2 M, save 1 M)

Q.6

a) Correlation coefficient and regression line.

6 M

$$n = 6, \bar{x} = 42/6 = 7, \bar{y} = 342/6 = 57$$

x	y	dx = x - $\bar{x}$	dy = y - $\bar{y}$	dx ²	dy ²	dx·dy
3	40	-4	-17	16	289	68
5	48	-2	-9	4	81	18
6	55	-1	-2	1	4	2
8	60	1	3	1	9	3
9	66	2	9	4	81	18
11	73	4	16	16	256	64
$\Sigma x = 42$	$\Sigma y = 342$			42	720	173

$r = \frac{\sum dx \cdot dy}{\sqrt{(\sum dx^2 \times \sum dy^2)}} = \frac{173}{\sqrt{(42 \times 720)}} = \mathbf{0.9948} \rightarrow$ very high positive correlation.

$b_{yx} = \frac{\sum dx \cdot dy}{\sum dx^2} = \frac{173}{42} = 4.1190$

Regression line of y on x: $y - 57 = 4.1190(x - 7) \rightarrow y = \mathbf{28.1667 + 4.1190x}$

Estimate at $x = 10$: $y = 28.1667 + 4.1190 \times 10 = \mathbf{69.36}$ marks

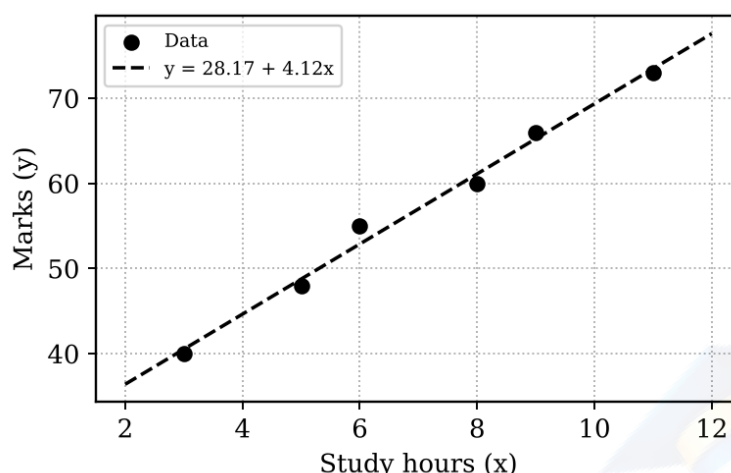


Fig.: Scatter diagram and regression line

(r 3 M, regression line 2 M, estimate 1 M)

b) Stepwise procedure — PivotTable of product-wise and region-wise amount.

6 M

1. Enter the data with headings Product, Region, Amount (say in A1:C11). Make sure there are no blank rows or columns.
2. Click any cell in the data. Choose **Insert** → **PivotTable** (Excel 365: *From Table/Range*).
3. In *Create PivotTable*, check range **Sheet1!\$A\$1:\$C\$11**; select **New Worksheet**; click **OK**.
4. In the **PivotTable Fields** pane, drag **Product** → **Rows**.
5. Drag **Region** → **Columns**.
6. Drag **Amount** → **Values**; it shows *Sum of Amount*. (Value Field Settings can change it to Average/Count and set number format.)
7. Grand totals for rows and columns appear automatically (Design → Grand Totals). Apply a PivotTable style and refresh after any change in data (PivotTable Analyze → Refresh).

PivotTable Fields

Choose fields to add to report:

☒ Product

☒ Region

☒ Amount

Drag fields between areas below:

▼ Filters

≡ Rows

Product

Columns

Region

Σ Values

Sum of Amount

Resulting PivotTable

Sum of Amount	Region ▼		
Product ▼	Nagpur	Nashik	Grand Total
Keyboard	...	...	...
Mouse	...	...	...
Grand Total	...	...	...

Rows area → row labels (Product)

Columns area → column labels (Region)

Values area → summarised numbers (Sum)

Fig.: PivotTable Fields pane — drag fields into the four areas

Resulting PivotTable (computed):

Sum of Amount / Product	Nagpur	Nashik	Pune	Grand Total
Keyboard	9,500	11,000	15,000	35,500
Mouse	7,000	6,500	12,000	25,500
Pen drive	14,000	9,000	12,000	35,000
Grand Total	30,500	26,500	39,000	96,000

(Steps 4 M, result 2 M)

c) Python program — bar chart with legend, ticks and limits.

6 M

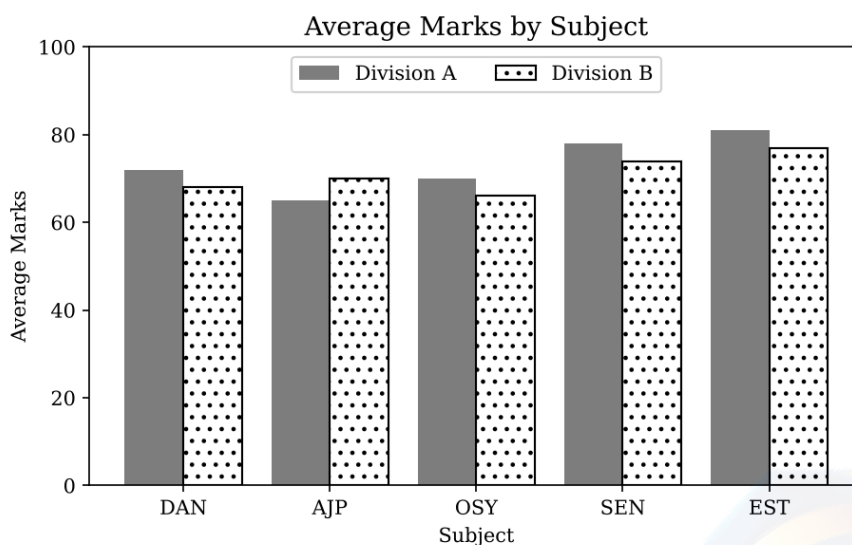
```
# Bar chart with title, labels, legend, custom ticks and axis limits
import matplotlib.pyplot as plt

subjects = ['DAN', 'AJP', 'OSY', 'SEN', 'EST']
div_a = [72, 65, 70, 78, 81]
div_b = [68, 70, 66, 74, 77]

pos = range(len(subjects))
plt.figure(figsize=(7, 4)) # width 7 in, height 4 in
plt.bar([p - 0.2 for p in pos], div_a, width=0.4, color='gray', label='Division A')
plt.bar([p + 0.2 for p in pos], div_b, width=0.4, color='white',
        edgecolor='black', hatch='..', label='Division B')

plt.title('Average Marks by Subject', fontsize=13)
plt.xlabel('Subject')
plt.ylabel('Average Marks')
plt.xticks(pos, subjects) # custom tick labels
plt.ylim(0, 100) # y-axis limits
plt.yticks(range(0, 101, 20)) # ticks at 0,20,...,100
```

```
plt.legend(loc='upper center', ncol=2) # legend in 2 columns at top
plt.show()
```



Output window (plt.show())

Functions used:

- `plt.figure(figsize=(7, 4))` — creates a figure 7 × 4 inches.
- `plt.bar(x, height, width, label=...)` — draws bars; two sets are shifted by ± 0.2 to stand side by side.
- `plt.title()`, `plt.xlabel()`, `plt.ylabel()` — chart title and axis labels.
- `plt.xticks(pos, subjects)` — places subject names as tick labels.
- `plt.ylim(0, 100)` and `plt.yticks(range(0, 101, 20))` — y-axis limits and tick positions.
- `plt.legend(loc='upper center', ncol=2)` — legend from the *label* of each series.
- `plt.show()` — displays the figure.

(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