Test_1_Resuelto

May 13, 2025

Creado por:

Isabel Maniega

1 Test 1

1.1 Question 1

```
Patricia is analyzing consumer trends in the fashion industry.
She needs to collect data on the types of clothing items that are currently popular.
Which method should Patricia use to gather this information?

a) Conducting in-deph interviews with fashion designers.
b) Analyzing sales data from major retail fashion stores.
c) Surveying consumers on their recent clothing purchases.
d) Reviewing fashion magazines and trend reports.
```

1.2 Solution 1

b

Analyzing sales data from major retail fashion stores Analyzing sales data from major retail fashion stores is the most effective method for Patricia to gather accurate information on currently popular clothing items. This method provides concrete data on consumer preferences and purchasing trends in the fashion industry.

1.3 Question 2

You're responsible for presenting the quarterly performance of your company's social media channels to a group of executives with limited technical knowledge. The key metrics include Total Followers, Engagement Rate, and Click-Through Rate. Here's the performance data for the year:

Quater	Total Followers	Engagement Rate (%)	Clicks-Through Rate (%)
Q1 (Jan-Mar)	20000	5.0	2.5
Q2 (Apr-Jun)	23500	6.0	3.0
Q3 (Jul-Sep)	27000	6.5	3.5
Q4 (Oct-Dec)	30000	7.0	3.8

How should you present this data to ensure clear understanding and actionable insights for your non-technical audience, especially focusing on the distribution and variability of the data? Select the best approach.

a) Use box plots to display the distribution and variability of Total Followers, engagement Rate, and Click-Through Rate for each quarter,paired with clear explanations to help the audience understand the central tendency, spread and outliers in the data.

b) Use a detailed spreadsheet format in the presentation that shows every metric's quarterly data, expecting stakeholders to draw conclusions on their own, demostrating the complexity of the data.

c) Create a complex infographic that combines all metrics using advanced visualizations and technical jargon, assuming the intricate design will impress and engage the executive.

d) Utilize simple line and bar graphs to display trends in Total Followers, Engagement Rate, and Click-Trough Rate, paired with brief explanations that provide context and highlight key takeaways, ensuring the data is easy to understand and actionable.

1.4 Solution 2

d

Simple line and bar graphs effectively communicate trends and comparisons over time, making it easier for a non-technical audience to grasp key insights. Pairing these visuals with concise narratives helps contextualize the data, making it more accessible and actionable for stakeholders.

1.5 Question 3

```
Sophia is tasked with combining quarterly financial data from different departments into a sing
To ensure data accuracy and consistency, what is the most important step Sophia should take?

a) Prioritize data from the department with the hihest revenue.
b) Standardize numerical formats and check for discrepancies in departmental data before mergi
c) Merge all data first and then address any discrepancies afterward.
d) Use the most recent quarter's data template for all department.
```

1.6 Solution 3

b

Standardize numerical formats and check for discrepancies in departmental data before merging. Standardizing numerical formats and checking for discrepancies before merging are crucial steps to ensure that the financial data from different departments is accurate and consistent when combined into a single report.

1.7 Question 4

You conducted a study and calculated a 95% confidence interval for the mean difference between two groups as (2, 8). What does this confidence interval indicate?

a) The true mean difference between the two groups is likely between 2 and 8.

b) The true mean difference the two groups is exactly 2.

c) There is a 95% chance that the true mean difference is between 2 and 8.

d) The study results are inconclusive about the mean difference between the two groups,

1.8 Solution 4

a

The true mean difference between the two groups is likely between 2 and 8. A 95% confidence interval indicates that we are 95% confident that the true parameter (mean difference in this case) falls within the interval (2, 8). It does not provide a probability or chance but rather a range within which the true parameter is likely to lie.

1.9 Question 5

```
Michael is studying the migration patterns of birds in a coastal area.
He needs to collect data on the number of different bird species present throughout the year.
What method should Michael employ to collect this data?

a) Reviewing literature on bird migration patterns.
b) Using autommated cameras to record bird movements.
c) Conducting monthly surveys through direct observation.
d) Analyzing satellite images of the area.
```

1.10 Solution 5

c

Conducting monthly surveys through direct observation Conducting monthly surveys through direct observation is the most effective method for Michael to collect accurate data on the number of different bird species present throughout the year. This method allows for detailed, species-specific observations over time.

1.11 Question 6

```
A dataset contains dates of customer transactions, and some dates are listed as '31st February

a) Incomplete data, as the year is not specified.
b) Irrelevant data, as the date may not be needed for analysis.
c) Erroneous data, due to an impossible date.
d) Duplicate data, if the same incorrect date appears multiple times.
```

1.12 Solution 6

c

Erroneous data, due to an impossible date Entries with dates like '31st February' are erroneous because they represent an impossible date (February never has 31 days). This indicates a fundamental issue with the data's validity.

1.13 Question 7

You are working with a Pandas DataFrame df containing a column named Scores with student scores ranging from 0 to 100. You want to normalize these scores to lie in the range of [0, 1]. Which of the following lines of code will accomplish this task correctly?

a) `df['Scores'] = df['Scores'] / df['Scores'].max()`

b) `df['Scores'] = df['Scores'] / 100`

c) `df['Scores'] = df['Scores'] - df['Scores'].min()) / df['Scores'].max()`

d) `df['Scores'] = df['Scores'].apply(lambda x : (x-df['Scores'].min()) / (df['Scores'].max() - df['Scores'].min()))`

1.14 Solution 7

d

This line of code correctly normalizes the scores in the "Scores" column of the DataFrame by applying a lambda function that subtracts the minimum score from each value and then divides by the range of scores (maximum score minus minimum score). This ensures that the normalized scores will fall within the range of [0, 1].

1.15 Question 8

```
In cleaning a dataset for a health study, a data analyst notices several instances where the 'a
How should these data points be treated?

a) As incomplete data, because the exact ages are not known.
b) As valid data, considering potential recording errors.
c) As erroneous data, given the unrealistic age values.
d) As duplicated data, if the same age appears multiple times.
```

1.16 Solution 8

c

As erroneous data, given the unrealistic age values. Ages over 150 years are biologically unrealistic and should be classified as erroneous data, indicating either a data entry mistake or a misinterpretation of the data field.

1.17 Question 9

When integrating time-series data from multiple sensors into a single dataset, what is essentia

a) Aligning all data entries to same time scale and format.
b) Focusing on the sensor with the highest frequency of data collection.
c) Summarizing the data from each sensor before merging to reduce complexity.
d) Choosing one sensor as the primary source and discarding data from others.

1.18 Solution 9

a

Aligning all data entries to the same time scale and format. Aligning data entries to the same time scale and format is crucial when integrating time-series data from multiple sources. This ensures consistency and accuracy, enabling coherent analysis across the combined dataset.

1.19 Question 10

Alex is integrating user interaction data from a website and a mobile app to analyze overall us

a) Aggregate all data points into a single average value for simplicity.
b) Validate and align interaction metrics from both sources to ensure they are on a similar.
c) Focus only on the platform with more user interactions for a biased analysis.
d) Manually enter website data to match the volume of app data.

1.20 Solution 10

b

Validate and align interaction metrics from both sources to ensure they are on a similar scale. Validating and aligning interaction metrics from both the website and mobile app are essential for Alex to ensure that the data is on a similar scale, making the integrated dataset reliable for analyzing overall user behavior.

1.21 Question 11

You have conducted a survey on customer satisfaction and created a visualization showing the ov

a) Avoid discussing the methodology and focus on the general trends.
b) Provide a high-level summary with key statistical metrics.
c) Present detailed analysis including survey methodology and statistical significance.
d) Use visual metaphors and analogies to explain the satisfaction levels.

1.22 Solution 11

c

Present detailed analysis including survey methodology and statistical significance. For a technical audience, presenting a detailed analysis including survey methodology, statistical significance, and key metrics helps communicate the insights effectively.

1.23 Question 12

```
You have analyzed sales data for a retail company and created a visualization showing the reve

a) Present complex statistical analysis with technical jargon.
b) Use clear and concise labels on the visualization to highlight key trends.
c) Provide raw data and detailed charts for audience exploration.
d) Use advanced machine learning algorithms to explain the revenue trends.
```

1.24 Solution 12

b

Use clear and concise labels on the visualization to highlight key trends. For a non-technical audience, using clear and concise labels on the visualization helps highlight key trends and insights without overwhelming them with technical jargon

---

1.25 Question 13

```
You are developing a Python script to process a large dataset and perform various data manipul

a) Using single-letter variable names for improved readability.
b) Writing long and complex functions to minimize the number of lines.
c) Adding descriptive comments to explain the purpose of each function and major code sections
d) Using global variables extensively to avoid passing arguments between functions.
```

1.26 Solution 13

c

Adding descriptive comments to explain the purpose of each function and major code sections Adding descriptive comments is a best practice for improving script maintainability as it helps other developers (or your future self) understand the code's purpose and functionality without needing to decipher each line of code.

1.27 Question 14

```
You have analyzed website traffic data and created a visualization showing the user engagement

a) Use technical terms and metrics without explanation.
b) Create a narrative around user behavior patterns and trends.
c) Provide raw data and detailed charts for audience exploration.
```

d) Focus on the aesthetics of the visualization rather than the insights.

1.28 Solution 14

b

Create a narrative around user behavior patterns and trends. For a non-technical audience, creating a narrative around user behavior patterns and trends helps contextualize the insights and make them more relatable. This approach allows the audience to understand the implications of the data in terms of user experience and engagement.

1.29 Question 15

You have analyzed a dataset containing customer purchase behavior and created a visualization s

a) Present detailed statistical analysis with technical terms.
b) Use storytelling and simple language to explain the trends and patterns.
c) Provide raw* data and complex charts for audience interpretation.
d) Focus only on the technical aspects without context or storytelling.

*raw: sin procesar

1.30 Solution 15

b

Use storytelling and simple language to explain the trends and patterns. For a non-technical audience, it's important to use storytelling techniques and simple language to convey the insights from visualizations effectively. This approach helps the audience understand the trends and patterns without overwhelming them with technical details.

1.31 Question 16

You are analyzing the performance of an email marketing campaign. Which of the following metri

a) Email Conversion Rate (Email Conversion Rate measures the perentage of recipients who take a
b) Time Spent on website.
c) Email Open Rate.
d) Bounce Rate (Bounce rate is a metric that reprensents the percentage of visitors who enter a

1.32 Solution 16

b

While it can indicate engagement, it doesn't directly reflect the effectiveness of the email marketing campaign in generating sales.

1.33 Question 17

```
In a data analysis project, you are aggregating data from various external web sources using Py

a) Limit data collection to a few sources for consistency.
b) Collect data in small batches and validate each batch.
c) Use threading to speed up data collection.
d) Verify data authenticity and integrity as you ingest it.
```

1.34 Solution 17

d

Collect data in small batches and validate each batch -> Partially correct. This method can help with managing data quality, but it doesn't focus on real-time validation during ingestion.

Verify data authenticity and integrity as you ingest it -> Correct. It ensures that any discrepancies or errors are identified and handled as the data is being collected.

1.35 Question 18

```
In the context of an ETL (Extract, Transform, Load) process, what is the primary purpose of the

a) Creating visualizations and reports for end-users.
b) Retrieving and reading data from multiple heterogeneous data sources.
c) Loading data into a data warehouse or database.
d) Performing data cleaning and preparation for analysis.
```

1.36 Solution 18

b

The 'Extract' phase is all about extracting data from various sources, which can include databases, APIs, flat files, etc. The key task is to efficiently retrieve data in its original format.

1.37 Question 19

```
When developing interactive web applications using Dash, how is the concept of 'Persistence' ut

a) To constantly update the application's data in real-time.
b) To secure the application against unauthorized access.
c) To maintain the database connection continuosly.
d) To remember the user's choices or data entries across multiple sessions.
```

1.38 Solution 19

d

In Dash applications, 'Persistence' refers to the ability of the application to remember the user's choices or data entries even after the application or browser is closed and reopened. This feature is essential for enhancing user experience by saving their preferences or the state of the application across sessions, without the need to start from scratch each time.

Other options, such as maintaining database connections, real-time data updates, or security measures, are important aspects of web application development but do not specifically describe the concept of 'Persistence' in Dash.

More about Dash:

Dash is an open-source web application framework for Python, created by Plotly. It is used primarily for building interactive web applications, especially those focused on data visualization and data-driven decision-making. Dash is designed to be simple and effective, allowing users to create applications with interactive graphs and visualizations using only Python code, without the need for extensive knowledge of HTML, CSS, or JavaScript.

Dash applications are composed of two parts: the layout and the callbacks. The layout defines the structure and appearance of the app and is usually composed of a set of components from Dash's core components and HTML components libraries. The callbacks provide interactivity to the application, defining the logic that connects the inputs (like button clicks, dropdown selections) with outputs (like charts, data tables).

1.39 Question 20

```
You are optimizing a Python script that processes a large dataset. You notice that certain func

a) Implement memoization to cache function results.
b) Rewrite the functions to perform parallel processing.
c) Increase the script's memory allocation for faster conputation.
d) Add more conditional statements to skip redundant computations.
```

1.40 Solution 20

a

Implement memoization to cache function results Implementing memoization allows you to cache function results based on input parameters, reducing redundant computations and improving script performance.

1.41 Question 21

```
You are analyzing sales data for a retail company, which includes daily sales figures for diffe

a) Summarizing
b) Filtering
c) Sorting
d) Grouping
```

1.42 Solution 21

d

Grouping Option D, Grouping, is the most appropriate data aggregation technique for calculating the total sales revenue for each product category across all stores. Grouping allows you to combine rows with similar attributes, in this case, product categories, and then apply an aggregation function (such as sum) to calculate the total sales revenue

---

1.43 Question 22

```
You are working on a data analysis script that involves multiple data processing steps. What is

a) Writing all code in a single massive script for simplicity.
b) Using short and cryptic function names to save space.
c) Breaking down the script into smaller functions with clear and descriptive names.
d) Embedding documentation within the script's code rather than using external documentation fi
```

1.44 Solution 22

c

Breaking down the script into smaller functions with clear and descriptive names Breaking down the script into smaller functions with clear and descriptive names is a recommended best practice as it improves readability, makes the code modular, and enhances maintainability by allowing easier debugging and modification of specific parts.

1.45 Question 23

```
You are debugging a Python script that is producing unexpected output. After thorough examinati

a) Add print statements to log variable values.
b) Comment out the suspected conditional statement.
c) Use the step-by-step debugger with breakpoint.
d) Rewrite the contitional statement using a different syntax.
```

1.46 Solution 23

c

Use the step-by-step debugger with breakpoints Using the step-by-step debugger with breakpoints allows you to trace the execution of the script line by line, inspect variable values, and identify the exact point where the unexpected output occurs.

1.47 Question 24

In a dataset containing customer purchase records, which data validation technique is most suitable for ensuring the accuracy of product prices?

a) Completeness validation

b) Range validation.

c) Consistency validation.

d) Cross-reference validation

1.48 Solution 24

b

Range validation Range validation checks if the values fall within expected ranges, making it suitable for verifying the accuracy of product prices.

1.49 Question 25

```
You are working on a Python script that is running slower than expected due to inefficient code


a) Move the loop outside the main function.
b) Use a generator expression instead of a list comprehension.
c) Cache the precomputed values in a dictionary.
d) Add more nested loops for a parallel procesing.
```

1.50 Solution 25

c

Cache the precomputed values in a dictionary Caching the precomputed values in a dictionary can significantly improve script performance by avoiding redundant calculations and storing results for reuse.

1.51 Question 26

You have a DataFrame named df with the following structure:

```
   ID  Name  Score  Subject
0   1   Tom     90  Math
1   2  Lisa     85  Math
2   1   Tom     92  History
3   2  Lisa     88  History
```

You want to reshape this DataFrame into a format that shows scores by Subject for each individual. Which of the following code snippets will achieve this?

a) `df.pivot(index='Name', columns='Subject', values='Score')`

b) `df.groupby(['Name', 'Subject']).Score.sum().unstack()`

c) `df.pivot_table(index='Name', columns='Subject', values='Score', aggfunc='mean')`

d) `df.set_index(['Name', 'Subject']).unstack()`

1.52 Solution 26

a

Using the pivot() function with the 'Name' as the index, 'Subject' as columns, and 'Score' as values will reshape the DataFrame to show scores by Subject for each individual. This method creates a new DataFrame with the specified structure, making it the correct choice for the given task.

1.53 Question 27

```
Sarah is developing a Python script for data analysis and visualization. She wants to ensure th

a) Using vague variable names to encourage code exploration.
b) Writing functions with long and convoluted logic to minimize the number of functions.
c) Dividing the script into smaller functions with clear names and specific responsibilities.
d) Avoiding names comments in the code to keep it concise.
```

1.54 Solution 27

c

Dividing the script into smaller functions with clear names and specific responsibilities Dividing the script into smaller functions with clear names and specific responsibilities is a recommended best practice for improving script maintainability. It helps in modularizing the code, making it easier to understand, debug, and modify

---

1.55 Question 28

```
You are debugging a Python script that is intended to calculate the average of a list of numbe

a) Add more test cases to cover a wider range of scenarios.
b) Print the list of numbers before and after the calculation.
c) Rewrite the calculation logic using a different algorithm.
d) Step through the script using a debugger and inspect variable values.
```

1.56 Solution 28

d

Step through the script using a debugger and inspect variable values Using a debugger allows you to step through the script, line by line, inspecting variable values and identifying any errors in the calculation logic that may be causing incorrect results

---

1.57 Question 29

You are working with a small dataset and want to assess your model's performance. Your colleague recommends using k-fold cross-validation. However, you are concerned about the reliability of the evaluation. What cross-validation technique would be most appropriate for your small dataset?

a) Shuffle-Split Cross-Validation

b) Time Series Cross-Validation

c) Leave-One-Out Cross-Validation

d) Stratified k-Fold Cross-Validation

1.58 Solution 29

c

Leave-One-Out Cross-Validation (LOOCV) is the most appropriate technique for small datasets as it involves training the model on all data points except one and then testing it on the left-out data point. This method provides a reliable evaluation of model performance without the need for a large amount of data.

1.59 Question 30

```
You are tasked with optimizing a Python script that processes a large dataset. During testing,

a) Add print statements to track memory usage.
b) Use a profiler to analyze memory usage patterns.
c) Comments out sections of code to isolate the issue.
d) Rewrite the script using a different programming paradigm.
```

1.60 Solution 30

b

Use a profiler to analyze memory usage patterns Using a profiler allows you to analyze memory usage patterns, identify areas of high memory consumption, and pinpoint the source of memory leaks for effective debugging and optimization

---

1.61 Question 31

```
Alex is tasked with optimizing a Python script for data processing. What coding practice should

a) Using global variables extensively to simplify data sharing between functions.
b) Creating functions with overly general names to accommodate multiple functionalities.
c) Implementing meaningful variable names and organizing code into logical sections.
```

d) Embedding documentation only in external files separate from the code.

1.62 Solution 31

c

Implementing meaningful variable names and organizing code into logical sections Implementing meaningful variable names and organizing code into logical sections is crucial for enhancing script maintainability. It improves readability, clarity, and ease of understanding for other developers working on the codebase.

1.63 Question 32

You are working with a dictionary in Python that stores information about students and their s

a) scores['John']
b) scores.get('John')
c) scores['scores']['John']
d) scores.get('John', 0)

1.64 Solution 32

a

Scores[“John”] In Python dictionaries, you can directly access a value associated with a key using square brackets notation (dict[key]), which is demonstrated in option A.

[1]:
```
scores = {'Anne': 25, 'Peter': 35, 'John': 50, 'Willy': 60}
scores['John']
```

[1]:
```
50
```

1.65 Question 33

You are debugging a Python script that is supposed to extract specific information from a JSON

a) Rewrite the entire script using a different programming paradigm for better error handling.
b) Utilize try-except blocks to catch and handle specific exceptions that occur during script
c) Increase the script's memory allocation to prevent runtime errors related to memory exhaust
d) Ignore the errors and proceed with running the script to see if it resolves itself.

1.66 Solution 33

b

Utilize try-except blocks to catch and handle specific exceptions that occur during script execution. Option B, utilizing try-except blocks, is the most effective approach for identifying and handling

errors that occur during script execution. Try-except blocks allow you to catch specific exceptions and implement error-handling logic.

[2]:
```
try:
    f = open('archivo_excepciones.json')  # El fichero no existe
except FileNotFoundError:
    print('¡El fichero no existe!')
else:
    print(f.read())
```

```
¡El fichero no existe!
```

1.67 Question 34

A data analyst is working with a dataset containing numerical values in the 'age' column. They

a) Regular expression matching to validate the age format.
b) Applying a lambda function to check for values outside the range.
c) Using the pd.cut() function to categorize ages into bins.
d) Using the pd.to_numeric() function with errors set to 'coerce' and then checking for NaN val

1.68 Solution 34

b

Applying a lambda function to check for values outside the range. A lambda function can be applied to the 'age' column to check if each value falls within the specified range of 18 to 65 years. This technique allows for precise validation of numerical data against a range.

1.69 Question 35

You conducted a correlation analysis between two variables and obtained a correlation coefficient of -0.75. What does this correlation coefficient value indicate about the relationship between the variables?

a) There is a strong positive correlation between the variables.

b) There is a moderate positive correlation between the variables.

c) There is a strong negative correlation between the variables.

d) There is no correlation between the variables.

1.70 Solution 35

c

There is a strong negative correlation between the variables. A correlation coefficient of -0.75 indicates a strong negative linear relationship between the two variables. As one variable increases, the other tends to decrease, and vice versa.

1.71 Question 36

You are tasked with validating a dataset containing ages of customers. Which data validation technique would be most appropriate to ensure the reliability and accuracy of the age data?

a) Completeness validation

b) Format validation.

c) Range validation.

d) Consistency validation

1.72 Solution 36

c

Range validation Range validation checks if values fall within expected ranges, such as valid age ranges, ensuring the reliability and accuracy of the collected age data

---

1.73 Question 37

You have employed 5-Fold Cross-Validation on a binary classification problem and received the following accuracy scores for each fold: [0.8, 0.85, 0.9, 0.7, 0.6]. What should be your next course of action?

a) Consider the model to be robust as the accuracy is above 50% for all folds.

b) Increase the number of folds to 10 for a more precise evaluation.

c) Investigate the cause of the variability in the accuracy scores across folds.

d) Pick the best model from the third fold, as it has the highest accuracy.

1.74 Solution 37

c

It is essential to investigate the cause of variability in accuracy scores across folds in 5-Fold Cross-Validation. Variability could indicate issues such as overfitting, data leakage, or model instability, which need to be addressed to ensure the reliability of the model.

1.75 Question 38

Suppose you have a DataFrame df with columns Name, Age, Gender, and Salary. Which of the following code snippets will filter the DataFrame to include only rows where Age is more than 25 and Salary is less than 50000, and also sort the resulting DataFrame by Name?

a) `df.filter('Age' > 25 & 'Salary' < 50000).sort_values(by='Name')`

b) `df[(df['Age'] > 25) & (df['Salary'] < 50000)].sort_values('Name')`

c) `df.sort_values('Name').where(df['Age'] > 25 & df['Salary'] < 50000)`

d) `df.query('Age > 25 and Salary < 50000').sort('Name')`

1.76 Solution 38

b

This code snippet correctly filters the DataFrame df to include only rows where Age is more than 25 and Salary is less than 50000 using boolean indexing with '&' operator. It then sorts the resulting DataFrame by the 'Name' column using the sort_values() function.

1.77 Question 39

Given the Python code snippet below, which utilizes Matplotlib to generate a plot, what type of chart will be produced?

```
import matplotlib.pyplot as plt

a = [2, 4, 6, 8, 10]
b = [1, 3, 5, 7, 9]

plt.plot(a, b, color='green', linestyle='solid', marker='s',
         markerfacecolor='yellow', mec='blue', linewidth=1.5, alpha=0.9,
         label='Custom Plot')
plt.xlabel('A Axis')
plt.ylabel('B Axis')
plt.title('Example Plot')
plt.legend(loc='lower left')
plt.show()
```

a) A scatter plot with square markers, green lines, and yellow marker faces.

b) A bar chart with green bars and yellow edges.

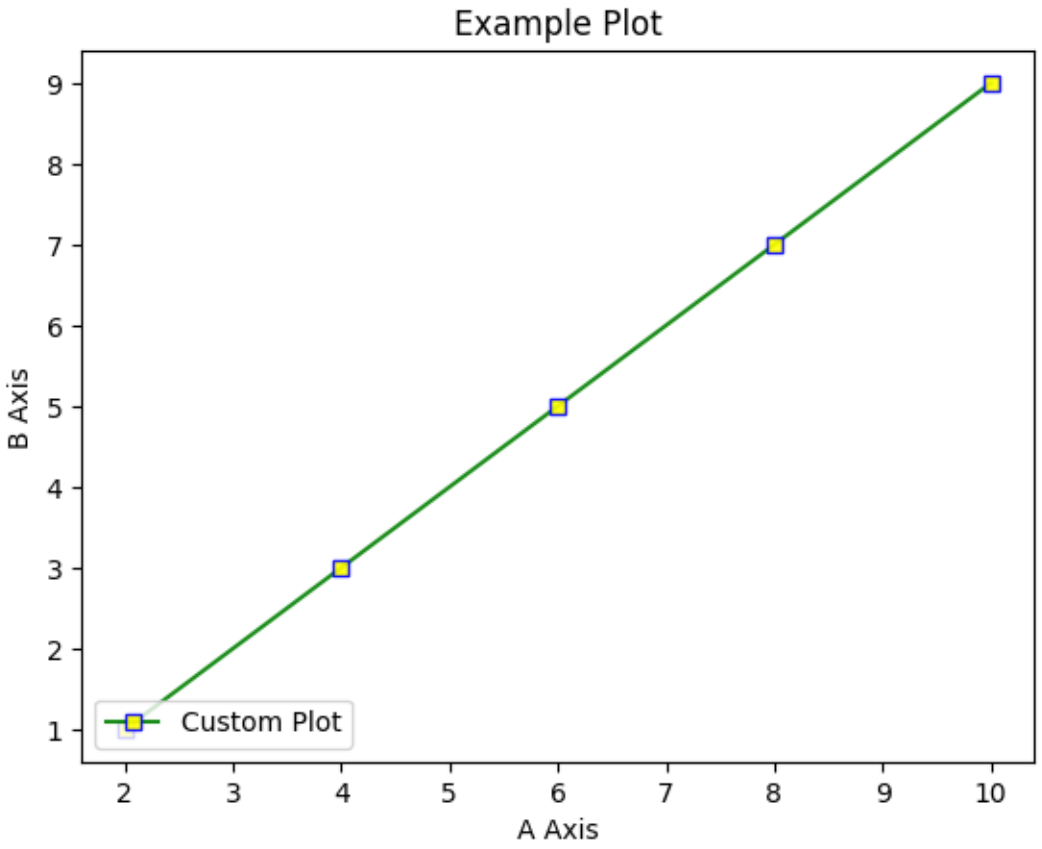
c) A line plot with square markers, green lines and yellow marker faces.

d) A pie chart with segment labled "Custom Plot".

1.78 Solution 39

[3]:
```
import matplotlib.pyplot as plt

a = [2, 4, 6, 8, 10]
b = [1, 3, 5, 7, 9]

plt.plot(a, b, color='green', linestyle='solid', marker='s',
         markerfacecolor='yellow', mec='blue', linewidth=1.5, alpha=0.9,
         label='Custom Plot')
```

```
plt.xlabel('A Axis')
plt.ylabel('B Axis')
plt.title('Example Plot')
plt.legend(loc='lower left')
plt.show()
```



c

The plt.plot() function in the provided code generates a line plot. The specified parameters (marker='s', color='green', markerfacecolor='yellow') customize the appearance of the plot, including square markers and line colors. The chart also includes a legend, axis labels, and a title.

1.79 Question 40

You have the following DataFrame containing sales data:

```
df = pd.DataFrame({
    'Month': ['Jan', 'Jan', 'Feb', 'Feb', 'Mar', 'Mar'],
    'Product': ['A', 'B', 'A', 'A', 'B', 'C'],
    'Sales': [100, 150, 200, 50, 300, 400]
```

```
})
```

You want to find the total sales for each month. Which code snippet will achieve this?

a) `df.groupby('Month').sum()`

b) `df.groupby('Product').sum('Sales')`

c) `df['Month'].agg({'Sales': 'sum'})`

d) `df.groupby('Month').agg({'Sales': 'sum'})`

1.80 Solution 40

[4]:
```
import pandas as pd

df = pd.DataFrame({
    'Month': ['Jan', 'Jan', 'Feb', 'Feb', 'Mar', 'Mar'],
    'Product': ['A', 'B', 'A', 'A', 'B', 'C'],
    'Sales': [100, 150, 200, 50, 300, 400]
})
```

[5]:
```
# a)

df.groupby('Month').sum()
```

[5]:
```
      Product  Sales
Month
Feb        AA    250
Jan        AB    250
Mar        BC    700
```

[6]:
```
# b)

df.groupby('Product').sum('Sales')
```

[6]:
```
         Sales
Product
A          350
B          450
C          400
```

[7]:
```
#c)
df['Month'].agg({'Sales': 'sum'})
```

[7]:
```
Sales    JanJanFebFebMarMar
Name: Month, dtype: object
```

[8]:
```
# d)
df.groupby('Month').agg({'Sales': 'sum'})
```

[8]:
```
        Sales
Month
Feb       250
Jan       250
Mar       700
```

d

This code snippet correctly groups the DataFrame by the 'Month' column and then calculates the sum of the 'Sales' column for each group, resulting in the total sales for each month.

1.81 Question 41

You have a DataFrame df with columns 'Quarter', 'Revenue', and 'Expenses'. You are tasked with calculating the quarterly profit margin as (Revenue - Expenses) / Revenue. Which of the following will correctly add a 'Profit Margin' column with these calculated values?

a) `df['Profit Margin'] = (df['Revenue'] - df['Expenses']) / df['Revenue']`

b) `df['Profit Margin'] = df['Revenue'] - df['Expenses'] / df['Revenue']`

c) `df['Profit Margin'] = df['Revenue'] / (df['Revenue'] - df['Expenses'])`

d)
```
df['Profit Margin'] = df.apply(lambda row: (row['Revenue'] -
row['Expenses']) / row['Revenue'], axis=1)
```

1.82 Solution 41

a

This choice correctly calculates the profit margin by subtracting 'Expenses' from 'Revenue' and then dividing the result by 'Revenue'. It assigns these calculated values to a new column 'Profit Margin' in the DataFrame.

1.83 Question 42

Your dataset contains monthly sales figures for different products over the past year. You want to visualize the sales trends for each product over time. Which type of visualization is most suitable for this task?

a) Line Plot

b) Box Plot

c) Pie Chart

d) Scatter Plot

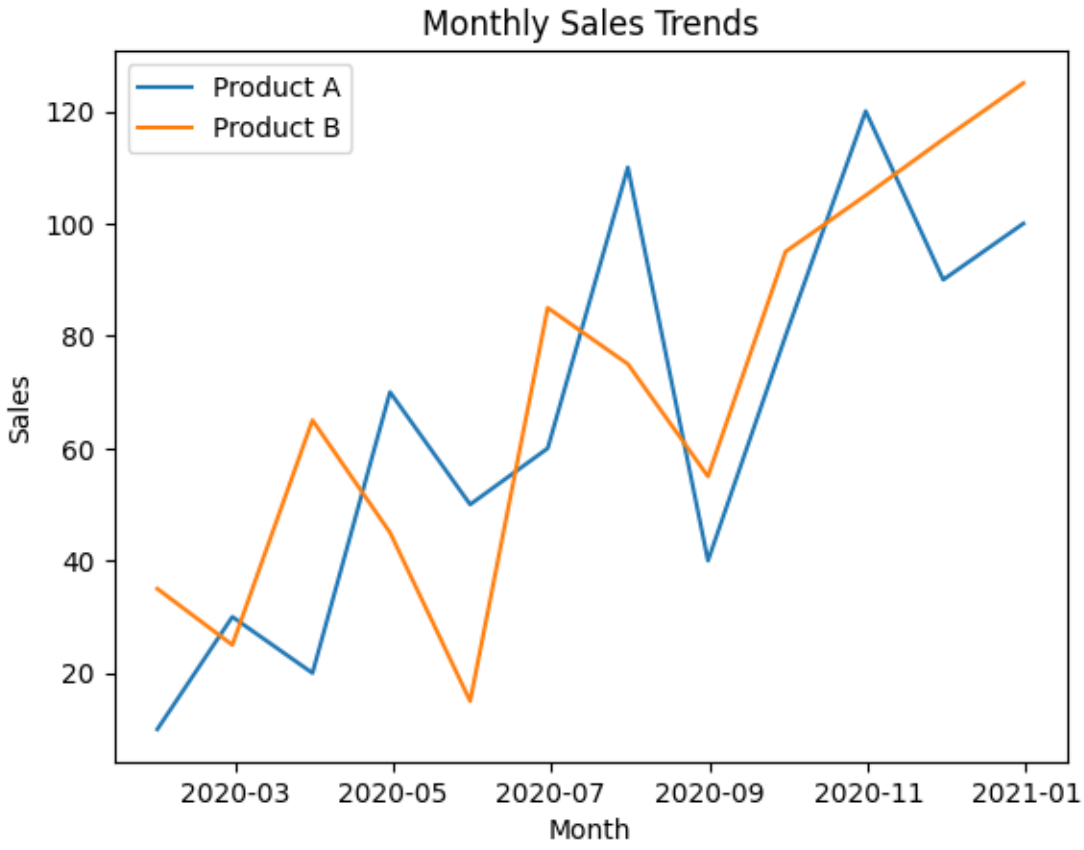
e) Bar chat

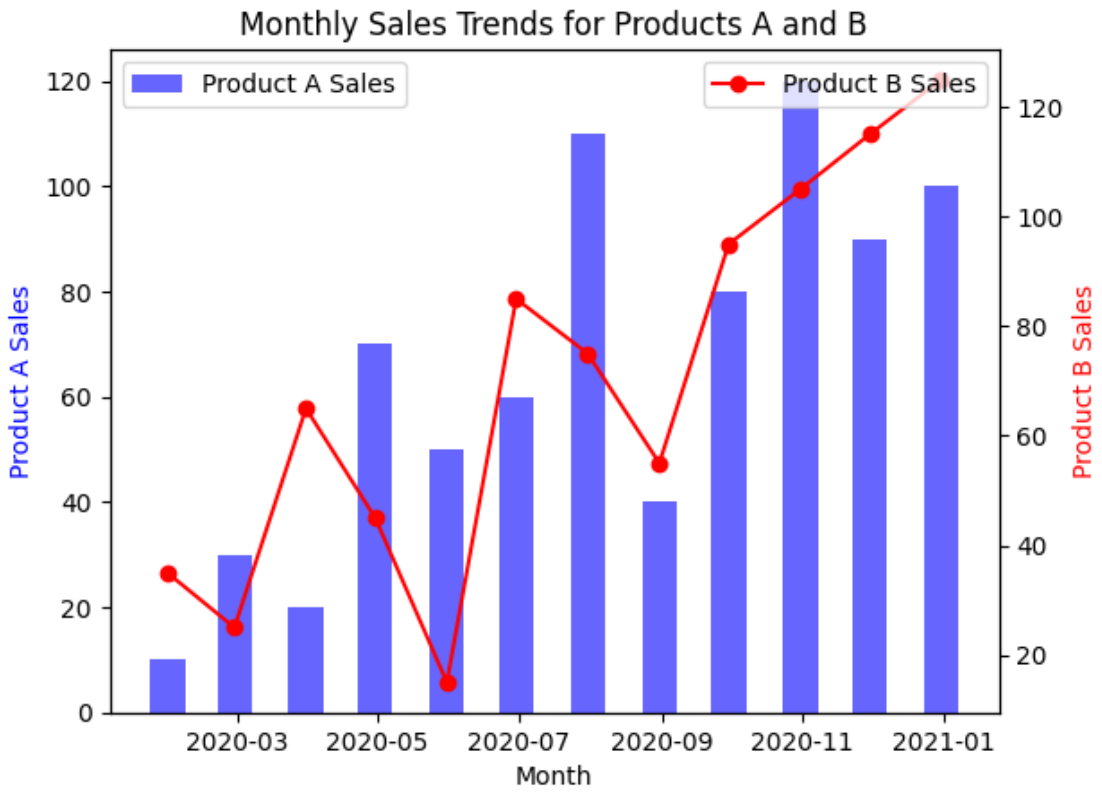
1.84 Solution 42

[9]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Sample data
data = {
    'Month': pd.date_range(start='2020-01-01', periods=12, freq='M'),
    'ProductA': [10, 30, 20, 70, 50, 60, 110, 40, 80, 120, 90, 100],
    'ProductB': [35, 25, 65, 45, 15, 85, 75, 55, 95, 105, 115, 125]
}
df = pd.DataFrame(data)

# Plotting the data
plt.plot(df['Month'], df['ProductA'], label='Product A')
plt.plot(df['Month'], df['ProductB'], label='Product B')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.title('Monthly Sales Trends')
plt.legend()
plt.show()

# Combined Bar and Line Chart
fig, ax1 = plt.subplots()

# Bar chart for ProductA
ax1.bar(df['Month'], df['ProductA'], width=15, label='Product A Sales',␣
 ↪color='blue', alpha=0.6)

# Line chart for ProductB on the same x-axis
ax2 = ax1.twinx()
ax2.plot(df['Month'], df['ProductB'], label='Product B Sales', color='red',␣
 ↪marker='o')

# Setting labels and title
ax1.set_xlabel('Month')
ax1.set_ylabel('Product A Sales', color='blue')
ax2.set_ylabel('Product B Sales', color='red')
plt.title('Monthly Sales Trends for Products A and B')

# Adding legends
ax1.legend(loc='upper left')
ax2.legend(loc='upper right')

# Display the plot
plt.show()
```

```
/tmp/ipykernel_6108/4176056615.py:6: FutureWarning: 'M' is deprecated and will
```

```
be removed in a future version, please use 'ME' instead.
  'Month': pd.date_range(start='2020-01-01', periods=12, freq='M'),
```





a

Ideal for displaying trends and changes over time for different products.

1.85 Question 43

You are analyzing a dataset containing daily temperature readings from multiple cities over a year. You want to visualize this data to compare the temperature trends across these cities. Which of the following visualization techniques would be most suitable for this purpose?

a) Barc Chart

b) Multiple Line Graphs on the same Plot

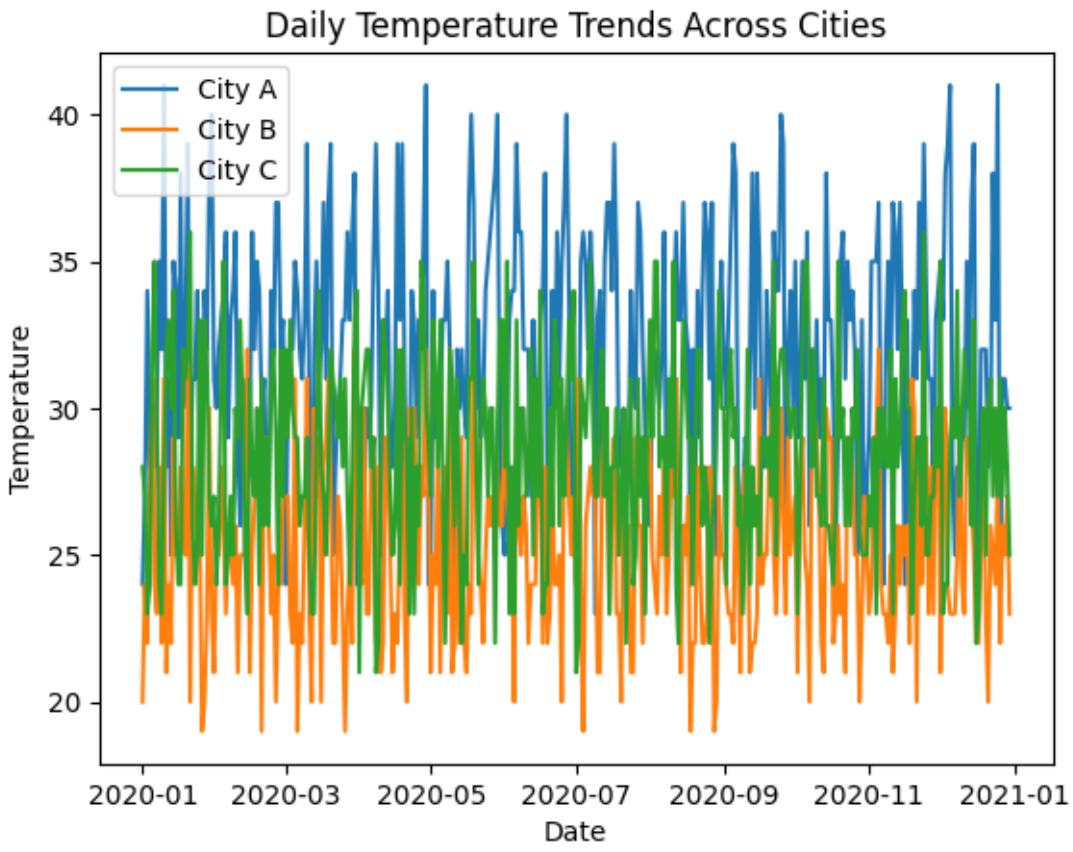
c) Scatter Plot

d) Pie Chart

e) Histogram

1.86 Solution 43

[10]:
```
import pandas as pd
import matplotlib.pyplot as plt
import random

# Sample data
data = {
    'Date': pd.date_range(start='2020-01-01', periods=365, freq='D'),
    'City_A': [22 + i % 10 + random.randint(1, 10) for i in range(365)],
    'City_B': [18 + i % 5 + random.randint(1, 10)for i in range(365)],
    'City_C': [20 + i % 7 + random.randint(1, 10)for i in range(365)]
}
df = pd.DataFrame(data)

# Plotting multiple line graphs
plt.plot(df['Date'], df['City_A'], label='City A')
plt.plot(df['Date'], df['City_B'], label='City B')
plt.plot(df['Date'], df['City_C'], label='City C')
plt.xlabel('Date')
plt.ylabel('Temperature')
plt.title('Daily Temperature Trends Across Cities')
plt.legend()
plt.show()
```



b

The best choice for comparing temperature trends across different cities over time. This method allows for clear visualization of the changes and comparisons over the same time period.

1.87 Question 44

In your climate study, you're examining the relationship between average yearly sunshine hours and average annual temperature in different countries. Additionally, you want to emphasize countries with the highest and lowest average yearly sunshine hours. What type of data visualization would be most suitable for this analysis?

a) Histogram

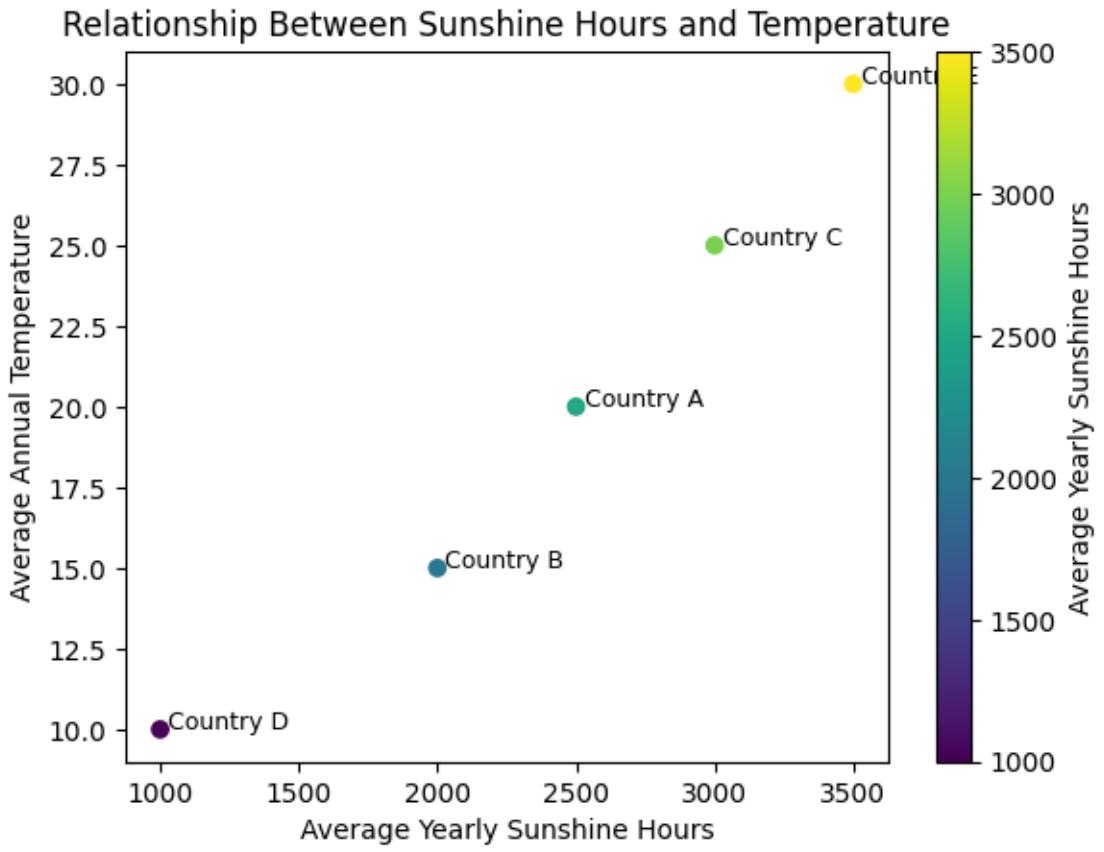
b) Scatter Plot with color Coding

c) Bubble Chart with Size Variation

d) Stacked Area Chart

1.88 Solution 44

[11]:
```python
import matplotlib.pyplot as plt

# Sample data
countries = ['Country A', 'Country B', 'Country C', 'Country D', 'Country E']
sunshine_hours = [2500, 2000, 3000, 1000, 3500]
average_temp = [20, 15, 25, 10, 30]

# Creating a scatter plot with color coding
plt.scatter(sunshine_hours, average_temp, c=sunshine_hours, cmap='viridis')
plt.colorbar(label='Average Yearly Sunshine Hours')

# Labeling
plt.xlabel('Average Yearly Sunshine Hours')
plt.ylabel('Average Annual Temperature')
plt.title('Relationship Between Sunshine Hours and Temperature')

# Annotating each country next to its point
for i, country in enumerate(countries):
    plt.text(sunshine_hours[i], average_temp[i], ' ' + country, fontsize=9)

plt.show()
```



b

A scatter plot is ideal for showing the relationship between two continuous variables (average yearly sunshine hours and average annual temperature). Using color coding can effectively highlight countries with extreme sunshine hours.

1.89 Question 45

In the context of an ETL (Extract, Transform, Load) process, what is the primary purpose of the 'Extract' phase?

a) Loading data into a data warehouse or database

b) Retrieving and reading data from multiple heterogeneous data sources

c) Creating visualizations and reports for end-users.

d) Performing data cleaning and preparation for analysis.

1.90 Solution 45

b

The 'Extract' phase is all about extracting data from various sources, which can include databases, APIs, flat files, etc. The key task is to efficiently retrieve data in its original format.

Creado por:

Isabel Maniega