Test_4_Resuelto

May 13, 2025

Creado por:

Isabel Maniega

1 Test 4

1.1 Question 1

```
Emma is cleaning a sales dataset and notices that the ‘sale_amount’ column has several entries

a) As valid data, because 'confidential' indicates a special data category.
b) As missing data, since 'confidential' does not provide a numerical value.
c) As duplicate data, assuming 'confidential' is a repeated placehoder.
d) As categorical data, because 'confidential' implies a different data grouping.
```

1.2 Solution 1

b

As missing data, since ‘confidential’ does not provide a numerical value Entries labeled as ‘confidential’ in the ‘sale_amount’ column, which should contain numerical values, should be treated as missing data. This is because ‘confidential’ does not represent a quantifiable amount and indicates the absence of disclosed numerical data.

1.3 Question 2

You have a list of strings and need to concatenate them into a single string with spaces between each element. Which code snippet achieves this task?

```
strings_list = ["hello", "world", "python"]
```

a) `"".join(strings_list)`

b) `"".concatenate(strings_list)`

c) `strings_list.join("")`

d) `strings_list.concatenate("")`

1.4 Solution 2

[1]:
```
strings_list = ["hello", "world", "python"]
```

[2]:
```
# a
"".join(strings_list)
```

[2]:
```
'helloworldpython'
```

[3]:
```
# b
"".concatenate(strings_list)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[3], line 2
      1 # b
----> 2 "".concatenate(strings_list)

AttributeError: 'str' object has no attribute 'concatenate'
```

[ ]:
```
# c
strings_list.join("")
```

[ ]:
```
# d
strings_list.concatenate("")
```

a

Option A correctly uses the join() method to concatenate the strings in strings_list with spaces between each element, resulting in a single string.

1.5 Question 3

You are creating a presentation and need to include a plot that shows the correlation between two numerical variables. You want to make sure the plot looks professional and can be easily understood when projected on a screen. What should you consider for this plot?

a) Use a 3D scatter plot with rainbow color map.

b) Use a 2D scatter plot with a single, high-contrast color and large markers.

c) Use a pie chart to represent each variable, side by side.

d) Use a line plot with highly stylized, decorative fonts.

1.6 Solution 3

b

Using a 2D scatter plot is the most appropriate choice for showing the correlation between two numerical variables. It allows for a clear visualization of the relationship between the variables, and using a single, high-contrast color with large markers will make the plot easily understandable, especially when projected on a screen.

1.7 Question 4

```
You need to create a dashboard to monitor key performance indicators (KPIs) for a marketing ca

a) Microsoft Power BI
b) Google Data Studio.
c) Tableau.
d) Qlik Sense
```

1.8 Solution 4

a

Microsoft Power BI Microsoft Power BI offers robust capabilities for integrating real-time data sources and automating data refreshes, making it suitable for monitoring KPIs with up-to-date insights in a dashboard.

1.9 Question 5

```
During the data preparation stage, Mike finds that several entries in the 'email' column of a

a) As complete data, because the email column is populated.
b) As valid data, assuming the '@' symbol was mistakenly omitted.
c) As missing data, since they do not represent valid email addresses.
d) As incorrect data, because the fail to meet the standard email format.
```

1.10 Solution 5

d

As incorrect data, because they fail to meet the standard email format Entries in the 'email' column that lack an '@' symbol should be classified as incorrect data during data cleaning, as they do not conform to the standard format for email addresses and likely represent data entry errors.

1.11 Question 6

You are analyzing a dataset containing customer ages, and you notice that some entries in the "Age" column are negative values. What would you consider as a common data quality issue in this scenario?

a) Entries with missing values in the "Age" column.

b) Entries with positive values in the “Age” column.

c) Entries with inconsistent formatting in the “Age” column.

d) Entries with incorrect values in the “Age” column

1.12 Solution 6

d

Entries with incorrect values in the “Age” column In this scenario, entries with negative values in the “Age” column are common data quality issues. Negative ages are invalid and likely indicate errors or anomalies in the dataset

---

1.13 Question 7

```
Rachel is analyzing a sales dataset and notices that some entries under the ‘sales_amount’ colu

a) Treat 'confidential' as a high-value sale to highlight potential maximum sales.
b) Replace 'confidential' with the average of the 'sales_amount' column.
c) Remove all records with 'confidential' sales amounts.
d) Consult with the data provider to understand the meaning of 'confidential' and decide on the
```

1.14 Solution 7

d

Consult with the data provider to understand the meaning of ‘confidential’ and decide on the most appropriate action. Consulting with the data provider is the best approach to understand why ‘confidential’ was used and determine the correct way to handle these entries, ensuring data integrity and accuracy.

1.15 Question 8

```
A research team is studying the dietary habits of teenagers in a particular region to identify

a) Direct observation in schools.
b) Online food diaries.
c) National heath surveys.
d) Public health records.
```

1.16 Solution 8

b

Online food diaries Online food diaries allow participants to record their daily dietary intake in real-time, providing detailed and personalized data over the extended period necessary for the study

---

1.17 Question 9

```
Samantha is cleaning a healthcare dataset and finds that the 'blood_pressure' column has some u

a) Leave the values as is, considering them as outliers that can be ignored.
b) replace these unrealistic values with the column average.
c) Mark these values as missing and handle them as part of the dataset's missing data.
d) Verify the values with healthcare professionals and correct them based on clinical advice.
```

1.18 Solution 9

d

Verify the values with healthcare professionals and correct them based on clinical advice. Verifying the unusually high values with healthcare professionals ensures that any corrections made are clinically sound and appropriate, maintaining the dataset's reliability and accuracy.

1.19 Question 10

Which one of the following methods returns the root element in the *xml.etree.ElementTree* module?

a) root

b) getroot

c) getparent

d) parent

1.20 Solution 10

b

1.21 Question 11

```
You are tasked with preparing a dataset for a machine learning model. The dataset contains miss

a) Remove rows with missing values.
b) Replace missing values with the mean of the column.
c) Use the most frequent value to replace missing values.
d) Apply linear regression to predict missing values.
```

1.22 Solution 11

c

Use the most frequent value to replace missing values Using the most frequent value (mode) to replace missing values is a common preprocessing technique that helps retain data integrity and minimize information loss.

1.23 Question 12

You are developing a dashboard for a retail company to track inventory and supply chain perfor

a) Inventory turnover ratio, stockout rate, lead time.
b) Customer retention rate, sales revenue, market share.
c) Employee productivity, training hours, satisfaction survey results.
d) Website traffic, social media followers, online sales conversion rate.

1.24 Solution 12

a

Inventory turnover ratio, stockout rate, lead time Option A includes metrics directly related to inventory management and supply chain performance, providing insights into inventory turnover, stockout incidents, and lead time efficiency.

1.25 Question 13

You are designing a dashboard for a transportation company to monitor fleet performance and lo

a) Vehicle utilization rate, on.time delivery rate, fuel efficiency.
b) Customer acquisition cost, revenue growth, market share.
c) Employee turnover rate, training completion rate, job satifaction score.
d) Website traffic, social media engagement, onlie sales revenue.

1.26 Solution 13

a

Vehicle utilization rate, on-time delivery rate, fuel efficiency Option A includes metrics directly related to fleet operations and logistics efficiency, providing insights into vehicle utilization, delivery punctuality, and fuel consumption efficiency

---

1.27 Question 14

A data analyst needs to source financial data for multiple companies over the last decade to p

a) Conducting interviews with indsutry experts.
b) Accessing a financial market database through an API.
c) Collecting data from printed financial magazines.
d) Using social media analytics

1.28 Solution 14

b

Accessing a financial market database through an API Accessing a financial market database through an API provides a direct, reliable, and efficient method to collect historical financial data for multiple companies, which is essential for conducting market trend analysis.

1.29 Question 15

In the csv module, saving data to a csv file is possible using:

(Select all that apply.)

a) The `DictWriter` class

b) The `write_csv` function

c) the `writer` function

d) the `Csv_writer` class

1.30 Solution 15

[ ]:
```
# a)
import csv

with open('test_exam.csv', 'w', newline='') as csvfile:
    fieldnames = ['Name', 'Phone']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

    writer.writeheader()
    writer.writerow({'Name': 'mother', 'Phone': '222-555-101'})
    writer.writerow({'Name': 'father', 'Phone': '222-555-102'})
    writer.writerow({'Name': 'wife', 'Phone': '222-555-103'})
    writer.writerow({'Name': 'mother-in-law', 'Phone': '222-555-104'})
    writer.writerow({'Name': 'grandmother, grandfather and auntie', 'Phone':␣
 ↪'222-555-105'})
```

[ ]:
```
# b)
import csv

with open('test_exam.csv', 'w', newline='') as csvfile:
    writer = csv.write_csv(csvfile, delimiter=',', quotechar='"', quoting=csv.
 ↪QUOTE_MINIMAL)

    writer.writerow(['Name', 'Phone'])
    writer.writerow(['mother', '222-555-101'])
    writer.writerow(['father', '222-555-102'])
    writer.writerow(['wife', '222-555-103'])
```

```
    writer.writerow(['mother-in-law', '222-555-104'])
    writer.writerow(['grandmother, grandfather', '222-555-105'])
```

[ ]:
```
# c)
import csv

with open('test_exam.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile, delimiter=',', quotechar='"', quoting=csv.
 ↪QUOTE_MINIMAL)

    writer.writerow(['Name', 'Phone'])
    writer.writerow(['mother', '222-555-101'])
    writer.writerow(['father', '222-555-102'])
    writer.writerow(['wife', '222-555-103'])
    writer.writerow(['mother-in-law', '222-555-104'])
    writer.writerow(['grandmother, grandfather', '222-555-105'])
```

[ ]:
```
# d)
import csv

with open('test_exam.csv', 'w', newline='') as csvfile:
    writer = csv.Csv_writer(csvfile, delimiter=',', quotechar='"', quoting=csv.
 ↪QUOTE_MINIMAL)

    writer.writerow(['Name', 'Phone'])
    writer.writerow(['mother', '222-555-101'])
    writer.writerow(['father', '222-555-102'])
    writer.writerow(['wife', '222-555-103'])
    writer.writerow(['mother-in-law', '222-555-104'])
    writer.writerow(['grandmother, grandfather', '222-555-105'])
```

a, c

1.31 Question 16

Which of the following methods exist in the `DictWriter` object provided by the `csv` module?

(Select all that apply.)

a) writeseparator

b) writerow

c) writeheader

d) write

1.32 Solution 16

b, c

1.33 Question 17

Select the XML modules available in the Python standard library:

(Select all that apply)

a) xml.parser

b) xml.etree.ElementTree

c) xml.processor

d) xml.dom.minidom

1.34 Solution 17

b, d

1.35 Question 18

In the csv module, reading data from a csv file is possible using:

(Select all that apply)

a) the `reader` function

b) the `CsvReader` class

c) the `DictReader` class

d) the `read_csv` function

1.36 Solution 18

a, c

1.37 Question 19

You have a dataset containing product sales information, including the product name, quantity sold, and price per unit. You need to calculate the total revenue generated from sales. Which of the following Python code snippets accomplishes this task?

a) `total_revenue = sum(product['quantity_sold'] * product['price_per_unit'] for product in products)`

b) `total_revenue = sum(product['quantity_sold'] * product['price_per_unit'] for product in products) / len(products)`

c) `total_revenue = sum(product['quantity_sold'] * product['price_per_unit'] for product in products) * len(products)`

d) `total_revenue = sum(product['quantity_sold'] * product['price_per_unit'] / len(products) for product in products)`

1.38 Solution 19

a

Option A correctly calculates the total revenue generated from sales by multiplying the quantity sold by the price per unit for each product and then summing these values.

1.39 Question 20

You are working with a Python dictionary representing student grades. Which code snippet correctly accesses the grade for the student with ID "123"?

```
student_grades = {"123": 85, "456": 92, "789": 78}
```

a) `student_grades[1]`

b) `student_grades.get("123")`

c) `student_grades["123"]`

d) `student_grades.get(123)`

1.40 Solution 20

[ ]:
```
student_grades = {"123": 85, "456": 92, "789": 78}
```

[ ]:
```
# a
student_grades[1]
```

[ ]:
```
# b
student_grades.get("123")
```

[ ]:
```
# c
student_grades["123"]
```

[ ]:
```
# d
student_grades.get(123)
```

c

Accesses the value associated with the key "123" in the dictionary using square brackets ([]), which is the standard syntax for accessing dictionary values by key

---

1.41 Question 21

You have a dictionary containing student names and their corresponding scores. You want to find the student with the highest score. Which code snippet accomplishes this task?

```
scores_dict = {"Alice": 85, "Bob": 92, "Charlie": 78}
```

a) `max(scores_dict, key=scores_dict.get)`

b) `max(scores_dict.values())`

c) `max(scores_dict.keys())`

d) `max(scores_dict.items(), key=lambda x: x[1])`

1.42 Solution 21

[ ]:
```
scores_dict = {"Alice": 85, "Bob": 92, "Charlie": 78}
```

[ ]:
```
# a
max(scores_dict, key=scores_dict.get)
```

[ ]:
```
# b
max(scores_dict.values())
```

[ ]:
```
# c
max(scores_dict.keys())
```

[ ]:
```
# d
max(scores_dict.items(), key=lambda x: x[1])
```

a

Option A correctly uses the max() function with the key argument to find the key (student name) corresponding to the highest value (score) in the dictionary.

1.43 Question 22

You want to create a Python function that takes two arguments, a list of numbers and a target value, and returns True if the target value is present in the list and False otherwise. Which code snippet defines this function correctly?

a)

```
def check_target_value(lst, target):
    return target in lst
```

b)

```
def check_target_value(lst, target):
    if target in lst:
        return True
    else:
```

```
        return False
```

c)

```
def check_target_value(lst, target):
    if target == lst:
        return True
    else:
        return False
```

d)

```
def check_target_value(lst, target):
    for num in lst:
        if num == target:
            return True
        return False
```

1.44 Solution 22

[ ]:
```
lst = [12, 24, 25, 40, 65, 48]
```

[ ]:
```
# a
def check_target_value(lst, target):
    return target in lst

print(check_target_value(lst, 40))
print(check_target_value(lst, 20))
```

[ ]:
```
# b
def check_target_value(lst, target):
    if target in lst:
        return True
    else:
        return False

print(check_target_value(lst, 40))
print(check_target_value(lst, 20))
```

[ ]:
```
# c
def check_target_value(lst, target):
    if target == lst:
        return True
    else:
        return False

print(check_target_value(lst, 40))
print(check_target_value(lst, 20))
```

[ ]:
```python
# d
def check_target_value(lst, target):
    for num in lst:
        if num == target:
            return True
        return False

print(check_target_value(lst, 40))
print(check_target_value(lst, 20))
```

a

Code snippet A defines the function check_target_value correctly using the in operator to check if the target value is present in the list and returns True or False accordingly.

1.45 Question 23

You have a Python script that processes a large dataset and performs multiple calculations. However, you notice that the script is running slower than expected, especially when handling large files. What is a recommended approach for optimizing the script's performance?

a) Add more comments and documentation to the script for better readability.

b) Use built-in Python functions and libraries optimazed for handling large dataset.

c) Break down complex calculations into smaller, manageable functions for better performance.

d) Increase the size of the dataset to test the script's scalability.

1.46 Solution 23

b

Use built-in Python functions and libraries optimized for handling large datasets. Option B, using built-in Python functions and libraries optimized for handling large datasets, is a recommended approach for optimizing script performance. Utilizing libraries like Pandas for data manipulation and NumPy for numerical computations can significantly improve performance.

1.47 Question 24

You are working on a Python script to parse and extract information from multiple JSON files in a directory. The script needs to be easily maintainable and scalable for handling new JSON files in the future. What is a recommended best practice for achieving these objectives?

a) Write a single monolithic script that processes all JSON files in the directory.

b) Use modular programming by diciding the script in to functions and modules for different tasks.

c) Hard-code file paths and names in the script for direct file access.

d) Avoid using error handling mechanisms to keep the script concise.

1.48 Solution 24

b

Use modular programming by dividing the script into functions and modules for different tasks. Option B, using modular programming by dividing the script into functions and modules, is a recommended best practice for scripting maintainability and scalability. This approach allows for easier maintenance, reusability of code, and scalability for handling new files or tasks.

1.49 Question 25

You are debugging a Python script that is supposed to extract specific information from a JSON file but encounters errors during execution. Which approach is most effective for identifying and fixing these errors?

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 execution.

c) Increase the script's memory allocation to prevent runtime errors related to memory exhaustion.

d) Ignore the errors and proceed with running the script to see if it resolvers itself.

1.50 Solution 25

b

Utilize try-except blocks to catch and handle specific exceptions that occur during script execution

---

1.51 Question 26

You've noticed that a Python script for data analysis is performing poorly and consuming a lot of memory. What would be the best approach to identify the root cause of the memory issue?

a) Re-write the entire script

b) Add more RAM to the system.

c) Use memory profilers to pinpoint memory-consuming operations.

d) Switch to a more powerful CPU

1.52 Solution 26

c

Using memory profilers is the best approach to identify memory-consuming operations in the Python script. Memory profilers can analyze the memory usage of the script, identify memory leaks, and

pinpoint specific operations or variables that are causing high memory consumption. This targeted approach helps in efficiently addressing the root cause of the memory issue without the need to rewrite the entire script or make hardware changes.

1.53 Question 27

You've written a Python script for data analysis, but it's running slower than expected. Which of the following is the best first step to take in optimizing the performance?

a) Convert the script to a lower-level language like C.

b) Refactor the code to use multithreading.

c) Increase the hardware resources (CPU, RAM).

d) Profile the code to identify bottlenecks.

1.54 Solution 27

d

Profiling the code is the best first step to take in optimizing performance as it helps identify specific areas of the code that are causing slowdowns. By pinpointing the bottlenecks, you can focus your optimization efforts on the most critical parts of the script.

1.55 Question 28

Which of the following best describes the practice of "version control" in the context of data scripting?

a) Update the Python interpreter regularly.

b) Use Git or another VCS to keep track of changes in your scripts.

c) Use comments in your code to describe changes made in different versions.

d) Always keep a backup copy of your original data.

1.56 Solution 28

b

This is the best practice for version control, as it allows you to track changes, collaborate with others, and revert to previous versions easily.

1.57 Question 29

You are tasked with creating a Python function that reads data from an SQLite database and returns it as a Pandas DataFrame. Which of the following implementations is correct?

a)

```
import sqlite3
import pandas as pd

def read_sqlite_to_df(database_path, query):
    con = sqlite3.connect(database_path)
    df = pd.read_sql_query(query, con)
    con.close()
    return df
```

b)

```
import sqlite3
import pandas as pd

def read_sqlite_to_df(database_path, query):
    con = sqlite3.connect(database_path)
    cursor = con.cursor()
    cursor.execute(query)
    df = cursor.fecth_pandas()
    con.close()
    return df
```

c)

```
import sqlite3
import pandas as pd

def read_sqlite_to_df(database_path, query):
    con = sqlite3.connect(database_path)
    df = pd.read_sql_query(query, con)
    return df
```

d)

```
import sqlite3
import pandas as pd

def read_sqlite_to_df(database_path, query):
    con = sqlite3.connect(database_path)
    df = pd.read_dataframe(query)
    con.close()
    return df
```

1.58 Solution 29

a

This is the correct method to read from an SQLite database and convert the data into a Pandas DataFrame. It utilizes the pd.read_sql_query() function from Pandas and safely closes the connection.

1.59 Question 30

When integrating time-series data from multiple sensors into a single dataset, what is essential to ensure data consistency and accuracy?

a) Aligning all data entries to the 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) Chossing one sensor as the primary source and discarding data from others.

1.60 Solution 30

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.

1.61 Question 31

In a dataset containing customer phone numbers, which data validation technique is most suitable for ensuring the reliability and accuracy of phone numbers?

a) Pattern validation

b) Completeness validation

c) Consistency validation.

d) Format validation.

1.62 Solution 31

d

Format validation Format validation checks if the phone numbers follow the expected format (e.g., +1-123-456-7890), ensuring the reliability and accuracy of the collected phone number data.

1.63 Question 32

You are working with a dataset that contains outliers in the "Temperature" column, affecting the statistical analysis. What method would you use to detect and handle these outliers?

a) Z-score method

b) Interquartile range (IQR)

c) Standard deviation method

d) To define the syntax and structure of the Python language.

1.64 Solution 32

b

Interquartile range (IQR) The Interquartile Range (IQR) method is commonly used to detect and handle outliers by defining a range based on the 25th and 75th percentiles of the data. Observations outside this range are considered outliers and can be managed accordingly

---

1.65 Question 33

You are tasked with merging two datasets, df1 and df2, on a common column 'id'. df1 has a column 'value1' and df2 has a column 'value2'. After merging, you need to create a new column called 'total_value' that sums 'value1' and 'value2'. Which of the following code snippets accomplishes this?

a)

```
df_merged = pd.merge(df1, df2, on='id')
df_merged['total_value'] = df_merged['value1'] + df_merged['value2']
```

b)

```
df_merged = df1.join(df2, on='id')
df_merged['total_value'] = df_merged['value1'] + df_merged['value2']
```

c)

```
df_merged = pd.concat([df1, df2], keys='id')
df_merged['total_value'] = df_merged['value1'] + df_merged['value2']
```

d)

```
df_merged = df1.merge(df2)
df_merged['total_value'] = df_merged['value1'] + df_merged['value2']
```

1.66 Solution 33

a

This code snippet correctly merges the two datasets df1 and df2 on the common column 'id' using the pd.merge() function. It then creates a new column 'total_value' in the merged dataframe by summing the 'value1' and 'value2' columns from df1 and df2, respectively.

1.67 Question 34

You are working on preprocessing a dataset for a classification problem. You notice that one of the numerical features has a heavily skewed distribution. Which of the following transformations is least likely to help in handling the skewness?

a) Z-score normalization

b) Box-Cox transformation

c) Logarithmic transformation

d) Square root transformation

1.68 Solution 34

a

Z-score normalization doesn't address skewness; it simply scales the data.

1.69 Question 35

You are working on a Python project using pandas and have loaded a DataFrame from a CSV file. The DataFrame has columns named ID, Name, Salary, and Age. After conducting an initial examination, you notice that the Salary column contains some NaN values. What is the most effective way to replace these NaN values with the mean salary without affecting the other columns?

a) `df['Salary'].fillna(df['Salary'].mean(), implace=True)`

b) `df['Salary'].fillna(df['Salary'].average(), implace=True)`

c) `df.fillna(df.mean(), axis='Salary')`

d) `df['Salary'].fillna(df['Salary'].mean())`

1.70 Solution 35

a

This choice correctly uses the `fillna()` method on the 'Salary' column of the DataFrame to replace NaN values with the mean salary. The `inplace=True` parameter ensures that the changes are made directly to the DataFrame without the need to assign the result back to the DataFrame variable.

1.71 Question 36

Consider the following Python code snippet that uses Pandas:

```
import pandas as pd

data = {
    'Department': ['HR', 'HR', 'Engineering', 'Engineering', 'Marketing'],
    'Salary': [50000, 60000, 80000, 90000, 70000]
}

df = pd.DataFrame(data)
grouped = df.groupby('Department').agg({'Salary': 'mean'})
result = grouped.loc['Engineering', 'Salary']
```

What will be the value of result after executing the code?

a) 85000

b) 75000

c) 80000

d) 70000

e) 90000

1.72 Solution 36

[ ]:
```
import pandas as pd

data = {
    'Department': ['HR', 'HR', 'Engineering', 'Engineering', 'Marketing'],
    'Salary': [50000, 60000, 80000, 90000, 70000]
}

df = pd.DataFrame(data)
grouped = df.groupby('Department').agg({'Salary': 'mean'})
print(grouped)
result = grouped.loc['Engineering', 'Salary']
result
```

a

The code snippet creates a DataFrame using the provided data and groups the data by the 'Department' column, calculating the mean salary for each department. Since 'Engineering' has salaries of 80000 and 90000, the mean of these values is 85000, which will be the value of 'result'.

1.73 Question 37

You are building a classification model using Scikit-learn and want to split your dataset into training and testing sets. Which code snippet correctly performs this task assuming X contains feature data and y contains target labels? from sklearn.model_selection import train_test_split

a) `X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)`

b) `X_train, y_train, X_test, y_test = train_test_split(X, y, test_size=0.2, random_state=42)`

c) `X_train, X_test= train_test_split(X, y, test_size=0.2, random_state=42), y_train, y_test`

d) `X_train, y_train= train_test_split(X, y, test_size=0.2, random_state=42), X_test, y_test`

1.74 Solution 37

a

Option A correctly uses the train_test_split() function from Scikit-learn to split the feature data X and target labels y into training and testing sets, with a test size of 20% (test_size=0.2) and a specified random state for reproducibility (random_state=42)

1.75 Question 38

When using Python to clean data, what is the correct way to replace all occurrences of a specific value in a pandas DataFrame with another value?

a) `df.replace('old_value', 'new_value')`

b) `df.update('old_value', 'new_value')`

c) `df.set_value('old_value', 'new_value')`

d) `df.modify('old_value', 'new_value')`

1.76 Solution 38

a

`df.replace(‘old_value’, ‘new_value’)` is the correct method in pandas for replacing all instances of a specific value within the DataFrame with another value, ensuring data consistency and accuracy.

1.77 Question 39

You are analyzing data related to annual rainfall and temperature for various cities. You want to show the relationship between rainfall and temperature for each city, as well as highlight the cities with the most and least rainfall. Which type of visualization would be most effective for this scenario?

a) Bar Graph

b) Scatter Plot with Color Coding

c) Line Graph

d) Pie chart

1.78 Solution 39

b

Can effectively show the relationship between rainfall and temperature, and color coding can highlight cities with extreme rainfall amounts.

1.79 Question 40

You are required to present a comparative analysis of the sales data of five different products over the last year. What would be the most effective way to visualize this data?

a) Heatmap

b) Multiple Line Graphs on the same plot

c) 3D Surface Plot

d) Radar Chart

1.80 Solution 40

b

Multiple Line Graphs on the same plot would effectively show the trends of different products over the same period, making it easier to perform a comparative analysis.

1.81 Question 41

You are tasked with visualizing a dataset containing the frequency distribution of student grades ('A', 'B', 'C', 'D', 'F') in a classroom. What is the most appropriate type of visualization to use?

a) Polar Plot

b) Bar Chart

c) Heatmap

d) Line Plot

e) 3D Scatter Plot

1.82 Solution 41

b

A bar chart is the most appropriate type of visualization for displaying the frequency distribution of categorical data, such as student grades ('A', 'B', 'C', 'D', 'F'). Each grade category can be represented by a separate bar, making it easy to compare the frequencies of each grade.

1.83 Question 42

You are working on a Python script to validate user inputs in a survey form. The survey includes a field for the email address. What would be the most robust way to validate the email addresses?

a) Use a regular expression to match the pattern of a valid email address.

b) Automatically accept any input in the email field and flag them for later review.

c) Use the `len()` function to check if the email address exceeds a certain length.

d) Use Python's `input()` funtion's validation.

1.84 Solution 42

a

Regular expressions provide a robust and precise way to validate the format of email addresses.

1.85 Question 43

You are working with a dataset that contains sales data for 50 products over a span of 12 months. You want to visualize how the sales of each product have changed over time. The emphasis is on understanding trends and spotting outliers. Which of the following visualization techniques would best serve your purpose?

a) Heatmap

b) Small Multiple (Facet Grids)

c) Candlestick Chart

d) Box Plot

1.86 Solution 43

b

Small Multiples (Facet Grids) are the most suitable visualization technique for comparing multiple sets of data, such as sales data for different products over time. They allow for easy comparison of trends and outliers across different products in a clear and concise manner.

1.87 Question 44

When dealing with a skewed numerical feature, which of the following transformation techniques is least likely to normalize the distribution of the feature?

a) Logarithmic transformation

b) Quantile transformation

c) Z-score normalization

d) Box-Cox transformation

1.88 Solution 44

c

Z-score normalization is a technique that standardizes the distribution of a numerical feature by subtracting the mean and dividing by the standard deviation. While it helps in centering the data around zero and scaling it, it may not effectively normalize a skewed distribution, especially if the skewness is extreme.

1.89 Question 45

When designing an interactive dashboard, what should be the primary focus to enhance user experience?

a) Adding as many widgets as possible.

b) Using complex and intricate visualizations.

c) Using a wide range of contrasting colors.

d) Focusing on quick load times and responsive design.

1.90 Solution 45

d

Focusing on quick load times and responsive design is crucial for enhancing user experience in an interactive dashboard. Users expect the dashboard to load quickly and be responsive to their interactions, providing a smooth and seamless experience.

Creado por:

Isabel Maniega