Test_3_Resuelto

May 13, 2025

Creado por:

Isabel Maniega

1 Test 3

1.1 Question 1

```
For a project analyzing traffic patterns in a city, what is the most effective method of data c

a) Administering questionnaires to drivers.
b) Using traffic cameras and sensors to monitor roads.
c) Conducting interviews with city planners.
d) Reviewing historical traffic data.
```

1.2 Solution 1

b

Using traffic cameras and sensors to monitor roads Using traffic cameras and sensors to monitor roads is the most effective method to obtain real-time traffic flow information. This approach allows for continuous and objective data collection, covering various locations and times without relying on self-reported data.

1.3 Question 2

```
You have to retrieve data from a database table named employees. The table contains columns emp

a) SELECT first_name, salary FROM employees WHERE department NOT 'HR';
b) SELECT first_name, salary FROM employees WHERE department = 'HR';
c) SELECT first_name, last_name, salary FROM employees WHERE department IS 'HR';
d) SELECT first_name, salary FROM employees WHERE department = 'HR' AND salary;
```

1.4 Solution 2

b

This query correctly selects the first_name and salary columns from the employees table where the department is 'HR'.

1.5 Question 3

```
You are validating data from various sources, including text files, CSV files, audio recordings
```

```
a) Checking data type consistency.
b) Verifying data completeness.
c) Ensuring data integrity.
d) Performing format validation.
```

1.6 Solution 3

d

Performing format validation Performing format validation is crucial for ensuring the reliability and accuracy of heterogeneous data from various sources like text files, CSV files, audio recordings, and social media APIs. It ensures that data follows specific formats or structures, reducing errors and maintaining data quality.

1.7 Question 4

```
Sophia is responsible for validating data from text files, Excel spreadsheets, audio clips, and
```

```
a) Ensuring data consistency
b) Performing data format validation
c) Verifying data completeness.
d) Conducting data integrity checks.
```

1.8 Solution 4

b

Performing data format validation Data format validation is crucial for ensuring the reliability and accuracy of data from different sources like text files, Excel spreadsheets, audio clips, and social media feeds. It ensures that data follows specific formats or structures, reducing errors and maintaining data quality.

1.9 Question 5

You are given a DataFrame df with columns 'Name', 'Age', and 'Occupation'. You want to filter the rows where 'Age' is greater than 30 and 'Occupation' is 'Engineer'. Which of the following code snippets will correctly achieve this?

a) `df[(df['Age'] > 30) and (df['Occupation'] == 'Engineer')]`

b) `df[df['Age'] > 30].df['Occupation'] == 'Engineer']`

c) `df[(df['Age'] > 30) & (df['Occupation'] == 'Engineer')]`

d) `df[(df['Age'] > 30) and (df['Occupation'] == Engineer)]`

1.10 Solution 5

c

This code snippet correctly filters the DataFrame 'df' based on the conditions where 'Age' is greater than 30 and 'Occupation' is 'Engineer'. The '&' operator is used for element-wise logical 'and' operation in pandas, which is the correct way to combine multiple conditions for filtering.

1.11 Question 6

```
In a dataset containing audio file references for a voice recognition system, how should a data

a) Listen to a sample of the audio files manually.
b) Verify the file extnsion matches the required audio format.
c) Check the metadata for each audio file to confirm format and quality specifications.
d) Transcribe the audi files using voice recognition software to ensure clarity.
```

1.12 Solution 6

c

Check the metadata for each audio file to confirm format and quality specifications. Checking the metadata for each audio file is an effective way to validate both the format and quality specifications of the audio files, ensuring they meet the requirements for voice recognition analysis without having to manually inspect each file.

1.13 Question 7

```
You are analyzing a dataset for predicting customer preferences in an e-commerce platform. The

a) Convert the "Product Category" feature into numerical values using label encoding.
b) Standarize the "Price" feature using Min-Max scaling to ensure consistent scales across feat
c) Drop the "Purchase History" feature as it is not relevant for predicting customer preferenc
d) Impute missing values in the "Customer Ratings" feature with the mean rating value.
```

1.14 Solution 7

a

Convert the "Product Category" feature into numerical values using label encoding. Option A, converting the "Product Category" feature into numerical values using label encoding, is recommended for handling categorical variables in a dataset for modeling customer preferences.

1.15 Question 8

```
You are working on a dataset for predicting customer churn in a telecommunications company. The

a) Convert the "Churn" column into numerical values using label encoding.
b) Impute missing values in the "Monthly Charges" column with the median value.
c) Scale the "Age" column using Min-Max scaling to ensure uniformity in the feature scales.
d) Remove outliers from the "Monthly Charges" column to improve model preformance.

*customer churn: perdida de clientes
```

1.16 Solution 8

b

Impute missing values in the "Monthly Charges" column with the median value. Option B is the recommended step for preparing the dataset for modeling as imputing missing values in the "Monthly Charges" column with the median value helps maintain data integrity and completeness

---

1.17 Question 9

```
You are tasked with validating data collected from a survey, including text responses, numeric

a) Ensuring data type consistency.
b) Validating data format.
c) Checking data range.
d) Cross-referencing data with external sources.
```

1.18 Solution 9

b

Validating data format Validating data format is crucial for ensuring the reliability and accuracy of data across different types such as text, numeric ratings, audio feedback, and video testimonials. It ensures that data adheres to specific formats or structures, reducing errors and inconsistencies.

1.19 Question 10

```
In a clinical trial dataset, some participants have missing values for their blood pressure rea

a) Assign the highest recorded blood pressure value to the missing entries.
b) Use interpolation methods to estimate the missing values based on nearby data points.
c) Exclude all participants with missing blood presure reading from the study.
d) Categorize the missing values as a separate group in the analysis.
```

1.20 Solution 10

b

Use interpolation methods to estimate the missing values based on nearby data points. Interpolation is a suitable method for estimating missing values in a sequence or time series, as it can provide a reasonable approximation based on the existing data, preserving the dataset's overall structure and continuity for analysis

---

1.21 Question 11

```
om is analyzing a dataset of video file paths for a digital media project. He needs to ensure t

a) Check that each file path ends with a video file extension (eg., mp4, avi)
b) Play each video file to check its validity manually.
c) Run a script that verifies each file path exists and points to a file the correct format.
d) Look at the file sizes to ensure they are within a reasonable range for video files.
```

1.22 Solution 11

c

Run a script that verifies each file path exists and points to a file of the correct format. Running a script to verify the existence of each file path and that it points to a file of the correct format is an efficient and reliable method for validating video files in a dataset, ensuring both the paths and file formats are correct

---

1.23 Question 12

```
Julia is processing a dataset for a social media analysis project and needs to verify that all

a) Manually search for each username on the social media platform.
b) Use the social media patform's API to check the status of each username.
c) Validate the format of the usernames using regular expressions.
d) Confirm the number of followers for each username to ensure activity.
```

1.24 Solution 12

b

Use the social media platform's API to check the status of each username. Using the social media platform's API is the most efficient way for Julia to validate that the usernames are not only valid but also active, as it allows for automated, large-scale verification directly with the source of the data

---

1.25 Question 13

```
Jason needs to collect historical weather data to analyze climate trends over the past 50 years

a) Local newspaper archives.
```

b) An online climate database accessed via API.
c) Personal interviews with long-time residents.
d) Social media posts related to weather events.

1.26 Solution 13

b

An online climate database accessed via API An online climate database accessed via API offers a comprehensive, reliable, and efficient way to gather historical weather data over extensive periods, making it suitable for analyzing long-term climate trends.

1.27 Question 14

You are creating a dashboard to analyze customer feedback data for a service-oriented company.

a) Microsoft Excel.
b) Google Data Studio.
c) Tableau.
d) Power BI

1.28 Solution 14

d

Power BI Microsoft Power BI offers robust capabilities for integrating data from multiple sources, including surveys, social media, and customer support tickets, making it suitable for creating comprehensive dashboards that combine diverse data sources.

1.29 Question 15

You are working as a data analyst for a retail company and have been tasked with creating a dashboard to visualize sales performance for different product categories over time. The dataset includes columns for "Date," "Product Category," and "Sales Amount." Which type of chart or visualization would be most suitable for this task?

a) Line chart showing sales amount over time with different lines for each product category.
b) Scatter plot showing sales amount against date for each product category.
c) Bar chart showing total sales amount for each product category on a specific date.
d) Pie chart showing the percentage contribution of each product category to total sales.

1.30 Solution 15

a

Line chart is the most suitable choice for visualizing sales performance over time, allowing easy comparison of sales trends for different product categories using different lines

---

1.31 Question 16

```
You need to create a dashboard to track social media metrics such as engagement rate, follower
```

```
a) Tableua
b) Google Data Studio
c) Microsoft Excel
d) Power BI
```

1.32 Solution 16

b

Google Data Studio Google Data Studio offers integrations with various social media platforms and provides specific connectors for visualizing social media analytics data, making it suitable for creating dashboards focused on social media metrics.

1.33 Question 17

```
Emily is working on a dataset that contains outliers in one of the numerical columns. What prep
```

```
a) Remove the outliers.
b) Replace outliers with the median of the column
c) Use z-score normalization to scale the column.
d) Apply clustering to identify and adjust outliers.
```

1.34 Solution 17

a

Remove the outliers Removing outliers is a common preprocessing technique to ensure that extreme values do not skew the model's performance and accuracy.

1.35 Question 18

Consider the following Python code snippet:

```
fruit = "banana"
numbers1 = (8, 3, 9, 2)
numbers2 = (5, 7, 6, 1)
combined_numbers = numbers1 + numbers2
unique_set = set([ord(fruit[2]), combined_numbers[7]])

print("Third Character:", fruit[2])
print("Eighth Number:", combined_numbers[7])
print("Unique Set:", unique_set)
```

Knowing that the ASCII value for the character 'n' is 110, what will be the output of this code?

a)

```
Third Character: n
Eighth Number: 1
Unique Set: {1, 110}
```

b)

```
Third Character: n
Eighth Number: 1
Unique Set: {n, 49}
```

c)

```
Third Character: n
Eighth Number: 1
Unique Set: {n, 1}
```

d)

```
Third Character: n
Eighth Number: 1
Unique Set: {110, 49}
```

1.36 Solution 18

[1]:
```
fruit = "banana"
numbers1 = (8, 3, 9, 2)
numbers2 = (5, 7, 6, 1)
combined_numbers = numbers1 + numbers2
unique_set = set([ord(fruit[2]), combined_numbers[7]])

print("Third Character:", fruit[2])
print("Eighth Number:", combined_numbers[7])
print("Unique Set:", unique_set)
```

```
Third Character: n
Eighth Number: 1
Unique Set: {1, 110}
```

a

The ASCII value for 'n' is 110, and the eighth number in the combined tuple is 1. Therefore, the unique set will contain 110 and 1. The combined tuple combined_numbers is created by concatenating numbers1 and numbers2, making the eighth element of combined_numbers the last element of numbers2, which is 1.

Recall that in a set notaion {1,110} is the same as {110,1}, i.e. the order in which the set elements are listed is not important

---

1.37 Question 19

To streamline library management, a librarian aims to highlight books with exceptional borrowing rates. The librarian has a dataset containing monthly borrowing counts for various books and decides to use Python to isolate books with counts exceeding a specific threshold, indicating high demand.

```
borrowing_counts = [5, 15, 8, 20, 12, 25]
popular_books = [count for count in borrowing_counts if count > 10]

print(popular_books)
```

What is the primary objective of this Python code in the context of data management principles?

a) Grouping borrowing counts into categories to classify books by their popularity.

b) Identifying the average borrowing count to find the mean demand for books.

c) Filtering the dataset to only include books that have borrowing counts above a certain threshold.

d) Creating a new list that inlcudes both the book titles and their borrowing counts.

1.38 Solution 19

[2]:
```
borrowing_counts = [5, 15, 8, 20, 12, 25]
popular_books = [count for count in borrowing_counts if count > 10]

print(popular_books)
```

```
[15, 20, 12, 25]
```

c

The code filters the borrowing counts to isolate books with counts above a specified threshold, focusing on identifying high-demand books.

1.39 Question 20

You have a list of numbers `[10, 20, 30, 40, 50]` in Python. You want to create a new list containing the squares of each number in the original list. Which of the following code snippets accomplishes this task?

a) `[num ** 2 for num in [10, 20, 30, 40, 50]]`

b) `[num * num for num in [10, 20, 30, 40, 50]]`

c) `[num ^ 2 for num in [10, 20, 30, 40, 50]]`

d) `[pow(num, 2) for num in [10, 20, 30, 40, 50]]`

1.40 Solution 20

[3]:
```
# a
[num ** 2 for num in [10, 20, 30, 40, 50]]
```

[3]:
```
[100, 400, 900, 1600, 2500]
```

[4]:
```
# b
[num * num for num in [10, 20, 30, 40, 50]]
```

[4]:
```
[100, 400, 900, 1600, 2500]
```

[5]:
```
# c
[num ^ 2 for num in [10, 20, 30, 40, 50]]
```

[5]:
```
[8, 22, 28, 42, 48]
```

[6]:
```
# d
[pow(num, 2) for num in [10, 20, 30, 40, 50]]
```

[6]:
```
[100, 400, 900, 1600, 2500]
```

a

[num**2 for num in [10, 20, 30, 40, 50]] Option A correctly uses list comprehension to create a new list containing the squares of each number in the original list.

1.41 Question 21

```
In a dataset containing details of online transactions, Jessica finds that the 'transaction_dat

a) As accurate data, since future transactions can be pre-recorded.
b) As complete data, because the date column is fully populated.
c) As irrelevant data, since they do not affect the analysis.
d) As erroneous data, because transaction dates chould not be in the future.
```

1.42 Solution 21

d

As erroneous data, because transaction dates should not be in the future Entries with future dates in the 'transaction_date' column are classified as erroneous data because transaction records should reflect past or current activities, not future dates. This indicates a potential data entry error or system issue that needs to be addressed during data cleaning.

1.43 Question 22

You are given a Python script that is supposed to calculate the average sales amount from a CSV file named "sales_data.csv" and print the result. However, when you run the script, it throws an error. Which approach is most appropriate for debugging this issue?

a) Use print statements to trace the execution flow and identify the error.

b) Utilize a Python debugger such as pdb to step through the code and find the error.

c) Rewrite the entire script from scratch to ensure error-free execution.

d) Ignore the error and proceed with running the script to see if it resolves itsef.

1.44 Solution 22

b

Utilize a Python debugger such as pdb to step through the code and find the error. Option B, utilizing a Python debugger such as pdb, is the most appropriate approach for debugging script errors. Using a debugger allows you to step through the code, inspect variables, and identify the exact point of failure.

1.45 Question 23

You are working with a dictionary in Python that contains student names as keys and their corresponding scores as values. You want to create a new dictionary that includes only students who scored above 90. Which of the following code snippets accomplishes this task?

a) `{student:score for student, score in student_scores.items() if score > 90}`

b) `{student:score if score > 90 else None for student, score in student_scores.items()}`

c) `{student:score for student, score in student_scores.items() if student_scores[student] > 90}`

d) `{student:score if student_scores[student] > 90 else None for student, score in student_scores.items()}`

1.46 Solution 23

[7]:
```
student_scores = {'Peter': 80, 'Alan': 60, 'John': 100, 'Anne': 95}
```

[8]:
```
# a
{student:score for student, score in student_scores.items() if score > 90}
```

[8]:
```
{'John': 100, 'Anne': 95}
```

[9]:
```
# b
{student:score if score > 90 else None for student, score in student_scores.
↪items()}
```

[9]:
```
{'Peter': None, 'Alan': None, 'John': 100, 'Anne': 95}
```

[10]:
```
# c
{student:score for student, score in student_scores.items() if␣
 ↪student_scores[student] > 90}
```

[10]:
```
{'John': 100, 'Anne': 95}
```

[11]:
```
# d
{student:score if student_scores[student] > 90 else None for student, score in␣
 ↪student_scores.items()}
```

[11]:
```
{'Peter': None, 'Alan': None, 'John': 100, 'Anne': 95}
```

a

{student: score for student, score in student_scores.items() if score > 90} Option A correctly uses dictionary comprehension with a conditional statement to filter and include only students who scored above 90 in the new dictionary.

1.47 Question 24

You have a SQLite database named "employees.db" that contains a table named "employees" with columns "id," "name," "age," and "salary." You need to write a Python script to connect to this database, retrieve all employees who are older than 40 years old and have a salary greater than $50,000, and then print their names. Which of the following code snippets correctly accomplishes this task using the sqlite3 library?

a)

```
import sqlite3

# Connect to database
conn = sqlite3.connect("employees.db")
cursor = conn.cursor()

# Retrieve employees based on criteria
cursor.execute("SELECT name FROM employees WHERE age > 40 AND salary > 50000")
employees = cursor.fetchall()

# Print names of employees
for employee in employees:
    print(employee[0])

# Close connection
conn.close()
```

b)

```
import sqlite3

# Connect to database
conn = sqlite3.connect("employees.db")
cursor = conn.cursor()

# Retrieve employees based on criteria
cursor.execute("SELECT name FROM employees WHERE age > 40 AND salary > 50000")
employees = [row[0] for row in cursor.fetchall()]

# Print names of employees
print(employees)

# Close connection
conn.close()
```

c)

```
import sqlite3

# Connect to database
conn = sqlite3.connect("employees.db")
cursor = conn.cursor()

# Retrieve employees based on criteria
cursor.execute("SELECT * FROM employees WHERE age > 40 AND salary > 50000")
employees = cursor.fetchall()

# Print names of employees
for employee in employees:
    print(employee["name"])

# Close connection
conn.close()
```

d)

```
import sqlite3

# Connect to database
conn = sqlite3.connect("employees.db")
cursor = conn.cursor()

# Retrieve employees based on criteria
cursor.execute("SELECT * FROM employees WHERE age > 40 AND salary > 50000")
employees = [row["name"] for row in cursor.fetchall()]

# Print names of employees
print(employees)
```

```
# Close connection
conn.close()
```

1.48 Solution 24

a

Option A correctly connects to the SQLite database, executes an SQL query to retrieve employees based on the specified criteria, fetches the results, and prints the names of eligible employees. This approach uses the correct SQL syntax and fetches the data efficiently.

1.49 Question 25

You are working with a list in Python that contains both strings and integers. You want to create a new list that includes only the integers from the original list. Which of the following code snippets accomplishes this task?

a) `[item for item in original_list if isinstance(item, int)]`

b) `[item if type(item) == int else None for item in original_list]`

c) `[item for item in original_list if type(item) == int]`

d) `[item if isinstance(item, int) else None for item in original_list]`

1.50 Solution 25

[12]:
```
original_list = ["hello", 25, 40, "test", 47, "name"]
```

[13]:
```
# a
[item for item in original_list if isinstance(item, int)]
```

[13]:
```
[25, 40, 47]
```

[14]:
```
# b
[item if type(item) == int else None for item in original_list]
```

[14]:
```
[None, 25, 40, None, 47, None]
```

[15]:
```
# c
[item for item in original_list if type(item) == int]
```

[15]:
```
[25, 40, 47]
```

[16]:
```
# d
[item if isinstance(item, int) else None for item in original_list]
```

[16]:
```
[None, 25, 40, None, 47, None]
```

a

Option A correctly uses list comprehension with isinstance() to filter and include only the integers from the original list.

1.51 Question 26

Which data validation technique is most suitable for ensuring that numeric data falls within a specific range?

a) Data type validation

b) Range validation.

c) Cross-reference validation

d) Format validation.

1.52 Solution 26

b

Range validation Range validation is used to check if data falls within a specified range, such as checking if numeric values are within a certain minimum and maximum range.

1.53 Question 27

You are tasked with optimizing a Python script that processes a large dataset. During testing, you notice that the script's memory usage keeps increasing over time, leading to potential memory leaks. What debugging technique should you use to identify and fix the memory leak issue?

a) Add print statements to track memory usage.

b) Use a profiler to analyze memory usage patterns.

c) Comment out sections of code to isolate the issue.

d) Rewrite the script using a diffetent programming paradigm.

1.54 Solution 27

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

---

1.55 Question 28

You are analyzing a dataset containing information about student scores, including their ID, subject, and score obtained. You need to calculate the average score for students who have scored above

80 in the 'Mathematics' subject. Which of the following Python code snippets accomplishes this task?

a)
```
average_score = sum(student['score'] for student in student_data
if student['subject'] == 'Mathematics' and student['score'] > 80
/len(student_data))
```

b)
```
average_score = sum(student['score'] for student in student_data if
student['subject'] == 'Mathematics' and student['score'] > 80)
```

c)
```
average_score = sum(student['score'] for student in student_data
if student['subject'] == 'Mathematics' and student['score'] < 80
/len(student_data))
```

d)
```
average_score = sum(student['score'] for student in student_data
if student['subject'] != 'Mathematics' and student['score'] > 80
/len(student_data))
```

1.56 Solution 28

a

Option A correctly filters the student data for those who scored above 80 in the 'Mathematics' subject and then calculates the average score by summing their scores and dividing by the number of such students.

1.57 Question 29

You are analyzing a dataset containing information about product sales, including the product ID, quantity sold, and price per unit. You need to calculate the total sales revenue for a specific product with ID 'ABC123'. Which of the following Python code snippets accomplishes this task?

a)
```
total_revenue = sum(item['quantity_sold'] * item['price_per_unit'] for item
in sales_data if item['product_id'] == 'ABC123')
```

b)
```
total_revenue = sum(item['quantity_sold'] * item['price_per_unit'] for item
in sales_data if item['product_id'] != 'ABC123')
```

c)
```
total_revenue = sum(item['quantity_sold'] * item['price_per_unit'] for item
in sales_data if item['product_id'] != 'ABC123') / len(sales_data)
```

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

1.58 Solution 29

a

Option A correctly filters the sales data for the specific product ID 'ABC123' and then calculates the total revenue by multiplying the quantity sold by the price per unit for that product.

1.59 Question 30

You are analyzing a dataset containing information about online orders, including the order ID, customer name, and total order amount. You need to calculate the total revenue generated from orders placed by customers with names starting with the letter 'A'. Which of the following Python code snippets accomplishes this task?

a) `total_revenue = sum(order['total_amount'] for order in orders_data if order['customer_name'].startswith('A'))`

b) `total_revenue = sum(order['total_amount'] for order in orders_data if order['customer_name'].startswith('B'))`

c) `total_revenue = sum(order['total_amount'] for order in orders_data if order['customer_name'] == 'A')`

d) `total_revenue = sum(order['total_amount'] for order in orders_data if order['customer_name']  != 'A')`

1.60 Solution 30

a

Option A correctly filters the orders data for customers whose names start with 'A' and then calculates the total revenue by summing their total order amounts. Option B is incorrect.

1.61 Question 31

You have a list of strings `['apple', 'banana', 'cherry', 'date']` in Python. You want to create a new list that includes only the strings starting with the letter 'b'. Which of the following code snippets accomplishes this task?

a) `[item for item in original_list if item.startswith('b')]`

b) `[item if item.startswith('b') else None for item in original_list]`

c) `[item for item in original_list if item[0] == 'b']`

d) `[item if item[0] == 'b' else None for item in original_list]`

1.62 Solution 31

[17]:
```
original_list = ['apple', 'banana', 'cherry', 'date']b
```

```
  Cell In[17], line 1
    original_list = ['apple', 'banana', 'cherry', 'date']b
                                                         ^
SyntaxError: invalid syntax
```

[ ]:
```python
# a
[item for item in original_list if item.startswith('b')]
```

[ ]:
```python
# b
[item if item.startswith('b') else None for item in original_list]
```

[ ]:
```python
# c
[item for item in original_list if item[0] == 'b']
```

[ ]:
```python
# d
[item if item[0] == 'b' else None for item in original_list]
```

a

[item for item in original_list if item.startswith('b')] Option A correctly uses list comprehension with startswith() to filter and include only the strings starting with 'b' from the original list.

1.63 Question 32

You are working with a dataset containing customer information, including their age, income, and purchase history. You need to calculate the average age of customers who have made at least one purchase. Which of the following Python code snippets accomplishes this task?

a) `average_age = sum(customer['age'] for customer in customers if customer['purchase_history']) / len(customers)`

b) `average_age = sum(customer['age'] for customer in customers) / len(customers)`

c) `average_age = sum(customer['age'] for customer in customers if customer['purchase_history']) / len(customers if customers['purchase_history'])`

d) `average_age = sum(customer['age'] for customer in customers if customer['purchase_history']) / len(customers if customers['purchase_history'] else 1)`

1.64 Solution 32

a

Option A correctly calculates the average age of customers who have made at least one purchase by filtering customers with non-empty purchase history and then dividing the sum of their ages by the total number of such customers.

1.65 Question 33

In cleaning a dataset for a health study, a data analyst notices several instances where the 'age' column contains values over 150 years. How should these data points be treated?

a) As incomplete data, because the exact ages are not known.

b) As valid data, considering potencial recording errors.

c) As erroneous data, given the unrealistic age values.

d) As duplicates data, if the same age appears multiple items.

1.66 Solution 33

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.

1.67 Question 34

You need to validate data from a dataset containing numeric values. Which data validation technique should you use to ensure that numeric data falls within a specific range, such as 0 to 100?

a) Completeness validation

b) Range validation.

c) Format validation.

d) Data type validation

1.68 Solution 34

b

Range validation Range validation ensures that numeric data falls within a specified range, such as 0 to 100, ensuring data accuracy and consistency

---

1.69 Question 35

A dataset used for analyzing traffic patterns has several entries where the 'vehicle_speed' column shows speeds exceeding the maximum possible speed for the road in question. How should these entries be classified?

a) As complete data, because every entry has a speed value.

b) As irrelevant data, assuming speed is not crucial for the analysis.

c) As erroneous data, because the speeds exceed the maximum possible for the road.

d) As outlier data, indicating potential data entry extremes.

1.70 Solution 35

c

As erroneous data, because the speeds exceed the maximum possible for the road. Speeds exceeding the maximum possible for a road indicate erroneous data, as they conflict with known physical and legal constraints, suggesting a recording or entry error.

1.71 Question 36

You are working on a predictive model to classify emails as either Spam or Not Spam. You have built a model and it has an accuracy of 99% on your training dataset. However, when you test it on a new set of emails, the accuracy drops to 70%. What is most likely happening?

a) Class Imbalance

b) Overfitting

c) Underfitting

d) High Bias

1.72 Solution 36

b

Overfitting occurs when a model performs extremely well on the training data but fails to generalize to new, unseen data. In this case, the high accuracy on the training dataset (99%) suggests that the model may have memorized the training data instead of learning the underlying patterns, leading to a drop in accuracy when tested on new emails.

1.73 Question 37

Which of the following methods would be the most appropriate for importing a moderately-sized CSV file (around 500MB) into a Pandas DataFrame?

a) `pd.read_csv('file.csv', chunksize=5000)`

b) `pd.read_fwf('file.csv')`

c) `pd.read_csv('file.csv')`

d) `pd.read_csv('file.csv', dtype=category)`

1.74 Solution 37

c

Using the pd.read_csv('file.csv') method without any additional parameters is the most appropriate way to import a moderately-sized CSV file into a Pandas DataFrame. This method reads the entire file into memory at once, which is suitable for a file size of around 500MB and allows for easy manipulation and analysis of the data.

1.75 Question 38

Which of the following methods is least likely to be effective when dealing with outliers in a dataset while using Python's Pandas library?

a) Transformation

b) Deleting the feature column.

c) Capping

d) Imputation

1.76 Solution 38

b

Deleting the feature column is the least effective method when dealing with outliers in a dataset as it involves removing valuable data that could potentially provide insights or patterns. It is generally not recommended to delete entire columns unless absolutely necessary, as it can lead to loss of important information.

1.77 Question 39

You have a DataFrame employee_data as follows:

```
employee_data = pd.DataFrame({'Name': ['Alice', 'Bob', 'Carol'], 'Skill':
['Python', 'Java', 'Python'], 'Experience': [3, 4, 5] })
```

You want to reshape this DataFrame to show the sum of Experience for each Skill. Which code snippet will accomplish this?

a) `employee_data.pivot_table(index='Skill', columns='Experience', values='Experience', aggfunc='sum')`

b) `employee_data.set_index(['Name','Skill']).unstack().fillna(0)`

c) `employee_data.pivot(index='Skill', columns='Experience', values='Experience').sum(axis=1)`

d) `employee_data.groupby('Skill').Experience.sum()`

1.78 Solution 39

[ ]:
```python
import pandas as pd

employee_data = pd.DataFrame({'Name': ['Alice', 'Bob', 'Carol'],
                              'Skill': ['Python', 'Java', 'Python'],
                              'Experience': [3, 4, 5] })
employee_data
```

```
[ ]: # a)
employee_data.pivot_table(index='Skill', columns='Experience',␣
  ↪values='Experience', aggfunc='sum')
```

```
[ ]: # b)
employee_data.set_index(['Name','Skill']).unstack().fillna(0)
```

```
[ ]: # c)
employee_data.pivot(index='Skill', columns='Experience', values='Experience').
  ↪sum(axis=1)
```

```
[ ]: # d)
employee_data.groupby('Skill').Experience.sum()
```

d

This code snippet correctly uses the `groupby` function on the 'Skill' column and then calculates the sum of the 'Experience' column for each group. It reshapes the DataFrame to show the sum of Experience for each Skill, as required.

1.79 Question 40

You are given a DataFrame df with a column 'Date' of type string in the format 'YYYY-MM-DD'. You want to filter the rows where the year is 2024. Which of the following code snippets is the most efficient way to do so?

a) `df[df['Date'].str.slice(0, 4) == '2024']`

b) `df[df['Date'].apply(lambda x: x.split('-')[0]) == '2024']`

c) `df[df['Date'].str.contains('2024')]`

d) `df[pd.to_datetime(df['Date']).dt.year == 2024]`

1.80 Solution 40

a

This code snippet efficiently filters the DataFrame by directly slicing the first four characters of the 'Date' column, which represent the year. By comparing this sliced year with '2024', it accurately filters the rows where the year is 2024 without the need for additional operations or conversions.

1.81 Question 41

Sarah has analyzed customer feedback data and created a bar chart showing customer satisfaction ratings for different products. How should Sarah effectively communicate the insight gained from this visualization to a non-technical audience?

a) Present detailed statistical analysis results.

b) Explain the methodology used to collect customer feedback.

c) Use plain language to describe the overall satisfaction levels and compare ratings across products.

d) Discuss the technical aspects of creating the bar chart

1.82 Solution 41

c

Use plain language to describe the overall satisfaction levels and compare ratings across products. When communicating with a non-technical audience like Sarah, it's important to use simple language to convey the main insights from the visualization, such as overall satisfaction levels and comparisons between products.

1.83 Question 42

Emma is presenting a scatter plot showing the correlation between marketing spend and sales revenue. How should Emma effectively communicate the insight gained from this visualization to a non-technical audience?

a) Provide detailed explanations of data cleaning techniques.

b) Use storytelling techniques to relate the scatter plot to real-word scenarios.

c) Discuss the technical aspects of creating the scatter plot.

d) Present statistical regression models used to analyze the correlation.

1.84 Solution 42

b

Use storytelling techniques to relate the scatter plot to real-world scenarios. When communicating with a non-technical audience like Emma, using storytelling techniques to relate the visualization to real-world scenarios can help convey the insights effectively.

1.85 Question 43

You are preparing a quarterly financial report for a company, which includes revenue, expenses, profit margins, and year-over-year growth rates. What visualization technique is most appropriate for comparing revenue and expenses over the past four quarters?

a) Line chart

b) Bar chart

c) Scatter plot

d) Pie Chart

1.86 Solution 43

a

Line chart, is the most appropriate visualization technique for comparing revenue and expenses over time. A line chart allows you to show trends and patterns in data over a continuous period, making it suitable for comparing financial metrics like revenue and expenses across quarters.

1.87 Question 44

Sarah has analyzed customer feedback data and created a bar chart showing customer satisfaction ratings for different products. How should Sarah effectively communicate the insight gained from this visualization to a non-technical audience?

a) Present detailed statistical analysis result.

b) Explain the methodology used to collect customer feedback.

c) Use plain language to describe the overall satisfaction levels compare ratings across products.

d) Discuss the technical aspects of creating the bar chart.

1.88 Solution 44

c

When communicating with a non-technical audience like Sarah, it's important to use simple language to convey the main insights from the visualization, such as overall satisfaction levels and comparisons between products

---

1.89 Question 45

You are consolidating data from multiple CSV and Excel files into a single Pandas DataFrame. What is the most effective way to validate that the final DataFrame contains no missing or duplicate values?

a) Use `DateFrame.describe()` to generate descriptive statistics for numerical columns.

b) Use `pd.isnull()` to check for missing values and `pd.duplicated()` to check fro duplicates.

c) Manually inspect the first and last rows of the DataFrame.

d) Assume that Pandas automatically handles missing and duplicate values.

1.90 Solution 45

b

Using pd.isnull() allows you to check for missing values in the DataFrame, while pd.duplicated() helps identify duplicate values. By combining these two methods, you can effectively validate that the final DataFrame contains no missing or duplicate values, ensuring data integrity and accuracy.

Creado por:

Isabel Maniega