Practical Exam: Spectrum Shades LLC

Spectrum Shades LLC is a prominent supplier of concrete color solutions, offering a wide range of pigments and coloring systems used in various concrete applications, including decorative concrete, precast concrete, and concrete pavers. The company prides itself on delivering high-quality colorants that meet the unique needs of its diverse clientele, including contractors, architects, and construction companies.

The company has recently observed a growing number of customer complaints regarding inconsistent color quality in their products. The discrepancies have led to a decline in customer satisfaction and a potential increase in product returns. By identifying and mitigating the factors causing color variations, the company can enhance product reliability, reduce customer complaints, and minimize return rates.

You are part of the data analysis team tasked with providing actionable insights to help Spectrum Shades LLC address the issues of inconsistent color quality and improve customer satisfaction.

Task 1

Before you can start any analysis, you need to confirm that the data is accurate and reflects what you expect to see.

It is known that there are some issues with the `production_data` table, and the data team have provided the following data description.

Write a query to return data matching this description. You must match all column names and description criteria.
Create a cleaned version of the dataframe.

- You should start with the data in the file "production_data.csv".
- Your output should be a dataframe named clean_data.
- All column names and values should match the table below.

Column Name	Criteria
batch_id	Discrete. Identifier for each batch. Missing values are not possible.
production_date	Date. Date when the batch was produced.
raw_material_supplier	Categorical. Supplier of the raw materials. (1='national_supplier', 2='international_supplier'). Missing values should be replaced with 'national_supplier'.
pigment_type	Nominal. Type of pigment used. ['type_a', 'type_b', 'type_c']. Missing values should be replaced with 'other'.
pigment_quantity	Continuous. Amount of pigment added (in kilograms) (Range: 1 - 100). Missing values should be replaced with median.

Column Name	Criteria
mixing_time	Continuous. Duration of the mixing process (in minutes). Missing values should be replaced with mean.
mixing_speed	Categorical. Speed of the mixing process represented as categories: 'Low', 'Medium', 'High'. Missing values should be replaced with 'Not Specified'.
product_quality_score	Continuous. Overall quality score of the final product (rating on a scale of 1 to 10). Missing values should be replaced with mean.

```python
import pandas as pd
import numpy as np

# Task 1: Clean the data
def clean_data_frame():
    # Load the data
    data = pd.read_csv('production_data.csv')

    # Handle missing values
    data['pigment_quantity'].fillna(data['pigment_quantity'].median(), inplace=True)
    data['mixing_time'].fillna(data['mixing_time'].mean(), inplace=True)
    data['product_quality_score'].fillna(data['product_quality_score'].mean(), inplace=True)

    # Handle raw_material_supplier
    data['raw_material_supplier'] = data['raw_material_supplier'].map({1:
'national_supplier', 2: 'international_supplier'})

    # Handle pigment_type
    valid_types = ['type_a', 'type_b', 'type_c']
    data['pigment_type'].fillna('other', inplace=True)

    # Handle mixing_speed
    valid_speeds = ['Low', 'Medium', 'High']
    data['mixing_speed'].fillna('Not Specified', inplace=True)

    return data

clean_data = clean_data_frame()
print("Clean Data Sample:")
print(clean_data.head())
```

```
Clean Data Sample:
   batch_id production_date  ... mixing_speed product_quality_score
0         1      2024-06-25  ...         High              7.165102
1         2      2023-11-23  ...         High              6.849126
2         3      2024-02-18  ...         High              5.661209
3         4      2023-11-11  ...         High              6.991735
4         5      2024-04-11  ...         High              7.095043

[5 rows x 8 columns]
```

Task 2

You want to understand how the supplier type and quantity of materials affect the final product attributes.

Calculate the average `product_quality_score` and `pigment_quantity` grouped by `raw_material_supplier`.

- You should start with the data in the file 'production_data.csv'.
- Your output should be a data frame named aggregated_data.
- It should include the three columns: `raw_material_supplier`, `avg_product_quality_score`, and `avg_pigment_quantity`.
- Your answers should be rounded to 2 decimal places.

```python
# Task 2
def aggregate_data(clean_data):
    aggregated_data = clean_data.groupby('raw_material_supplier').agg({
        'pigment_quantity': 'mean',
        'product_quality_score': 'mean'
    }).reset_index()

    return aggregated_data.round(2)
aggregated_data = aggregate_data(clean_data)
print(aggregated_data)
```

```
    raw_material_supplier  pigment_quantity  product_quality_score
0  international_supplier             34.91                   5.97
1       national_supplier             44.73                   8.02
```

Task 3

Identify all `product_quality_score` values for batches with a `raw_material_supplier` of 2 and a `pigment_quantity` greater than 35 kg. Use the original production data table, not the output of Task 2.

- You should start with the data in the file 'production_data.csv'.
- Your output should be a data frame named pigment_data.
- It should include the three columns: `raw_material_supplier`, `pigment_quantity`, and `product_quality_score`.

- Your answers should be rounded to 3 decimal places.

```python
# Task 3

# Task 3: Filter data
def filter_data(clean_data):
    pigment_data = clean_data[
        (clean_data['raw_material_supplier'] == 'international_supplier') &
        (clean_data['pigment_quantity'] > 35)
    ][['raw_material_supplier', 'pigment_quantity', 'product_quality_score']]

    return pigment_data
filtered_pigment_data = filter_data(clean_data)
print(filtered_pigment_data)
```

```
       raw_material_supplier  pigment_quantity  product_quality_score
1     international_supplier         42.873479               6.849126
4     international_supplier         36.205108               7.095043
6     international_supplier         35.941439               5.735791
7     international_supplier         40.497203               5.510766
8     international_supplier         36.015111               4.959952
...                      ...               ...                    ...
1985  international_supplier         40.484362               7.027202
1986  international_supplier         41.978065               6.735215
1994  international_supplier         37.483930               4.500927
1995  international_supplier         42.916500               5.043882
1998  international_supplier         46.148489               5.055166

[619 rows x 3 columns]
```

Task 4

In order to proceed with further analysis later, you need to analyze how various factors relate to product quality. Start by calculating the mean and standard deviation for the following columns: `pigment_quantity`, and `product_quality_score`.
These statistics will help in understanding the central tendency and variability of the data related to product quality.

Next, calculate the Pearson correlation coefficient between the following variables: `pigment_quantity`, and `product_quality_score`.
These correlation coefficients will provide insights into the strength and direction of the relationships between the factors and overall product quality.

- You should start with the data in the file 'production_data.csv'.
- Calculate the mean and standard deviation for the columns pigment_quantity and product_quality_score as: `product_quality_score_mean`, `product_quality_score_sd`, `pigment_quantity_mean`, `pigment_quantity_sd`.
- Calculate the Pearson correlation coefficient between pigment_quantity and product_quality_score as: `corr_coef`

- Your output should be a data frame named product_quality.
- It should include the columns: `product_quality_score_mean`, `product_quality_score_sd`, `pigment_quantity_mean`, `pigment_quantity_sd`, `corr_coef`.
- Ensure that your answers are rounded to 2 decimal places.

```
# Task 4
# Task 4: Calculate statistics
def calculate_statistics(clean_data):
    stats = {
        'product_quality_score_mean': clean_data['product_quality_score'].mean(),
        'product_quality_score_sd': clean_data['product_quality_score'].std(),
        'pigment_quantity_mean': clean_data['pigment_quantity'].mean(),
        'pigment_quantity_sd': clean_data['pigment_quantity'].std(),
        'corr_coef':
clean_data['pigment_quantity'].corr(clean_data['product_quality_score'])
    }
```