```python
# finding datatype of columns in a dataset using pandas
import pandas as pd
df = pd.read_csv('retail_sales_dataset.csv')
print(df.dtypes)
```

```
Transaction ID       int64
Date                object
Customer ID         object
Gender              object
Age                  int64
Product Category    object
Quantity             int64
Price per Unit       int64
Total Amount         int64
dtype: object
```

HANDLING MISSING VALUES

```python
# CHECKING MISSING VALUES AND REMOVING THEM
print(df.isnull().sum())
df = df.dropna()
print(df.isnull().sum())
```

```
Transaction ID      0
Date                0
Customer ID         0
Gender              0
Age                 0
Product Category    0
Quantity            0
Price per Unit      0
Total Amount        0
dtype: int64
Transaction ID      0
Date                0
Customer ID         0
Gender              0
Age                 0
Product Category    0
Quantity            0
Price per Unit      0
Total Amount        0
dtype: int64
```

```python
import numpy as np

# Replace 2% of values with NaN
df = df.mask(np.random.random(df.shape) < .02)
```

```
# CHECKING MISSING VALUES AND REMOVING THEM
print(df.isnull().sum())
print("\nnow removing them\n")
df = df.dropna()
print(df.isnull().sum())
```

```
Transaction ID      8
Date                5
Customer ID         4
Gender              4
Age                 5
Product Category    3
Quantity            3
Price per Unit      7
Total Amount        2
dtype: int64

now removing them

Transaction ID      0
Date                0
Customer ID         0
Gender              0
Age                 0
Product Category    0
Quantity            0
Price per Unit      0
Total Amount        0
dtype: int64
```

QUESTION1: WHICH COLUMN HAVE MISSING VALUES

... initially none of the columns had missing value so, I created 2% missing values in all of them

QUESTION2: HOW AND WHY I HANDLED:

I handled missing values using dropna() function <<< it removes the entitty with attributes having missing values>>>. This is done to increase i accuracy and performance of a future model trained on this dataset etc. Also it makes the dataset accurate.

DATA CLEANING

```
#  Identifying and correcting data entries that are out-of-range or incorrect
```

```python
# identifying out of range values
import pandas as pd
df = pd.read_csv('retail_sales_dataset.csv')
print(df.describe())

# df[(df['Age'] < 18) | (df['Age'] > 50)]
# print(ov)

df['Age'] = np.where((df['Age'] < 18) | (df['Age'] > 50), 25,
df['Age'])
print(df)
```

```
       Transaction ID          Age     Quantity  Price per Unit  Total
Amount
count     1000.000000  1000.00000  1000.000000     1000.000000
1000.000000
mean       500.500000    41.39200     2.514000      179.890000
456.000000
std        288.819436    13.68143     1.132734      189.681356
559.997632
min          1.000000    18.00000     1.000000       25.000000
25.000000
25%        250.750000    29.00000     1.000000       30.000000
60.000000
50%        500.500000    42.00000     3.000000       50.000000
135.000000
75%        750.250000    53.00000     4.000000      300.000000
900.000000
max       1000.000000    64.00000     4.000000      500.000000
2000.000000
     Transaction ID        Date Customer ID  Gender  Age Product
Category  \
0                 1  2023-11-24     CUST001    Male   34
Beauty
1                 2  2023-02-27     CUST002  Female   26
Clothing
2                 3  2023-01-13     CUST003    Male   50
Electronics
3                 4  2023-05-21     CUST004    Male   37
Clothing
4                 5  2023-05-06     CUST005    Male   30
Beauty
..              ...         ...         ...     ...  ...
...
995             996  2023-05-16     CUST996    Male   25
Clothing
996             997  2023-11-17     CUST997    Male   25
Beauty
997             998  2023-10-29     CUST998  Female   23
```

```
Beauty
998             999  2023-12-05     CUST999  Female   36
Electronics
999            1000  2023-04-12    CUST1000    Male   47
Electronics

     Quantity  Price per Unit  Total Amount
0           3              50           150
1           2             500          1000
2           1              30            30
3           1             500           500
4           2              50           100
..        ...             ...           ...
995         1              50            50
996         3              30            90
997         4              25           100
998         3              50           150
999         4              30           120

[1000 rows x 9 columns]
```

QUESTION1: CRITERIA USED TO IDENTIFY INCORRECT DATA. i used an age limit for the customers. Minimum age tobe 18 and maximum obe 50. then displayed the data set falling out of this category.

QUESTION2: how to corect. there a many ways to correct it like replacing it with age 25 which i have done by first locating them and checking by upper and lower bound values. the if it is true so we replace else we ignore

DATA ANALYSIS

```
# ANAlyse data to find insights such total sales by product or average
sales per month
# total sales by product
total_sales_by_product = df.groupby('Product Category')
['Quantity'].sum()
print(total_sales_by_product)
```

```
Product Category
Beauty         771
Clothing       894
Electronics    849
Name: Quantity, dtype: int64
```

QUESTION1: CALOTHING PRODUCTS HAVE THE HIGHEST SALE

```
# HOW do average sals vary by month

from statistics import mean
df['Date'] = pd.to_datetime(df['Date'])
df['Month'] = df['Date'].dt.month
average_sales_by_month = df.groupby('Month')['Total Amount'].mean()
print(average_sales_by_month)
print(mean(average_sales_by_month))
```

```
Month
1     474.102564
2     518.352941
3     397.123288
4     393.837209
5     506.190476
6     476.818182
7     492.569444
8     393.191489
9     363.384615
10    485.208333
11    447.692308
12    491.098901
Name: Total Amount, dtype: float64
453.29747929804626
```

Data correlations

```
correlation = df['Total Amount'].corr(df['Quantity'])
print(correlation)
```

```
0.3737070541214061
```

by implementing a default function corelation()i interpret

The correlation of 0.3737 indicates a moderate positive correlation between sales amount and quantity sold. This suggests a tendency for sales to increase with higher quantities, but the relationship isn't very strong. Other factors like pricing and discounts likely play a significant role. Further analysis and visualization can provide deeper insights.

```

```