Assignment_3ML_B

Harshitha Boyapati

700761387

+ Code + Text

[84]
```
import pandas as pd
from google.colab import drive

#Loading the data into the drive
drive.mount('/drive')
dc = pd.read_csv('/drive/MyDrive/data.csv')

# Showing basic statistical description of the data using the description() function
print(dc.describe())
```

```
Drive already mounted at /drive; to attempt to forcibly remount, call drive.mount("/drive", force_remount=True).
          Duration       Pulse    Maxpulse     Calories
count  169.000000  169.000000  169.000000   164.000000
mean    63.846154  107.461538  134.047337   375.790244
std     42.299949   14.510259   16.450434   266.379919
min     15.000000   80.000000  100.000000    50.300000
25%     45.000000  100.000000  124.000000   250.925000
50%     60.000000  105.000000  131.000000   318.600000
75%     60.000000  111.000000  141.000000   387.600000
max    300.000000  159.000000  184.000000  1860.400000
```

[86]
```
# Check if the data has null values.
print('Are there any null values present in data: ',dc.isnull().values.any())
# Replace the null values with the mean
dc.fillna(dc.mean(),inplace=True)
print('Are there any null values after using fillna: ',dc.isnull().values.any())
```

```
[99] # Check if the data has null values.
     print('Are there any null values present in data: ',dc.isnull().values.any())
     # Replace the null values with the mean
     dc.fillna(dc.mean(),inplace=True)
     print('Are there any null values after using fillna: ',dc.isnull().values.any())
```

```
Are there any null values present in data:  False
Are there any null values after using fillna:  False
```

```
# Select at least two columns and aggregate the data using: min, max, count, mean.
aggregat = dc.groupby('Duration').agg({'Calories':['mean','min','max','count']})
aggregat
```

	Calories			
	mean	min	max	count
Duration				
15	87.000000	50	124	2
20	151.222222	50	229	9
25	244.000000	244	244	1
30	191.812500	86	319	16
45	278.885714	100	406	35

+ Code + Text

45	278.885714	100	406	35
60	340.797468	215	486	79
75	325.000000	320	330	2
80	643.000000	643	643	1
90	541.625000	466	700	8
120	666.666667	500	1000	3
150	939.250000	816	1115	4
160	943.500000	853	1034	2
180	733.333333	600	800	3
210	1618.000000	1376	1860	2
270	1729.000000	1729	1729	1
300	1500.000000	1500	1500	1

```
[89] # Filter the dataframe to select the rows with calories values between 500 and 1000
     dc[(dc['Calories']>=500) & (dc['Calories']<=1000)]
```

	Duration	Pulse	Maxpulse	Calories
51	80	123	146	643.1

[89]

	Duration	Pulse	Maxpulse	Calories
51	80	123	146	643.1
62	160	109	135	853.0
65	180	90	130	800.4
66	150	105	135	873.4
67	150	107	130	816.0
72	90	100	127	700.0
73	150	97	127	953.2
75	90	98	125	563.2
78	120	100	130	500.4
83	120	100	130	500.0
90	180	101	127	600.1
99	90	93	124	604.1
101	90	90	110	500.0
102	90	90	100	500.0
103	90	90	100	500.4

+ Code + Text

[89]

102	90	90	100	500.0
103	90	90	100	500.4
106	180	90	120	800.3
108	90	90	120	500.3

[90]
```
# Filter the dataframe to select the rows with calories values > 500 and pulse < 100
dc[(dc['Calories']>500) & (dc['Pulse']<100)]
```

	Duration	Pulse	Maxpulse	Calories
65	180	90	130	800.4
70	150	97	129	1115.0
73	150	97	127	953.2
75	90	98	125	563.2
99	90	93	124	604.1
103	90	90	100	500.4
106	180	90	120	800.3
108	90	90	120	500.3

```
[92] # Create a new "dc_modified" dataframe that contains all the columns from df except for "Maxpulse"
     dc_modified = dc[['Duration', 'Pulse', 'Calories']]
     dc_modified
```

	Duration	Pulse	Calories
0	60	110	409.1
1	60	117	479.0
2	60	103	340.0
3	45	109	282.4
4	45	117	406.0
...	...	...	...
164	60	105	290.8
165	60	110	300.0
166	60	115	310.2
167	75	120	320.4
168	75	125	330.4

169 rows × 3 columns

```
[93] # Delete the "Maxpulse" column from the main df dataframe
     dc = dc.drop('Maxpulse', axis=1)
     dc
```

	Duration	Pulse	Calories
0	60	110	409.1
1	60	117	479.0
2	60	103	340.0
3	45	109	282.4
4	45	117	406.0
...	...	...	...
164	60	105	290.8
165	60	110	300.0
166	60	115	310.2
167	75	120	320.4
168	75	125	330.4

169 rows × 3 columns

```
[94] # Convert the datatype of Calories column to int datatype
     dc['Calories'] = dc['Calories'].astype('int64')
     dc.dtypes
```

```
Duration    int64
Pulse       int64
Calories    int64
dtype: object
```