1_BasicDataManipulation

August 27, 2024

[5]:
```
import pandas as pd;
# This file includes basic data manipulation

# Creating series manually
data = pd.Series([5,2,8])
print(data)
```

```
0    5
1    2
2    8
dtype: int64
```

[7]:
```
# Creating a dataframe manually
data = {
    'ID': [1,2,3,4,5,6,7,8,9,10,11,12],
    'Weight': [92,70,58,99,55,76,62,92,71,70,77,79],
    'Exercise': [6,6,6,2,8,4,6,6,5,6,4,4],
    'Cholesterol': [4.8,5.1,6.4,6.5,2.3,5.7,4.2,6.9,4.8,4.8,7.7,5.7],
    'Income': [2060,2660,2530,1740,3520,3750,2720,3130,2100,3340,2430,2700],
    'Happiness': [49,36,49,28,77,55,43,39,54,29,53,47],
    'Birthyear': [1953,1955,1939,1942,1989,1937,1979,1905,1995,1966,1938,1993]
}
df = pd.DataFrame(data)
df.set_index('ID', inplace=True)
df
```

[7]:

ID	Weight	Exercise	Cholesterol	Income	Happiness	Birthyear
1	92	6	4.8	2060	49	1953
2	70	6	5.1	2660	36	1955
3	58	6	6.4	2530	49	1939
4	99	2	6.5	1740	28	1942
5	55	8	2.3	3520	77	1989
6	76	4	5.7	3750	55	1937
7	62	6	4.2	2720	43	1979
8	92	6	6.9	3130	39	1905
9	71	5	4.8	2100	54	1995
10	70	6	4.8	3340	29	1966

11	77	4	7.7	2430	53	1938
12	79	4	5.7	2700	47	1993

[9]:
```
# Printing basic statistics of the dataframe
df.describe()
```

[9]:

	Weight	Exercise	Cholesterol	Income	Happiness	Birthyear
count	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000
mean	75.083333	5.250000	5.408333	2723.333333	46.583333	1957.583333
std	13.813421	1.544786	1.419000	614.807188	13.283334	27.566311
min	55.000000	2.000000	2.300000	1740.000000	28.000000	1905.000000
25%	68.000000	4.000000	4.800000	2347.500000	38.250000	1938.750000
50%	73.500000	6.000000	5.400000	2680.000000	48.000000	1954.000000
75%	82.250000	6.000000	6.425000	3182.500000	53.250000	1981.500000
max	99.000000	8.000000	7.700000	3750.000000	77.000000	1995.000000

[11]:
```
# Checking if the income of a person is larger or smaller than the average
incomeMean = df.Income.mean()
for i, Income in df.Income.items():
    word = "at"
    if Income < incomeMean:
        word = "below"
    elif Income > incomeMean:
        word = "above"
    print(f'Person {i}: {word} average income.')
```

```
Person 1: below average income.
Person 2: below average income.
Person 3: below average income.
Person 4: below average income.
Person 5: above average income.
Person 6: above average income.
Person 7: below average income.
Person 8: above average income.
Person 9: below average income.
Person 10: above average income.
Person 11: below average income.
Person 12: below average income.
```