python3_data_analysis_1

January 22, 2024

[1]:
```
from datetime import datetime
import pytz
```

[4]:
```
time_zone=pytz.timezone("US/Eastern")
current_time=datetime.now(time_zone)
print(current_time)
```

```
2024-01-21 20:46:39.675890-05:00
```

[5]:
```
name = input("Name? ")

print(f"Hi, my name is {name}! Welcome to my GitHub!")
```

```
Name? Sukhdeep
Hi, my name is Sukhdeep! Welcome to my GitHub!
```

[6]:
```
import pandas as pd
import numpy as np
```

[7]:
```
my_dictionary={'A': [1,2,3,4,5], 'B': [6,7,8,9,10], 'C': [11,12,13,14,15]}
df=pd.DataFrame(data=my_dictionary)
df
```

[7]:
```
   A   B   C
0  1   6  11
1  2   7  12
2  3   8  13
3  4   9  14
4  5  10  15
```

[8]:
```
df.columns
```

[8]:
```
Index(['A', 'B', 'C'], dtype='object')
```

[9]:
```
df
```

[9]:
```
   A   B   C
0  1   6  11
```

```
1  2   7  12
2  3   8  13
3  4   9  14
4  5  10  15
```

[10]:
```
#best practices to clean data
#remove spaces
#change to lower cases

df.columns = [x.strip() for x in df.columns]
df.columns = [x.lower() for x in df.columns]
```

[11]:
```
df.columns
```

[11]:
```
Index(['a', 'b', 'c'], dtype='object')
```

[12]:
```
df
```

[12]:
```
   a   b   c
0  1   6  11
1  2   7  12
2  3   8  13
3  4   9  14
4  5  10  15
```

[ ]:
```
#lets get to know our data...
```

[14]:
```
#(sample size, variable) or (rows, columns)

df.shape
```

[14]:
```
(5, 3)
```

[17]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   a       5 non-null      int64
 1   b       5 non-null      int64
 2   c       5 non-null      int64
dtypes: int64(3)
memory usage: 248.0 bytes
```

[18]:
```
df.index
```

```
[18]: RangeIndex(start=0, stop=5, step=1)
```

```
[19]: type(df)
```

```
[19]: pandas.core.frame.DataFrame
```

```
[20]: #lets change the row names

      df.index=['customer_1', 'customer_2', 'customer_3', 'customer_4', 'customer_5']
```

```
[21]: df
```

```
[21]:             a   b   c
      customer_1  1   6  11
      customer_2  2   7  12
      customer_3  3   8  13
      customer_4  4   9  14
      customer_5  5  10  15
```

```
[25]: #lets change the columns name #inplace=True makes it permanent!

      df.rename(columns={'a': 'Product_Category'}, inplace=True)
      df.rename(columns={'b': 'revenue'}, inplace=True)
      df.rename(columns={'c': 'cost'}, inplace=True)

      df
```

```
[25]:             Product_Category  revenue  cost
      customer_1                 1        6    11
      customer_2                 2        7    12
      customer_3                 3        8    13
      customer_4                 4        9    14
      customer_5                 5       10    15
```

```
[28]: #find the sum of revenue

      df.revenue.sum()
```

```
[28]: 40
```

```
[29]: #find the sum of cost

      df.cost.sum()
```

```
[29]: 65
```

[30]:
```
#feeling lazy?

df.sum()
```

[30]:
```
Product_Category    15
revenue             40
cost                65
dtype: int64
```

[31]:
```
df
```

[31]:
```
            Product_Category  revenue  cost
customer_1                 1        6    11
customer_2                 2        7    12
customer_3                 3        8    13
customer_4                 4        9    14
customer_5                 5       10    15
```

[33]:
```
#turns out customer_2 actually had a revenue of 250...lets change it

df.replace([7],[250], inplace=True)

df
```

[33]:
```
            Product_Category  revenue  cost
customer_1                 1        6    11
customer_2                 2      250    12
customer_3                 3        8    13
customer_4                 4        9    14
customer_5                 5       10    15
```

[34]:
```
#turns out customer_3 and customer_4 both had a revenue of 300

df.replace([8,9], [300,300], inplace=True)

df
```

[34]:
```
            Product_Category  revenue  cost
customer_1                 1        6    11
customer_2                 2      250    12
customer_3                 3      300    13
customer_4                 4      300    14
customer_5                 5       10    15
```

[35]:
```
#Profit?
#profit = revenue - cost
```

```
df['profit']=df['revenue']-df['cost']

df
```

[35]:
```
            Product_Category  revenue  cost  profit
customer_1                 1        6    11      -5
customer_2                 2      250    12     238
customer_3                 3      300    13     287
customer_4                 4      300    14     286
customer_5                 5       10    15      -5
```

[37]:
```
#adding customer_6

df.loc['customer_6']=["2", 400, 20, 380]

df
```

[37]:
```
           Product_Category  revenue  cost  profit
customer_1                1        6    11      -5
customer_2                2      250    12     238
customer_3                3      300    13     287
customer_4                4      300    14     286
customer_5                5       10    15      -5
customer_6                2      400    20     380
```

[38]:
```
#removing a column

df.drop('Product_Category', axis=1, inplace=True)

df
```

[38]:
```
            revenue  cost  profit
customer_1        6    11      -5
customer_2      250    12     238
customer_3      300    13     287
customer_4      300    14     286
customer_5       10    15      -5
customer_6      400    20     380
```

[39]:
```
#removing a row

df.drop('customer_5', axis=0, inplace=True)

df
```

[39]:
```
            revenue  cost  profit
customer_1        6    11      -5
```

```
customer_2       250    12     238
customer_3       300    13     287
customer_4       300    14     286
customer_6       400    20     380
```