C29 - Inclass Assignment (pandas)

author & date

- author: Akshar Patel
- date: 4/27/2022

Q1. Update author name and date

In [1]:

```
# Run this line first

import pandas as pd
```

Q2. Create a variable `even` that contains a Pandas series of the list [2, 4, 6, 8, 10]

In [2]:

```
even = [2, 4, 6, 8, 10]
```

Q3. Create a variable `even_indexed` with a list [2, 4, 6, 8, 10] and a list of index ["s1", "s2", "s3", "s4", "s5"], and print `even_indexed`

In [3]:

```
even_indexed = pd.Series([2, 4, 6, 8, 10], index=["s1", "s2", "s3", "s4", "s5"])
print(even_indexed)
```

```
s1     2
s2     4
s3     6
s4     8
s5    10
dtype: int64
```

Q4. Run the following code and find the error message. Copy and paste the error message & interpret the meaning of the error message in your own language.

In [4]:

```
pd.Series([1, 3, 5, 7, 9], index=["A", "b", "C", "d"])
```

```
# error message: Length of passed values is 5, index implies 4.

# interpretation: Length of values (5) does not match length of index (4)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-4-39db537f58b4> in <module>
----> 1 pd.Series([1, 3, 5, 7, 9], index=["A", "b", "C", "d"])
      2 
      3 # error message: Length of passed values is 5, index implies 4.
      4 
      5 # interpretation: Length of values (5) does not match length of index (4)

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\series.py in __init__(self, data, index, dtype, name, copy, fastpath)
    348                 try:
    349                     if len(index) != len(data):
--> 350                         raise ValueError(
    351                             f"Length of passed values is {len(data)}, "
    352                             f"index implies {len(index)}."

ValueError: Length of passed values is 5, index implies 4.
```

Run the following chunk for Q5

In [13]:

```
dict_a = {"today": 5, "yesterday": 7, "tomorrow": 1}
series_a = pd.Series(dict_a)
```

Q5. A line of code to return the value of `yesterday`

In [14]:

```
series_a["yesterday"]
```

Out[14]:

```
7
```

Run the following chunk for Q6

In [18]:

```
item_dict = {
    "item1": pd.Series(['one', 'two', 3, 'four'], index=["a", "b", "c", "d"]),
    "item2": pd.Series([5, 6, "seven", 8], index=["a", "b", "c", "d"]),
}
```

Q6. Create a pandas dataframe `item_df` by passing `item_dict`, and print it.

In [22]:

```
item_df = pd.DataFrame(item_dict)
print(item_df)
```

```
  item1  item2
a   one      5
b   two      6
c     3  seven
d  four      8
```

Q7. Two ways of selecting `item1` column of `item_df`

In [23]:

```
item_df.item1
```

Out[23]:

```
a     one
b     two
c       3
d    four
Name: item1, dtype: object
```

In [24]:

```
item_df['item1']
```

Out[24]:

```
a     one
b     two
c       3
d    four
Name: item1, dtype: object
```

Q8. A line of code to select the row `c` only using the row name

In [25]:

```
item_df.loc['c']
```

Out[25]:

```
item1        3
item2    seven
Name: c, dtype: object
```

Q9. A line of code to select the row `c` only using the index number

In [33]:

```
item_df.iloc[2]
```

Out[33]:

```
item1        3
item2    seven
Name: c, dtype: object
```

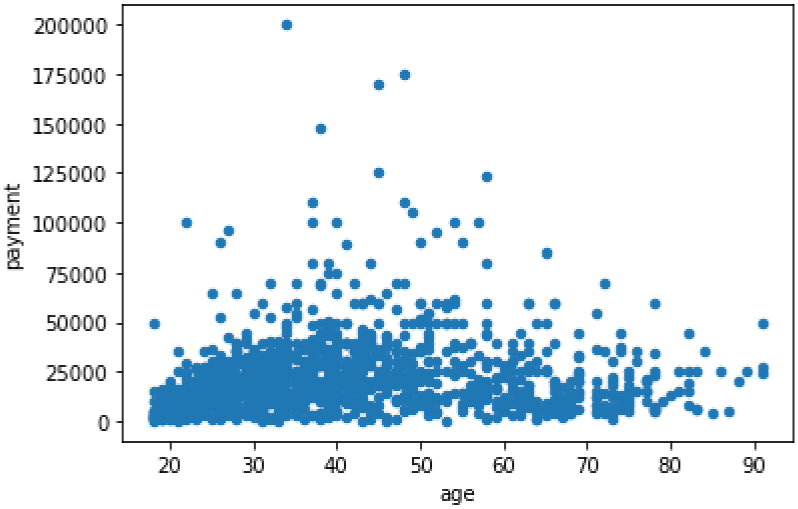
Run the following code for Q10. If you don't have `payment.csv` file, please download it from the last Hands-on Assignment and upload it to your Jupyter working directory.

```
In [30]: payment = pd.read_csv("payment.csv")
```

Q10. A line of code to display a scatter plot. X-axis is `age` column, and y-axis is `payment` column.

```
In [31]: payment.plot.scatter(x="age", y= "payment")
```

```
Out[31]: <AxesSubplot:xlabel='age', ylabel='payment'>
```


payment
age