Discount_changingdataincolumns

November 21, 2016

In [1]:
```
import pandas as pd
```

In [2]:
```
import pandas as pd
purchase_1 = pd.Series({'Name' : 'Chris',
                        'Item Purchased' : 'Dog Food',
                        'Cost' : 22.50})
purchase_2 = pd.Series({'Name' : 'Kevyn',
                        'Item Purchased' : 'Kitty Litter',
                        'Cost' : 2.50})
purchase_3 = pd.Series({'Name' : 'Vinod',
                        'Item Purchased' : 'Bird Seed',
                        'Cost' : 5.00})
df = pd.DataFrame([purchase_1, purchase_2, purchase_3], index = ['Store 1',
df.head()
```

Out[2]:

	Cost	Item Purchased	Name
Store 1	22.5	Dog Food	Chris
Store 1	2.5	Kitty Litter	Kevyn
Store 2	5.0	Bird Seed	Vinod

In [3]:
```
df.T # transpose
```

Out[3]:

	Store 1	Store 1	Store 2
Cost	22.5	2.5	5
Item Purchased	Dog Food	Kitty Litter	Bird Seed
Name	Chris	Kevyn	Vinod

In [4]:
```
df.T.loc['Cost']
```

Out[4]:
```
Store 1    22.5
Store 1     2.5
Store 2       5
Name: Cost, dtype: object
```

In [5]:
```
# try to avoid chaining
df.loc['Store 1']['Cost'] # not the best approach, selects copy not view
```

Out[5]:
```
Store 1    22.5
Store 1     2.5
Name: Cost, dtype: float64
```

```
In [6]: df.loc[:, ['Name', 'Cost']] # using .loc and slicing

Out[6]:           Name  Cost
        Store 1  Chris  22.5
        Store 1  Kevyn   2.5
        Store 2  Vinod   5.0

In [7]: # returns copy of dataframe
        df.drop('Store 1')

Out[7]:          Cost Item Purchased   Name
        Store 2   5.0      Bird Seed  Vinod

In [9]: # drop has axes and inplace
        copy_df = df.copy()
        copy_df = copy_df.drop('Store 1')
        copy_df

Out[9]:          Cost Item Purchased   Name
        Store 2   5.0      Bird Seed  Vinod

In [10]: # immediate effect on data and doesn't return a view
         del copy_df['Name']
         copy_df

Out[10]:          Cost Item Purchased
         Store 2   5.0      Bird Seed

In [12]: df['Location'] = None
         df

Out[12]:          Cost Item Purchased   Name Location
         Store 1  22.5       Dog Food  Chris     None
         Store 1   2.5   Kitty Litter  Kevyn     None
         Store 2   5.0      Bird Seed  Vinod     None

In [14]: purchase_1 = pd.Series({'Name': 'Chris',
                                 'Item Purchased': 'Dog Food',
                                 'Cost': 22.50})
         purchase_2 = pd.Series({'Name': 'Kevyn',
                                 'Item Purchased': 'Kitty Litter',
                                 'Cost': 2.50})
         purchase_3 = pd.Series({'Name': 'Vinod',
                                 'Item Purchased': 'Bird Seed',
                                 'Cost': 5.00})


         df = pd.DataFrame([purchase_1, purchase_2, purchase_3], index=['Store 1',

In [15]: df
```

```
Out[15]:          Cost Item Purchased   Name
         Store 1  22.5       Dog Food  Chris
         Store 1   2.5   Kitty Litter  Kevyn
         Store 2   5.0      Bird Seed  Vinod

In [23]: df['Cost']

Out[23]: Store 1    22.5
         Store 1     2.5
         Store 2     5.0
         Name: Cost, dtype: float64

In [30]: # evaluated 20 percent discount
         df['Cost'] = [30.5, 2, 4]

In [28]: df

Out[28]:          Cost Item Purchased   Name
         Store 1  30.5       Dog Food  Chris
         Store 1   2.0   Kitty Litter  Kevyn
         Store 2   4.0      Bird Seed  Vinod

In [31]: # alternate answer of applying 20 percent discount
         df['Cost'] *= 0.8
         print(df)

         Cost Item Purchased   Name
Store 1  24.4       Dog Food  Chris
Store 1   1.6   Kitty Litter  Kevyn
Store 2   3.2      Bird Seed  Vinod


In [ ]:
```