Module 7: Developing Web Maps and Representing information using Plots

Case Study Solution

Case Study Solution

Operations on Pandas

1. Create a pandas dataframe having following structure

	float_col	int_col	str_col
0	0.1	1	a
1	0.2	2	b
2	0.2	6	None
3	10.1	8	c
4	NaN	-1	a

Solution

```
import pandas

x=pandas.DataFrame({'int_col' : [10,20,60,80,-10], 'float_col' : [0.1, 0.5,0.9,10.9,None], 'str_col' :['a','b',None,'c','a']})

print(x)
```


2. Filter the columns 'float_col', 'int_col' from the dataframe in one query. Hint-use ix method of dataframes. Also print without using ix method

Solution

```
print(x.ix[:,['float_col','int_col']] )

#or

print(x[['float_col','int_col']])
```

3. Filter the records from float_col having value greater than 0.15 and in separate query filter float_col value equal to 0.1

Solution

```
x[x['float_col'] > 0.15] and  x[x['float_col'] == 0.1]
```

4. Filter the records from data frame which satisfies both the conditions float_col greater than 0.1 and int_col greater than 2

Solution

```
print(x[(x['float_col'] > 0.1) & (x['int_col']>2)])
```

5. Filter the records from data frame which satisfies both the conditions float_col greater than 0.1 or int_col greater than 2

Solution

```
print(x[(x['float_col'] > 0.1) | (x['int_col']>2)])
```

6. Filter the records from data frame which satisfies the conditions float_col not greater than 0.1

Solution

```
print(x[~(x['float_col'] > 0.1)])
```

7. Create a new data frame in which column int_col is renamed to new_name.

Solution

```
x1 = x.rename(columns={'int_col' : 'new_name'})
```

```
print(x1)
```

8. Modify the existing data frame and rename the column int_col to new_name

Solution

```
print(x.rename(columns={'int_col' : 'new_name'}, inplace = True))
```

9. Drop the rows where any value is missing from the data frame

Solution

```
print(x.dropna())
```


10. Change the missing value in column float_col as mean value of the float_col

Solution

```
print(x['float_col'].fillna(x['float_col'].mean))
```

11. Change all the values of str_col with new value and drop the missing values. New value should have prefix map_ and original value. Eg map_a, map_b

Solution

```
print(x['str_col'].dropna().map(lambda a:'map_'+a))
```

12. Group all the values of str_col and find the mean of float_col in all the groups respectively.

Solution

```
grouped = x['float_col'].groupby(x['str_col'])

print(grouped.mean())
```

13. Find the covariance of float_col and int_col

Solution

```
print(x.cov())
```

14. Find the correlation of float_col and int_col

Solution

```
print(x.corr())
```


15. Create a data frame 'other' having columns some_val and str_col having values given below

	some_val	str_col
0	1	a
1	2	b

Perform inner join, outer join, left join and right join with data frame x

Solution

```
other = DataFrame({'str_col' : ['a','b'], 'some_val' : [1, 2]})

print(pandas.merge(x,other,on='str_col',how='inner'))

print(pandas.merge(x,other,on='str_col',how='left'))

print(pandas.merge(x,other,on='str_col',how='right'))
```

16. When we want to send the same invitations to many people, the body of the mail does not change. Only the name (and maybe address) needs to be changed.

    Mail merge is a process of doing this. Instead of writing each mail separately, we have a template for body of the mail and a list of names that we merge together to form all the mails.

    Create a text file "names.txt" having the names.

    Anil
    sunita
    suman
    lokesh
    Sumita
    John
    Johny

    Create a text file "body.txt" having the body of email.

    I am going to Delhi. Lets meet on 7th Jan 2018
    Have a great day
    Regards
    Team Victory

    Write a program which should create separate files Anil.txt, sunita.txt, suman.txt etc after picking names from names.txt. content of these files looks like –

    Anil.txt

    Hello Anil
    I am going to Delhi. Lets meet on 7th Jan 2018
    Have a great day
    Regards
    Team Victory
    ----

    sunita.txt

    Hello Anil
    I am going to Delhi. Lets meet on 7th Jan 2018
    Have a great day
    Regards
    Team Victory

Solution

```
with open("names.txt",'r') as names_file:
  # open body.txt for reading
  with open("body.txt",'r') as body_file:
    # read entire content of the body
    body = body_file.read()
    # iterate over names
    for name in names_file:
      mail = "Hello "+name+body
      # write the mails to individual files
      with open(name.strip()+".txt",'w') as mail_file:
        mail_file.write(mail)
```