|  |
| --- |
| Module 9: Developing Web Maps and Representing information using Plots |
| Case Study I Solution |

|  |
| --- |
|  |



© Brain4ce Education Solutions Pvt. Ltd.

# 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)

1. 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']])

1. 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]

1. 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)])

1. 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)])

1. 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)])

1. 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)

1. 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))

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

# Solution

print(x.dropna())

1. 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))

1. 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))

1. 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())

1. Find the covariance of float\_col and int\_col

# Solution

print(x.cov())

1. Find the correlation of float\_col and int\_col

# Solution

print(x.corr())

1. 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'))

1. 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)