Interactive exercise week #7b

Liping Wu 300-958-061

In this exercise we will do the following:

Handle Basics of data – summary, dimensions, and structure:

- To check whether the data has read in correctly or not
- To determine how the data looks; its shape and size
- To summarize and visualize the data
- To get the column names and summary statistics of numerical variables

Pre-requisites:

1- Install Anoconda
2- We will be using a lot of Public datasets these datasets are available at:

   https://goo.gl/zjS4C6

   Under a folder named "Datasets for Predictive Modelling with Python", the datasets are organized in the order of the third text book chapters:
   Python: Advanced Predictive Analytics, by Joseph Babcock and Ashish Kumar. Published by Packt Publishing Ltd ISBN: 9781788992367.(12/2017) **For this exercise we need the files of chapter # 2**

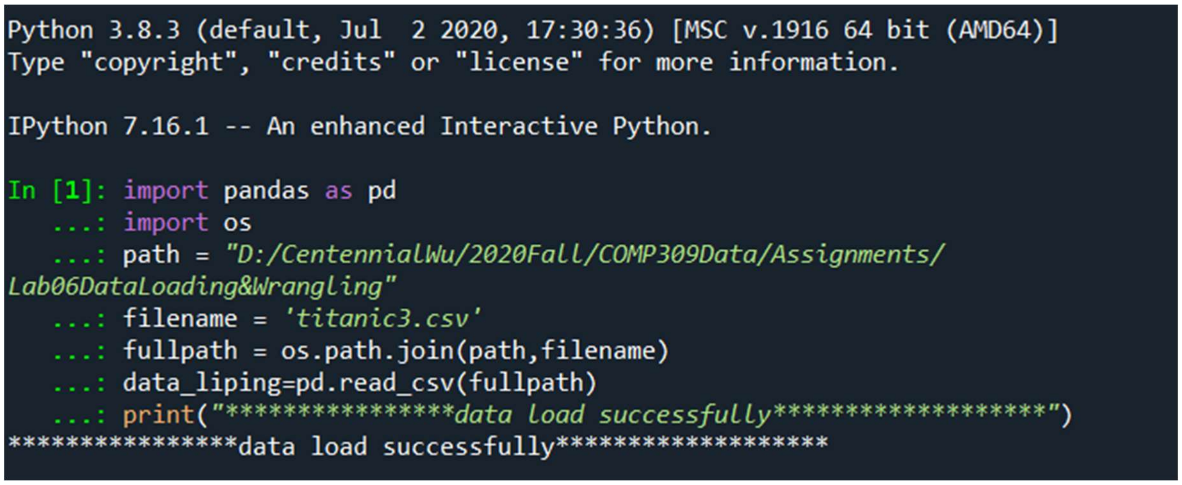
Steps for handling data basics:

1- Open your spyder IDE
2- Load the 'titanic3.csv' file into a dataframe name *the dataframe data_firstname where first name is your first name* carry out the following activities:
   a. Get the first five records
   b. Get the shape of the data
   c. Get the column values
   d. Create statistic summaries
   e. Get the types of columns

   Following is the code, *make sure you update the path to the correct path where you placed the files*:

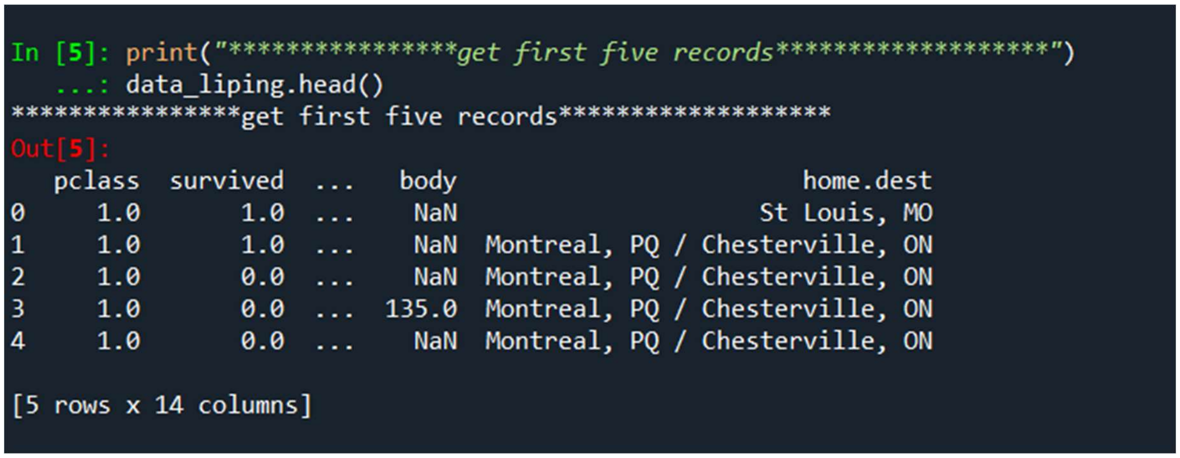
```
import pandas as pd
import os
path = "D:/CentennialWu/2020Fall/COMP309Data/Assignments/Lab06DataLoading&Wrangling"
filename = 'titanic3.csv'
fullpath = os.path.join(path,filename)
data_liping=pd.read_csv(fullpath)
```

print("***************data load successfully******************")



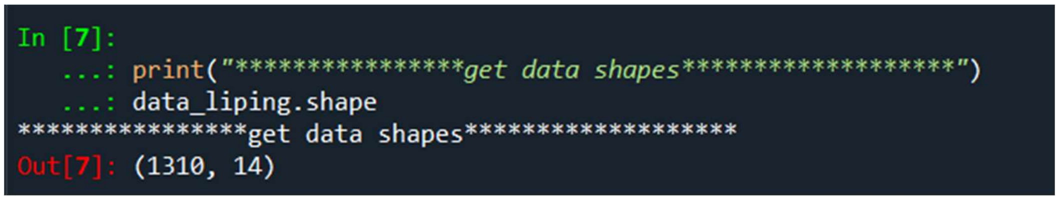
```
Python 3.8.3 (default, Jul  2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.16.1 -- An enhanced Interactive Python.

In [1]: import pandas as pd
   ...: import os
   ...: path = "D:/CentennialWu/2020Fall/COMP309Data/Assignments/
Lab06DataLoading&Wrangling"
   ...: filename = 'titanic3.csv'
   ...: fullpath = os.path.join(path,filename)
   ...: data_liping=pd.read_csv(fullpath)
   ...: print("***************data load successfully******************")
***************data load successfully******************
```

print("***************get first five records******************")
data_liping.head()



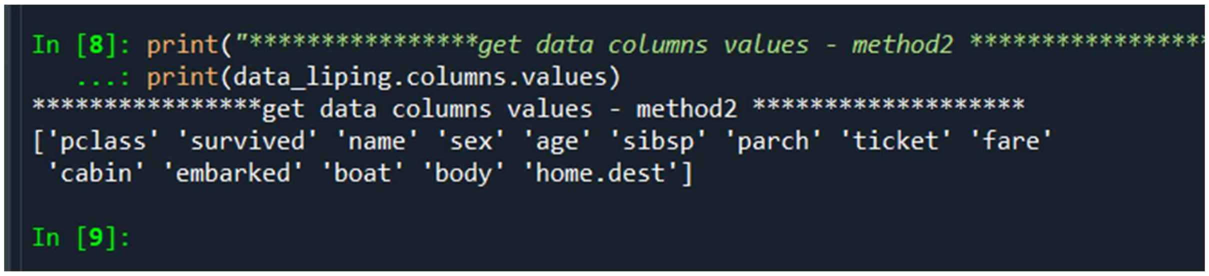
```
In [5]: print("***************get first five records******************")
   ...: data_liping.head()
***************get first five records******************
Out[5]:
   pclass  survived  ...   body                         home.dest
0     1.0       1.0  ...    NaN                      St Louis, MO
1     1.0       1.0  ...    NaN  Montreal, PQ / Chesterville, ON
2     1.0       0.0  ...    NaN  Montreal, PQ / Chesterville, ON
3     1.0       0.0  ...  135.0  Montreal, PQ / Chesterville, ON
4     1.0       0.0  ...    NaN  Montreal, PQ / Chesterville, ON

[5 rows x 14 columns]
```

print("***************get data shapes******************")
data_liping.shape



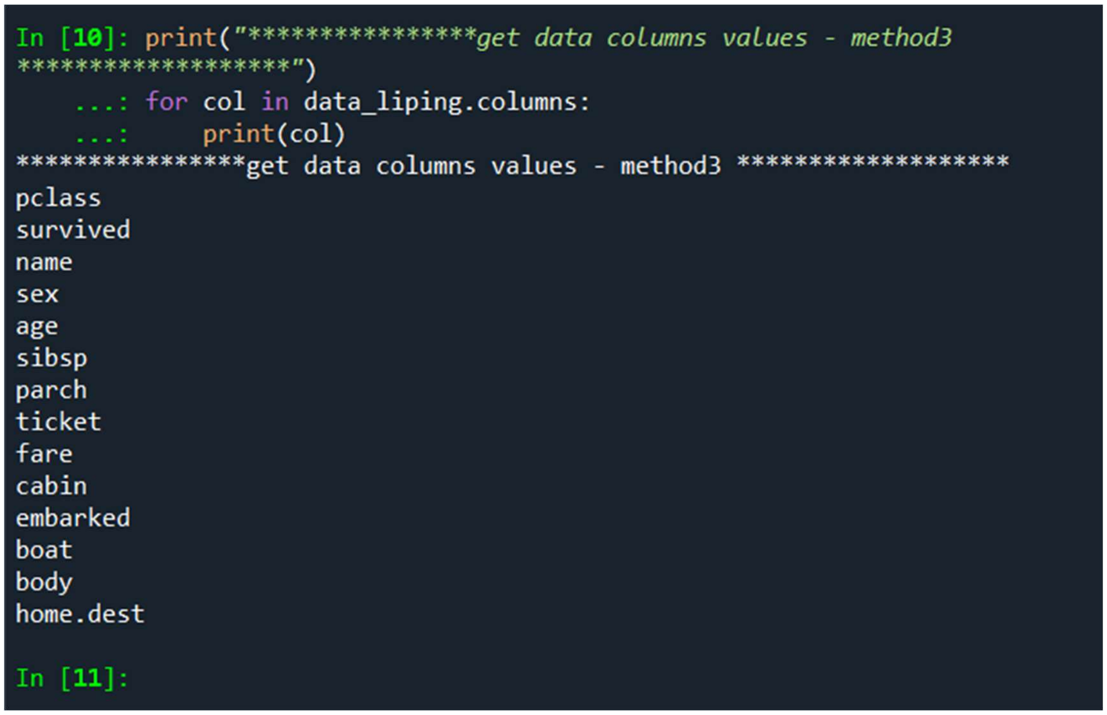
```
In [7]:
   ...: print("***************get data shapes******************")
   ...: data_liping.shape
***************get data shapes******************
Out[7]: (1310, 14)
```

print("***************get data columns values - method1 ******************")
data_liping.columns.values



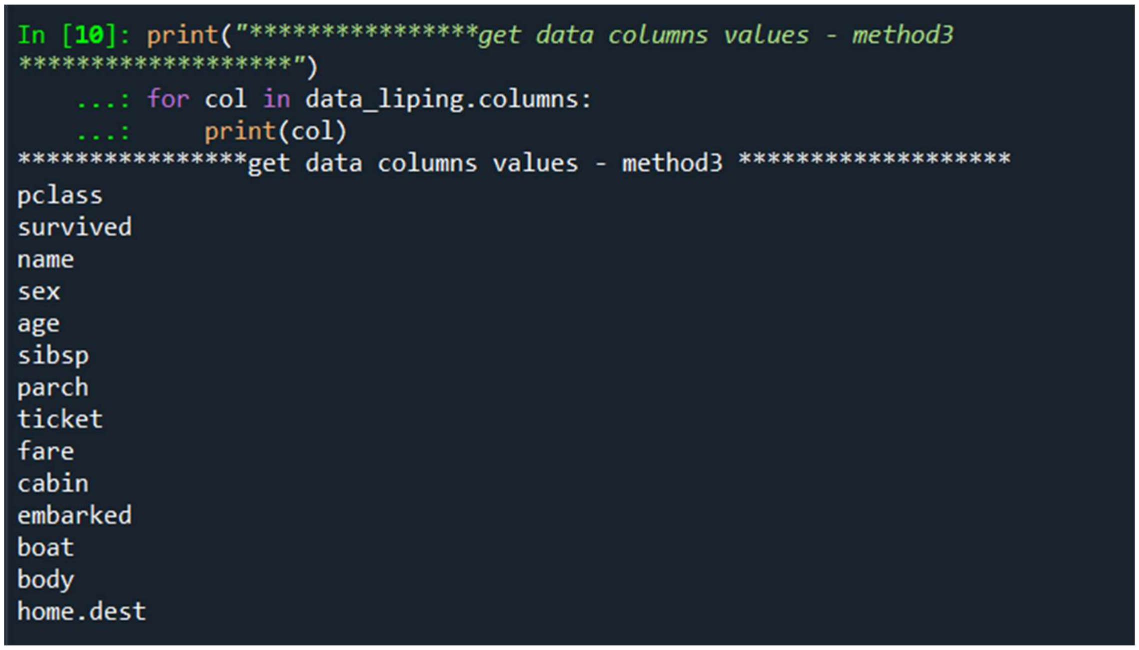
```
In [8]: print("***************get data columns values - method2 ***************
   ...: print(data_liping.columns.values)
***************get data columns values - method2 ******************
['pclass' 'survived' 'name' 'sex' 'age' 'sibsp' 'parch' 'ticket' 'fare'
 'cabin' 'embarked' 'boat' 'body' 'home.dest']

In [9]:
```

```
print("***************get data columns values - method2 ******************")
print(data_liping.columns.values)
```



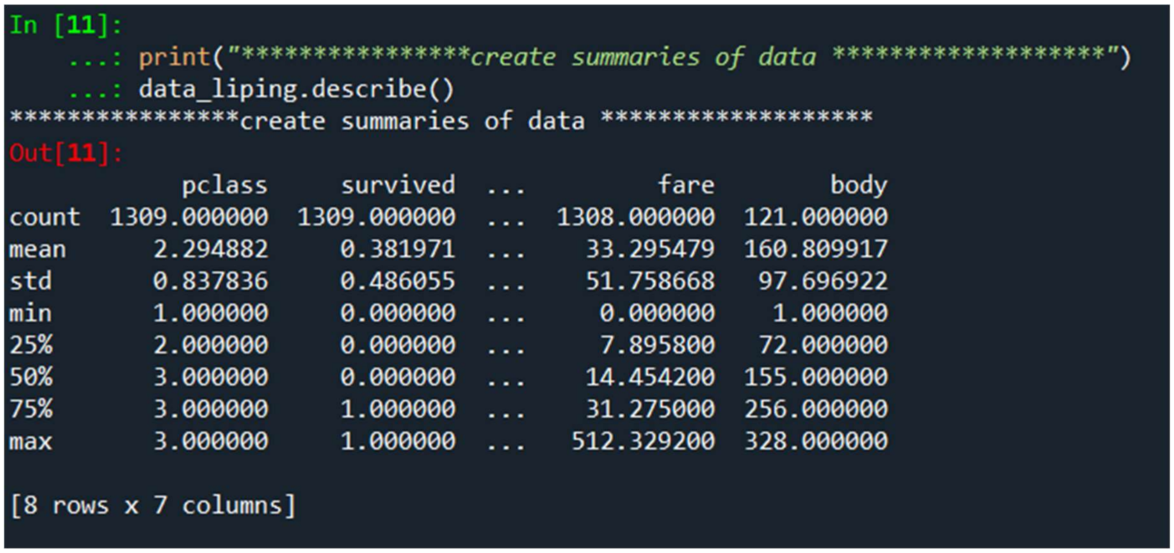
```
In [10]: print("***************get data columns values - method3
******************")
    ...: for col in data_liping.columns:
    ...:     print(col)
***************get data columns values - method3 ******************
pclass
survived
name
sex
age
sibsp
parch
ticket
fare
cabin
embarked
boat
body
home.dest

In [11]:
```

```
print("***************get data columns values - method3 ******************")
for col in data_liping.columns:
   print(col)
```



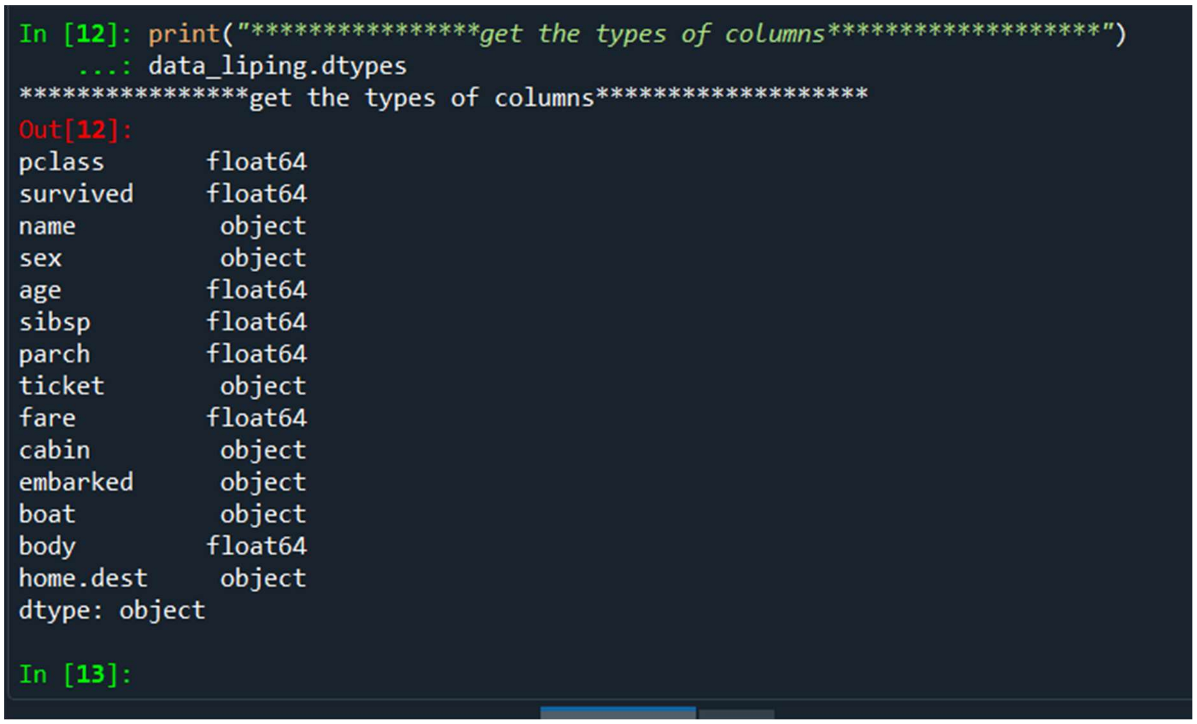
```
In [10]: print("***************get data columns values - method3
******************")
    ...: for col in data_liping.columns:
    ...:     print(col)
***************get data columns values - method3 ******************
pclass
survived
name
sex
age
sibsp
parch
ticket
fare
cabin
embarked
boat
body
home.dest
```

```
print("***************create summaries of data ******************")
data_liping.describe()
```



```
In [11]:
    ...: print("***************create summaries of data ******************")
    ...: data_liping.describe()
***************create summaries of data ******************
Out[11]:
            pclass     survived  ...         fare        body
count  1309.000000  1309.000000  ...  1308.000000  121.000000
mean      2.294882     0.381971  ...    33.295479  160.809917
std       0.837836     0.486055  ...    51.758668   97.696922
min       1.000000     0.000000  ...     0.000000    1.000000
25%       2.000000     0.000000  ...     7.895800   72.000000
50%       3.000000     0.000000  ...    14.454200  155.000000
75%       3.000000     1.000000  ...    31.275000  256.000000
max       3.000000     1.000000  ...   512.329200  328.000000

[8 rows x 7 columns]
```

```
print("***************get the types of columns******************")
data_liping.dtypes
```

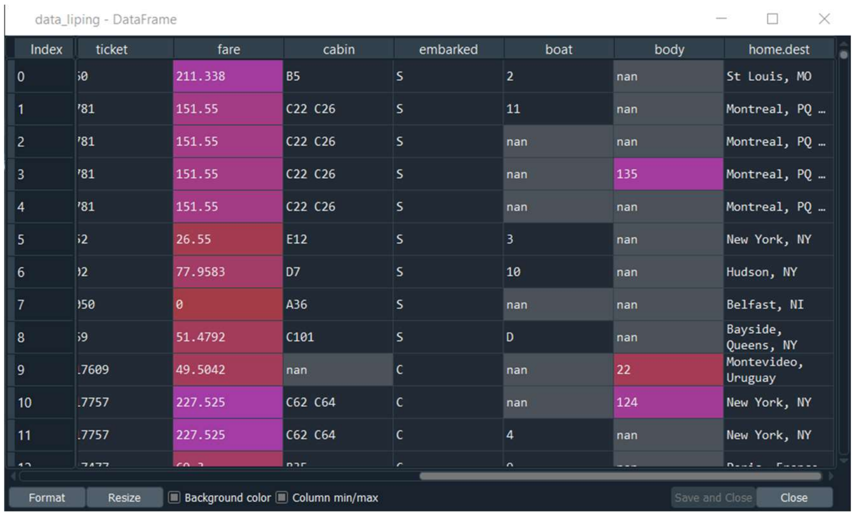


```
In [12]: print("***************get the types of columns******************")
    ...: data_liping.dtypes
***************get the types of columns******************
Out[12]:
pclass       float64
survived     float64
name          object
sex           object
age          float64
sibsp        float64
parch        float64
ticket        object
fare         float64
cabin         object
embarked      object
boat          object
body         float64
home.dest     object
dtype: object

In [13]:
```

3- Handling missing values as follows:
   a. Fill missing values with zero
   b. Fill missing values with a string and generate a new dataframe
   c. Fill the missing values for a specific column with a string
   d. Fill the missing values for a specific column with the mean of the column

Following is the code, *make sure you update the path to the correct path where you placed the files.*

```
####Imputation

# Fill the missing values with zeros

import pandas as pd

import os

path = "D:/CentennialWu/2020Fall/COMP309Data/Assignments/Lab06DataLoading&Wrangling"

filename = 'titanic3.csv'

fullpath = os.path.join(path,filename)

data_liping=pd.read_csv(fullpath)
```

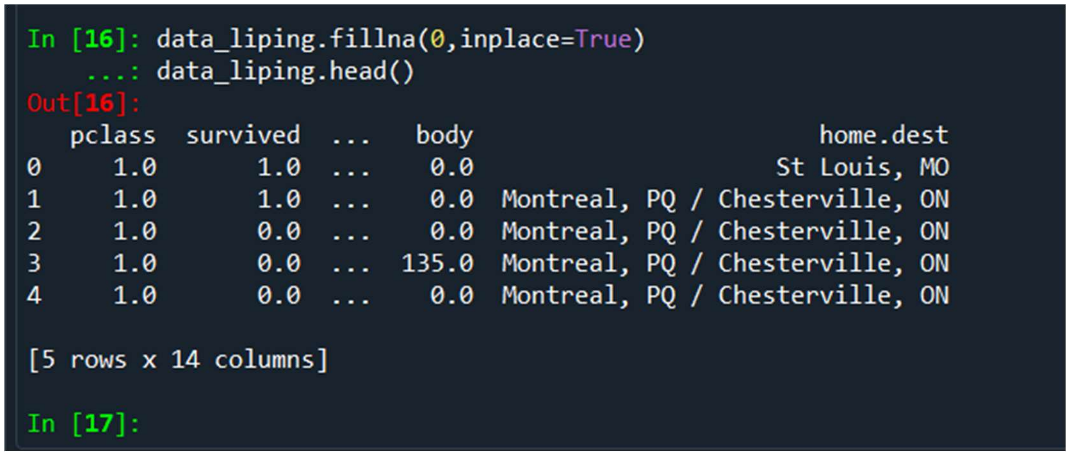


data_liping - DataFrame

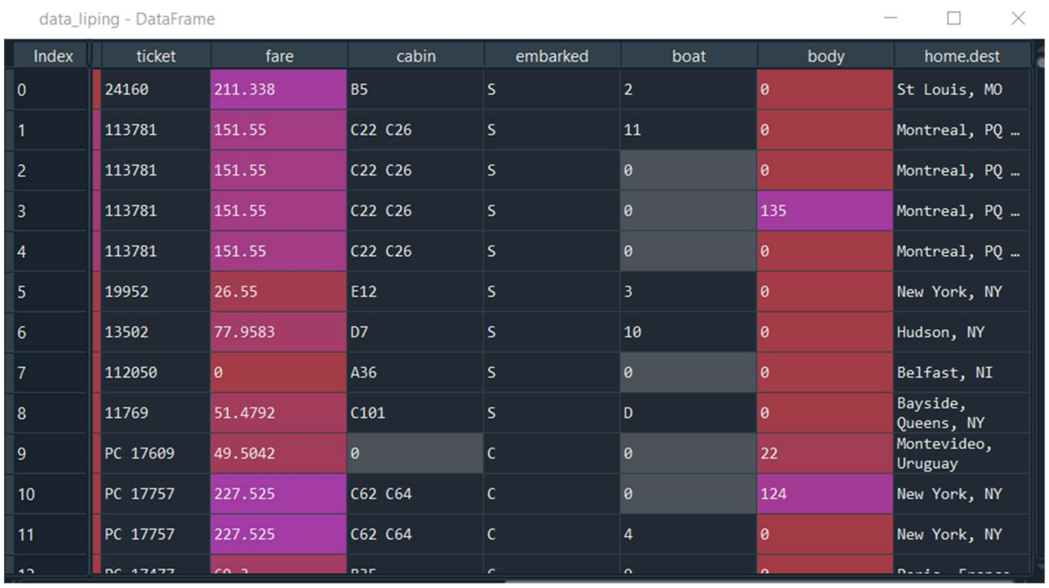
Index	ticket	fare	cabin	embarked	boat	body	home.dest
0	50	211.338	B5	S	2	nan	St Louis, MO
1	781	151.55	C22 C26	S	11	nan	Montreal, PQ …
2	781	151.55	C22 C26	S	nan	nan	Montreal, PQ …
3	781	151.55	C22 C26	S	nan	135	Montreal, PQ …
4	781	151.55	C22 C26	S	nan	nan	Montreal, PQ …
5	52	26.55	E12	S	3	nan	New York, NY
6	02	77.9583	D7	S	10	nan	Hudson, NY
7	050	0	A36	S	nan	nan	Belfast, NI
8	59	51.4792	C101	S	D	nan	Bayside, Queens, NY
9	17609	49.5042	nan	C	nan	22	Montevideo, Uruguay
10	17757	227.525	C62 C64	C	nan	124	New York, NY
11	17757	227.525	C62 C64	C	4	nan	New York, NY

data_liping.fillna(0,inplace=True)

data_liping.head()



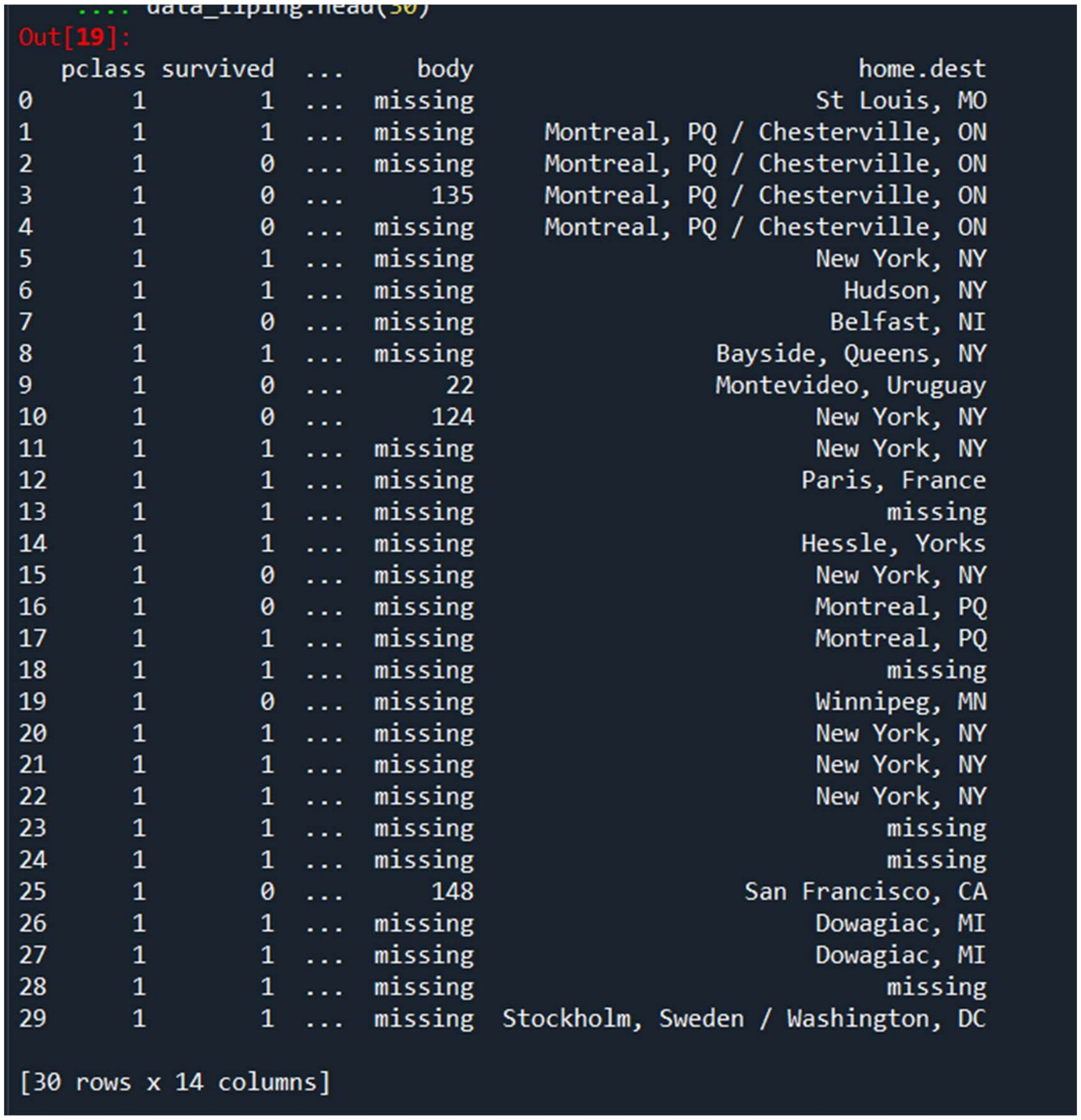
```
In [16]: data_liping.fillna(0,inplace=True)
    ...: data_liping.head()
Out[16]:
   pclass  survived  ...   body                         home.dest
0     1.0       1.0  ...    0.0                      St Louis, MO
1     1.0       1.0  ...    0.0  Montreal, PQ / Chesterville, ON
2     1.0       0.0  ...    0.0  Montreal, PQ / Chesterville, ON
3     1.0       0.0  ...  135.0  Montreal, PQ / Chesterville, ON
4     1.0       0.0  ...    0.0  Montreal, PQ / Chesterville, ON

[5 rows x 14 columns]

In [17]:
```



data_liping - DataFrame

Index	ticket	fare	cabin	embarked	boat	body	home.dest
0	24160	211.338	B5	S	2	0	St Louis, MO
1	113781	151.55	C22 C26	S	11	0	Montreal, PQ …
2	113781	151.55	C22 C26	S	0	0	Montreal, PQ …
3	113781	151.55	C22 C26	S	0	135	Montreal, PQ …
4	113781	151.55	C22 C26	S	0	0	Montreal, PQ …
5	19952	26.55	E12	S	3	0	New York, NY
6	13502	77.9583	D7	S	10	0	Hudson, NY
7	112050	0	A36	S	0	0	Belfast, NI
8	11769	51.4792	C101	S	D	0	Bayside, Queens, NY
9	PC 17609	49.5042	0	C	0	22	Montevideo, Uruguay
10	PC 17757	227.525	C62 C64	C	0	124	New York, NY
11	PC 17757	227.525	C62 C64	C	4	0	New York, NY

```
# Fill the missing values with "missing"
import pandas as pd
import os
path = "D:/CentennialWu/2020Fall/COMP309Data/Assignments/Lab06DataLoading&Wrangling"
filename = 'titanic3.csv'
fullpath = os.path.join(path,filename)
data_liping=pd.read_csv(fullpath)
data_liping.fillna("missing",inplace=True)
data_liping.head(30)
```



```
    ...: data_liping.head(30)
Out[19]:
    pclass survived  ...     body                        home.dest
0        1        1  ...  missing                     St Louis, MO
1        1        1  ...  missing  Montreal, PQ / Chesterville, ON
2        1        0  ...  missing  Montreal, PQ / Chesterville, ON
3        1        0  ...      135  Montreal, PQ / Chesterville, ON
4        1        0  ...  missing  Montreal, PQ / Chesterville, ON
5        1        1  ...  missing                     New York, NY
6        1        1  ...  missing                       Hudson, NY
7        1        0  ...  missing                      Belfast, NI
8        1        1  ...  missing              Bayside, Queens, NY
9        1        0  ...       22              Montevideo, Uruguay
10       1        0  ...      124                     New York, NY
11       1        1  ...  missing                     New York, NY
12       1        1  ...  missing                    Paris, France
13       1        1  ...  missing                          missing
14       1        1  ...  missing                    Hessle, Yorks
15       1        0  ...  missing                     New York, NY
16       1        0  ...  missing                     Montreal, PQ
17       1        1  ...  missing                     Montreal, PQ
18       1        1  ...  missing                          missing
19       1        0  ...  missing                     Winnipeg, MN
20       1        1  ...  missing                     New York, NY
21       1        1  ...  missing                     New York, NY
22       1        1  ...  missing                     New York, NY
23       1        1  ...  missing                          missing
24       1        1  ...  missing                          missing
25       1        0  ...      148                San Francisco, CA
26       1        1  ...  missing                     Dowagiac, MI
27       1        1  ...  missing                     Dowagiac, MI
28       1        1  ...  missing                          missing
29       1        1  ...  missing Stockholm, Sweden / Washington, DC

[30 rows x 14 columns]
```

# fill only a column

import pandas as pd

import os

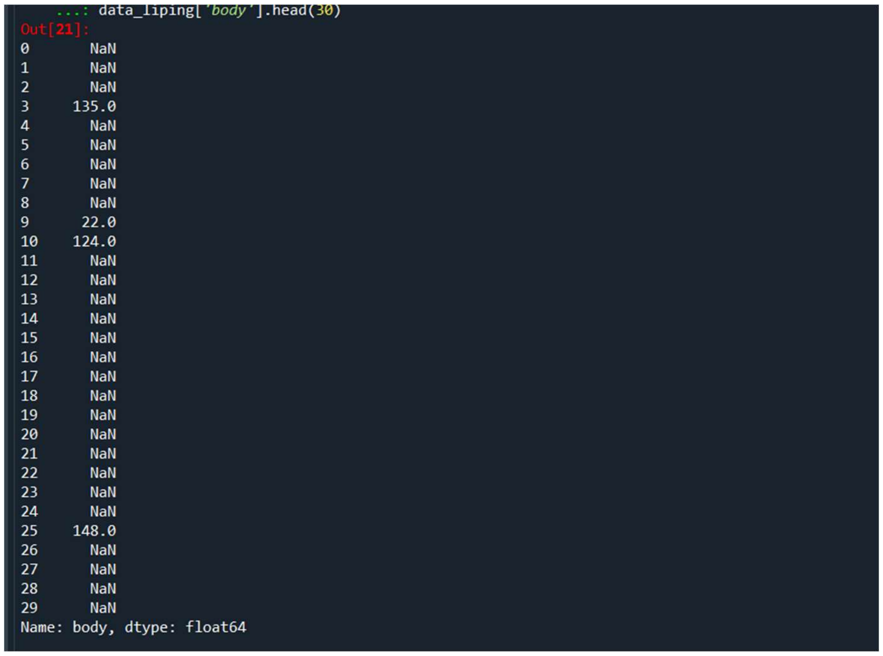
path = "D:/CentennialWu/2020Fall/COMP309Data/Assignments/Lab06DataLoading&Wrangling"

filename = 'titanic3.csv'

fullpath = os.path.join(path,filename)

data_liping=pd.read_csv(fullpath)

data_liping['body'].head(30)

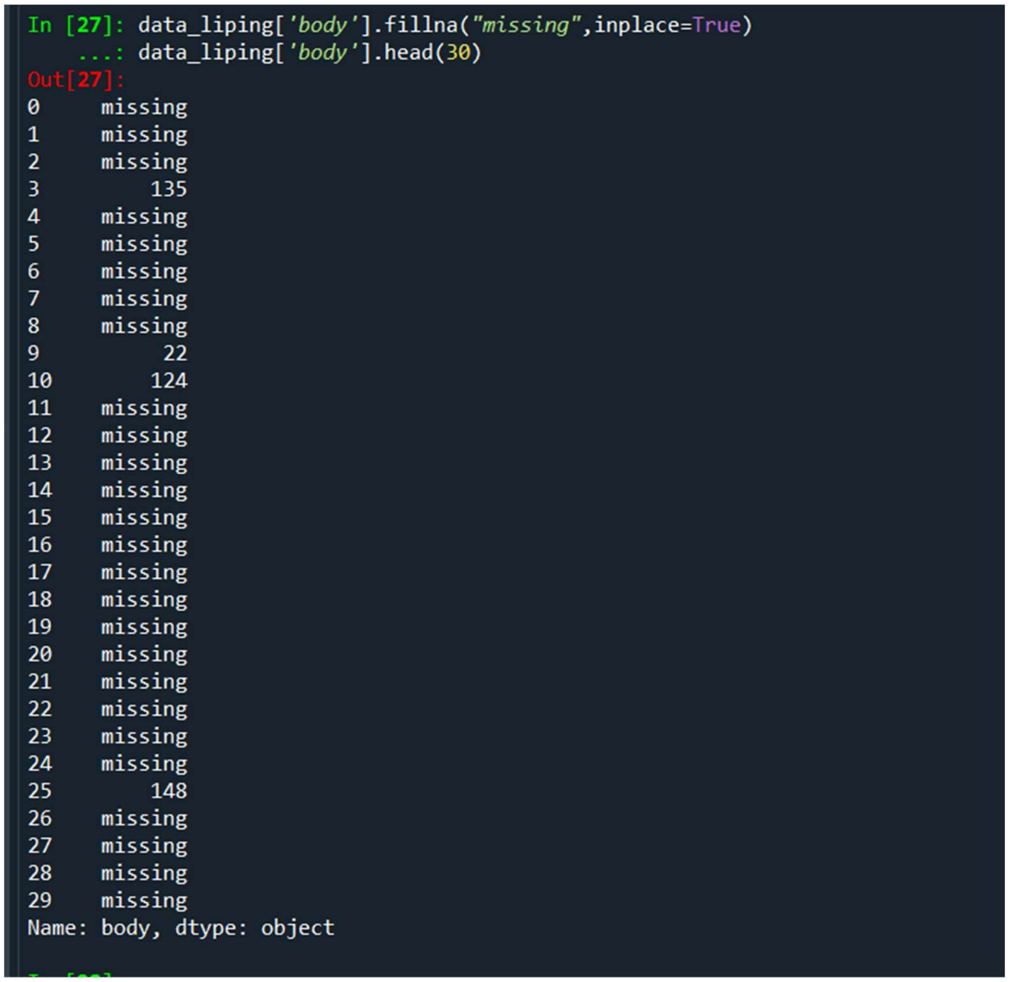


```
     ...: data_liping['body'].head(30)
Out[21]:
0        NaN
1        NaN
2        NaN
3      135.0
4        NaN
5        NaN
6        NaN
7        NaN
8        NaN
9       22.0
10     124.0
11       NaN
12       NaN
13       NaN
14       NaN
15       NaN
16       NaN
17       NaN
18       NaN
19       NaN
20       NaN
21       NaN
22       NaN
23       NaN
24       NaN
25     148.0
26       NaN
27       NaN
28       NaN
29       NaN
Name: body, dtype: float64
```

##

data_liping['body'].fillna("missing",inplace=True)

data_liping['body'].head(30)



```
In [27]: data_liping['body'].fillna("missing",inplace=True)
    ...: data_liping['body'].head(30)
Out[27]:
0     missing
1     missing
2     missing
3         135
4     missing
5     missing
6     missing
7     missing
8     missing
9          22
10        124
11    missing
12    missing
13    missing
14    missing
15    missing
16    missing
17    missing
18    missing
19    missing
20    missing
21    missing
22    missing
23    missing
24    missing
25        148
26    missing
27    missing
28    missing
29    missing
Name: body, dtype: object
```

# use the average to fill in the missing age

import pandas as pd

import os

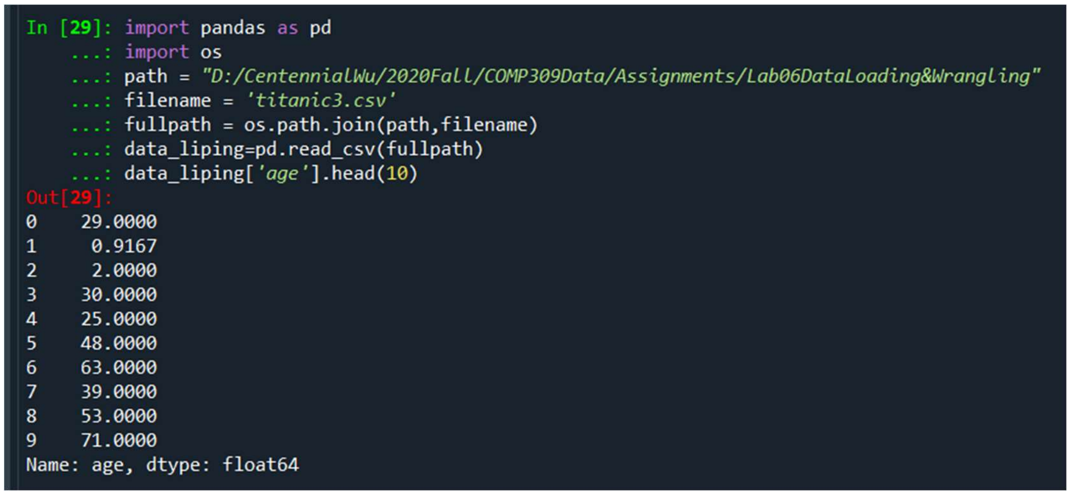
path = "D:/CentennialWu/2020Fall/COMP309Data/Assignments/Lab06DataLoading&Wrangling"

filename = 'titanic3.csv'

fullpath = os.path.join(path,filename)

data_liping=pd.read_csv(fullpath)

data_liping['age'].head(10)

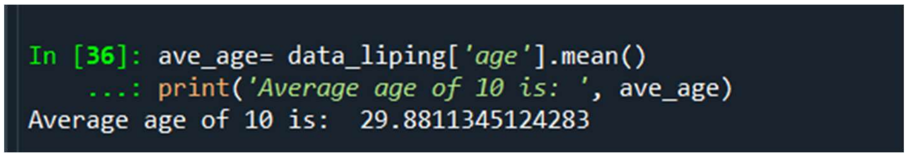


```
In [29]: import pandas as pd
    ...: import os
    ...: path = "D:/CentennialWu/2020Fall/COMP309Data/Assignments/Lab06DataLoading&Wrangling"
    ...: filename = 'titanic3.csv'
    ...: fullpath = os.path.join(path,filename)
    ...: data_liping=pd.read_csv(fullpath)
    ...: data_liping['age'].head(10)
Out[29]:
0    29.0000
1     0.9167
2     2.0000
3    30.0000
4    25.0000
5    48.0000
6    63.0000
7    39.0000
8    53.0000
9    71.0000
Name: age, dtype: float64
```

## get the age mean

ave_age= data_liping['age'].mean()

print('Average age of 10 is: ', ave_age)

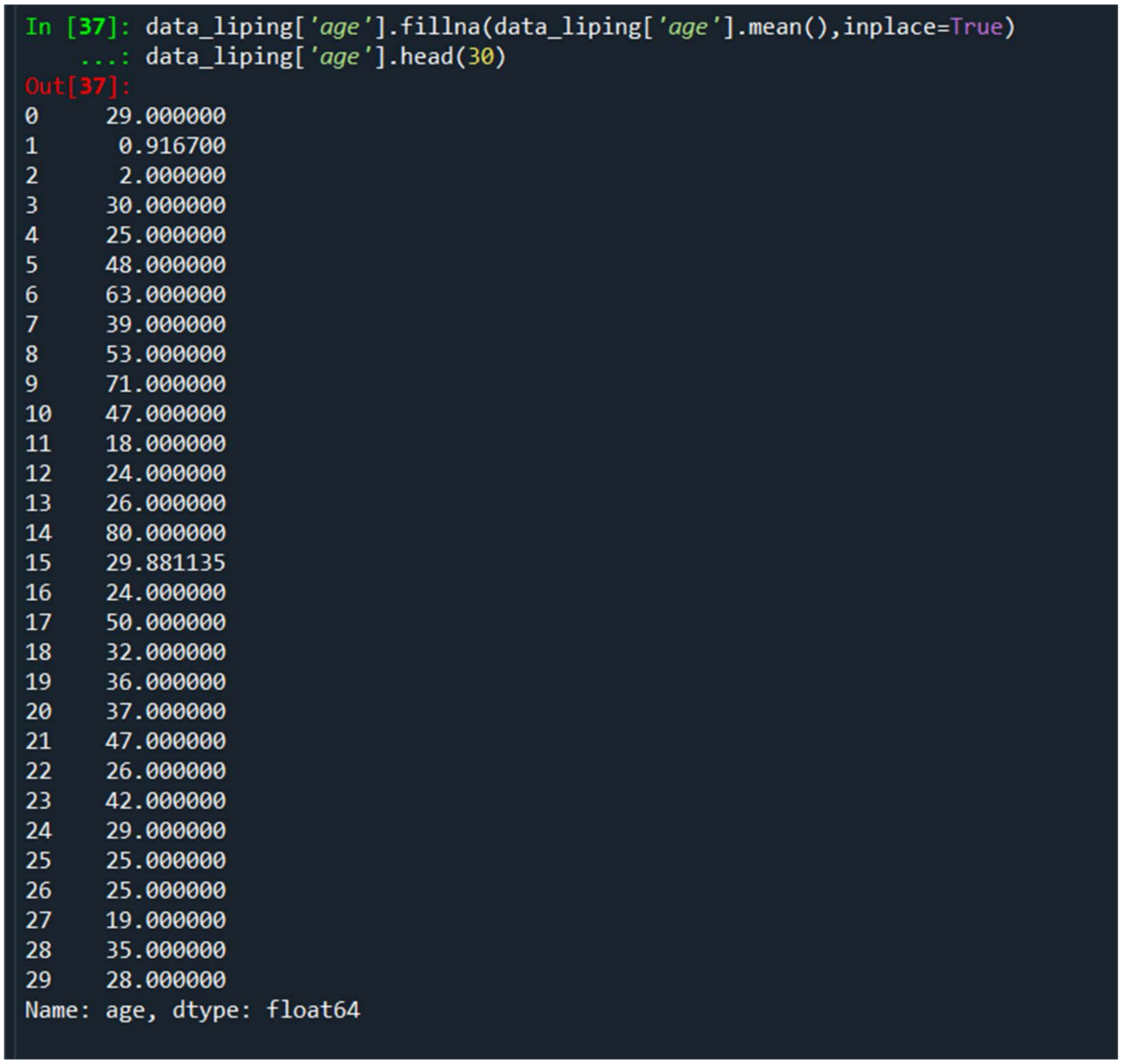


```
In [36]: ave_age= data_liping['age'].mean()
    ...: print('Average age of 10 is: ', ave_age)
Average age of 10 is:  29.8811345124283
```

##

data_liping['age'].fillna(data_liping['age'].mean(),inplace=True)

data_liping['age'].head(30)



```
In [37]: data_liping['age'].fillna(data_liping['age'].mean(),inplace=True)
    ...: data_liping['age'].head(30)
Out[37]:
0      29.000000
1       0.916700
2       2.000000
3      30.000000
4      25.000000
5      48.000000
6      63.000000
7      39.000000
8      53.000000
9      71.000000
10     47.000000
11     18.000000
12     24.000000
13     26.000000
14     80.000000
15     29.881135
16     24.000000
17     50.000000
18     32.000000
19     36.000000
20     37.000000
21     47.000000
22     26.000000
23     42.000000
24     29.000000
25     25.000000
26     25.000000
27     19.000000
28     35.000000
29     28.000000
Name: age, dtype: float64
```

4- Creating Dummy variables for categorical data, as follows:

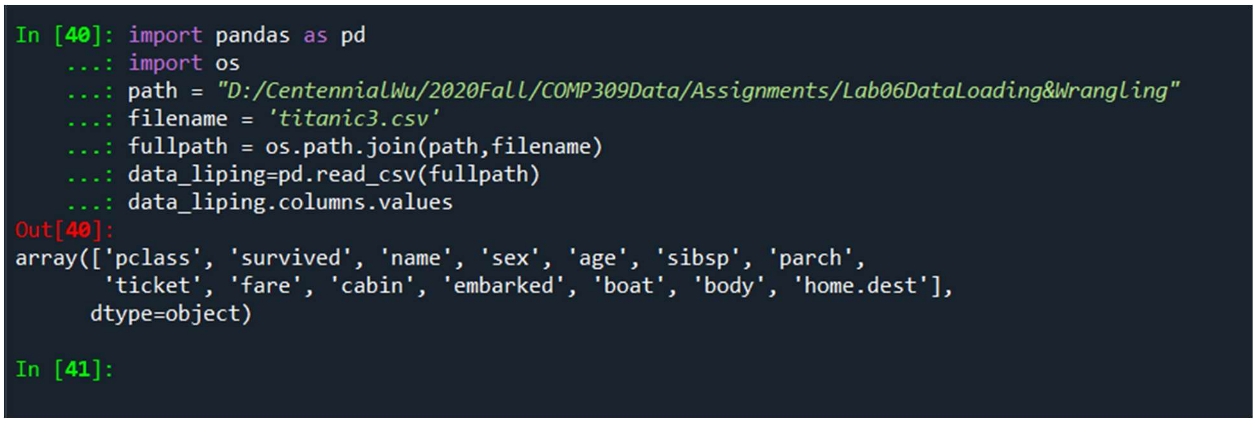
a. Load the titanic file
b. Create a dummy data frame
c. Remove the old column and join the new columns

Following is the code, *make sure you update the path to the correct path where you placed the files change my firstname to your firstname:*

```
import pandas as pd

import os

path = "D:/CentennialWu/2020Fall/COMP309Data/Assignments/Lab06DataLoading&Wrangling"

filename = 'titanic3.csv'

fullpath = os.path.join(path,filename)
```

data_liping=pd.read_csv(fullpath)

data_liping.columns.values

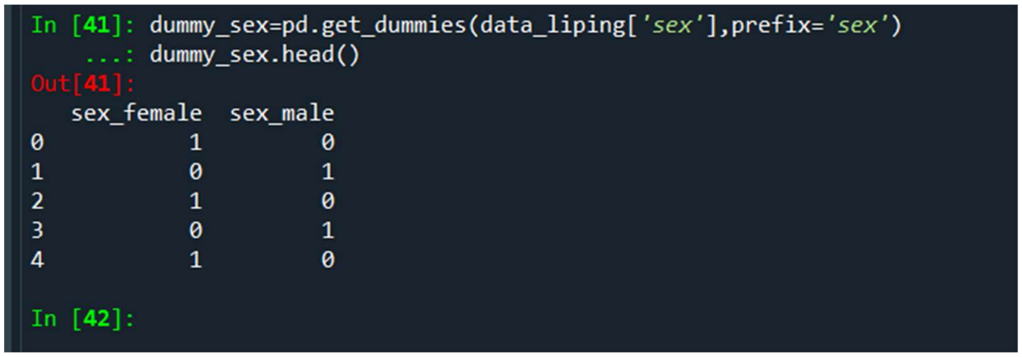


```
In [40]: import pandas as pd
    ...: import os
    ...: path = "D:/CentennialWu/2020Fall/COMP309Data/Assignments/Lab06DataLoading&Wrangling"
    ...: filename = 'titanic3.csv'
    ...: fullpath = os.path.join(path,filename)
    ...: data_liping=pd.read_csv(fullpath)
    ...: data_liping.columns.values
Out[40]:
array(['pclass', 'survived', 'name', 'sex', 'age', 'sibsp', 'parch',
       'ticket', 'fare', 'cabin', 'embarked', 'boat', 'body', 'home.dest'],
      dtype=object)

In [41]:
```

# create dummy dataframe

dummy_sex=pd.get_dummies(data_liping['sex'],prefix='sex')

dummy_sex.head()



```
In [41]: dummy_sex=pd.get_dummies(data_liping['sex'],prefix='sex')
    ...: dummy_sex.head()
Out[41]:
   sex_female  sex_male
0           1         0
1           0         1
2           1         0
3           0         1
4           1         0

In [42]:
```

# join the dummy dataframe to the original dataset and remove the original column

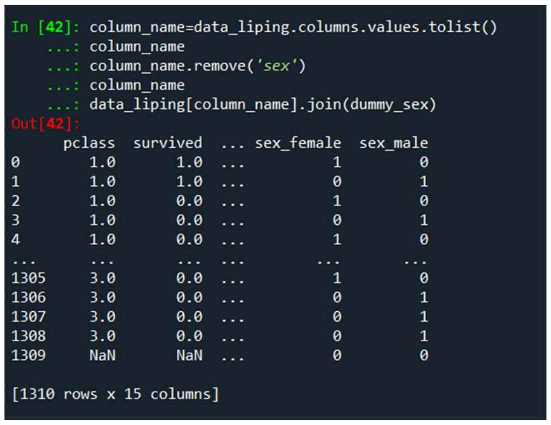
column_name=data_liping.columns.values.tolist()

column_name

column_name.remove('sex')

column_name

data_liping[column_name].join(dummy_sex)



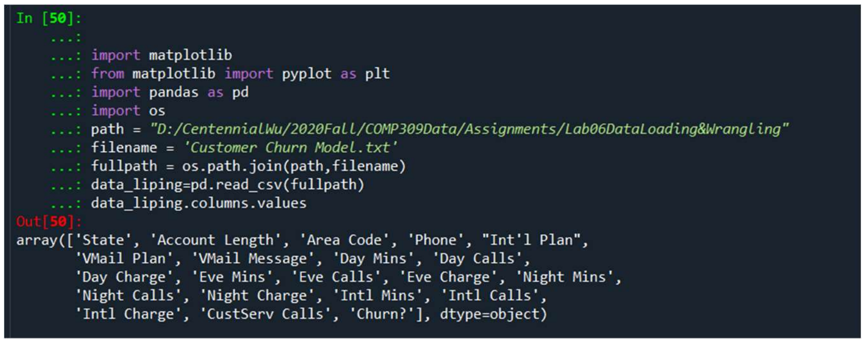
```
In [42]: column_name=data_liping.columns.values.tolist()
    ...: column_name
    ...: column_name.remove('sex')
    ...: column_name
    ...: data_liping[column_name].join(dummy_sex)
Out[42]:
      pclass  survived  ... sex_female  sex_male
0        1.0       1.0  ...          1         0
1        1.0       1.0  ...          0         1
2        1.0       0.0  ...          1         0
3        1.0       0.0  ...          0         1
4        1.0       0.0  ...          1         0
...      ...       ...  ...        ...       ...
1305     3.0       0.0  ...          1         0
1306     3.0       0.0  ...          0         1
1307     3.0       0.0  ...          0         1
1308     3.0       0.0  ...          0         1
1309     NaN       NaN  ...          0         0

[1310 rows x 15 columns]
```

5- Visualize the data using basic plots as follows:

a. Load the data
b. Create a scatter plot
c. Save a scatter plot
d. Create a multiple scatter plot
e. Create a histogram
f. Create a boxplot

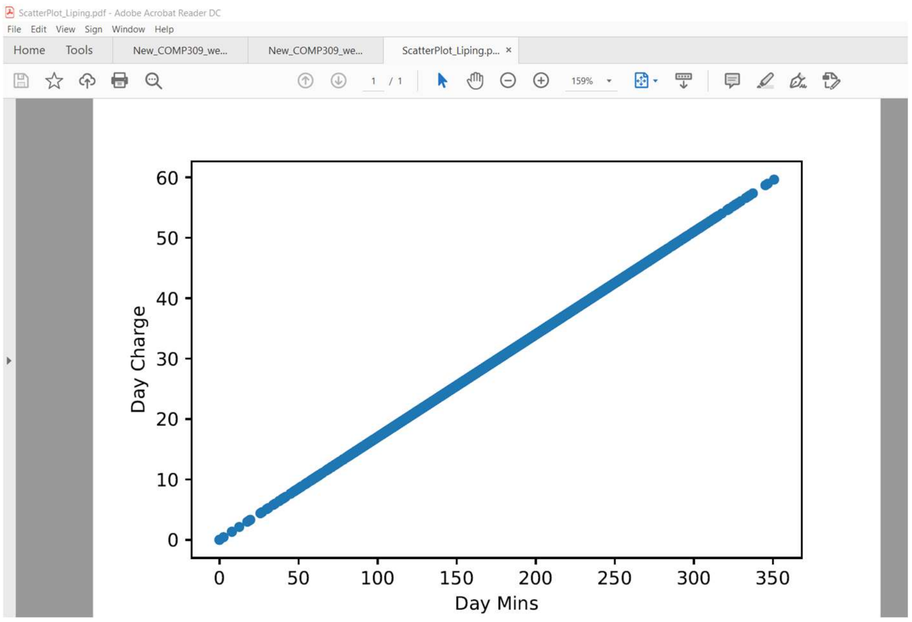
Following is the code, *make sure you update the path to the correct path where you placed the files change my firstname to your firstname:*

```
import matplotlib
from matplotlib import pyplot as plt
import pandas as pd
import os
path = "D:/CentennialWu/2020Fall/COMP309Data/Assignments/Lab06DataLoading&Wrangling"
filename = 'Customer Churn Model.txt'
fullpath = os.path.join(path,filename)
data_liping=pd.read_csv(fullpath)
data_liping.columns.values
```

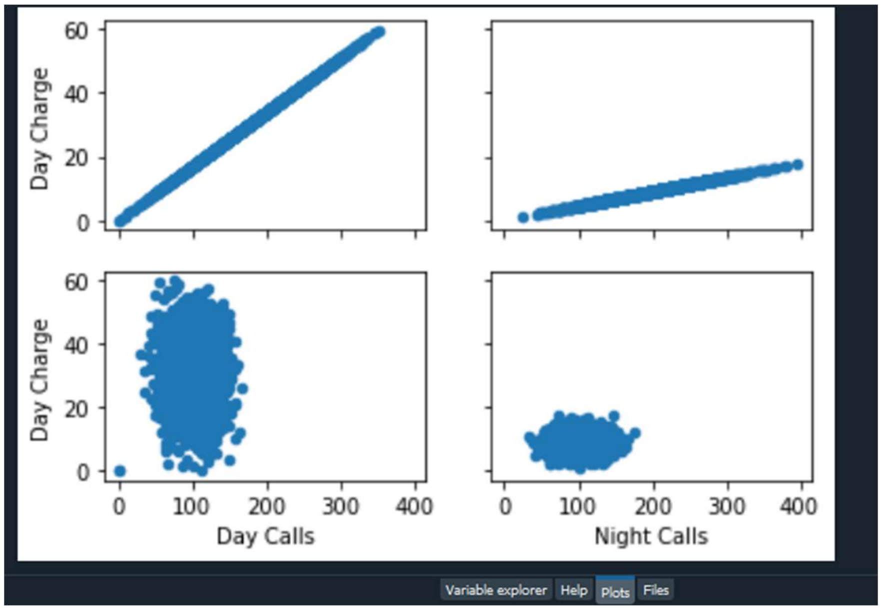


```
In [50]:
    ...:
    ...: import matplotlib
    ...: from matplotlib import pyplot as plt
    ...: import pandas as pd
    ...: import os
    ...: path = "D:/CentennialWu/2020Fall/COMP309Data/Assignments/Lab06DataLoading&Wrangling"
    ...: filename = 'Customer Churn Model.txt'
    ...: fullpath = os.path.join(path,filename)
    ...: data_liping=pd.read_csv(fullpath)
    ...: data_liping.columns.values
Out[50]:
array(['State', 'Account Length', 'Area Code', 'Phone', "Int'l Plan",
       'VMail Plan', 'VMail Message', 'Day Mins', 'Day Calls',
       'Day Charge', 'Eve Mins', 'Eve Calls', 'Eve Charge', 'Night Mins',
       'Night Calls', 'Night Charge', 'Intl Mins', 'Intl Calls',
       'Intl Charge', 'CustServ Calls', 'Churn?'], dtype=object)
```

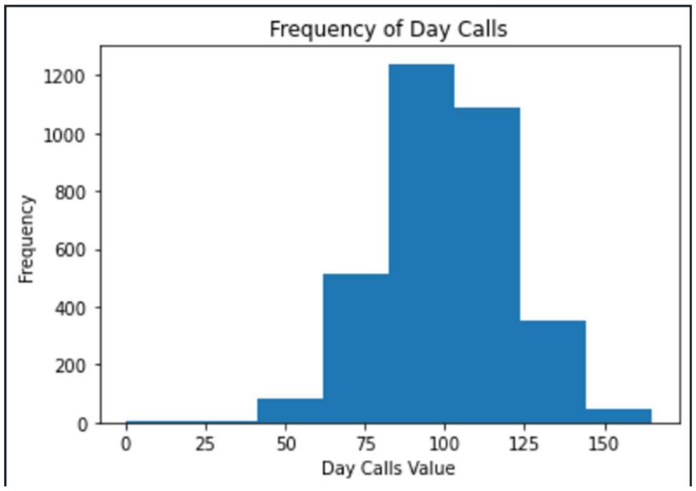
```
#create a scatterplot
fig_liping = data_liping.plot(kind='scatter',x='Day Mins',y='Day Charge')

# Save the scatter plot
figfilename = "ScatterPlot_Liping.pdf"
figfullpath = os.path.join(path, figfilename)
fig_liping.figure.savefig(figfullpath)
```


ScatterPlot_Liping.pdf - Adobe Acrobat Reader DC
File Edit View Sign Window Help
Home Tools New_COMP309_we... New_COMP309_we... ScatterPlot_Liping.p...
1 / 1
159%
Day Charge
Day Mins
0 10 20 30 40 50 60
0 50 100 150 200 250 300 350

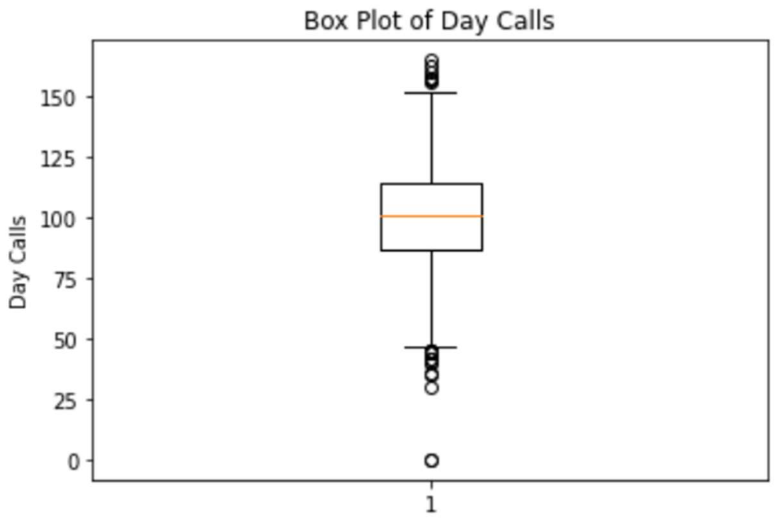
```
# Plot multiple charts
help(plt.subplot)
import matplotlib.pyplot as plt
figure_liping,axs = plt.subplots(2, 2,sharey=True,sharex=True)
data_liping.plot(kind='scatter',x='Day Mins',y='Day Charge',ax=axs[0][0])
data_liping.plot(kind='scatter',x='Night Mins',y='Night Charge',ax=axs[0][1])
data_liping.plot(kind='scatter',x='Day Calls',y='Day Charge',ax=axs[1][0])
data_liping.plot(kind='scatter',x='Night Calls',y='Night Charge',ax=axs[1][1])
```



```
# Plot a histogram
import matplotlib.pyplot as plt
hist_liping= plt.hist(data_liping['Day Calls'],bins=8)
plt.xlabel('Day Calls Value')
plt.ylabel('Frequency')
plt.title('Frequency of Day Calls')
```



```
# Plot a boxplot

import matplotlib.pyplot as plt

plt.boxplot(data_liping['Day Calls'])

plt.ylabel('Day Calls')
plt.title('Box Plot of Day Calls')
```

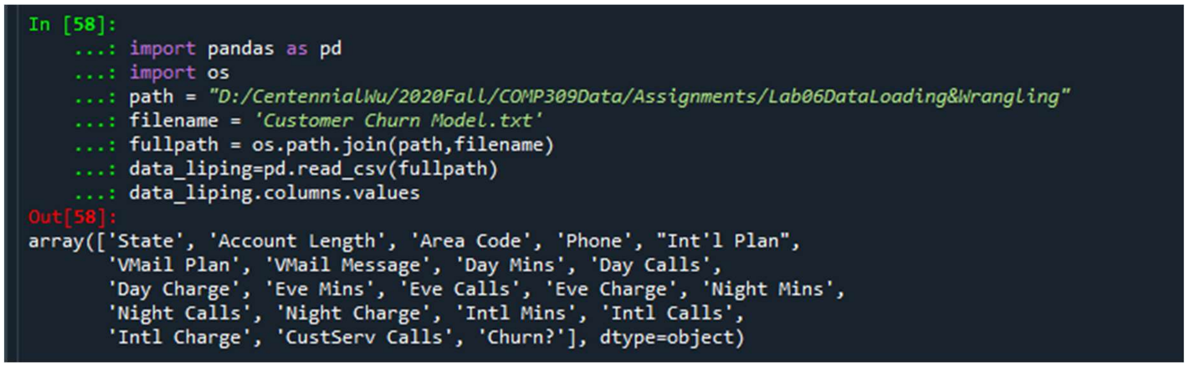


Slice and dice the data as follows:

g. Load the Churn data file
h. extract one column
i. extract many columns into a new dataframe
j. create a list of wanted columns
k. Select the first 50 rows
l. Select 50 rows starting at 25
m. Filter the rows that have clocked day Mins to be greater than 350
n. Filter the rows for which the state is VA
o. Filter the rows that have clocked day Mins to be greater than 150 and the state value is VA
p. Create a new column for total minutes

Following is the code, *make sure you update the path to the correct path where you placed the files change my firstname to your firstname*

```
#Sub setting the data slicing and dicing
import pandas as pd
import os
path = "D:/CentennialWu/2020Fall/COMP309Data/Assignments/Lab06DataLoading&Wrangling"
filename = 'Customer Churn Model.txt'
fullpath = os.path.join(path,filename)
data_liping=pd.read_csv(fullpath)
data_liping.columns.values
```



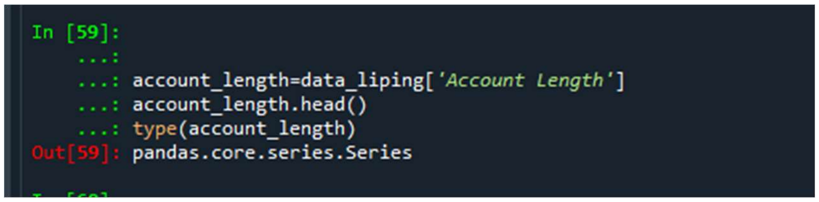
```
In [58]:
    ...: import pandas as pd
    ...: import os
    ...: path = "D:/CentennialWu/2020Fall/COMP309Data/Assignments/Lab06DataLoading&Wrangling"
    ...: filename = 'Customer Churn Model.txt'
    ...: fullpath = os.path.join(path,filename)
    ...: data_liping=pd.read_csv(fullpath)
    ...: data_liping.columns.values
Out[58]:
array(['State', 'Account Length', 'Area Code', 'Phone', "Int'l Plan",
       'VMail Plan', 'VMail Message', 'Day Mins', 'Day Calls',
       'Day Charge', 'Eve Mins', 'Eve Calls', 'Eve Charge', 'Night Mins',
       'Night Calls', 'Night Charge', 'Intl Mins', 'Intl Calls',
       'Intl Charge', 'CustServ Calls', 'Churn?'], dtype=object)
```

# extract one column (i.e. a series)

account_length=data_liping['Account Length']

account_length.head()

type(account_length)



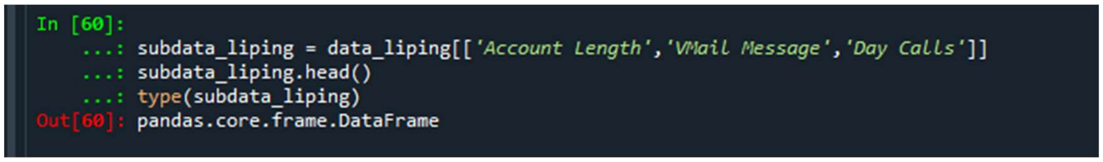
```
In [59]:
    ...:
    ...: account_length=data_liping['Account Length']
    ...: account_length.head()
    ...: type(account_length)
Out[59]: pandas.core.series.Series
```

#extract many columns into a new dataframe

subdata_liping = data_liping[['Account Length','VMail Message','Day Calls']]

subdata_liping.head()

type(subdata_liping)



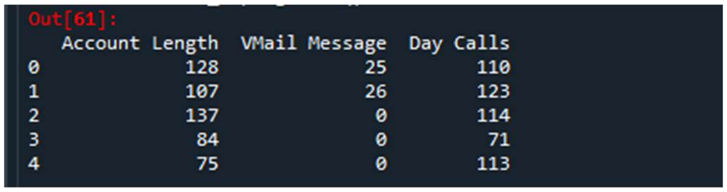
```
In [60]:
    ...: subdata_liping = data_liping[['Account Length','VMail Message','Day Calls']]
    ...: subdata_liping.head()
    ...: type(subdata_liping)
Out[60]: pandas.core.frame.DataFrame
```

# Create a list of wanted columns

wanted_columns=['Account Length','VMail Message','Day Calls']

subdata_liping=data_liping[wanted_columns]

subdata_liping.head()



```
Out[61]:
   Account Length  VMail Message  Day Calls
0             128             25        110
1             107             26        123
2             137              0        114
3              84              0         71
4              75              0        113
```

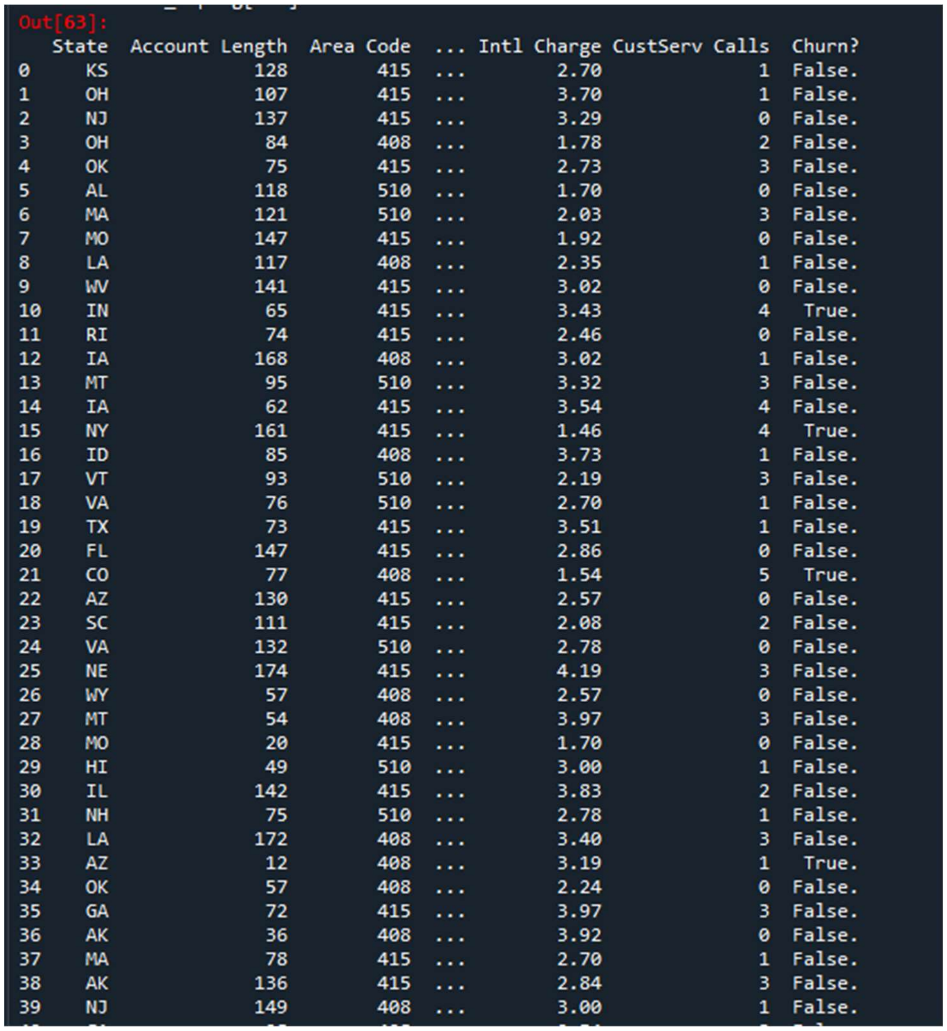
## Another way useful when many columns

wanted=['Account Length','VMail Message','Day Calls']

column_list=data_liping.columns.values.tolist()

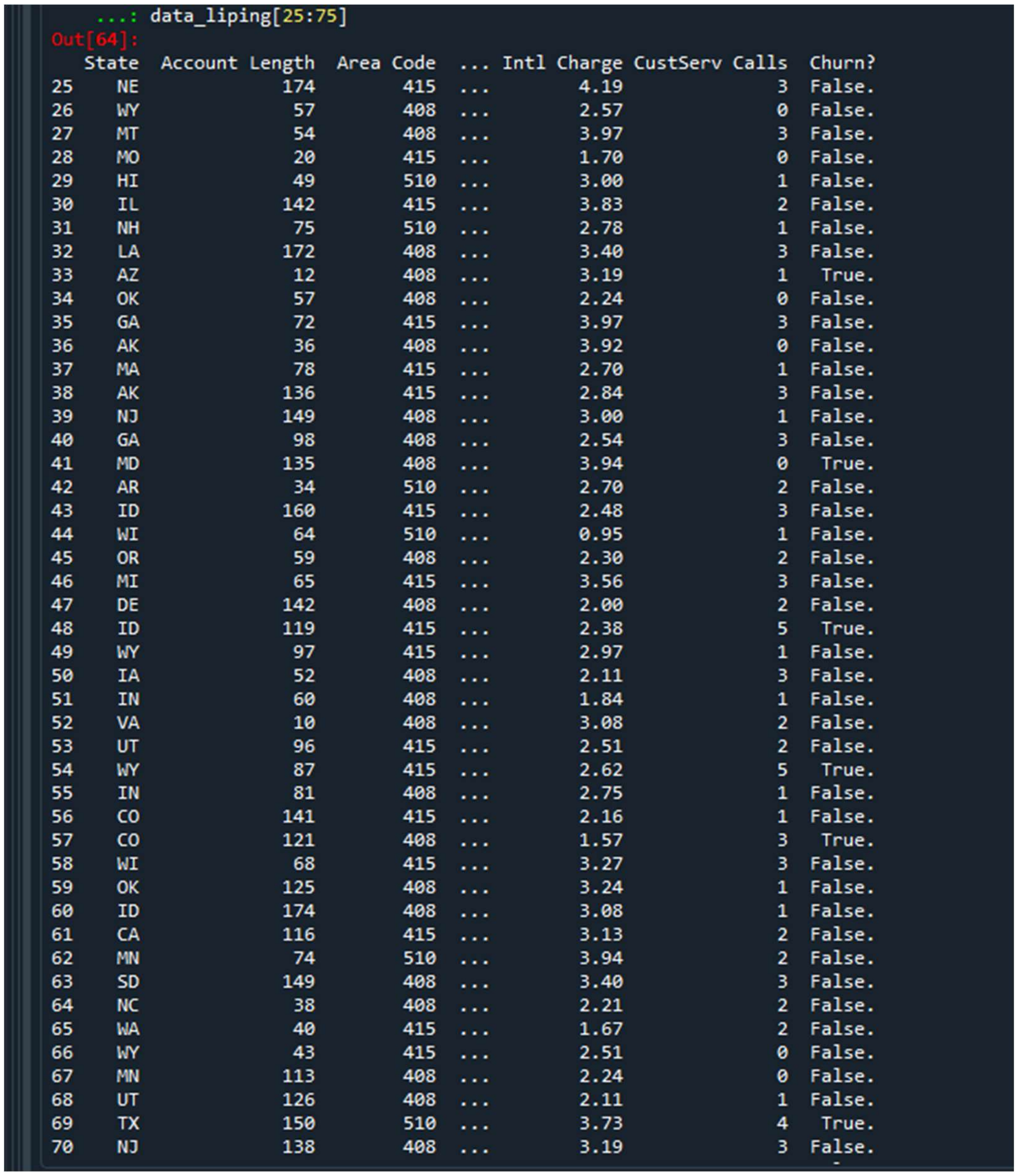
sublist=[x for x in column_list if x not in wanted]
subdata=data_liping[sublist]
subdata_liping.head()

## Rows
#Select the first 50 rows
data_liping[:50]



```
Out[63]:
    State  Account Length  Area Code  ...  Intl Charge CustServ Calls  Churn?
0      KS             128        415  ...         2.70              1  False.
1      OH             107        415  ...         3.70              1  False.
2      NJ             137        415  ...         3.29              0  False.
3      OH              84        408  ...         1.78              2  False.
4      OK              75        415  ...         2.73              3  False.
5      AL             118        510  ...         1.70              0  False.
6      MA             121        510  ...         2.03              3  False.
7      MO             147        415  ...         1.92              0  False.
8      LA             117        408  ...         2.35              1  False.
9      WV             141        415  ...         3.02              0  False.
10     IN              65        415  ...         3.43              4   True.
11     RI              74        415  ...         2.46              0  False.
12     IA             168        408  ...         3.02              1  False.
13     MT              95        510  ...         3.32              3  False.
14     IA              62        415  ...         3.54              4  False.
15     NY             161        415  ...         1.46              4   True.
16     ID              85        408  ...         3.73              1  False.
17     VT              93        510  ...         2.19              3  False.
18     VA              76        510  ...         2.70              1  False.
19     TX              73        415  ...         3.51              1  False.
20     FL             147        415  ...         2.86              0  False.
21     CO              77        408  ...         1.54              5   True.
22     AZ             130        415  ...         2.57              0  False.
23     SC             111        415  ...         2.08              2  False.
24     VA             132        510  ...         2.78              0  False.
25     NE             174        415  ...         4.19              3  False.
26     WY              57        408  ...         2.57              0  False.
27     MT              54        408  ...         3.97              3  False.
28     MO              20        415  ...         1.70              0  False.
29     HI              49        510  ...         3.00              1  False.
30     IL             142        415  ...         3.83              2  False.
31     NH              75        510  ...         2.78              1  False.
32     LA             172        408  ...         3.40              3  False.
33     AZ              12        408  ...         3.19              1   True.
34     OK              57        408  ...         2.24              0  False.
35     GA              72        415  ...         3.97              3  False.
36     AK              36        408  ...         3.92              0  False.
37     MA              78        415  ...         2.70              1  False.
38     AK             136        415  ...         2.84              3  False.
39     NJ             149        408  ...         3.00              1  False.
```

# select 50 rows starting at 25
data_liping[25:75]



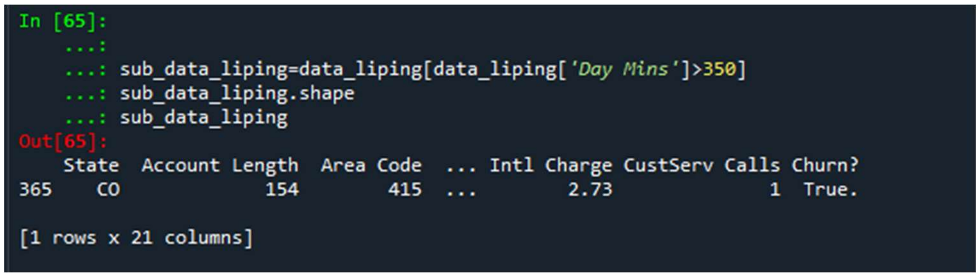
```
    ...: data_liping[25:75]
Out[64]:
   State  Account Length  Area Code  ... Intl Charge CustServ Calls  Churn?
25    NE             174        415  ...        4.19              3  False.
26    WY              57        408  ...        2.57              0  False.
27    MT              54        408  ...        3.97              3  False.
28    MO              20        415  ...        1.70              0  False.
29    HI              49        510  ...        3.00              1  False.
30    IL             142        415  ...        3.83              2  False.
31    NH              75        510  ...        2.78              1  False.
32    LA             172        408  ...        3.40              3  False.
33    AZ              12        408  ...        3.19              1   True.
34    OK              57        408  ...        2.24              0  False.
35    GA              72        415  ...        3.97              3  False.
36    AK              36        408  ...        3.92              0  False.
37    MA              78        415  ...        2.70              1  False.
38    AK             136        415  ...        2.84              3  False.
39    NJ             149        408  ...        3.00              1  False.
40    GA              98        408  ...        2.54              3  False.
41    MD             135        408  ...        3.94              0   True.
42    AR              34        510  ...        2.70              2  False.
43    ID             160        415  ...        2.48              3  False.
44    WI              64        510  ...        0.95              1  False.
45    OR              59        408  ...        2.30              2  False.
46    MI              65        415  ...        3.56              3  False.
47    DE             142        408  ...        2.00              2  False.
48    ID             119        415  ...        2.38              5   True.
49    WY              97        415  ...        2.97              1  False.
50    IA              52        408  ...        2.11              3  False.
51    IN              60        408  ...        1.84              1  False.
52    VA              10        408  ...        3.08              2  False.
53    UT              96        415  ...        2.51              2  False.
54    WY              87        415  ...        2.62              5   True.
55    IN              81        408  ...        2.75              1  False.
56    CO             141        415  ...        2.16              1  False.
57    CO             121        408  ...        1.57              3   True.
58    WI              68        415  ...        3.27              3  False.
59    OK             125        408  ...        3.24              1  False.
60    ID             174        408  ...        3.08              1  False.
61    CA             116        415  ...        3.13              2  False.
62    MN              74        510  ...        3.94              2  False.
63    SD             149        408  ...        3.40              3  False.
64    NC              38        408  ...        2.21              2  False.
65    WA              40        415  ...        1.67              2  False.
66    WY              43        415  ...        2.51              0  False.
67    MN             113        408  ...        2.24              0  False.
68    UT             126        408  ...        2.11              1  False.
69    TX             150        510  ...        3.73              4   True.
70    NJ             138        408  ...        3.19              3  False.
```

# filter the rows that have clocked day Mins to be greater than 350.

sub_data_liping=data_liping[data_liping['Day Mins']>350]

sub_data_liping.shape

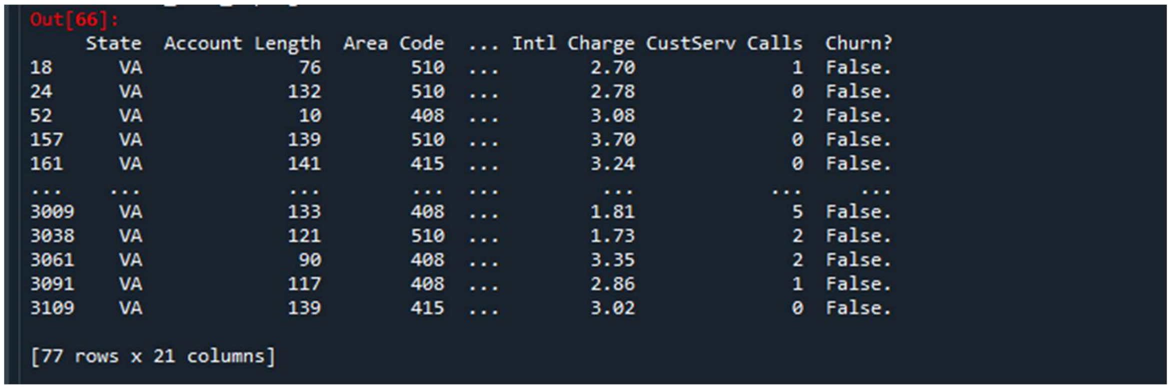
sub_data_liping



```
In [65]:
    ...:
    ...: sub_data_liping=data_liping[data_liping['Day Mins']>350]
    ...: sub_data_liping.shape
    ...: sub_data_liping
Out[65]:
    State  Account Length  Area Code  ... Intl Charge CustServ Calls Churn?
365    CO             154        415  ...        2.73              1  True.

[1 rows x 21 columns]
```

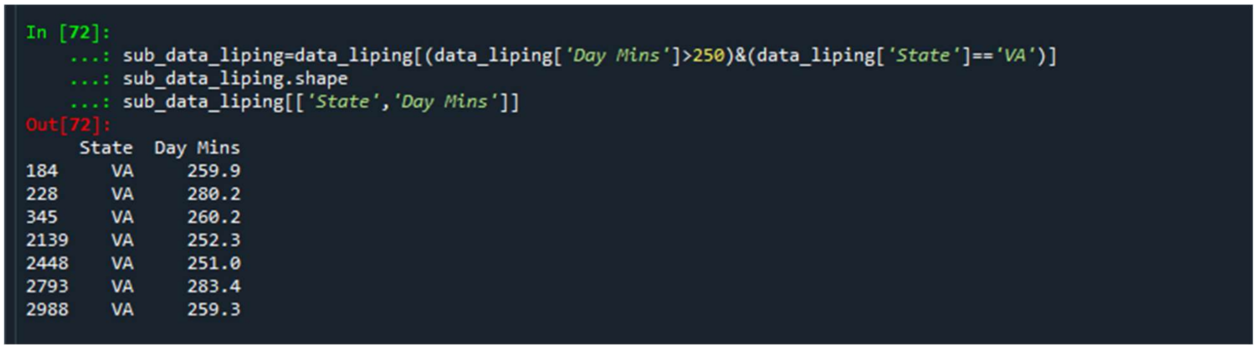
#filter the rows for which the state is VA:

sub_data_liping=data_liping[data_liping['State']=='VA']
sub_data_liping.shape
sub_data_liping



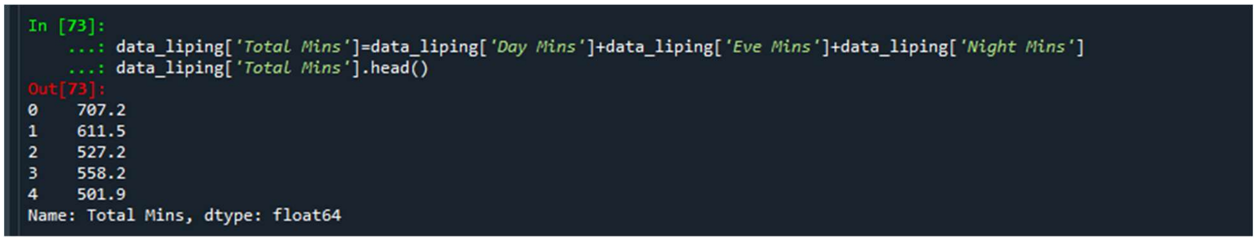
```
Out[66]:
     State  Account Length  Area Code  ... Intl Charge CustServ Calls  Churn?
18      VA              76        510  ...        2.70              1  False.
24      VA             132        510  ...        2.78              0  False.
52      VA              10        408  ...        3.08              2  False.
157     VA             139        510  ...        3.70              0  False.
161     VA             141        415  ...        3.24              0  False.
...    ...             ...        ...  ...         ...            ...     ...
3009    VA             133        408  ...        1.81              5  False.
3038    VA             121        510  ...        1.73              2  False.
3061    VA              90        408  ...        3.35              2  False.
3091    VA             117        408  ...        2.86              1  False.
3109    VA             139        415  ...        3.02              0  False.

[77 rows x 21 columns]
```

#filter the rows that have clocked day Mins to be greater than 150 and the state value is VA
sub_data_liping=data_liping[(data_liping['Day Mins']>250)&(data_liping['State']=='VA')]
sub_data_liping.shape
sub_data_liping[['State','Day Mins']]



```
In [72]:
    ...: sub_data_liping=data_liping[(data_liping['Day Mins']>250)&(data_liping['State']=='VA')]
    ...: sub_data_liping.shape
    ...: sub_data_liping[['State','Day Mins']]
Out[72]:
     State  Day Mins
184     VA     259.9
228     VA     280.2
345     VA     260.2
2139    VA     252.3
2448    VA     251.0
2793    VA     283.4
2988    VA     259.3
```

## Create a new column for total minutes
data_liping['Total Mins']=data_liping['Day Mins']+data_liping['Eve Mins']+data_liping['Night Mins']
data_liping['Total Mins'].head()



```
In [73]:
    ...: data_liping['Total Mins']=data_liping['Day Mins']+data_liping['Eve Mins']+data_liping['Night Mins']
    ...: data_liping['Total Mins'].head()
Out[73]:
0    707.2
1    611.5
2    527.2
3    558.2
4    501.9
Name: Total Mins, dtype: float64
```