Tutorial Week 3: Pandas for Unstructured Data (Solutions)

Click to download the Jupyter Notebook files (.ipynb)

Example: Load the automobile dataset and print the first 5 rows

Hint:

- The data are available in the files automobile_data.csv.
- Use Pandas `read_csv()` to load the automobile dataset.
- Use Pandas `head()` to return the first `n` rows.

[1]:
```
import pandas as pd

### Start your code here ###

df = pd.read_csv("automobile_data.csv")
df.head(5)

### End your code here ###
```

[1]:
```
   index      company   body-style  wheel-base  length engine-type  \
0      0  alfa-romero  convertible        88.6   168.8        dohc
1      1  alfa-romero  convertible        88.6   168.8        dohc
2      2  alfa-romero    hatchback        94.5   171.2        ohcv
3      3         audi        sedan        99.8   176.6         ohc
4      4         audi        sedan        99.4   176.6         ohc

  num-of-cylinders  horsepower  average-mileage  price
0             four         111               21  13495
1             four         111               21  16500
2              six         154               19  16500
3             four         102               24  13950
4             five         115               18  17450
```

1 Questions

Hint:

- The data are available in the files automobile_data.csv.
- Write **your code** between two comment lines: `### Start/End your code here ###`.
- **Expected output** is shown at the end of each question (directly below the code cell).

1.1 Drop the rows where at least one element is missing

Hint:

- Use Pandas `dropna()`.

[2]:
```
import pandas as pd
import numpy as np

# dictionary with list object in values
details = {
    'Name' : ['Ankit', 'Aishwarya', 'Shaurya', 'Shivangi'],
    'Age' : [23, np.nan, 22, 21],
    'University' : ['BHU', 'JNU', np.nan, 'BHU'],
}

# creating a Dataframe object
df = pd.DataFrame(details)

### Start your code here ###

df1 = df.dropna()
df1

### End your code here ###
```

[2]:
```
       Name   Age University
0     Ankit  23.0        BHU
3  Shivangi  21.0        BHU
```

1.2 Print all details of Toyota cars

Hint:

- Use Pandas selection method.

[3]:
```
import pandas as pd

df = pd.read_csv("automobile_data.csv")

### Start your code here ###

df1 = df.loc[df['company']=='toyota']
df1

### End your code here ###
```

[3]:
```
    index company body-style  wheel-base  length engine-type num-of-cylinders␣
↪ \
44     66  toyota  hatchback        95.7   158.7         ohc             four
45     67  toyota  hatchback        95.7   158.7         ohc             four
46     68  toyota  hatchback        95.7   158.7         ohc             four
47     69  toyota      wagon        95.7   169.7         ohc             four
48     70  toyota      wagon        95.7   169.7         ohc             four
49     71  toyota      wagon        95.7   169.7         ohc             four
50     79  toyota      wagon       104.5   187.8        dohc              six
```

```
    horsepower  average-mileage  price
44          62               35   5348
45          62               31   6338
46          62               31   6488
47          62               31   6918
48          62               27   7898
49          62               27   8778
50         156               19  15750
```

1.3 Find the most expensive car's company name

- Print most expensive car's company name and price.

[4]:
```
import pandas as pd

df = pd.read_csv("automobile_data.csv")

### Start your code here ###

df1 = df.loc[df['price']==df['price'].max()]
df2 = df1[['company','price']]
df2

### End your code here ###
```

[4]:
```
           company  price
32  mercedes-benz  45400
```

1.4 Count total cars per company

Hint:

- Use Pandas `value_counts()`.

[5]:
```
import pandas as pd

df = pd.read_csv("automobile_data.csv")

### Start your code here ###

df1 = df['company'].value_counts()
df1

### End your code here ###
```

[5]:
```
toyota           7
bmw              6
mazda            5
nissan           5
audi             4
mercedes-benz    4
```

```
mitsubishi       4
volkswagen       4
alfa-romero      3
honda            3
jaguar           3
chevrolet        2
dodge            2
porsche          2
volvo            2
isuzu            1
Name: company, dtype: int64
```

1.5 Find each company's Higesht price car

[6]:
```
import pandas as pd

df = pd.read_csv("automobile_data.csv")

### Start your code here ###

df1 = df.groupby('company')['price'].max()
df1

### End your code here ###
```

[6]:
```
company
alfa-romero      16500
audi             18920
bmw              41315
chevrolet         6575
dodge             6377
honda            12945
isuzu             6785
jaguar           36000
mazda            18344
mercedes-benz    45400
mitsubishi        8189
nissan           13499
porsche          37028
toyota           15750
volkswagen        9995
volvo            13415
Name: price, dtype: int64
```

1.6 Find the average mileage of each car making company

[7]:
```
import pandas as pd

df = pd.read_csv("automobile_data.csv")

### Start your code here ###
```

```
df1 = df.groupby('company')['average-mileage'].mean()
df1

### End your code here ###
```

[7]:
```
company
alfa-romero      20.333333
audi             20.000000
bmw              19.000000
chevrolet        38.000000
dodge            31.000000
honda            26.333333
isuzu            24.000000
jaguar           14.333333
mazda            28.000000
mercedes-benz    18.000000
mitsubishi       29.500000
nissan           31.400000
porsche          17.000000
toyota           28.714286
volkswagen       31.750000
volvo            23.000000
Name: average-mileage, dtype: float64
```

1.7 Sort all cars by Price column in descending order

Hint:

- Use Pandas `sort_values()`.
- Print the first 5 rows of the sorted `Dataframe`.

[8]:
```
import pandas as pd

df = pd.read_csv("automobile_data.csv")

### Start your code here ###

df1 = df.sort_values(by=['price'], ascending=False)
df1.head(5)

### End your code here ###
```

[8]:

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
32	47	mercedes-benz	hardtop	112.0	199.2	ohcv	eight	184	14	45400
11	14	bmw	sedan	103.5	193.8	ohc				
31	46	mercedes-benz	sedan	120.9	208.1	ohcv				
43	62	porsche	convertible	89.5	168.9	ohcf				
12	15	bmw	sedan	110.0	197.0	ohc				

```
11            six        182              16  41315
31          eight        184              14  40960
43            six        207              17  37028
12            six        182              15  36880
```

1.8 Concatenate two Dataframes and reset the index of the combined Dataframe

Hint:

- Use Pandas `concat()` and its `ignore_index` option.

[9]:
```
import pandas as pd

df1 = pd.DataFrame({'Company': ['Ford', 'Mercedes', 'BMV', 'Audi'], 'Price':␣
↪[23845, 171995, 135925 , 71400]})

df2 = pd.DataFrame({'Company': ['Toyota', 'Honda', 'Nissan', 'Mitsubishi '],␣
↪'Price': [29995, 23600, 61500 , 58900]})

### Start your code here ###

df3 = pd.concat([df1, df2], ignore_index=True)
df3

### End your code here ###
```

[9]:
```
       Company   Price
0         Ford   23845
1     Mercedes  171995
2          BMV  135925
3         Audi   71400
4       Toyota   29995
5        Honda   23600
6       Nissan   61500
7  Mitsubishi    58900
```

1.9 Merge two data frames using the following condition

- Create two `DataFrames` using the following two `Dictionarys`.
- Merge two `DataFrames` on the `Company` column.

[10]:
```
import pandas as pd

dict1 = {'Company': ['Toyota', 'Honda', 'BMV', 'Audi', 'Jaguar'], 'Price':␣
↪[23845, 17995, 135925 , 71400, 23725]}
df1 = pd.DataFrame(dict1)

dict2 = {'Company': ['Toyota', 'Honda', 'BMV', 'Audi', 'Jaguar'],␣
↪'horsepower': [141, 80, 182 , 160, 220]}
df2 = pd.DataFrame(dict2)
```

```
### Start your code here ###

df3 = pd.merge(df1, df2, on="Company")
df3

### End your code here ###
```

[10]:

	Company	Price	horsepower
0	Toyota	23845	141
1	Honda	17995	80
2	BMV	135925	182
3	Audi	71400	160
4	Jaguar	23725	220