

# Wine

## Introduction:

This exercise is a adaptation from the UCI Wine dataset. The only pupose is to practice deleting data with pandas.

## Step 1. Import the necessary libraries

```
In [71]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

## Step 2. Import the dataset from this [address](https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data).

```
In [72]: df=pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data",sep=',',header=None)
```

```
In [73]: df.head()
```

Out[73]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735

## Step 3. Assign it to a variable called wine

```
In [74]: wine=df
```

```
In [75]: wine.head()
```

Out[75]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735

## Step 4. Delete the first, fourth, seventh, ninth, eleventh, thirteenth and fourteenth columns

```
In [76]: wine.drop(wine.columns[[0,3,6,8,10,12,13]], axis = 1, inplace = True)
```

```
In [77]: wine
```

Out[77]:

	1	2	4	5	7	9	11
0	14.23	1.71	15.6	127	3.06	2.29	1.04
1	13.20	1.78	11.2	100	2.76	1.28	1.05
2	13.16	2.36	18.6	101	3.24	2.81	1.03
3	14.37	1.95	16.8	113	3.49	2.18	0.86
4	13.24	2.59	21.0	118	2.69	1.82	1.04
...	...	...	...	...	...	...	...
173	13.71	5.65	20.5	95	0.61	1.06	0.64
174	13.40	3.91	23.0	102	0.75	1.41	0.70
175	13.27	4.28	20.0	120	0.69	1.35	0.59
176	13.17	2.59	20.0	120	0.68	1.46	0.60
177	14.13	4.10	24.5	96	0.76	1.35	0.61

178 rows × 7 columns

```
In [78]: wine.shape[1]

Out[78]: 7
```

Step 5. Assign the columns as below:

The attributes are (donated by Riccardo Leardi, riclea '@' anchem.unige.it):

- 1) Alcohol
- 2) Malic acid
- 3) Ash
- 4) Alcalinity of ash
- 5) Magnesium
- 6) Total phenols
- 7) Flavanoids
- 8) Nonflavanoid phenols
- 9) Proanthocyanins
- 10)Color intensity
- 11)Hue
- 12)OD280/OD315 of diluted wines
- 13)Proline

NOTE: 0th attribute is class identifier - Type of cultivator (1-3)

```
In [79]: wine.columns=["alcohol","malic_acid","alcalinity_of_ash","magnesium","flavanoids","proanthocyanins","hue"]

In [80]: wine.head()
```

Out[80]:

	alcohol	malic_acid	alcalinity_of_ash	magnesium	flavanoids	proanthocyanins	hue
0	14.23	1.71	15.6	127	3.06	2.29	1.04
1	13.20	1.78	11.2	100	2.76	1.28	1.05
2	13.16	2.36	18.6	101	3.24	2.81	1.03
3	14.37	1.95	16.8	113	3.49	2.18	0.86
4	13.24	2.59	21.0	118	2.69	1.82	1.04

Step 6. Set the values of the first 3 rows from alcohol as NaN

```
In [81]: wine.alcohol.iloc[:3] = np.NaN

/tmp/ipykernel_32/2227344411.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  wine.alcohol.iloc[:3] = np.NaN

In [82]: wine
```

Out[82]:

	alcohol	malic_acid	alcalinity_of_ash	magnesium	flavanoids	proanthocyanins	hue
0	NaN	1.71	15.6	127	3.06	2.29	1.04
1	NaN	1.78	11.2	100	2.76	1.28	1.05
2	NaN	2.36	18.6	101	3.24	2.81	1.03
3	14.37	1.95	16.8	113	3.49	2.18	0.86
4	13.24	2.59	21.0	118	2.69	1.82	1.04
...	...	...	...	...	...	...	...
173	13.71	5.65	20.5	95	0.61	1.06	0.64
174	13.40	3.91	23.0	102	0.75	1.41	0.70
175	13.27	4.28	20.0	120	0.69	1.35	0.59
176	13.17	2.59	20.0	120	0.68	1.46	0.60
177	14.13	4.10	24.5	96	0.76	1.35	0.61

178 rows × 7 columns

Step 7. Now set the value of the rows 3 and 4 of magnesium as NaN

```
In [83]: wine.magnesium.iloc[3:5]=np.nan
```

```
/tmp/ipykernel_32/2337123237.py:1: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver  
sus-a-copy  
    wine.magnesium.iloc[3:5]=np.nan
```

```
In [84]: wine.head()
```

Out[84]:

	alcohol	malic_acid	alcalinity_of_ash	magnesium	flavanoids	proanthocyanins	hue
0	NaN	1.71	15.6	127.0	3.06	2.29	1.04
1	NaN	1.78	11.2	100.0	2.76	1.28	1.05
2	NaN	2.36	18.6	101.0	3.24	2.81	1.03
3	14.37	1.95	16.8	NaN	3.49	2.18	0.86
4	13.24	2.59	21.0	NaN	2.69	1.82	1.04

```
In [85]: wine.isna().sum()
```

Out[85]:

```
alcohol      3  
malic_acid    0  
alcalinity_of_ash  0  
magnesium     2  
flavanoids    0  
proanthocyanins 0  
hue           0  
dtype: int64
```

Step 8. Fill the value of NaN with the number 10 in alcohol and 100 in magnesium

```
In [86]: wine.alcohol=wine.alcohol.fillna(10)  
wine.magnesium=wine.magnesium.fillna(100)  
wine
```

Out[86]:

	alcohol	malic_acid	alcalinity_of_ash	magnesium	flavanoids	proanthocyanins	hue
0	10.00	1.71	15.6	127.0	3.06	2.29	1.04
1	10.00	1.78	11.2	100.0	2.76	1.28	1.05
2	10.00	2.36	18.6	101.0	3.24	2.81	1.03
3	14.37	1.95	16.8	100.0	3.49	2.18	0.86
4	13.24	2.59	21.0	100.0	2.69	1.82	1.04
...	...	...	...	...	...	...	...
173	13.71	5.65	20.5	95.0	0.61	1.06	0.64
174	13.40	3.91	23.0	102.0	0.75	1.41	0.70
175	13.27	4.28	20.0	120.0	0.69	1.35	0.59
176	13.17	2.59	20.0	120.0	0.68	1.46	0.60
177	14.13	4.10	24.5	96.0	0.76	1.35	0.61

178 rows × 7 columns

Step 9. Count the number of missing values

```
In [87]: wine.isnull().sum().sum()
```

Out[87]: 0

Step 10. Create an array of 10 random numbers up until 10

```
In [88]: randNum = np.random.randint(0,11,10)
```

```
In [89]: randNum
```

Out[89]: array([7, 0, 0, 8, 2, 9, 7, 2, 1, 6])

Step 11. Use random numbers you generated as an index and assign NaN value to each of cell.

```
In [90]: wine.iloc[randNum]=np.nan
```

Step 12. How many missing values do we have?

```
In [91]: wine.isnull().sum().sum()
```

Out[91]: 49

Step 13. Delete the rows that contain missing values

```
In [92]: wine.dropna(inplace=True)
```

```
In [93]: wine
```

Out[93]:

	alcohol	malic_acid	alcalinity_of_ash	magnesium	flavanoids	proanthocyanins	hue
3	14.37	1.95	16.8	100.0	3.49	2.18	0.86
4	13.24	2.59	21.0	100.0	2.69	1.82	1.04
5	14.20	1.76	15.2	112.0	3.39	1.97	1.05
10	14.10	2.16	18.0	105.0	3.32	2.38	1.25
11	14.12	1.48	16.8	95.0	2.43	1.57	1.17
...	...	...	...	...	...	...	...
173	13.71	5.65	20.5	95.0	0.61	1.06	0.64
174	13.40	3.91	23.0	102.0	0.75	1.41	0.70
175	13.27	4.28	20.0	120.0	0.69	1.35	0.59
176	13.17	2.59	20.0	120.0	0.68	1.46	0.60
177	14.13	4.10	24.5	96.0	0.76	1.35	0.61

171 rows × 7 columns

```
In [94]: wine.isnull().sum().sum()
```

Out[94]: 0

Step 14. Print only the non-null values in alcohol

```
In [97]: wine
```

Out[97]:

	alcohol	malic_acid	alcalinity_of_ash	magnesium	flavanoids	proanthocyanins	hue
3	14.37	1.95	16.8	100.0	3.49	2.18	0.86
4	13.24	2.59	21.0	100.0	2.69	1.82	1.04
5	14.20	1.76	15.2	112.0	3.39	1.97	1.05
10	14.10	2.16	18.0	105.0	3.32	2.38	1.25
11	14.12	1.48	16.8	95.0	2.43	1.57	1.17
...	...	...	...	...	...	...	...
173	13.71	5.65	20.5	95.0	0.61	1.06	0.64
174	13.40	3.91	23.0	102.0	0.75	1.41	0.70
175	13.27	4.28	20.0	120.0	0.69	1.35	0.59
176	13.17	2.59	20.0	120.0	0.68	1.46	0.60
177	14.13	4.10	24.5	96.0	0.76	1.35	0.61

171 rows × 7 columns

Step 15. Reset the index, so it starts with 0 again

```
In [98]: wine.reset_index(drop=True)
```

Out[98]:

	alcohol	malic_acid	alcalinity_of_ash	magnesium	flavanoids	proanthocyanins	hue
<b>0</b>	14.37	1.95	16.8	100.0	3.49	2.18	0.86
<b>1</b>	13.24	2.59	21.0	100.0	2.69	1.82	1.04
<b>2</b>	14.20	1.76	15.2	112.0	3.39	1.97	1.05
<b>3</b>	14.10	2.16	18.0	105.0	3.32	2.38	1.25
<b>4</b>	14.12	1.48	16.8	95.0	2.43	1.57	1.17
...	...	...	...	...	...	...	...
<b>166</b>	13.71	5.65	20.5	95.0	0.61	1.06	0.64
<b>167</b>	13.40	3.91	23.0	102.0	0.75	1.41	0.70
<b>168</b>	13.27	4.28	20.0	120.0	0.69	1.35	0.59
<b>169</b>	13.17	2.59	20.0	120.0	0.68	1.46	0.60
<b>170</b>	14.13	4.10	24.5	96.0	0.76	1.35	0.61

171 rows × 7 columns

In [ ]: