tips_and_tricks

July 25, 2022

1 Name:Ammar Zafar

2 Email:Ammarzafar028@gmail.com

3 01- How to find the Version

```
[ ]: # First We have to import the pandas library and then check the version
     import pandas as pd
     pd.__version__
```

```
[ ]: '1.4.1'
```

```
[ ]: # Another way to check the version
     pd.show_versions()
```

```
INSTALLED VERSIONS
------------------
commit           : 06d230151e6f18fdb8139d09abf539867a8cd481
python           : 3.10.1.final.0
python-bits      : 64
OS               : Windows
OS-release       : 10
Version          : 10.0.19044
machine          : AMD64
processor        : Intel64 Family 6 Model 42 Stepping 7, GenuineIntel
byteorder        : little
LC_ALL           : None
LANG             : None
LOCALE           : English_United States.1252

pandas           : 1.4.1
numpy            : 1.22.2
pytz             : 2021.3
dateutil         : 2.8.2
pip              : 22.1.2
setuptools       : 58.1.0
Cython           : None
```

```
pytest             : None
hypothesis         : None
sphinx             : None
blosc              : None
feather            : None
xlsxwriter         : None
lxml.etree         : None
html5lib           : None
pymysql            : None
psycopg2           : None
jinja2             : 3.0.3
IPython            : 8.1.1
pandas_datareader: None
bs4                : 4.10.0
bottleneck         : None
fastparquet        : None
fsspec             : 2022.02.0
gcsfs              : None
matplotlib         : 3.5.1
numba              : None
numexpr            : None
odfpy              : None
openpyxl           : 3.0.10
pandas_gbq         : None
pyarrow            : None
pyreadstat         : None
pyxlsb             : None
s3fs               : 2022.02.0
scipy              : 1.8.0
sqlalchemy         : None
tables             : None
tabulate           : 0.8.9
xarray             : 2022.3.0
xlrd               : None
xlwt               : None
zstandard          : None
```

4 02- Make a dataframe

```
[ ]: # now we are making a dataframe
     df=pd.DataFrame({"A Col":[1,2,3,7,9,22],"B Col":[4,5,6,8,12,20]})
     df
```

```
[ ]:    A Col  B Col
     0      1      4
     1      2      5
     2      3      6
```

```
3      7      8
4      9     12
5     22     20
```

[ ]:

```python
# Numpy array use to create DataFrame
import numpy as np
array=np.array([[1,2,3],[4,5,6],[7,8,9]])
pd.DataFrame(array)
```

[ ]:

```
   0  1  2
0  1  2  3
1  4  5  6
2  7  8  9
```

[ ]:

```python
# We are making numpy array dataframe
pd.DataFrame(np.random.rand(4,8))
```

[ ]:

```
          0         1         2         3         4         5         6  \
0  0.227438  0.768964  0.067451  0.951509  0.481932  0.034696  0.974427
1  0.053828  0.293163  0.193856  0.945921  0.106795  0.194545  0.116258
2  0.992958  0.427184  0.109688  0.878474  0.507883  0.708144  0.450596
3  0.086460  0.303217  0.615265  0.186380  0.866725  0.939049  0.964431

          7
0  0.101165
1  0.026892
2  0.561832
3  0.821946
```

[ ]:

```python
pd.DataFrame(np.random.rand(4,8), columns=list("ABCDEFGH"))
```

[ ]:

```
          A         B         C         D         E         F         G  \
0  0.799661  0.509111  0.825871  0.319241  0.935529  0.616984  0.546212
1  0.606471  0.480119  0.635631  0.741752  0.817759  0.743296  0.083758
2  0.555395  0.516987  0.847938  0.291029  0.748031  0.422014  0.392527
3  0.089612  0.273183  0.456888  0.974178  0.436739  0.039706  0.925629

          H
0  0.871902
1  0.730319
2  0.410851
3  0.429255
```

5 03- How to rename columns

```
[ ]: df=pd.DataFrame({"A Col":[1,2,3,7,9,22],"B Col":[4,5,6,8,12,20]})
     df
```

```
[ ]:    A Col  B Col
     0      1      4
     1      2      5
     2      3      6
     3      7      8
     4      9     12
     5     22     20
```

```
[ ]: # we are doing rename columns
     df.rename(columns={"A Col":"col_a","col_b":"B_Col"},inplace=True)
     df
```

```
[ ]:    col_a  B Col
     0      1      4
     1      2      5
     2      3      6
     3      7      8
     4      9     12
     5     22     20
```

```
[ ]: # rename columns
     df.columns=["col_aa","col_bb"]
     df
```

```
[ ]:    col_aa  col_bb
     0       1       4
     1       2       5
     2       3       6
     3       7       8
     4       9      12
     5      22      20
```

```
[ ]: df.columns=df.columns.str.replace("_","")
     df
```

```
[ ]:    colaa  colbb
     0      1      4
     1      2      5
     2      3      6
     3      7      8
     4      9     12
     5     22     20
```

```
[ ]: # Adding prefix to columns
     df=df.add_prefix("baba_")
     df
```

```
[ ]:    baba_colaa  baba_colbb
     0           1           4
     1           2           5
     2           3           6
     3           7           8
     4           9          12
     5          22          20
```

```
[ ]: # We are adding suffix
     df=df.add_suffix("_baba")
     df
```

```
[ ]:    baba_colaa_baba  baba_colbb_baba
     0                1                4
     1                2                5
     2                3                6
     3                7                8
     4                9               12
     5               22               20
```

```
[ ]: df.columns=["col_aa","col_bb"]
     df
```

```
[ ]:    col_aa  col_bb
     0       1       4
     1       2       5
     2       3       6
     3       7       8
     4       9      12
     5      22      20
```

6 04- Using Template Data

```
[ ]: # We are importing libraries and dataset
     import numpy as np
     import pandas as pd
     import seaborn as sns

     df=sns.load_dataset("tips")
     df.head()
```

```
[ ]:    total_bill   tip     sex smoker  day    time  size
     0       16.99  1.01  Female     No  Sun  Dinner     2
```

```
1        10.34  1.66    Male     No  Sun  Dinner     3
2        21.01  3.50    Male     No  Sun  Dinner     3
3        23.68  3.31    Male     No  Sun  Dinner     2
4        24.59  3.61  Female     No  Sun  Dinner     4
```

```
[ ]: # Checking stats
     df.describe()

     # column names
     df.columns
```

```
[ ]: Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'],
     dtype='object')
```

```
[ ]: # Saving a dataset
     df.to_csv("tips_save.csv")

     df.to_excel("tips.save.xlsx")
```

7 05-Using your own data

```
[ ]: import pandas as pd
     #df=pd.read_csv("tips_save.csv")
     #df.head()
     df=pd.read_excel("tips.save.xlsx")
     df.head()
```

```
[ ]:    Unnamed: 0  total_bill   tip     sex smoker  day    time  size
     0           0       16.99  1.01  Female     No  Sun  Dinner     2
     1           1       10.34  1.66    Male     No  Sun  Dinner     3
     2           2       21.01  3.50    Male     No  Sun  Dinner     3
     3           3       23.68  3.31    Male     No  Sun  Dinner     2
     4           4       24.59  3.61  Female     No  Sun  Dinner     4
```

8 06- Reverse Row as Order

```
[ ]: import seaborn as sns
     import pandas as pd

     df=sns.load_dataset("titanic")
     df.head()
```

```
[ ]:    survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
     0         0       3    male  22.0      1      0   7.2500        S  Third
     1         1       1  female  38.0      1      0  71.2833        C  First
     2         1       3  female  26.0      0      0   7.9250        S  Third
```

```
3          1       1  female  35.0      1      0  53.1000        S  First
4          0       3    male  35.0      0      0   8.0500        S  Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True
```

```
[ ]: df.loc[::-1].head()
```

```
[ ]:
     survived  pclass     sex   age  sibsp  parch   fare embarked   class  \
890         0       3    male  32.0      0      0   7.75        Q   Third
889         1       1    male  26.0      0      0  30.00        C   First
888         0       3  female   NaN      1      2  23.45        S   Third
887         1       1  female  19.0      0      0  30.00        S   First
886         0       2    male  27.0      0      0  13.00        S  Second

        who  adult_male deck  embark_town alive  alone
890    man        True  NaN   Queenstown    no   True
889    man        True    C    Cherbourg   yes   True
888  woman       False  NaN  Southampton    no  False
887  woman       False    B  Southampton   yes   True
886    man        True  NaN  Southampton    no   True
```

```
[ ]: df.loc[::-1].reset_index(drop=True).head()
```

```
[ ]:
   survived  pclass     sex   age  sibsp  parch   fare embarked   class  \
0         0       3    male  32.0      0      0   7.75        Q   Third
1         1       1    male  26.0      0      0  30.00        C   First
2         0       3  female   NaN      1      2  23.45        S   Third
3         1       1  female  19.0      0      0  30.00        S   First
4         0       2    male  27.0      0      0  13.00        S  Second

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN   Queenstown    no   True
1    man        True    C    Cherbourg   yes   True
2  woman       False  NaN  Southampton    no  False
3  woman       False    B  Southampton   yes   True
4    man        True  NaN  Southampton    no   True
```

9 07- Reverse Column Order

```
[ ]: df.loc[:,::-1].head()
[ ]:    alone alive  embark_town deck  adult_male    who  class embarked     fare  \
     0  False    no  Southampton  NaN        True    man  Third        S   7.2500
     1  False   yes    Cherbourg    C       False  woman  First        C  71.2833
     2   True   yes  Southampton  NaN       False  woman  Third        S   7.9250
     3  False   yes  Southampton    C       False  woman  First        S  53.1000
     4   True    no  Southampton  NaN        True    man  Third        S   8.0500

        parch  sibsp   age     sex  pclass  survived
     0      0      1  22.0    male       3         0
     1      0      1  38.0  female       1         1
     2      0      0  26.0  female       3         1
     3      0      1  35.0  female       1         1
     4      0      0  35.0    male       3         0
```

10 08- Select a column by dtype

```
[ ]: df.dtypes
```

```
[ ]: survived          int64
     pclass            int64
     sex              object
     age             float64
     sibsp             int64
     parch             int64
     fare            float64
     embarked         object
     class          category
     who              object
     adult_male         bool
     deck           category
     embark_town      object
     alive            object
     alone              bool
     dtype: object
```

```
[ ]: # Only select those have numeric types
     df.select_dtypes(include=["number"]).head()
```

```
[ ]:    survived  pclass   age  sibsp  parch     fare
     0         0       3  22.0      1      0   7.2500
     1         1       1  38.0      1      0  71.2833
     2         1       3  26.0      0      0   7.9250
     3         1       1  35.0      1      0  53.1000
     4         0       3  35.0      0      0   8.0500
```

```
[ ]: # Only select those have categorical types
     df.select_dtypes(include=["object"]).head()
```

```
[ ]:       sex embarked    who  embark_town alive
     0    male        S    man  Southampton    no
     1  female        C  woman    Cherbourg   yes
     2  female        S  woman  Southampton   yes
     3  female        S  woman  Southampton   yes
     4    male        S    man  Southampton    no
```

```
[ ]: # Only select those have multiple types
     df.select_dtypes(include=["category","object","number"]).head()
```

```
[ ]:    survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
     0         0       3    male  22.0      1      0   7.2500        S  Third
     1         1       1  female  38.0      1      0  71.2833        C  First
     2         1       3  female  26.0      0      0   7.9250        S  Third
     3         1       1  female  35.0      1      0  53.1000        S  First
     4         0       3    male  35.0      0      0   8.0500        S  Third

          who deck  embark_town alive
     0    man  NaN  Southampton    no
     1  woman    C    Cherbourg   yes
     2  woman  NaN  Southampton   yes
     3  woman    C  Southampton   yes
     4    man  NaN  Southampton    no
```

```
[ ]: # Exclude numeric
     df.select_dtypes(exclude=["number"]).head()
```

```
[ ]:       sex embarked  class    who  adult_male deck  embark_town alive  alone
     0    male        S  Third    man        True  NaN  Southampton    no  False
     1  female        C  First  woman       False    C    Cherbourg   yes  False
     2  female        S  Third  woman       False  NaN  Southampton   yes   True
     3  female        S  First  woman       False    C  Southampton   yes  False
     4    male        S  Third    man        True  NaN  Southampton    no   True
```

```
[ ]: # Exclude categorical
     df.select_dtypes(exclude=["object","category"]).head()
```

```
[ ]:    survived  pclass   age  sibsp  parch     fare  adult_male  alone
     0         0       3  22.0      1      0   7.2500        True  False
     1         1       1  38.0      1      0  71.2833       False  False
     2         1       3  26.0      0      0   7.9250       False   True
     3         1       1  35.0      1      0  53.1000       False  False
     4         0       3  35.0      0      0   8.0500        True   True
```

```
[ ]: df=pd.DataFrame({"A_Col":["1.2","2","3","7","9","22"],"B_Col":
      ↪["4","5","6","8","12","20"]})
     df
```

```
[ ]:   A_Col B_Col
     0   1.2     4
     1     2     5
     2     3     6
     3     7     8
     4     9    12
     5    22    20
```

```
[ ]: df.dtypes
```

```
[ ]: A_Col    object
     B_Col    object
     dtype: object
```

```
[ ]: df.astype({"A_Col":"float64","B_Col":"int64"}).dtypes
```

```
[ ]: A_Col    float64
     B_Col      int64
     dtype: object
```

```
[ ]: pd.to_numeric(df["A_Col"],errors="coerce")
     pd.to_numeric(df["B_Col"],errors="coerce")
```

```
[ ]: 0     4
     1     5
     2     6
     3     8
     4    12
     5    20
     Name: B_Col, dtype: int64
```

11 10 - Reduce Dataframe Size

```
[ ]: df=sns.load_dataset("titanic")
     df.shape
```

```
[ ]: (891, 15)
```

```
[ ]: df.sample(frac=0.1).shape
     df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object  
 3   age          714 non-null    float64 
 4   sibsp        891 non-null    int64   
 5   parch        891 non-null    int64   
 6   fare         891 non-null    float64 
 7   embarked     889 non-null    object  
 8   class        891 non-null    category
 9   who          891 non-null    object  
 10  adult_male   891 non-null    bool    
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object  
 13  alive        891 non-null    object  
 14  alone        891 non-null    bool    
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

```
[ ]:
```