Day_35_301123

January 23, 2024

[131]:
```
import pandas as pd
```

[132]:
```
df = pd.read_csv("mckinsey (1).csv")
```

[133]:
```
df.head()
```

[133]:
```
       country  year  population continent  life_exp     gdp_cap
0  Afghanistan  1952     8425333      Asia    28.801  779.445314
1  Afghanistan  1957     9240934      Asia    30.332  820.853030
2  Afghanistan  1962    10267083      Asia    31.997  853.100710
3  Afghanistan  1967    11537966      Asia    34.020  836.197138
4  Afghanistan  1972    13079460      Asia    36.088  739.981106
```

[134]:
```
df.shape
```

[134]:
```
(1704, 6)
```

1 Adding duplicates

[135]:
```
df.loc[1704] = ['India',1933,89778854,'Asia',86.23,897.956]
df.loc[1705] = ['India',1933,89778854,'Asia',86.23,897.956]
df.loc[1706] = ['India',1933,89778854,'Asia',86.23,897.956]
df.loc[1707] = ['India',1933,89778854,'Asia',86.23,897.956]
df.loc[1708] = ['India',1933,89778854,'Asia',86.23,897.956]
df.loc[1709] = ['India',1933,89778854,'Asia',86.23,897.956]
df.loc[1710] = ['India',1933,89778854,'Asia',86.23,897.956]
```

[136]:
```
df.tail()
```

[136]:
```
     country  year  population continent  life_exp  gdp_cap
1706   India  1933    89778854      Asia     86.23  897.956
1707   India  1933    89778854      Asia     86.23  897.956
1708   India  1933    89778854      Asia     86.23  897.956
1709   India  1933    89778854      Asia     86.23  897.956
1710   India  1933    89778854      Asia     86.23  897.956
```

[137]:
```
df.duplicated()
```

```
[137]: 0       False
       1       False
       2       False
       3       False
       4       False
               …
       1706     True
       1707     True
       1708     True
       1709     True
       1710     True
       Length: 1711, dtype: bool
```

```
[138]: df.loc[df.duplicated()]
```

```
[138]:      country  year  population continent  life_exp  gdp_cap
       1705   India  1933    89778854      Asia     86.23  897.956
       1706   India  1933    89778854      Asia     86.23  897.956
       1707   India  1933    89778854      Asia     86.23  897.956
       1708   India  1933    89778854      Asia     86.23  897.956
       1709   India  1933    89778854      Asia     86.23  897.956
       1710   India  1933    89778854      Asia     86.23  897.956
```

2 Removing duplicated

3 Drop duplicated and keep last one

```
[139]: df.drop_duplicates(keep='last')
```

```
[139]:           country  year  population continent  life_exp     gdp_cap
       0      Afghanistan  1952     8425333      Asia    28.801  779.445314
       1      Afghanistan  1957     9240934      Asia    30.332  820.853030
       2      Afghanistan  1962    10267083      Asia    31.997  853.100710
       3      Afghanistan  1967    11537966      Asia    34.020  836.197138
       4      Afghanistan  1972    13079460      Asia    36.088  739.981106
       …              …    …           …         …         …           …
       1700      Zimbabwe  1992    10704340    Africa    60.377  693.420786
       1701      Zimbabwe  1997    11404948    Africa    46.809  792.449960
       1702      Zimbabwe  2002    11926563    Africa    39.989  672.038623
       1703      Zimbabwe  2007    12311143    Africa    43.487  469.709298
       1710         India  1933    89778854      Asia    86.230  897.956000

       [1705 rows x 6 columns]
```

4 Drop everything which are duplicated

[140]:

```
df.drop_duplicates(keep=False,inplace=True)
```

5 Working with columns and rows using Slicing

[141]:

```
df.iloc[:4,:3]
```

[141]:

```
       country  year  population
0  Afghanistan  1952     8425333
1  Afghanistan  1957     9240934
2  Afghanistan  1962    10267083
3  Afghanistan  1967    11537966
```

[142]:

```
df.loc[1:5,['country','life_exp']]
```

[142]:

```
       country  life_exp
1  Afghanistan    30.332
2  Afghanistan    31.997
3  Afghanistan    34.020
4  Afghanistan    36.088
5  Afghanistan    38.438
```

[143]:

```
df.loc[1:5,'country':'life_exp']
```

[143]:

```
       country  year  population continent  life_exp
1  Afghanistan  1957     9240934      Asia    30.332
2  Afghanistan  1962    10267083      Asia    31.997
3  Afghanistan  1967    11537966      Asia    34.020
4  Afghanistan  1972    13079460      Asia    36.088
5  Afghanistan  1977    14880372      Asia    38.438
```

[144]:

```
df.iloc[[1,3,5],[2,4,5]]
```

[144]:

```
   population  life_exp     gdp_cap
1     9240934    30.332  820.853030
3    11537966    34.020  836.197138
5    14880372    38.438  786.113360
```

[145]:

```
df.loc[1:10:2,'country':'gdp_cap':2]
```

[145]:

```
       country  population  life_exp
1  Afghanistan     9240934    30.332
3  Afghanistan    11537966    34.020
5  Afghanistan    14880372    38.438
7  Afghanistan    13867957    40.822
9  Afghanistan    22227415    41.763
```

```
[146]: df.loc[[3,4,5],'country':'gdp_cap':2]
```

```
[146]:        country  population  life_exp
       3  Afghanistan    11537966    34.020
       4  Afghanistan    13079460    36.088
       5  Afghanistan    14880372    38.438
```

6 Sorting

```
[147]: df.sort_values(['year','life_exp'],ascending=[False,True])
```

```
[147]:            country  year  population continent  life_exp       gdp_cap
       1463      Swaziland  2007     1133066    Africa    39.613   4513.480643
       1043     Mozambique  2007    19951656    Africa    42.082    823.685621
       1691         Zambia  2007    11746035    Africa    42.384   1271.211593
       1355   Sierra Leone  2007     6144562    Africa    42.568    862.540756
       887         Lesotho  2007     2012649    Africa    42.592   1569.331442
       …               …    …           …         …         …           …
       408         Denmark  1952     4334000    Europe    70.780   9692.385245
       1464         Sweden  1952     7124673    Europe    71.860   8527.844662
       1080    Netherlands  1952    10381988    Europe    72.130   8941.571858
       684         Iceland  1952      147962    Europe    72.490   7267.688428
       1140         Norway  1952     3327728    Europe    72.670  10095.421720

       [1704 rows x 6 columns]
```

```
[148]: df.sort_values(['gdp_cap','population']).head()
```

```
[148]:              country  year  population continent  life_exp     gdp_cap
       334  Congo, Dem. Rep.  2002    55379852    Africa    44.966  241.165876
       335  Congo, Dem. Rep.  2007    64606759    Africa    46.462  277.551859
       876           Lesotho  1952      748747    Africa    42.138  298.846212
       624     Guinea-Bissau  1952      580653    Africa    32.500  299.850319
       333  Congo, Dem. Rep.  1997    47798986    Africa    42.587  312.188423
```

```
[149]: df.sort_values(['gdp_cap','population'],ascending=[False,True]).head()
```

```
[149]:     country  year  population continent  life_exp        gdp_cap
       853  Kuwait  1957      212846      Asia    58.033  113523.13290
       856  Kuwait  1972      841934      Asia    67.712  109347.86700
       852  Kuwait  1952      160000      Asia    55.565  108382.35290
       854  Kuwait  1962      358266      Asia    60.470   95458.11176
       855  Kuwait  1967      575003      Asia    64.624   80894.88326
```

7 Mathematical Functions

```
[150]: le = df['life_exp']
```

```
[151]: le.min()
```

```
[151]: 23.599
```

```
[152]: le.max()
```

```
[152]: 82.603
```

```
[153]: le.mean()
```

```
[153]: 59.474439366197174
```

```
[154]: le.std()
```

```
[154]: 12.917107415241192
```

```
[155]: le.var()
```

```
[155]: 166.851663976879
```

```
[156]: le.mode()
```

```
[156]: 0    69.39
        Name: life_exp, dtype: float64
```

```
[157]: le.count()
```

```
[157]: 1704
```

```
[158]: pop = df['population']
```

```
[159]: pop.min()
```

```
[159]: 60011
```

```
[160]: pop.max()
```

```
[160]: 1318683096
```

```
[161]: pop.mean()
```

```
[161]: 29601212.324530516
```

```
[162]: pop.sum()
```

```
[162]: 50440465801
```

```
[163]: gdp = df['gdp_cap']
```

```
[164]: gdp.min()
```

```
[164]: 241.1658765
```

```
[165]: gdp.max()
```

```
[165]: 113523.1329
```

```
[166]: gdp.mean()
```

```
[166]: 7215.327081212149
```

```
[167]: gdp.sum()
```

```
[167]: 12294917.346385501
```

8 Joining & Merging Tables

```
[168]: users = pd.DataFrame(
           {
               'user_id':[1,2,3,4,5],
               'name':['Sai','Preethi','Shamika','Veenasree','Sharan']
           }
       )
```

```
[169]: users
```

```
[169]:    user_id       name
       0        1        Sai
       1        2    Preethi
       2        3    Shamika
       3        4  Veenasree
       4        5     Sharan
```

```
[170]: msgs = pd.DataFrame(
           {
               'user_id':[1,1,3,4,2],
               'message':['hi','hello','Fine!','How are you ?','Bye']
           }
       )
```

```
[171]: msgs
```

```
[171]:    user_id        message
       0        1             hi
       1        1          hello
       2        3          Fine!
       3        4  How are you ?
       4        2            Bye
```

```
[172]: pd.concat([users,msgs],ignore_index=True)  # Union, vstack, full join
```

```
[172]:    user_id       name        message
       0        1        Sai            NaN
       1        2    Preethi            NaN
       2        3    Shamika            NaN
       3        4  Veenasree            NaN
       4        5     Sharan            NaN
       5        1        NaN             hi
       6        1        NaN          hello
       7        3        NaN          Fine!
       8        4        NaN  How are you ?
       9        2        NaN            Bye
```

```
[173]: pd.concat([users,msgs],axis=1) #hstack
```

```
[173]:    user_id       name  user_id        message
       0        1        Sai        1             hi
       1        2    Preethi        1          hello
       2        3    Shamika        3          Fine!
       3        4  Veenasree        4  How are you ?
       4        5     Sharan        2            Bye
```

9 Joining two tables

- ### 9.0.1 pd.merge(table1, table2, on='comman_column', how='Type_of_join')

- ### 9.0.2 table1.merge(table2, on='comman_column', how='Type_of_join')

```
[174]: pd.merge(users,msgs,on='user_id')
```

```
[174]:    user_id       name        message
       0        1        Sai             hi
       1        1        Sai          hello
       2        2    Preethi            Bye
       3        3    Shamika          Fine!
       4        4  Veenasree  How are you ?
```

```
[175]: users.merge(msgs,on='user_id',how='outer')
```

```
[175]:    user_id       name        message
       0        1        Sai             hi
       1        1        Sai          hello
       2        2    Preethi            Bye
       3        3    Shamika          Fine!
       4        4  Veenasree  How are you ?
       5        5     Sharan            NaN
```

```
[176]: users.merge(msgs,on='user_id',how='right')
```

```
[176]:    user_id       name        message
       0        1        Sai             hi
       1        1        Sai          hello
       2        3    Shamika          Fine!
       3        4  Veenasree  How are you ?
       4        2    Preethi            Bye
```

```
[179]: users.rename(columns={'user_id':'id'},inplace=True)
```

```
[180]: users
```

```
[180]:    id       name
       0   1        Sai
       1   2    Preethi
       2   3    Shamika
       3   4  Veenasree
       4   5     Sharan
```

```
[186]: users.merge(msgs,left_on='id',right_on='user_id')
```

```
[186]:    id       name  user_id        message
       0   1        Sai        1             hi
       1   1        Sai        1          hello
       2   2    Preethi        2            Bye
       3   3    Shamika        3          Fine!
       4   4  Veenasree        4  How are you ?
```

```
[187]: !gdown 1s2TkjSpzNc4SyxqRrQleZyDIHlc7bxnd
```

```
Downloading…
From: https://drive.google.com/uc?id=1s2TkjSpzNc4SyxqRrQleZyDIHlc7bxnd
To: C:\Data\Data_science\Data Science RIA\3 Python\Pandas\Codes\movies.csv

  0%|          | 0.00/112k [00:00<?, ?B/s]
100%|##########| 112k/112k [00:00<00:00, 1.16MB/s]
```

[188]:
```
!gdown 1Ws-_s1fHZ9nHfGLVUQurbHDvStePlEJm
```

```
Downloading…
From: https://drive.google.com/uc?id=1Ws-_s1fHZ9nHfGLVUQurbHDvStePlEJm
To: C:\Data\Data_science\Data Science RIA\3 Python\Pandas\Codes\directors.csv

  0%|          | 0.00/65.4k [00:00<?, ?B/s]
100%|##########| 65.4k/65.4k [00:00<00:00, 1.53MB/s]
```

[223]:
```
movies = pd.read_csv("movies.csv") # to choose index col throw an argument␣
  ↪index_col = 0
```

[224]:
```
directors = pd.read_csv("directors.csv")
```

[225]:
```
movies.shape
```

[225]:
```
(1465, 12)
```

[226]:
```
directors.shape
```

[226]:
```
(2349, 4)
```

[227]:
```
movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1465 entries, 0 to 1464
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Unnamed: 0    1465 non-null   int64
 1   id            1465 non-null   int64
 2   budget        1465 non-null   int64
 3   popularity    1465 non-null   int64
 4   revenue       1465 non-null   int64
 5   title         1465 non-null   object
 6   vote_average  1465 non-null   float64
 7   vote_count    1465 non-null   int64
 8   director_id   1465 non-null   int64
 9   year          1465 non-null   int64
 10  month         1465 non-null   object
 11  day           1465 non-null   object
dtypes: float64(1), int64(8), object(3)
memory usage: 137.5+ KB
```

[228]:
```
directors.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2349 entries, 0 to 2348
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Unnamed: 0     2349 non-null   int64
 1   director_name  2349 non-null   object
 2   id             2349 non-null   int64
 3   gender         1724 non-null   object
dtypes: int64(2), object(2)
memory usage: 73.5+ KB
```

```
[229]: movies.ndim
```

```
[229]: 2
```

```
[230]: directors.ndim
```

```
[230]: 2
```

```
[231]: movies.drop('Unnamed: 0',axis=1,inplace=True)
```

```
[232]: directors.drop('Unnamed: 0',axis=1,inplace=True)
```

```
[234]: movies.sort_values('vote_count',ascending=False)
```

```
[234]:         id     budget  popularity     revenue  \
       59    43693  160000000         167   825532764
       45    43662  185000000         187  1004558444
       0     43597  237000000         150  2787965087
       58    43692  165000000         724   675120017
       178   43884  100000000          82   425368238
       …       …          …           …          …
       1431  47962          0           0           0
       879   45373          0           0           0
       1438  48145     500000           0           0
       1440  48155          0           0           0
       1378  47387          0           0           0

                                title  vote_average  vote_count  director_id  \
       59                   Inception           8.1       13752         4765
       45             The Dark Knight           8.2       12002         4765
       0                       Avatar           7.2       11800         4762
       58                Interstellar           8.1       10867         4765
       178           Django Unchained           7.8       10099         4927
       …                          …            …           …           …
       1431        Walking and Talking           6.6           7         6204
       879              The Magic Flute           6.9           6         4847
       1438      Everything Put Together           5.0           2         4773
       1440  Alleluia! The Devil's Carnival           6.0           2         6056
       1378        An Everlasting Piece           6.0           1         5037
```

```
      year month        day
59    2010   Jul  Wednesday
45    2008   Jul  Wednesday
0     2009   Dec   Thursday
58    2014   Nov  Wednesday
178   2012   Dec    Tuesday
…     …   …         …
1431  1996   Jul  Wednesday
879   2006   Sep   Thursday
1438  2001   Nov     Friday
1440  2016   Mar    Tuesday
1378  2000   Dec     Friday

[1465 rows x 11 columns]
```

[ ]:

[ ]:

[ ]: