13_MultiIndexing

August 18, 2022

1 MultiIndexes

1.1 AKA Heirarchical Indexes

[3]:
```
import pandas as pd
```

[4]:
```
carstocks = pd.read_csv("C:/Users/ashuv/Desktop/DataAnalysis/data/car_stocks.
↪csv")
```

[5]:
```
titanic = pd.read_csv("C:/Users/ashuv/Desktop/DataAnalysis/data/titanic.csv")
titanic['age'] = titanic["age"].replace(['?'], [None]).astype('float')
titanic['fare'] = titanic["fare"].replace(['?'], [None]).astype('float')
```

[6]:
```
s1 = titanic.groupby("sex")["age"].mean()
```

[5]:
```
s1.index
```

[5]:
```
Index(['female', 'male'], dtype='object', name='sex')
```

[7]:
```
s1
```

[7]:
```
sex
female    28.687071
male      30.585233
Name: age, dtype: float64
```

1.2 Grouping By Multiple Columns!

[7]:
```
df = titanic.groupby(["pclass", "sex"]).mean()
```

[8]:
```
df.index
```

[8]:
```
MultiIndex([(1, 'female'),
            (1,   'male'),
            (2, 'female'),
            (2,   'male'),
            (3, 'female'),
            (3,   'male')],
```

```
            names=['pclass', 'sex'])
```

[9]:
```
titanic.index
```

[9]:
```
RangeIndex(start=0, stop=1309, step=1)
```

[10]:
```
df
```

[10]:
```
                survived        age     sibsp     parch        fare
pclass sex
1      female   0.965278  37.037594  0.555556  0.472222  109.412385
       male     0.340782  41.029250  0.340782  0.279330   69.888385
2      female   0.886792  27.499191  0.500000  0.650943   23.234827
       male     0.146199  30.815401  0.327485  0.192982   19.904946
3      female   0.490741  22.185307  0.791667  0.731481   15.324250
       male     0.152130  25.962273  0.470588  0.255578   12.415462
```

[9]:
```
titanic.groupby(["age", "sex"]).mean()
```

[9]:
```
                pclass  survived  sibsp  parch     fare
age     sex
0.1667  female     3.0       1.0    1.0    2.0  20.5750
0.3333  male       3.0       0.0    0.0    2.0  14.4000
0.4167  male       3.0       1.0    0.0    1.0   8.5167
0.6667  male       2.0       1.0    1.0    1.0  14.5000
0.7500  female     3.0       1.0    2.0    1.0  19.2583
…                  …         …      …      …       …
70.5000 male       3.0       0.0    0.0    0.0   7.7500
71.0000 male       1.0       0.0    0.0    0.0  42.0792
74.0000 male       3.0       0.0    0.0    0.0   7.7750
76.0000 female     1.0       1.0    1.0    0.0  78.8500
80.0000 male       1.0       1.0    0.0    0.0  30.0000

[166 rows x 5 columns]
```

1.3 Creating Your Own MultiIndex

[11]:
```
pops = pd.read_csv("C:/Users/ashuv/Desktop/DataAnalysis/data/state_pops.csv")
```

[12]:
```
pops
```

[12]:
```
     state  year   population
0       AL  2012    4817528.0
1       AL  2010    4785570.0
2       AL  2011    4801627.0
3       AL  2009    4757938.0
4       AL  2013    4833722.0
…    …   …           …
```

```
1267   USA  2013  316128839.0
1268   USA  2009  306771529.0
1269   USA  2010  309326295.0
1270   USA  2011  311582564.0
1271   USA  2012  313873685.0

[1272 rows x 3 columns]
```

[13]:
```
pops.set_index("state")
```

[13]:
```
        year   population
state
AL      2012    4817528.0
AL      2010    4785570.0
AL      2011    4801627.0
AL      2009    4757938.0
AL      2013    4833722.0
…       …          …
USA     2013  316128839.0
USA     2009  306771529.0
USA     2010  309326295.0
USA     2011  311582564.0
USA     2012  313873685.0

[1272 rows x 2 columns]
```

[14]:
```
pops.set_index("year")
```

[14]:
```
      state   population
year
2012    AL    4817528.0
2010    AL    4785570.0
2011    AL    4801627.0
2009    AL    4757938.0
2013    AL    4833722.0
…     …          …
2013   USA  316128839.0
2009   USA  306771529.0
2010   USA  309326295.0
2011   USA  311582564.0
2012   USA  313873685.0

[1272 rows x 2 columns]
```

[15]:
```
pops.set_index(["state", "year"])
```

[15]:
```
              population
state year
AL    2012    4817528.0
      2010    4785570.0
      2011    4801627.0
      2009    4757938.0
      2013    4833722.0
…                   …
USA   2013  316128839.0
      2009  306771529.0
      2010  309326295.0
      2011  311582564.0
      2012  313873685.0

[1272 rows x 1 columns]
```

[16]:
```
pops.set_index(["year", "state"])
```

[16]:
```
              population
year state
2012 AL        4817528.0
2010 AL        4785570.0
2011 AL        4801627.0
2009 AL        4757938.0
2013 AL        4833722.0
…                   …
     USA     316128839.0
2009 USA     306771529.0
2010 USA     309326295.0
2011 USA     311582564.0
2012 USA     313873685.0

[1272 rows x 1 columns]
```

[17]:
```
pops.set_index(["state", "year"], inplace=True)
```

[18]:
```
pops
```

[18]:
```
              population
state year
AL    2012    4817528.0
      2010    4785570.0
      2011    4801627.0
      2009    4757938.0
      2013    4833722.0
…                   …
USA   2013  316128839.0
```

```
      2009  306771529.0
      2010  309326295.0
      2011  311582564.0
      2012  313873685.0

[1272 rows x 1 columns]
```

1.4 Sorting A MultiIndex

[19]:
```
pops.sort_index()
```

[19]:
```
            population
state year
AK    1990    553290.0
      1991    570193.0
      1992    588736.0
      1993    599434.0
      1994    603308.0
…                …
WY    2009    559851.0
      2010    564222.0
      2011    567329.0
      2012    576626.0
      2013    582658.0

[1272 rows x 1 columns]
```

[20]:
```
pops.sort_index(ascending=False)
```

[20]:
```
            population
state year
WY    2013    582658.0
      2012    576626.0
      2011    567329.0
      2010    564222.0
      2009    559851.0
…                …
AK    1994    603308.0
      1993    599434.0
      1992    588736.0
      1991    570193.0
      1990    553290.0

[1272 rows x 1 columns]
```

[21]:
```
pops.sort_index(level=1)
```

[21]:
```
            population
state year
AK    1990    553290.0
AL    1990   4050055.0
AR    1990   2356586.0
AZ    1990   3684097.0
CA    1990  29959515.0
…                …
VT    2013    626630.0
WA    2013   6971406.0
WI    2013   5742713.0
WV    2013   1854304.0
WY    2013    582658.0

[1272 rows x 1 columns]
```

[23]:
```
pops.sort_index(level=[1,0],ascending=[False,True])
```

[23]:
```
            population
state year
AK    2013    735132.0
AL    2013   4833722.0
AR    2013   2959373.0
AZ    2013   6626624.0
CA    2013  38332521.0
…                …
VT    1990    564798.0
WA    1990   4903043.0
WI    1990   4904562.0
WV    1990   1792548.0
WY    1990    453690.0

[1272 rows x 1 columns]
```

[22]:
```
pops.sort_index(inplace=True)
```

[23]:
```
pops
```

[23]:
```
            population
state year
AK    1990    553290.0
      1991    570193.0
      1992    588736.0
      1993    599434.0
      1994    603308.0
…                …
WY    2009    559851.0
```

```
      2010    564222.0
      2011    567329.0
      2012    576626.0
      2013    582658.0

[1272 rows x 1 columns]
```

1.5 loc [] with MultiIndexes

[24]:
```
pops.loc["CA"]
```

[24]:
```
      population
year
1990  29959515.0
1991  30470736.0
1992  30974659.0
1993  31274928.0
1994  31484435.0
1995  31696582.0
1996  32018834.0
1997  32486010.0
1998  32987675.0
1999  33499204.0
2000  33987977.0
2001  34479458.0
2002  34871843.0
2003  35253159.0
2004  35574576.0
2005  35827943.0
2006  36021202.0
2007  36250311.0
2008  36604337.0
2009  36961229.0
2010  37333601.0
2011  37668681.0
2012  37999878.0
2013  38332521.0
```

[25]:
```
pops.loc[["CA", "AK"]]
```

[25]:
```
            population
state year
CA    1990  29959515.0
      1991  30470736.0
      1992  30974659.0
      1993  31274928.0
      1994  31484435.0
```

```
          1995  31696582.0
          1996  32018834.0
          1997  32486010.0
          1998  32987675.0
          1999  33499204.0
          2000  33987977.0
          2001  34479458.0
          2002  34871843.0
          2003  35253159.0
          2004  35574576.0
          2005  35827943.0
          2006  36021202.0
          2007  36250311.0
          2008  36604337.0
          2009  36961229.0
          2010  37333601.0
          2011  37668681.0
          2012  37999878.0
          2013  38332521.0
AK        1990    553290.0
          1991    570193.0
          1992    588736.0
          1993    599434.0
          1994    603308.0
          1995    604412.0
          1996    608569.0
          1997    612968.0
          1998    619933.0
          1999    624779.0
          2000    627963.0
          2001    633714.0
          2002    642337.0
          2003    648414.0
          2004    659286.0
          2005    666946.0
          2006    675302.0
          2007    680300.0
          2008    687455.0
          2009    698895.0
          2010    713868.0
          2011    723375.0
          2012    730307.0
          2013    735132.0
```

[28]:
```
pops.loc["NM": "TX"]
```

[28]:
```
            population
state year
NM    1990   1521574.0
      1991   1555305.0
      1992   1595442.0
      1993   1636453.0
      1994   1682398.0
…                …
TX    2009  24801761.0
      2010  25245178.0
      2011  25640909.0
      2012  26060796.0
      2013  26448193.0

[312 rows x 1 columns]
```

[29]:
```
pops.loc[("MT",1992)]
```

[29]:
```
population    825770.0
Name: (MT, 1992), dtype: float64
```

[30]:
```
pops.loc[("CA", 2013)]
```

[30]:
```
population    38332521.0
Name: (CA, 2013), dtype: float64
```

[26]:
```
pops.loc[("AK", 1990):("AK",1995)]
```

[26]:
```
            population
state year
AK    1990    553290.0
      1991    570193.0
      1992    588736.0
      1993    599434.0
      1994    603308.0
      1995    604412.0
```

[32]:
```
pops.info()
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 1272 entries, ('AK', 1990) to ('WY', 2013)
Data columns (total 1 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   population  1262 non-null   float64
dtypes: float64(1)
memory usage: 48.0+ KB
```

[27]:
```
pops.loc[("AK", 2011):("AL",1993)]
```

[27]:
```
            population
state year
AK    2011    723375.0
      2012    730307.0
      2013    735132.0
AL    1990   4050055.0
      1991   4099156.0
      1992   4154014.0
      1993   4214202.0
```

[29]:
```
# This won't work!
pops.loc[:,1990,:]
```

[29]:
```
        population
state
AK        553290.0
AL       4050055.0
AR       2356586.0
AZ       3684097.0
CA      29959515.0
CO       3307618.0
CT       3291967.0
DC        605321.0
DE        669567.0
FL      13033307.0
GA       6512602.0
HI       1113491.0
IA       2781018.0
ID       1012384.0
IL      11453316.0
IN       5557798.0
KS       2481349.0
KY       3694048.0
LA       4221532.0
MA       6022639.0
MD       4799770.0
ME       1231719.0
MI       9311319.0
MN       4389857.0
MO       5128880.0
MS       2578897.0
MT        800204.0
NC       6664016.0
ND        637685.0
NE       1581660.0
```

```
NH       1112384.0
NJ       7762963.0
NM       1521574.0
NV       1220695.0
NY      18020784.0
OH      10864162.0
OK       3148825.0
OR       2860375.0
PA      11903299.0
PR             NaN
RI       1005995.0
SC       3501155.0
SD        697101.0
TN       4894492.0
TX      17056755.0
USA    249622814.0
UT       1731223.0
VA       6216884.0
VT        564798.0
WA       4903043.0
WI       4904562.0
WV       1792548.0
WY        453690.0
```

[35]:
```
titanic.loc[20, "name"]
```

[35]:
```
'Beckwith, Mr. Richard Leonard'
```

[31]:
```
pops.loc["AL", 1990, :]
```

[31]:
```
            population
state year
AL    1990   4050055.0
```

[ ]:
```
pops.loc[:, [1990,1991], :]
```

[ ]:
```
pops.loc[slice(None), [1990,1991], :]
```

[ ]:
```
pops.loc[:,2013,:]
```

1.6 The .xs( ) Method

[33]:
```
pops.xs(2013, level="year").mean()
```

[33]:
```
population    1.199760e+07
dtype: float64
```

[34]:
```
pops.loc[:,2013,:].mean()
```

[34]:
```
population    1.199760e+07
dtype: float64
```

[ ]:
```
pops.xs(2013, level=1)
```

1.7 get_level_values( )

[ ]:
```
pops.index.levels
```

[ ]:
```
pops.index.get_level_values(0)
```

[ ]:
```
pops.index.get_level_values(1)
```

[ ]:
```
pops.loc[:, [1990, 1992, 1994], :]
```

[ ]:
```
even_years = pops.index.get_level_values(1) % 2 == 0
```

[ ]:
```
len(even_years)
```

[ ]:
```
len(pops)
```

[ ]:
```
even_years
```

[ ]:
```
pops[even_years]
```

[ ]:
```
pops[pops["population"] % 2 == 0]
```

[ ]:
```
even_pops = pops["population"] % 2 == 0
```

[ ]:
```
pops[even_years & even_pops]
```

[ ]:
```
ends_with_a = pops.index.get_level_values(0).str[1] == "A"
```

[ ]:
```
pops[ends_with_a]
```

1.8 Heirarchical Columns!

[ ]:
```
df = titanic.groupby(["pclass", "sex"]).mean()
```

[ ]:
```
df.loc[(2, "male")]
```

[ ]:
```
df
```

[ ]:
```
df = titanic.groupby("sex").agg({
    "age": ["min", "max", "mean"],
    "fare": ["min", "max", "mean"],
```

```
    "survived": ["mean"]
})
```

[ ]:
```
df
```

[ ]:
```
df.index
```

[ ]:
```
df.columns
```

[ ]:
```
df[("age", "mean")]
```

[ ]:
```
df["age"]["mean"]
```

[ ]:
```
df
```

1.9 Stack( ) and Unstack( )

[ ]:
```
pops
```

[ ]:
```
unstacked_df = pops.unstack(level="state")
```

[ ]:
```
unstacked_df.stack().unstack()
```

[ ]:
```
titanic.groupby(["pclass", "sex"])["age"].mean()
```

[ ]:
```
titanic.groupby(["pclass", "sex"])["age"].mean().plot(kind="bar")
```

[ ]:
```
titanic.groupby(["pclass", "sex"])["age"].mean().unstack()
```

[ ]:
```
titanic.groupby(["pclass", "sex"])["age"].mean().unstack().plot(kind="bar")
```

[ ]:
```
titanic.groupby(["pclass", "sex"])["age"].mean().unstack(level="pclass")
```

[ ]:
```
titanic.groupby(["pclass", "sex"])["age"].mean().unstack(level="pclass").
↪plot(kind="bar")
```

[ ]:
```
titanic.groupby(["sex", "survived"])["age"].mean()
```

[ ]:
```
titanic.groupby(["sex", "survived"])["age"].mean().unstack()
```

[ ]:
```
titanic.groupby(["sex", "survived"])["age"].mean().unstack().plot(kind="bar")
```

[ ]:
```
df = titanic.groupby(["sex", "survived"])["age"].mean()
df.unstack().rename(columns={0: "Died", 1: "Survived"}).plot(kind="bar")
```

[ ]:
```
pops.groupby(level=1).sum()
```

[ ]:
```
pops.groupby(level=0).min()
```

[ ]:
```
pops.groupby(level=[1,0]).min()
```

[ ]:
```
pops.index
```

[ ]:
```
pops.groupby(["year", "state"]).min()
```