Lecture5

January 24, 2018

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import sqlalchemy

In [2]: # let's continue on with our adventures with dataframes

In [3]: sailors = pd.read_csv("/Users/mikejcarey/Desktop/teaching/STATS170ab/Sailors2.csv")

In [4]: sailors
```

Out[4]:

	sid	sname	rating	age	major	hobbies
0	22	Dustin	7.0	45.0	CS	sailing,surfing
1	29	Brutus	1.0	33.0	EE	music
2	31	Lubber	8.0	55.5	Econ	coins,stamps,investing
3	32	Andy	8.0	25.5	Math	running
4	58	Rusty	10.0	35.0	CS	games
5	64	Horatio	7.0	35.0	CS	music,games
6	71	Zorba	10.0	16.0	CS	karate,running,music
7	74	Horatio	9.0	35.0	Math	music
8	85	Art	4.0	25.5	Music	NaN
9	95	Bob	3.0	63.5	Econ	sailing
10	101	Joan	3.0	NaN	Math	surfing,running,music
11	107	Johannes	NaN	35.0	NaN	music,karate

```
In [5]: boats = pd.read_json("/Users/mikejcarey/Desktop/teaching/STATS170ab/Boats.json", lines='

In [6]: boats
```

Out[6]:

	bid	bname	color
0	101	Interlake	blue
1	102	Interlake	red
2	103	Clipper	green
3	104	Marine	red

```
In [7]: from sqlalchemy import create_engine
        engine = create_engine('postgresql+psycopg2://mikejcarey:postquel@localhost/mikejcarey')

In [8]: reserves = pd.read_sql_query('SELECT * from reserves',con=engine)
```

```
In [9]: reserves
```

Out[9]:

	sid	bid	date
0	22.0	101.0	1998-10-10
1	22.0	102.0	1998-10-10
2	22.0	103.0	1998-10-08
3	22.0	104.0	1998-10-07
4	31.0	102.0	1998-10-11
5	31.0	103.0	1998-11-06
6	31.0	104.0	1998-11-12
7	64.0	101.0	1998-09-05
8	64.0	102.0	1998-09-02
9	74.0	103.0	1993-09-08
10	NaN	103.0	1998-09-09
11	1.0	NaN	2001-01-11
12	1.0	NaN	2002-02-02

```
In [10]: moreboats = pd.read_json("/Users/mikejcarey/Desktop/teaching/STATS170ab/MoreBoats.json"
```

```
In [11]: moreboats
```

Out[11]:

	bid	bname	color
0	201	Sunfish	blue
1	202	Sunfish	red
2	203	Yacht	green
3	204	Barge	red

```
In [12]: # we left off having just done relational-style unions (and more)
```

```
In [13]: pd.concat([boats, moreboats])
```

Out[13]:

	bid	bname	color
0	101	Interlake	blue
1	102	Interlake	red
2	103	Clipper	green
3	104	Marine	red
0	201	Sunfish	blue
1	202	Sunfish	red
2	203	Yacht	green
3	204	Barge	red

```
In [14]: pd.concat([boats, moreboats], ignore_index=True)
```

Out[14]:

	bid	bname	color
0	101	Interlake	blue
1	102	Interlake	red
2	103	Clipper	green
3	104	Marine	red
4	201	Sunfish	blue
5	202	Sunfish	red
6	203	Yacht	green
7	204	Barge	red

```
In [15]: pd.concat([boats, moreboats], axis=1)
```

```
Out[15]:    bid      bname  color  bid    bname  color
         0  101  Interlake   blue  201  Sunfish   blue
         1  102  Interlake    red  202  Sunfish    red
         2  103    Clipper  green  203    Yacht  green
         3  104     Marine    red  204    Barge    red
```

```
In [16]: # we can do relational-style grouping and aggregation (and more)

         bymajor = sailors.groupby('major')
```

```
In [17]: bymajor
```

```
Out[17]: <pandas.core.groupby.DataFrameGroupBy object at 0x10cb4db00>
```

```
In [18]: bymajor.size()
```

```
Out[18]: major
         CS       4
         EE       1
         Econ     2
         Math     3
         Music    1
         dtype: int64
```

```
In [19]: bymajor.count()
```

```
Out[19]:        sid  sname  rating  age  hobbies
         major
         CS       4      4       4    4        4
         EE       1      1       1    1        1
         Econ     2      2       2    2        2
         Math     3      3       3    2        3
         Music    1      1       1    1        0
```

```
In [20]: bymajor.rating.max()
```

```
Out[20]: major
         CS       10.0
         EE        1.0
         Econ      8.0
         Math      9.0
         Music     4.0
         Name: rating, dtype: float64
```

```
In [21]: bymajor.max()[['rating','age']]
```

```
Out[21]:        rating   age
         major
         CS       10.0  45.0
         EE        1.0  33.0
         Econ      8.0  63.5
         Math      9.0  35.0
         Music     4.0  25.5

In [22]: bymajor.agg({'rating' : ['min', 'max'], 'age' : ['count','mean','std']})

Out[22]:       rating         age
                  min   max count   mean        std
         major
         CS       7.0  10.0     4  32.75  12.120919
         EE       1.0   1.0     1  33.00        NaN
         Econ     3.0   8.0     2  59.50   5.656854
         Math     3.0   9.0     2  30.25   6.717514
         Music    4.0   4.0     1  25.50        NaN

In [23]: # we can do relational-style ordering and limiting as well
         bymajor.rating.max().sort_values(ascending=False)[:3]

Out[23]: major
         CS      10.0
         Math     9.0
         Econ     8.0
         Name: rating, dtype: float64

In [24]: # we can also extend a dataframe with a new column, e.g., dogage
         sailors.age / 7

Out[24]: 0     6.428571
         1     4.714286
         2     7.928571
         3     3.642857
         4     5.000000
         5     5.000000
         6     2.285714
         7     5.000000
         8     3.642857
         9     9.071429
         10         NaN
         11    5.000000
         Name: age, dtype: float64

In [25]: sailors.assign(dogage = sailors.age / 7)

Out[25]:     sid     sname  rating   age  major                hobbies    dogage
         0    22    Dustin     7.0  45.0     CS         sailing,surfing  6.428571
```

```
1    29    Brutus     1.0  33.0     EE                   music  4.714286
2    31    Lubber     8.0  55.5   Econ  coins,stamps,investing  7.928571
3    32      Andy     8.0  25.5   Math                 running  3.642857
4    58     Rusty    10.0  35.0     CS                   games  5.000000
5    64   Horatio     7.0  35.0     CS            music,games  5.000000
6    71     Zorba    10.0  16.0     CS   karate,running,music  2.285714
7    74   Horatio     9.0  35.0   Math                   music  5.000000
8    85       Art     4.0  25.5  Music                     NaN  3.642857
9    95       Bob     3.0  63.5   Econ                 sailing  9.071429
10  101      Joan     3.0   NaN   Math  surfing,running,music       NaN
11  107  Johannes     NaN  35.0    NaN           music,karate  5.000000
```

In [26]:
```
# and remember, as always, Python and dataframes are functional in nature, so...
sailors
```

Out[26]:
```
    sid     sname  rating   age  major                 hobbies
0    22    Dustin     7.0  45.0     CS         sailing,surfing
1    29    Brutus     1.0  33.0     EE                   music
2    31    Lubber     8.0  55.5   Econ  coins,stamps,investing
3    32      Andy     8.0  25.5   Math                 running
4    58     Rusty    10.0  35.0     CS                   games
5    64   Horatio     7.0  35.0     CS            music,games
6    71     Zorba    10.0  16.0     CS   karate,running,music
7    74   Horatio     9.0  35.0   Math                   music
8    85       Art     4.0  25.5  Music                     NaN
9    95       Bob     3.0  63.5   Econ                 sailing
10  101      Joan     3.0   NaN   Math  surfing,running,music
11  107  Johannes     NaN  35.0    NaN           music,karate
```

In [27]:
```
# we can also tackle nested data in Pandas (with some effort)
```

In [28]:
```
# one approach to handling nested data is not to (i.e., we can normalize it)
sclean = sailors.fillna(value = '')
sarray = sclean.hobbies.str.split(',')
s1NF = sclean.loc[sclean.index.repeat(sarray.str.len())].assign(hobbies=np.concatenate(
s1NF.reset_index(drop = True, inplace = True)
```

In [29]:
```
s1NF
```

Out[29]:
```
    sid     sname rating   age  major    hobbies
0    22    Dustin      7    45     CS    sailing
1    22    Dustin      7    45     CS    surfing
2    29    Brutus      1    33     EE      music
3    31    Lubber      8  55.5   Econ      coins
4    31    Lubber      8  55.5   Econ     stamps
5    31    Lubber      8  55.5   Econ  investing
6    32      Andy      8  25.5   Math    running
7    58     Rusty     10    35     CS      games
8    64   Horatio      7    35     CS      music
```

```
9    64   Horatio      7    35     CS      games
10   71     Zorba     10    16     CS     karate
11   71     Zorba     10    16     CS    running
12   71     Zorba     10    16     CS      music
13   74   Horatio      9    35   Math      music
14   85       Art      4  25.5  Music
15   95       Bob      3  63.5   Econ    sailing
16  101      Joan      3         Math    surfing
17  101      Joan      3         Math    running
18  101      Joan      3         Math      music
19  107  Johannes           35              music
20  107  Johannes           35             karate
```

```
In [30]: # let's break this magic down into its steps
```

```
In [31]: sclean = sailors.fillna(value = '')
         sclean
```

Out[31]:

```
    sid     sname rating   age  major                 hobbies
0    22    Dustin      7    45     CS         sailing,surfing
1    29    Brutus      1    33     EE                   music
2    31    Lubber      8  55.5   Econ  coins,stamps,investing
3    32      Andy      8  25.5   Math                 running
4    58     Rusty     10    35     CS                   games
5    64   Horatio      7    35     CS             music,games
6    71     Zorba     10    16     CS    karate,running,music
7    74   Horatio      9    35   Math                   music
8    85       Art      4  25.5  Music
9    95       Bob      3  63.5   Econ                 sailing
10  101      Joan      3         Math   surfing,running,music
11  107  Johannes           35                  music,karate
```

```
In [32]: sarray = sclean.hobbies.str.split(',')
         sarray
```

Out[32]:

```
0              [sailing, surfing]
1                         [music]
2     [coins, stamps, investing]
3                       [running]
4                         [games]
5                  [music, games]
6       [karate, running, music]
7                         [music]
8                              []
9                       [sailing]
10     [surfing, running, music]
11                [music, karate]
Name: hobbies, dtype: object
```

```
In [33]: sarray.str.len()
```

```
Out[33]: 0     2
         1     1
         2     3
         3     1
         4     1
         5     2
         6     3
         7     1
         8     1
         9     1
         10    3
         11    2
         Name: hobbies, dtype: int64
```

```
In [34]: sclean.index.repeat(sarray.str.len())
```

```
Out[34]: Int64Index([0, 0, 1, 2, 2, 2, 3, 4, 5, 5, 6, 6, 6, 7, 8, 9, 10, 10, 10, 11,
                     11],
                    dtype='int64')
```

```
In [35]: sclean.loc[sclean.index.repeat(sarray.str.len())]
```

Out[35]:

	sid	sname	rating	age	major	hobbies
0	22	Dustin	7	45	CS	sailing,surfing
0	22	Dustin	7	45	CS	sailing,surfing
1	29	Brutus	1	33	EE	music
2	31	Lubber	8	55.5	Econ	coins,stamps,investing
2	31	Lubber	8	55.5	Econ	coins,stamps,investing
2	31	Lubber	8	55.5	Econ	coins,stamps,investing
3	32	Andy	8	25.5	Math	running
4	58	Rusty	10	35	CS	games
5	64	Horatio	7	35	CS	music,games
5	64	Horatio	7	35	CS	music,games
6	71	Zorba	10	16	CS	karate,running,music
6	71	Zorba	10	16	CS	karate,running,music
6	71	Zorba	10	16	CS	karate,running,music
7	74	Horatio	9	35	Math	music
8	85	Art	4	25.5	Music	
9	95	Bob	3	63.5	Econ	sailing
10	101	Joan	3		Math	surfing,running,music
10	101	Joan	3		Math	surfing,running,music
10	101	Joan	3		Math	surfing,running,music
11	107	Johannes		35		music,karate
11	107	Johannes		35		music,karate

```
In [36]: np.concatenate(sarray)
```

```
Out[36]: array(['sailing', 'surfing', 'music', 'coins', 'stamps', 'investing',
                'running', 'games', 'music', 'games', 'karate', 'running', 'music',
                'music', '', 'sailing', 'surfing', 'running', 'music', 'music',
                'karate'],
               dtype='<U9')
```

```
In [37]: sclean.loc[sclean.index.repeat(sarray.str.len())].assign(hobbies=np.concatenate(sarray)
```

Out[37]:

	sid	sname	rating	age	major	hobbies
0	22	Dustin	7	45	CS	sailing
0	22	Dustin	7	45	CS	surfing
1	29	Brutus	1	33	EE	music
2	31	Lubber	8	55.5	Econ	coins
2	31	Lubber	8	55.5	Econ	stamps
2	31	Lubber	8	55.5	Econ	investing
3	32	Andy	8	25.5	Math	running
4	58	Rusty	10	35	CS	games
5	64	Horatio	7	35	CS	music
5	64	Horatio	7	35	CS	games
6	71	Zorba	10	16	CS	karate
6	71	Zorba	10	16	CS	running
6	71	Zorba	10	16	CS	music
7	74	Horatio	9	35	Math	music
8	85	Art	4	25.5	Music	
9	95	Bob	3	63.5	Econ	sailing
10	101	Joan	3		Math	surfing
10	101	Joan	3		Math	running
10	101	Joan	3		Math	music
11	107	Johannes		35		music
11	107	Johannes		35		karate

```
In [38]: # one more time, with understanding...!  :-)
         sclean = sailors.fillna(value = '')
         sarray = sclean.hobbies.str.split(',')
         s1NF = sclean.loc[sclean.index.repeat(sarray.str.len())].assign(hobbies=np.concatenate(
         s1NF.reset_index(drop = True, inplace = True)
         s1NF
```

Out[38]:

	sid	sname	rating	age	major	hobbies
0	22	Dustin	7	45	CS	sailing
1	22	Dustin	7	45	CS	surfing
2	29	Brutus	1	33	EE	music
3	31	Lubber	8	55.5	Econ	coins
4	31	Lubber	8	55.5	Econ	stamps
5	31	Lubber	8	55.5	Econ	investing
6	32	Andy	8	25.5	Math	running
7	58	Rusty	10	35	CS	games
8	64	Horatio	7	35	CS	music
9	64	Horatio	7	35	CS	games

```
10   71     Zorba     10    16     CS     karate
11   71     Zorba     10    16     CS    running
12   71     Zorba     10    16     CS      music
13   74   Horatio      9    35   Math      music
14   85       Art      4  25.5  Music
15   95       Bob      3  63.5   Econ    sailing
16  101      Joan      3         Math    surfing
17  101      Joan      3         Math    running
18  101      Joan      3         Math      music
19  107  Johannes           35              music
20  107  Johannes           35             karate
```

In [39]:
```
# another approach to nested categorical data is to convert it to "dummy" variables (in
sailors
```

Out[39]:

	sid	sname	rating	age	major	hobbies
0	22	Dustin	7.0	45.0	CS	sailing,surfing
1	29	Brutus	1.0	33.0	EE	music
2	31	Lubber	8.0	55.5	Econ	coins,stamps,investing
3	32	Andy	8.0	25.5	Math	running
4	58	Rusty	10.0	35.0	CS	games
5	64	Horatio	7.0	35.0	CS	music,games
6	71	Zorba	10.0	16.0	CS	karate,running,music
7	74	Horatio	9.0	35.0	Math	music
8	85	Art	4.0	25.5	Music	NaN
9	95	Bob	3.0	63.5	Econ	sailing
10	101	Joan	3.0	NaN	Math	surfing,running,music
11	107	Johannes	NaN	35.0	NaN	music,karate

In [40]:
```
sailors['hobbies'].str.get_dummies(sep=',')
```

Out[40]:

	coins	games	investing	karate	music	running	sailing	stamps	surfing
0	0	0	0	0	0	0	1	0	1
1	0	0	0	0	1	0	0	0	0
2	1	0	1	0	0	0	0	1	0
3	0	0	0	0	0	1	0	0	0
4	0	1	0	0	0	0	0	0	0
5	0	1	0	0	1	0	0	0	0
6	0	0	0	1	1	1	0	0	0
7	0	0	0	0	1	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0
10	0	0	0	0	1	1	0	0	1
11	0	0	0	1	1	0	0	0	0

In [41]:
```
pd.concat([sailors, sailors['hobbies'].str.get_dummies(sep=',')], axis=1)
```

Out[41]:

	sid	sname	rating	age	major	hobbies	coins	games	\
0	22	Dustin	7.0	45.0	CS	sailing,surfing	0	0	

1	29	Brutus	1.0	33.0	EE	music	0	0
2	31	Lubber	8.0	55.5	Econ	coins,stamps,investing	1	0
3	32	Andy	8.0	25.5	Math	running	0	0
4	58	Rusty	10.0	35.0	CS	games	0	1
5	64	Horatio	7.0	35.0	CS	music,games	0	1
6	71	Zorba	10.0	16.0	CS	karate,running,music	0	0
7	74	Horatio	9.0	35.0	Math	music	0	0
8	85	Art	4.0	25.5	Music	NaN	0	0
9	95	Bob	3.0	63.5	Econ	sailing	0	0
10	101	Joan	3.0	NaN	Math	surfing,running,music	0	0
11	107	Johannes	NaN	35.0	NaN	music,karate	0	0

	investing	karate	music	running	sailing	stamps	surfing
0	0	0	0	0	1	0	1
1	0	0	1	0	0	0	0
2	1	0	0	0	0	1	0
3	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0
5	0	0	1	0	0	0	0
6	0	1	1	1	0	0	0
7	0	0	1	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
10	0	0	1	1	0	0	1
11	0	1	1	0	0	0	0

In [42]:
```
# before we move on from Pandas wrangling basics, let's look at some cleaning primitive
sailors
```

Out[42]:

	sid	sname	rating	age	major	hobbies
0	22	Dustin	7.0	45.0	CS	sailing,surfing
1	29	Brutus	1.0	33.0	EE	music
2	31	Lubber	8.0	55.5	Econ	coins,stamps,investing
3	32	Andy	8.0	25.5	Math	running
4	58	Rusty	10.0	35.0	CS	games
5	64	Horatio	7.0	35.0	CS	music,games
6	71	Zorba	10.0	16.0	CS	karate,running,music
7	74	Horatio	9.0	35.0	Math	music
8	85	Art	4.0	25.5	Music	NaN
9	95	Bob	3.0	63.5	Econ	sailing
10	101	Joan	3.0	NaN	Math	surfing,running,music
11	107	Johannes	NaN	35.0	NaN	music,karate

In [43]:
```
sailors.dropna()
```

Out[43]:

	sid	sname	rating	age	major	hobbies
0	22	Dustin	7.0	45.0	CS	sailing,surfing
1	29	Brutus	1.0	33.0	EE	music
2	31	Lubber	8.0	55.5	Econ	coins,stamps,investing

```
3   32     Andy     8.0  25.5  Math                 running
4   58    Rusty    10.0  35.0    CS                   games
5   64  Horatio     7.0  35.0    CS             music,games
6   71    Zorba    10.0  16.0    CS    karate,running,music
7   74  Horatio     9.0  35.0  Math                   music
9   95      Bob     3.0  63.5  Econ                 sailing
```

In [44]:
```
sailors.dropna(subset=['age'])
```

Out[44]:

	sid	sname	rating	age	major	hobbies
0	22	Dustin	7.0	45.0	CS	sailing,surfing
1	29	Brutus	1.0	33.0	EE	music
2	31	Lubber	8.0	55.5	Econ	coins,stamps,investing
3	32	Andy	8.0	25.5	Math	running
4	58	Rusty	10.0	35.0	CS	games
5	64	Horatio	7.0	35.0	CS	music,games
6	71	Zorba	10.0	16.0	CS	karate,running,music
7	74	Horatio	9.0	35.0	Math	music
8	85	Art	4.0	25.5	Music	NaN
9	95	Bob	3.0	63.5	Econ	sailing
11	107	Johannes	NaN	35.0	NaN	music,karate

In [45]:
```
sailors.fillna({"rating":0, "age": sailors.age.mean(), "major":"Undeclared", "hobbies":
```

Out[45]:

	sid	sname	rating	age	major	hobbies
0	22	Dustin	7.0	45.000000	CS	sailing,surfing
1	29	Brutus	1.0	33.000000	EE	music
2	31	Lubber	8.0	55.500000	Econ	coins,stamps,investing
3	32	Andy	8.0	25.500000	Math	running
4	58	Rusty	10.0	35.000000	CS	games
5	64	Horatio	7.0	35.000000	CS	music,games
6	71	Zorba	10.0	16.000000	CS	karate,running,music
7	74	Horatio	9.0	35.000000	Math	music
8	85	Art	4.0	25.500000	Music	
9	95	Bob	3.0	63.500000	Econ	sailing
10	101	Joan	3.0	36.727273	Math	surfing,running,music
11	107	Johannes	0.0	35.000000	Undeclared	music,karate

In [46]:
```
rooms = pd.read_csv("/Users/mikejcarey/Desktop/teaching/STATS170ab/Rooms.csv")
rooms
```

Out[46]:

	room	temp
0	101	71.0
1	102	72.0
2	103	73.0
3	104	NaN
4	105	75.0
5	106	76.0

In [47]:
```
rooms.fillna(method="ffill")
```

```
Out[47]:    room  temp
         0   101  71.0
         1   102  72.0
         2   103  73.0
         3   104  73.0
         4   105  75.0
         5   106  76.0

In [48]: rooms.fillna(method="bfill")

Out[48]:    room  temp
         0   101  71.0
         1   102  72.0
         2   103  73.0
         3   104  75.0
         4   105  75.0
         5   106  76.0

In [49]: rooms.interpolate()

Out[49]:    room  temp
         0   101  71.0
         1   102  72.0
         2   103  73.0
         3   104  74.0
         4   105  75.0
         5   106  76.0

In [50]: # that's enough for now about wrangling (as you can read/web-surf to learn more)
```