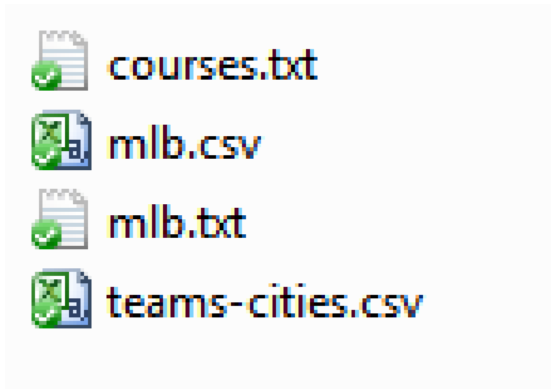
Lect 19 – Data Aggregation

Rob Capra

INLS 490-172

Data files for today

- Sakai → Resources → Lectures → lect18_data.zip



courses.txt
mlb.csv
mlb.txt
teams-cities.csv

MLB stats from last lecture

```
In [25]: !type mlb.csv
team,league,wins,losses,rs,ra
yankees,al,6,6,46,52
nationals,nl,7,5,60,50
cardinals,nl,7,5,48,48
redsox,al,5,7,44,50
braves,nl,8,4,46,33
cubs,nl,4,8,47,55
tigers,al,6,4,40,39

In [26]: stats_df = pd.read_csv('mlb.csv')

In [27]: print stats_df
        team league  wins  losses  rs  ra
0    yankees     al     6       6  46  52
1  nationals     nl     7       5  60  50
2  cardinals     nl     7       5  48  48
3     redsox     al     5       7  44  50
4     braves     nl     8       4  46  33
5       cubs     nl     4       8  47  55
6     tigers     al     6       4  40  39
```

teams-cities.csv

- Suppose we have another, related set of data that indicates the city and state for each team

```
In [28]: !type teams-cities.csv
team,city,state
yankees,new york,ny
nationals,washington,dc
cardinals,st. louis,mo
redsox,boston,ma
braves,atlanta,ga
cubs,chicago,il
tigers,detroit,mi

In [29]: city_df = pd.read_csv('teams-cities.csv')

In [30]: print city_df
        team        city state
0    yankees    new york    ny
1  nationals  washington    dc
2  cardinals   st. louis    mo
3     redsox      boston    ma
4     braves     atlanta    ga
5       cubs     chicago    il
6     tigers     detroit    mi
```

merge

- Database-style join/merge on DataFrames.

```
In [31]: print stats_df
        team league  wins  losses  rs  ra
0    yankees     al     6       6  46  52
1  nationals     nl     7       5  60  50
2  cardinals     nl     7       5  48  48
3     redsox     al     5       7  44  50
4     braves     nl     8       4  46  33
5       cubs     nl     4       8  47  55
6     tigers     al     6       4  40  39

In [32]: print city_df
        team        city state
0    yankees    new york    ny
1  nationals  washington    dc
2  cardinals   st. louis    mo
3     redsox      boston    ma
4     braves     atlanta    ga
5       cubs     chicago    il
6     tigers     detroit    mi

In [33]: print pd.merge(stats_df, city_df)
        team league  wins  losses  rs  ra        city state
0    yankees     al     6       6  46  52    new york    ny
1  nationals     nl     7       5  60  50  washington    dc
2  cardinals     nl     7       5  48  48   st. louis    mo
3     redsox     al     5       7  44  50      boston    ma
4     braves     nl     8       4  46  33     atlanta    ga
5       cubs     nl     4       8  47  55     chicago    il
6     tigers     al     6       4  40  39     detroit    mi
```

Merge on index

```
In [5]: stats_df = pd.read_csv('mlb.csv', index_col='team')

In [6]: print stats_df
          league  wins  losses  rs  ra
team
yankees       al     6       6  46  52
nationals     nl     7       5  60  50
cardinals     nl     7       5  48  48
redsox        al     5       7  44  50
braves        nl     8       4  46  33
cubs          nl     4       8  47  55
tigers        al     6       4  40  39

In [7]: city_df = pd.read_csv('teams-cities.csv', index_col='team')

In [8]: print city_df
                 city state
team
yankees      new york    ny
nationals  washington    dc
cardinals   st. louis    mo
redsox         boston    ma
braves        atlanta    ga
cubs          chicago    il
tigers        detroit    mi

In [9]: print pd.merge(stats_df, city_df, left_index=True, right_index=True)
          league  wins  losses  rs  ra        city state
team
yankees       al     6       6  46  52    new york    ny
nationals     nl     7       5  60  50  washington    dc
cardinals     nl     7       5  48  48   st. louis    mo
redsox        al     5       7  44  50      boston    ma
braves        nl     8       4  46  33     atlanta    ga
cubs          nl     4       8  47  55     chicago    il
tigers        al     6       4  40  39     detroit    mi
```

Merge on index and column

```
In [11]: print stats_df
          league  wins  losses  rs  ra
team
yankees       al     6       6  46  52
nationals     nl     7       5  60  50
cardinals     nl     7       5  48  48
redsox        al     5       7  44  50
braves        nl     8       4  46  33
cubs          nl     4       8  47  55
tigers        al     6       4  40  39

In [12]: city_df = pd.read_csv('teams-cities.csv')

In [13]: print city_df
        team        city state
0    yankees    new york    ny
1  nationals  washington    dc
2  cardinals   st. louis    mo
3     redsox      boston    ma
4     braves     atlanta    ga
5       cubs     chicago    il
6     tigers     detroit    mi

In [14]: print pd.merge(stats_df, city_df, left_index=True, right_on='team')
  league  wins  losses  rs  ra       team        city state
0     al     6       6  46  52    yankees    new york    ny
1     nl     7       5  60  50  nationals  washington    dc
2     nl     7       5  48  48  cardinals   st. louis    mo
3     al     5       7  44  50     redsox      boston    ma
4     nl     8       4  46  33     braves     atlanta    ga
5     nl     4       8  47  55       cubs     chicago    il
6     al     6       4  40  39     tigers     detroit    mi
```

Using merge to filter

```
In [16]: print stats_df
          league  wins  losses  rs  ra
team
yankees       al     6       6  46  52
nationals     nl     7       5  60  50
cardinals     nl     7       5  48  48
redsox        al     5       7  44  50
braves        nl     8       4  46  33
cubs          nl     4       8  47  55
tigers        al     6       4  40  39

In [17]: z = city_df[:3]

In [18]: print z
        team        city state
0    yankees    new york    ny
1  nationals  washington    dc
2  cardinals   st. louis    mo

In [19]: print pd.merge(stats_df, z, left_index=True, right_on='team')
  league  wins  losses  rs  ra       team        city state
0     al     6       6  46  52    yankees    new york    ny
1     nl     7       5  60  50  nationals  washington    dc
2     nl     7       5  48  48  cardinals   st. louis    mo
```

Recall – Read CSV + Hirearchical Index

- Remember that we can read in a CSV file
- And create a hierarchical index

```
In [44]: !type courses2.csv
course,sem,enroll,assign
inls101,f12,12,3
inls161,f12,18,4
inls382,f12,15,4
inls101,f13,17,4
inls161,f13,19,3
inls382,f13,21,5

In [45]: df = pd.read_csv('courses2.csv', index_col=['sem', 'course'])

In [46]: print df
             enroll  assign
sem course
f12 inls101      12       3
    inls161      18       4
    inls382      15       4
f13 inls101      17       4
    inls161      19       3
    inls382      21       5
```

Recall – Summary Stats

- Many summary stats functions have a ***level*** option that can be used with a hierarchical index

```
In [51]: print df
             enroll  assign
sem course
f12 inls101      12       3
    inls161      18       4
    inls382      15       4
f13 inls101      17       4
    inls161      19       3
    inls382      21       5

In [52]: print df.sum(level=0)
     enroll  assign
sem
f12      45      11
f13      57      12

In [53]: print df.sum(level=1)
         enroll  assign
course
inls101      29       7
inls161      37       7
inls382      36       9
```

```
In [54]: print
df['enroll'].sum(level=0)
sem
f12    45
f13    57
dtype: int64

In [55]: print
df.sum(level=0)['enroll']
sem
f12    45
f13    57
Name: enroll, dtype: int64
```

Aggregation – GroupBy

- As an alternative to creating and using a hierarchical index to do aggregation, we can use groupby.
- Group by uses a split-apply-combine process.

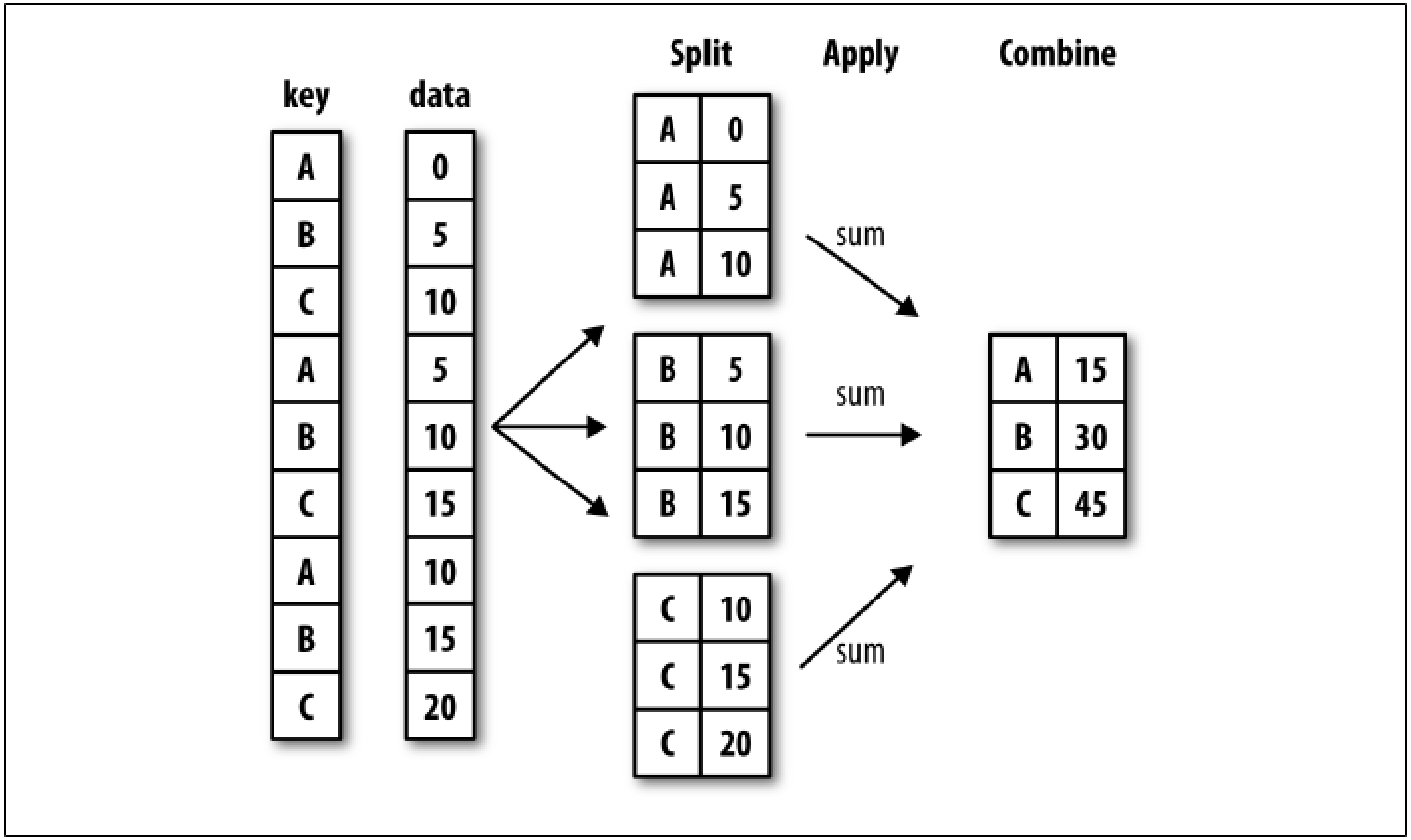


Figure 9-1. Illustration of a group aggregation

Aggregation – GroupBy

- In our previous examples, we created summary statistics by relying on a hierarchical index
- Using GroupBy, we will not need to have a hierarchical index.

Previous way – Hierarchical Index

```
In [62]: !type courses2.csv
course,sem,enroll,assign
inls101,f12,12,3
inls161,f12,18,4
inls382,f12,15,4
inls101,f13,17,4
inls161,f13,19,3
inls382,f13,21,5

In [63]: df =
pd.read_csv('courses2.csv',
index_col=['sem', 'course'])

In [64]: print df
             enroll  assign
sem course
f12 inls101      12       3
    inls161      18       4
    inls382      15       4
f13 inls101      17       4
    inls161      19       3
    inls382      21       5
```

New way – No hierarchical index

```
In [77]: df = pd.read_csv('courses2.csv')

In [78]: print df
    course  sem  enroll  assign
0  inls101  f12      12       3
1  inls161  f12      18       4
2  inls382  f12      15       4
3  inls101  f13      17       4
4  inls161  f13      19       3
5  inls382  f13      21       5
```

GroupBy Objects

- .groupby() on a DataFrame returns a GroupBy object
- GroupBy objects have methods such as .sum() and .mean()

```
In [84]: print df
    course  sem  enroll  assign
0  inls101  f12      12       3
1  inls161  f12      18       4
2  inls382  f12      15       4
3  inls101  f13      17       4
4  inls161  f13      19       3
5  inls382  f13      21       5

In [85]: g = df['enroll'].groupby(df['sem'])

In [86]: print g
<pandas.core.groupby.SeriesGroupBy object at 0x0000000011422860>

In [87]: print g.sum()
sem
f12    45
f13    57
dtype: int64

In [88]: print g.mean()
sem
f12    15
f13    19
dtype: int64

In [95]: print type(g.sum())
<class 'pandas.core.series.Series'>

In [103]: print g.sum().index
Index([u'f12', u'f13'], dtype=object)
```

Creates a GroupBy object grouped by the semester. This object can be used later to do operations (such as sum and mean on the groups.

Also notice that g.sum() returns a Series with semester as its index

GroupBy

- If you groupby multiple columns, when you perform an operation such as .sum(), the result will have a hierarchical index

```
In [96]: print df
    course  sem  enroll  assign
0  inls101  f12      12       3
1  inls161  f12      18       4
2  inls382  f12      15       4
3  inls101  f13      17       4
4  inls161  f13      19       3
5  inls382  f13      21       5

In [97]: g = df['enroll'].groupby([df['sem'], df['course']])

In [98]: print g.sum()
sem  course
f12  inls101    12
     inls161    18
     inls382    15
f13  inls101    17
     inls161    19
     inls382    21
dtype: int64

In [99]: print type(g.sum())
<class 'pandas.core.series.Series'>

In [100]: print g.sum().index
MultiIndex
[(u'f12', u'inls101'), (u'f12', u'inls161'), (u'f12', u'inls382'), (u'f13',
u'inls101'), (u'f13', u'inls161'), (u'f13', u'inls382')]
```

Groupby + Unstack

- After doing a groupby with two columns, you may want to use unstack

```
In [111]: print df
    course  sem  enroll  assign
0  inls101  f12      12       3
1  inls161  f12      18       4
2  inls382  f12      15       4
3  inls101  f13      17       4
4  inls161  f13      19       3
5  inls382  f13      21       5

In [112]: g = df['enroll'].groupby([df['sem'], df['course']])

In [113]: print g.sum()
sem  course
f12  inls101    12
     inls161    18
     inls382    15
f13  inls101    17
     inls161    19
     inls382    21
dtype: int64

In [114]: print g.sum().unstack()
course  inls101  inls161  inls382
sem
f12          12       18       15
f13          17       19       21

In [115]: print g.sum().unstack()['inls382']['f13']
21
```

Groupby shorthand

- If the grouping information is in the same DataFrame as the data being aggregated, you can use a shorthand notation.

```
In [119]: print df
    course  sem  enroll  assign
0  inls101  f12      12       3
1  inls161  f12      18       4
2  inls382  f12      15       4
3  inls101  f13      17       4
4  inls161  f13      19       3
5  inls382  f13      21       5

In [120]: print df.groupby(df['sem']).sum()
     enroll  assign
sem
f12      45      11
f13      57      12

In [121]: print df.groupby('sem').sum()
     enroll  assign
sem
f12      45      11
f13      57      12
```

Watch out!

```
In [130]: print df
    course  sem  enroll  assign
0  inls101  f12      12       3
1  inls161  f12      18       4
2  inls382  f12      15       4
3  inls101  f13      17       4
4  inls161  f13      19       3
5  inls382  f13      21       5

In [131]: print df.groupby('sem').sum()
     enroll  assign
sem
f12      45      11
f13      57      12

In [132]: print df.groupby('sem').sum()['enroll']
sem
f12    45
f13    57
Name: enroll, dtype: int64

In [133]: print df['enroll'].groupby('sem').sum()
```

X

Line 133 does not work!
Why not?

```
In [134]: print df['enroll'].groupby(df['sem']).sum()
sem
f12    45
f13    57
dtype: int64
```

```
In [146]: df.groupby('sem')['enroll'].sum()
Out[146]:
sem
f12    45
f13    57
Name: enroll, dtype: int64
```

Line 146 is another syntax that works

Using the results from groupby

```
In [136]: print df
    course  sem  enroll  assign
0  inls101  f12      12       3
1  inls161  f12      18       4
2  inls382  f12      15       4
3  inls101  f13      17       4
4  inls161  f13      19       3
5  inls382  f13      21       5

In [137]: z = df['enroll'].groupby(df['course']).sum()

In [138]: print z
course
inls101    29
inls161    37
inls382    36
dtype: int64

In [139]: z.sort(ascending=False)

In [140]: print z
course
inls161    37
inls382    36
inls101    29
dtype: int64
```

```
In [141]: type(z)
Out[141]: pandas.core.series.Series

In [143]: z[:2]
Out[143]:
course
inls161 37
inls382 36
dtype: int64


In [144]: for course, enroll in z[:2].iteritems():
     ...: print course, enroll
     ...:
inls161 37
inls382 36
```

Manipulating DF, Series, Groupby

```
In [210]: print df
    course  sem  enroll  assign
0  inls101  f12      12       3
1  inls161  f12      18       4
2  inls382  f12      15       4
3  inls101  f13      17       4
4  inls161  f13      19       3
5  inls382  f13      21       5

In [211]: z =
df.groupby('course')['enroll'].sum().order
(ascending=False)[:2]

In [212]: print z
course
inls161    37
inls382    36
Name: enroll, dtype: int64

In [213]: zdf = DataFrame(z.values,
index=z.index)

In [214]: print zdf
          0
course
inls161  37
inls382  36
```

```
In [215]: zdf.columns = ['enroll']

In [216]: print zdf
         enroll
course
inls161      37
inls382      36


In [218]: y = DataFrame(['Tools',
'InfoSys'], index=['inls161', 'inls382'])

In [219]: print y
               0
inls161    Tools
inls382  InfoSys

In [220]: y.columns = ['coursename']

In [221]: print y
        coursename
inls161      Tools
inls382    InfoSys

In [222]: zdfy = zdf.join(y)

In [223]: print zdfy
         enroll coursename
course
inls161      37      Tools
inls382      36    InfoSys
```

GroupBy Exercise
(not to turn in)

- Create a DataFrame using with the following data:
- Do NOT create a hierarchcal index

- After creating the DF, use groupby to:
  1. Output a summary table of the total plays for each uid for each month (i.e. collapse artists)
  2. Output a summary table of the total plays for each artist for each month (i.e. collapse uids)
  3. Output a summary table of the total plays for each uid for each artists (i.e. collapse months)

		Aug	Sep	Nov
uid123	Bowie	12	15	26
uid123	Gaga	2	0	4
uid123	Spears	1	0	3
uid345	Bowie	3	0	4
uid345	Gaga	24	18	31
uid345	Spears	8	12	5
uid678	Bowie	6	3	0
uid678	Gaga	8	14	27
uid678	Spears	28	21	16