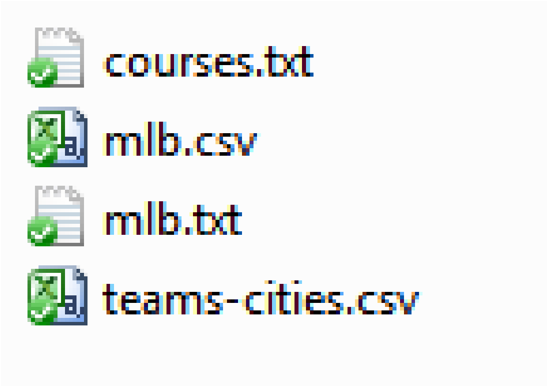
Lect 18 – Data Loading

Rob Capra

INLS 490-172

Data files for today

- Sakai → Resources → Lectures → lect18_data.zip



courses.txt

mlb.csv

mlb.txt

teams-cities.csv

Reading CSV files with pandas

- pd.read_csv()

`!type foo.txt` will show the contents of foo.txt
On a Mac, try: `!cat foo.txt`

```
In [14]: !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 [15]: df = pd.read_csv('mlb.csv')

In [16]: print 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
```

Note: The first row was automatically used for the column labels.

Integers were used for the row index

Reading delimited files

- pd.read_table()

```
In [32]: !type mlb.txt
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 [33]: df = pd.read_table('mlb.txt', sep='|', header=None)

In [34]: print df
           0   1  2  3   4   5
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
```

Indicate the separator (delimiter) character

This file does not have a header line, so indicate that with `header=None`

read_table() will create column labels for us

Adding column labels

- We can add column labels after reading in the DF

```
In [35]: mylabels = ['team', 'league', 'wins', 'losses', 'rs', 'ra']

In [36]: df.columns = mylabels

In [37]: print 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
```

Naming the index

- Recall that you can name the index

```
In [33]: df = pd.read_csv('mlb.csv')

In [34]: print 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 [35]: df.index.name = 'fred'

In [36]: print df
           team league  wins  losses  rs  ra
fred
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
```

Hierarchical Index Naming

- You can also give names to level of a hierarchical index

```
In [49]: df = DataFrame({'a':[1, 2, 3, 4], 'b':[5, 6, 7, 8]},
index=[['r','r','s','s'],['x', 'y', 'x', 'y']])

In [50]: print df
     a  b
r x  1  5
  y  2  6
s x  3  7
  y  4  8

In [51]: df.index.names = ['rors', 'xory']

In [52]: print df
           a  b
rors xory
r    x     1  5
     y     2  6
s    x     3  7
     y     4  8
```

Read and specify a row index

- When reading, we can specify a column to use as the row index

```
In [44]: !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 [45]: df = pd.read_csv('mlb.csv', index_col='team')

In [46]: print 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
```

Read and set a hierarchical index

- Two or more columns can be set a hierarchical index

```
In [47]: !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 [48]: df = pd.read_csv('mlb.csv', index_col=['league', 'team'])

In [49]: print df
                  wins  losses  rs  ra
league team
al     yankees       6       6  46  52
nl     nationals     7       5  60  50
       cardinals     7       5  48  48
al     redsox        5       7  44  50
nl     braves        8       4  46  33
       cubs          4       8  47  55
al     tigers        6       4  40  39
```

Hmm... this looks weird.

Write out a data frame

- .to_csv() will save a data frame to disk

```
In [55]: df = pd.read_table('mlb.txt', sep='|', header=None)

In [56]: df.columns = mylabels

In [57]: print 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 [58]: df.to_csv('mlb2.txt', sep='#')

In [59]: !type mlb2.txt
#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
```

to_csv() options

```
In [62]: print 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 [63]: df.to_csv('mlb3.txt', index=False, header=False)

In [64]: !type mlb3.txt
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
```

What about Series?

- .to_csv() works pretty much as you would expect with a Series
- You can use read_csv to create a Series, but it requires some work
  - No header
  - First column should be set as the index
- There is also a `series.from_csv()` method

Reading CSV Exercise

- For this exercise, use read_csv to read in the file courses.txt

```
In [24]: !type courses.txt
inls101:f12:12:3
inls161:f12:18:4
inls 382:f12:15:4
inls101:f13:17:4
inls382:f13:21:4
```

- Use read_csv and other manipulations to produce a DF with a hierarchical index as shown below.
  - Start by creating the DF with a hierarchical index
  - Then add names/labels to the index and columns

```
In [27]: print df
                   enrollment  assignments
semester course
f12      inls101           12            3
         inls161           18            4
         inls 382          15            4
f13      inls101           17            4
         inls382           21            4
```