Lect 20 – Pivot Tables

Rob Capra

INLS 490-172

PDA Ch.8: tips.csv

- Dataset about tipping on meals:

```
total_bill,tip,sex,smoker,day,time,size
16.99,1.01,Female,No,Sun,Dinner,2
10.34,1.66,Male,No,Sun,Dinner,3
21.01,3.5,Male,No,Sun,Dinner,3
23.68,3.31,Male,No,Sun,Dinner,2
24.59,3.61,Female,No,Sun,Dinner,4
25.29,4.71,Male,No,Sun,Dinner,4
8.77,2.0,Male,No,Sun,Dinner,2
```

- You can download this dataset from:
  - https://raw.githubusercontent.com/pydata/pydata-book/master/ch08/tips.csv

Pivot Tables

- Pivot tables are a data summarization tool
- Common in spreadsheets and data analysis software
- Aggregates data based on one or more keys
- Puts results into a rectangle
  - Some of the grouped keys on the rows, some on cols
- DataFrames have a .pivot_table() method
  - Under the hood, it uses groupby, reshape, and hierarchical indexing

tips.csv data

- First, load the tips.csv data:

```
In [17]: from pandas import Series, DataFrame
    ...: import pandas as pd
    ...: from numpy.random import randn
    ...: import numpy as np
    ...:

In [18]: tips = pd.read_csv('tips.csv')

In [19]: print tips[:10]
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
5       25.29  4.71    Male     No  Sun  Dinner     4
6        8.77  2.00    Male     No  Sun  Dinner     2
7       26.88  3.12    Male     No  Sun  Dinner     4
8       15.04  1.96    Male     No  Sun  Dinner     2
9       14.78  3.23    Male     No  Sun  Dinner     2
```

Add a column for tip percentage

- tip_pct = tip / total_bill

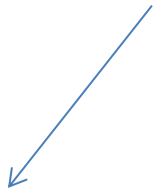
```
In [20]: tips['tip_pct'] = tips['tip'] / tips['total_bill']

In [21]: print tips[:10]
   total_bill   tip     sex smoker  day    time  size   tip_pct
0       16.99  1.01  Female     No  Sun  Dinner     2  0.059447
1       10.34  1.66    Male     No  Sun  Dinner     3  0.160542
2       21.01  3.50    Male     No  Sun  Dinner     3  0.166587
3       23.68  3.31    Male     No  Sun  Dinner     2  0.139780
4       24.59  3.61  Female     No  Sun  Dinner     4  0.146808
5       25.29  4.71    Male     No  Sun  Dinner     4  0.186240
6        8.77  2.00    Male     No  Sun  Dinner     2  0.228050
7       26.88  3.12    Male     No  Sun  Dinner     4  0.116071
8       15.04  1.96    Male     No  Sun  Dinner     2  0.130319
9       14.78  3.23    Male     No  Sun  Dinner     2  0.218539
```

Remember groupby

- We could use groupby to analyze the data

```
In [22]: g = tips.groupby(['sex', 'smoker'])

In [23]: gpct = g['tip_pct']

In [24]: type(gpct)
Out[24]: pandas.core.groupby.SeriesGroupBy

In [25]: print gpct.mean()
sex     smoker
Female  No        0.156921
        Yes       0.182150
Male    No        0.160669
        Yes       0.152771
Name: tip_pct, dtype: float64

In [26]: print tips.groupby(['sex','smoker'], as_index=False).mean()
      sex smoker  total_bill       tip      size   tip_pct
0  Female     No   18.105185  2.773519  2.592593  0.156921
1  Female    Yes   17.977879  2.931515  2.242424  0.182150
2    Male     No   19.791237  3.113402  2.711340  0.160669
3    Male    Yes   22.284500  3.051167  2.500000  0.152771
```

as_index=False prevents an index from being created based on the group key combinations; a simple integer index is created instead



.pivot_table()

- DataFrames have a pivot_table() method
- Default aggregation is group mean (average)
- Can specify groups for rows and/or cols

```
In [28]: print tips[:3]
   total_bill   tip     sex smoker  day    time  size   tip_pct
0       16.99  1.01  Female     No  Sun  Dinner     2  0.059447
1       10.34  1.66    Male     No  Sun  Dinner     3  0.160542
2       21.01  3.50    Male     No  Sun  Dinner     3  0.166587
```

```
In [29]: print tips.pivot_table(rows=['sex','smoker'])

                    size       tip   tip_pct  total_bill
sex    smoker
Female No      2.592593  2.773519  0.156921   18.105185
       Yes     2.242424  2.931515  0.182150   17.977879
Male   No      2.711340  3.113402  0.160669   19.791237
       Yes     2.500000  3.051167  0.152771   22.284500
```

.pivot_table()

- DataFrames have a pivot_table() method
- Default aggregation is group mean (average)
- Can specify groups for rows and/or cols

```
In [28]: print tips[:3]
   total_bill   tip     sex smoker  day    time  size   tip_pct
0       16.99  1.01  Female     No  Sun  Dinner     2  0.059447
1       10.34  1.66    Male     No  Sun  Dinner     3  0.160542
2       21.01  3.50    Male     No  Sun  Dinner     3  0.166587
```

```
In [29]: print tips.pivot_table(rows=['sex','smoker'])
```

```
                size       tip   tip_pct  total_bill
sex    smoker
Female No       2.592593  2.773519  0.156921   18.105185
       Yes      2.242424  2.931515  0.182150   17.977879
Male   No       2.711340  3.113402  0.160669   19.791237
       Yes      2.500000  3.051167  0.152771   22.284500
```

We could have done this just with groupby – next, pivot table magic!

Groups on rows and cols

- Aggregate stats on tip_pct and size
- Group by day & sex (rows), and smoker (cols)

```
In [31]: print tips[:3]
   total_bill   tip     sex smoker  day    time  size   tip_pct
0       16.99  1.01  Female     No  Sun  Dinner     2  0.059447
1       10.34  1.66    Male     No  Sun  Dinner     3  0.160542
2       21.01  3.50    Male     No  Sun  Dinner     3  0.166587

In [32]: print tips.pivot_table(['tip_pct', 'size'], rows=['sex', 'day'],
cols='smoker')
             tip_pct                size
smoker            No       Yes        No       Yes
sex    day
Female Fri   0.165296  0.209129  2.500000  2.000000
       Sat   0.147993  0.163817  2.307692  2.200000
       Sun   0.165710  0.237075  3.071429  2.500000
       Thur  0.155971  0.163073  2.480000  2.428571
Male   Fri   0.138005  0.144730  2.000000  2.125000
       Sat   0.162132  0.139067  2.656250  2.629630
       Sun   0.158291  0.173964  2.883721  2.600000
       Thur  0.165706  0.164417  2.500000  2.300000
```

Margins

- Include partial total using margins=True

```
In [33]: print tips.pivot_table(['tip_pct', 'size'], rows=['sex', 'day'],
cols='smoker', margins=True)
             tip_pct                          size
smoker            No       Yes       All        No       Yes       All
sex    day
Female Fri   0.165296  0.209129  0.199388  2.500000  2.000000  2.111111
       Sat   0.147993  0.163817  0.156470  2.307692  2.200000  2.250000
       Sun   0.165710  0.237075  0.181569  3.071429  2.500000  2.944444
       Thur  0.155971  0.163073  0.157525  2.480000  2.428571  2.468750
Male   Fri   0.138005  0.144730  0.143385  2.000000  2.125000  2.100000
       Sat   0.162132  0.139067  0.151577  2.656250  2.629630  2.644068
       Sun   0.158291  0.173964  0.162344  2.883721  2.600000  2.810345
       Thur  0.165706  0.164417  0.165276  2.500000  2.300000  2.433333
All          0.159328  0.163196  0.160803  2.668874  2.408602  2.569672
```

aggfunc

- Use a different aggregation function

```
In [35]: print tips[:3]
   total_bill   tip     sex smoker  day    time  size   tip_pct
0       16.99  1.01  Female     No  Sun  Dinner     2  0.059447
1       10.34  1.66    Male     No  Sun  Dinner     3  0.160542
2       21.01  3.50    Male     No  Sun  Dinner     3  0.166587

In [36]: print tips.pivot_table('tip_pct', rows=['sex', 'smoker'],
cols='day', aggfunc=len, margins=True)
day             Fri  Sat  Sun  Thur  All
sex    smoker
Female No         2   13   14    25   54
       Yes        7   15    4     7   33
Male   No         2   32   43    20   97
       Yes        8   27   15    10   60
All              19   87   76    62  244
```

Pivot Table Exercise
(not to turn in)

- Create a DataFrame using the tips.csv dataset that looks like the one below:

sex		Female	Male	All
time	smoker			
Dinner	No	0.158347	0.166093	0.163919
	Yes	0.194904	0.156298	0.170480
Lunch	No	0.161637	0.172176	0.166455
	Yes	0.180075	0.159816	0.168346
All		0.173009	0.163311	0.167009