*You are currently looking at **version 1.0** of this notebook. To download notebooks and datafiles, as well as get help on Jupyter notebooks in the Coursera platform, visit the Jupyter Notebook FAQ (https://www.coursera.org/learn/python-data-analysis/resources/0dhYG) course resource.*

Merging Dataframes

In [27]:

```
import pandas as pd

df = pd.DataFrame([{'Name': 'Chris', 'Item Purchased': 'Sponge', 'Cost': 22.50},
                   {'Name': 'Kevyn', 'Item Purchased': 'Kitty Litter', 'Cost': 2.50},
                   {'Name': 'Filip', 'Item Purchased': 'Spoon', 'Cost': 5.00}],
                  index=['Store 1', 'Store 1', 'Store 2'])
df
```

Out[27]:

	Cost	Item Purchased	Name
Store 1	22.5	Sponge	Chris
Store 1	2.5	Kitty Litter	Kevyn
Store 2	5.0	Spoon	Filip

In [3]:

```
df['Date'] = ['December 1', 'January 1', 'mid-May']
df
```

Out[3]:

	Cost	Item Purchased	Name	Date
Store 1	22.5	Sponge	Chris	December 1
Store 1	2.5	Kitty Litter	Kevyn	January 1
Store 2	5.0	Spoon	Filip	mid-May

In [4]:

```
df['Delivered'] = True
df
```

Out[4]:

	Cost	Item Purchased	Name	Date	Delivered
Store 1	22.5	Sponge	Chris	December 1	True
Store 1	2.5	Kitty Litter	Kevyn	January 1	True
Store 2	5.0	Spoon	Filip	mid-May	True

In [5]:

```
df['Feedback'] = ['Positive', None, 'Negative']
df
```

Out[5]:

	Cost	Item Purchased	Name	Date	Delivered	Feedback
Store 1	22.5	Sponge	Chris	December 1	True	Positive
Store 1	2.5	Kitty Litter	Kevyn	January 1	True	None
Store 2	5.0	Spoon	Filip	mid-May	True	Negative

In [6]:

```
adf = df.reset_index()
adf['Date'] = pd.Series({0: 'December 1', 2: 'mid-May'})
adf
```

Out[6]:

	index	Cost	Item Purchased	Name	Date	Delivered	Feedback
0	Store 1	22.5	Sponge	Chris	December 1	True	Positive
1	Store 1	2.5	Kitty Litter	Kevyn	NaN	True	None
2	Store 2	5.0	Spoon	Filip	mid-May	True	Negative

In [7]:

```
staff_df = pd.DataFrame([{'Name': 'Kelly', 'Role': 'Director of HR'},
                         {'Name': 'Sally', 'Role': 'Course liasion'},
                         {'Name': 'James', 'Role': 'Grader'}])
staff_df = staff_df.set_index('Name')
student_df = pd.DataFrame([{'Name': 'James', 'School': 'Business'},
                           {'Name': 'Mike', 'School': 'Law'},
                           {'Name': 'Sally', 'School': 'Engineering'}])
student_df = student_df.set_index('Name')
print(staff_df.head())
print()
print(student_df.head())
```

```
                Role
Name
Kelly  Director of HR
Sally  Course liasion
James          Grader

            School
Name
James     Business
Mike           Law
Sally  Engineering
```

In [8]:

```
pd.merge(staff_df, student_df, how='outer', left_index=True, right_index=True)
```

Out[8]:

	Role	School
Name		
James	Grader	Business
Kelly	Director of HR	NaN
Mike	NaN	Law
Sally	Course liasion	Engineering

In [9]:

```
pd.merge(staff_df, student_df, how='inner', left_index=True, right_index=True)
```

Out[9]:

	Role	School
Name		
James	Grader	Business
Sally	Course liasion	Engineering

In [10]:

```
pd.merge(staff_df, student_df, how='left', left_index=True, right_index=True)
```

Out[10]:

	Role	School
Name		
Kelly	Director of HR	NaN
Sally	Course liasion	Engineering
James	Grader	Business

In [11]:

```
pd.merge(staff_df, student_df, how='right', left_index=True, right_index=True)
```

Out[11]:

	Role	School
Name		
James	Grader	Business
Mike	NaN	Law
Sally	Course liasion	Engineering

In [12]:

```
staff_df = staff_df.reset_index()
student_df = student_df.reset_index()
pd.merge(staff_df, student_df, how='left', left_on='Name', right_on='Name')
```

Out[12]:

	Name	Role	School
0	Kelly	Director of HR	NaN
1	Sally	Course liasion	Engineering
2	James	Grader	Business

In [13]:

```
staff_df = pd.DataFrame([{'Name': 'Kelly', 'Role': 'Director of HR', 'Location': 'State Str
                         {'Name': 'Sally', 'Role': 'Course liasion', 'Location': 'Washingtc
                         {'Name': 'James', 'Role': 'Grader', 'Location': 'Washington Avenue
student_df = pd.DataFrame([{'Name': 'James', 'School': 'Business', 'Location': '1024 Billia
                           {'Name': 'Mike', 'School': 'Law', 'Location': 'Fraternity House
                           {'Name': 'Sally', 'School': 'Engineering', 'Location': '512 Wils
pd.merge(staff_df, student_df, how='left', left_on='Name', right_on='Name')
```

Out[13]:

	Location_x	Name	Role	Location_y	School
0	State Street	Kelly	Director of HR	NaN	NaN
1	Washington Avenue	Sally	Course liasion	512 Wilson Crescent	Engineering
2	Washington Avenue	James	Grader	1024 Billiard Avenue	Business

In [17]:

```
staff_df = pd.DataFrame([{'First Name': 'Kelly', 'Last Name': 'Desjardins', 'Role': 'Direct
                         {'First Name': 'Sally', 'Last Name': 'Brooks', 'Role': 'Course lia
                         {'First Name': 'James', 'Last Name': 'Wilde', 'Role': 'Grader'}])
student_df = pd.DataFrame([{'First Name': 'James', 'Last Name': 'Hammond', 'School': 'Busir
                           {'First Name': 'Mike', 'Last Name': 'Smith', 'School': 'Law'},
                           {'First Name': 'Sally', 'Last Name': 'Brooks', 'School': 'Engine
print(staff_df)
print()
print(student_df)
print()
pd.merge(staff_df, student_df, how='inner', left_on=['First Name','Last Name'], right_on=['
```

```
  First Name   Last Name            Role
0      Kelly  Desjardins  Director of HR
1      Sally      Brooks  Course liasion
2      James       Wilde          Grader

  First Name Last Name       School
0      James   Hammond     Business
1       Mike     Smith          Law
2      Sally    Brooks  Engineering
```

Out[17]:

	First Name	Last Name	Role	School
0	Sally	Brooks	Course liasion	Engineering

Idiomatic Pandas: Making Code Pandorable

In [28]:

```
import pandas as pd
df = pd.read_csv('census.csv')
df
```

Out[28]:

	SUMLEV	REGION	DIVISION	STATE	COUNTY	STNAME	CTYNAME	CENSUS2010POP	ESTI
0	40	3	6	1	0	Alabama	Alabama	4779736	4780
1	50	3	6	1	1	Alabama	Autauga County	54571	5457
2	50	3	6	1	3	Alabama	Baldwin County	182265	1822
3	50	3	6	1	5	Alabama	Barbour County	27457	2745
4	50	3	6	1	7	Alabama	Bibb County	22915	2291
5	50	3	6	1	9	Alabama	Blount	57322	5732

In [9]:

```
(df.where(df['SUMLEV']==50)
    .dropna()
    .set_index(['STNAME','CTYNAME'])
    .rename(columns={'ESTIMATESBASE2010': 'Estimates Base 2010'}))
#print(df.drop(df[df['Quantity'] == 0].index).rename(columns={'Weight': 'Weight (oz.)'}))
```

Out[9]:

		SUMLEV	REGION	DIVISION	STATE	COUNTY	CENSUS2010POP	Estimate Base 2010
STNAME	**CTYNAME**							
Alabama	**Autauga County**	50.0	3.0	6.0	1.0	1.0	54571.0	54571.0
	Baldwin County	50.0	3.0	6.0	1.0	3.0	182265.0	182265.0
	Barbour County	50.0	3.0	6.0	1.0	5.0	27457.0	27457.0
	Bibb County	50.0	3.0	6.0	1.0	7.0	22915.0	22919.0

In [10]:

```
df = df[df['SUMLEV']==50]
df.set_index(['STNAME','CTYNAME'], inplace=True)
df.rename(columns={'ESTIMATESBASE2010': 'Estimates Base 2010'})
```

Out[10]:

		SUMLEV	REGION	DIVISION	STATE	COUNTY	CENSUS2010POP	Estimate Base 2010
STNAME	**CTYNAME**							
Alabama	**Autauga County**	50	3	6	1	1	54571	54571
	Baldwin County	50	3	6	1	3	182265	182265
	Barbour County	50	3	6	1	5	27457	27457
	Bibb County	50	3	6	1	7	22915	22919

In [11]:

```
import numpy as np
def min_max(row):
    data = row[['POPESTIMATE2010',
                'POPESTIMATE2011',
                'POPESTIMATE2012',
                'POPESTIMATE2013',
                'POPESTIMATE2014',
                'POPESTIMATE2015']]
    return pd.Series({'min': np.min(data), 'max': np.max(data)})
```

In [17]:

```
df.apply?
```

In [16]:

```
df.apply(min_max, axis=1)
```

Out[16]:

		max	min
STNAME	**CTYNAME**		
Alabama	**Autauga County**	55347.0	54660.0
	Baldwin County	203709.0	183193.0
	Barbour County	27341.0	26489.0
	Bibb County	22861.0	22512.0
	Blount County	57776.0	57373.0
	Bullock County	10887.0	10606.0
	Butler County	20944.0	20154.0
	Calhoun County	118437.0	115620.0
	Chambers County	34153.0	33993.0
	Cherokee County	26084.0	25859.0
	Chilton County	43943.0	43665.0
	Choctaw County	13841.0	13170.0
	Clarke County	25767.0	24675.0
	Clay County	13880.0	13456.0
	Cleburne County	15072.0	14921.0
	Coffee County	51211.0	50177.0
	Colbert County	54514.0	54354.0
	Conecuh County	13208.0	12662.0
	Coosa County	11758.0	10724.0
	Covington County	38060.0	37796.0
	Crenshaw County	13963.0	13853.0
	Cullman County	82005.0	80374.0
	Dale County	50358.0	49501.0
	Dallas County	43803.0	41131.0
	DeKalb County	71387.0	70869.0
	Elmore County	81468.0	79465.0
	Escambia County	38309.0	37784.0
	Etowah County	104442.0	103057.0
	Fayette County	17231.0	16759.0
	Franklin County	31734.0	31507.0
...	...	...	...

Wisconsin	**Washburn County**	15930.0	15552.0

		max	min
STNAME	**CTYNAME**		
	Washington County	133674.0	131967.0
	Waukesha County	396488.0	390076.0
	Waupaca County	52422.0	51945.0
	Waushara County	24581.0	24033.0
	Winnebago County	169639.0	167059.0
	Wood County	74807.0	73435.0
Wyoming	**Albany County**	37956.0	36428.0
	Big Horn County	12022.0	11672.0
	Campbell County	49220.0	46244.0
	Carbon County	15856.0	15559.0
	Converse County	14343.0	13728.0
	Crook County	7444.0	7114.0
	Fremont County	41129.0	40222.0
	Goshen County	13666.0	13383.0
	Hot Springs County	4846.0	4741.0
	Johnson County	8636.0	8552.0
	Laramie County	97121.0	92271.0
	Lincoln County	18722.0	17943.0
	Natrona County	82178.0	75472.0
	Niobrara County	2548.0	2475.0
	Park County	29237.0	28259.0
	Platte County	8812.0	8678.0
	Sheridan County	30020.0	29146.0
	Sublette County	10418.0	9899.0
	Sweetwater County	45162.0	43593.0
	Teton County	23125.0	21297.0
	Uinta County	21102.0	20822.0
	Washakie County	8545.0	8316.0
	Weston County	7234.0	7065.0

3142 rows × 2 columns

In [18]:

```
import numpy as np
def min_max(row):
    data = row[['POPESTIMATE2010',
                'POPESTIMATE2011',
                'POPESTIMATE2012',
                'POPESTIMATE2013',
                'POPESTIMATE2014',
                'POPESTIMATE2015']]
    row['max'] = np.max(data)
    row['min'] = np.min(data)
    return row
df.apply(min_max, axis=1)
```

Out[18]:

		SUMLEV	REGION	DIVISION	STATE	COUNTY	CENSUS2010POP	ESTIMAT
STNAME	**CTYNAME**							
Alabama	**Autauga County**	50.0	3.0	6.0	1.0	1.0	54571.0	54571.0
	Baldwin County	50.0	3.0	6.0	1.0	3.0	182265.0	182265.0
	Barbour County	50.0	3.0	6.0	1.0	5.0	27457.0	27457.0
	Bibb County	50.0	3.0	6.0	1.0	7.0	22915.0	22919.0
	Blount	50.0	3.0	6.0	1.0	9.0	57322.0	57322.0

In [19]:

```
rows = ['POPESTIMATE2010',
        'POPESTIMATE2011',
        'POPESTIMATE2012',
        'POPESTIMATE2013',
        'POPESTIMATE2014',
        'POPESTIMATE2015']
df.apply(lambda x: np.max(x[rows]), axis=1)
```

Out[19]:

STNAME	CTYNAME	
Alabama	Autauga County	55347.0
	Baldwin County	203709.0
	Barbour County	27341.0
	Bibb County	22861.0
	Blount County	57776.0
	Bullock County	10887.0
	Butler County	20944.0
	Calhoun County	118437.0
	Chambers County	34153.0
	Cherokee County	26084.0
	Chilton County	43943.0
	Choctaw County	13841.0
	Clarke County	25767.0
	Clay County	13880.0
	Cleburne County	15072.0
	Coffee County	51211.0
	Colbert County	54514.0
	Conecuh County	13208.0
	Coosa County	11758.0
	Covington County	38060.0
	Crenshaw County	13963.0
	Cullman County	82005.0
	Dale County	50358.0
	Dallas County	43803.0
	DeKalb County	71387.0
	Elmore County	81468.0
	Escambia County	38309.0
	Etowah County	104442.0
	Fayette County	17231.0
	Franklin County	31734.0
		...
Wisconsin	Washburn County	15930.0
	Washington County	133674.0
	Waukesha County	396488.0
	Waupaca County	52422.0
	Waushara County	24581.0
	Winnebago County	169639.0
	Wood County	74807.0
Wyoming	Albany County	37956.0
	Big Horn County	12022.0
	Campbell County	49220.0
	Carbon County	15856.0
	Converse County	14343.0
	Crook County	7444.0
	Fremont County	41129.0
	Goshen County	13666.0
	Hot Springs County	4846.0
	Johnson County	8636.0
	Laramie County	97121.0

```
Lincoln County          18722.0
Natrona County          82178.0
Niobrara County          2548.0
Park County             29237.0
Platte County            8812.0
Sheridan County         30020.0
Sublette County         10418.0
Sweetwater County       45162.0
Teton County            23125.0
Uinta County            21102.0
Washakie County          8545.0
Weston County            7234.0
dtype: float64
```

Group by

In [29]:

```
import pandas as pd
import numpy as np
df = pd.read_csv('census.csv')
df = df[df['SUMLEV']==50]
df
```

Out[29]:

	SUMLEV	REGION	DIVISION	STATE	COUNTY	STNAME	CTYNAME	CENSUS2010POP	ESTI
1	50	3	6	1	1	Alabama	Autauga County	54571	5457
2	50	3	6	1	3	Alabama	Baldwin County	182265	1822
3	50	3	6	1	5	Alabama	Barbour County	27457	2745
4	50	3	6	1	7	Alabama	Bibb County	22915	2291
5	50	3	6	1	9	Alabama	Blount County	57322	5732

In [21]:

```
%%timeit -n 10
for state in df['STNAME'].unique():
    avg = np.average(df.where(df['STNAME']==state).dropna()['CENSUS2010POP'])
    print('Counties in state ' + state + ' have an average population of ' + str(avg))
```

```
Counties in state Alabama have an average population of 71339.3432836
Counties in state Alaska have an average population of 24490.7241379
Counties in state Arizona have an average population of 426134.466667
Counties in state Arkansas have an average population of 38878.9066667
Counties in state California have an average population of 642309.586207
Counties in state Colorado have an average population of 78581.1875
Counties in state Connecticut have an average population of 446762.125
Counties in state Delaware have an average population of 299311.333333
Counties in state District of Columbia have an average population of 60172
3.0
Counties in state Florida have an average population of 280616.567164
Counties in state Georgia have an average population of 60928.6352201
Counties in state Hawaii have an average population of 272060.2
Counties in state Idaho have an average population of 35626.8636364
Counties in state Illinois have an average population of 125790.509804
Counties in state Indiana have an average population of 70476.1086957
Counties in state Iowa have an average population of 30771.2626263
Counties in state Kansas have an average population of 27172.552381
Counties in state Kentucky have an average population of 36161.3916667
Counties in state Louisiana have an average population of 70833.9375
```

```
Counties in state Georgia have an average population of 60928.6352201

Counties in state Hawaii have an average population of 272060.2
Counties in state Idaho have an average population of 35626.8636364
Counties in state Illinois have an average population of 125790.509804
Counties in state Indiana have an average population of 70476.1086957
Counties in state Iowa have an average population of 30771.2626263
Counties in state Kansas have an average population of 27172.552381
Counties in state Kentucky have an average population of 36161.3916667
Counties in state Louisiana have an average population of 70833.9375
Counties in state Maine have an average population of 83022.5625
Counties in state Maryland have an average population of 240564.666667
Counties in state Massachusetts have an average population of 467687.785714
Counties in state Michigan have an average population of 119080.0
Counties in state Minnesota have an average population of 60964.6551724
Counties in state Mississippi have an average population of 36186.5487805
Counties in state Missouri have an average population of 52077.626087
Counties in state Montana have an average population of 17668.125
Counties in state Nebraska have an average population of 19638.0752688
Counties in state Nevada have an average population of 158855.941176
Counties in state New Hampshire have an average population of 131647.0
Counties in state New Jersey have an average population of 418661.619048
Counties in state New Mexico have an average population of 62399.3636364
Counties in state New York have an average population of 312550.032258
Counties in state North Carolina have an average population of 95354.83
Counties in state North Dakota have an average population of 12690.3962264
Counties in state Ohio have an average population of 131096.636364
Counties in state Oklahoma have an average population of 48718.8441558
Counties in state Oregon have an average population of 106418.722222
Counties in state Pennsylvania have an average population of 189587.746269
Counties in state Rhode Island have an average population of 210513.4
Counties in state South Carolina have an average population of 100551.391304
Counties in state South Dakota have an average population of 12336.0606061
Counties in state Tennessee have an average population of 66801.1052632
Counties in state Texas have an average population of 98998.2716535
Counties in state Utah have an average population of 95306.3793103
Counties in state Vermont have an average population of 44695.7857143
Counties in state Virginia have an average population of 60111.2932331
Counties in state Washington have an average population of 172424.102564
Counties in state West Virginia have an average population of 33690.8
Counties in state Wisconsin have an average population of 78985.9166667
Counties in state Wyoming have an average population of 24505.4782609
10 loops, best of 3: 25.2 ms per loop
```

In [23]:

```
df.head()
```

Out[23]:

	SUMLEV	REGION	DIVISION	STATE	COUNTY	STNAME	CTYNAME	CENSUS2010PO
1	50	3	6	1	1	Alabama	Autauga County	54571
2	50	3	6	1	3	Alabama	Baldwin County	182265
3	50	3	6	1	5	Alabama	Barbour County	27457
4	50	3	6	1	7	Alabama	Bibb County	22915
5	50	3	6	1	9	Alabama	Blount County	57322

5 rows × 100 columns

In [30]:

```
df = df.set_index('STNAME')

def fun(item):
    if item[0]<'M':
        return 0
    if item[0]<'Q':
        return 1
    return 2

for group, frame in df.groupby(fun):
    print('There are ' + str(len(frame)) + ' records in group ' + str(group) + ' for proces
```

```
There are 1177 records in group 0 for processing.
There are 1134 records in group 1 for processing.
There are 831 records in group 2 for processing.
```

In [31]:

```
df = pd.read_csv('census.csv')
df = df[df['SUMLEV']==50]
```

In [32]:

```
df.groupby('STNAME').agg({'CENSUS2010POP': np.average})
```

Out[32]:

	CENSUS2010POP
STNAME	
Alabama	71339.343284
Alaska	24490.724138
Arizona	426134.466667
Arkansas	38878.906667
California	642309.586207
Colorado	78581.187500
Connecticut	446762.125000
Delaware	299311.333333
District of Columbia	601723.000000
Florida	280616.567164
Georgia	60928.635220
Hawaii	272060.200000
Idaho	35626.863636
Illinois	125790.509804
Indiana	70476.108696
Iowa	30771.262626
Kansas	27172.552381
Kentucky	36161.391667
Louisiana	70833.937500
Maine	83022.562500
Maryland	240564.666667
Massachusetts	467687.785714
Michigan	119080.000000
Minnesota	60964.655172
Mississippi	36186.548780
Missouri	52077.626087
Montana	17668.125000
Nebraska	19638.075269
Nevada	158855.941176
New Hampshire	131647.000000
New Jersey	418661.619048

New Mexico	62399.363636

	CENSUS2010POP
STNAME	
New York	312550.032258
North Carolina	95354.830000
North Dakota	12690.396226
Ohio	131096.636364
Oklahoma	48718.844156
Oregon	106418.722222
Pennsylvania	189587.746269
Rhode Island	210513.400000
South Carolina	100551.391304
South Dakota	12336.060606
Tennessee	66801.105263
Texas	98998.271654
Utah	95306.379310
Vermont	44695.785714
Virginia	60111.293233
Washington	172424.102564
West Virginia	33690.800000
Wisconsin	78985.916667
Wyoming	24505.478261

In [33]:

```
print(type(df.groupby(level=0)['POPESTIMATE2010','POPESTIMATE2011']))
print(type(df.groupby(level=0)['POPESTIMATE2010']))
```

```
<class 'pandas.core.groupby.DataFrameGroupBy'>
<class 'pandas.core.groupby.SeriesGroupBy'>
```

In [34]:

```
(df.set_index('STNAME').groupby(level=0)['CENSUS2010POP']
    .agg({'avg': np.average, 'sum': np.sum}))
```

Out[34]:

	avg	sum
STNAME		
Alabama	71339.343284	4779736
Alaska	24490.724138	710231
Arizona	426134.466667	6392017
Arkansas	38878.906667	2915918
California	642309.586207	37253956
Colorado	78581.187500	5029196
Connecticut	446762.125000	3574097
Delaware	299311.333333	897934
District of Columbia	601723.000000	601723
Florida	280616.567164	18801310
Georgia	60928.635220	9687653
Hawaii	272060.200000	1360301
Idaho	35626.863636	1567582
Illinois	125790.509804	12830632
Indiana	70476.108696	6483802
Iowa	30771.262626	3046355
Kansas	27172.552381	2853118
Kentucky	36161.391667	4339367
Louisiana	70833.937500	4533372
Maine	83022.562500	1328361
Maryland	240564.666667	5773552
Massachusetts	467687.785714	6547629
Michigan	119080.000000	9883640
Minnesota	60964.655172	5303925
Mississippi	36186.548780	2967297
Missouri	52077.626087	5988927
Montana	17668.125000	989415
Nebraska	19638.075269	1826341
Nevada	158855.941176	2700551
New Hampshire	131647.000000	1316470

New Jersey	418661.619048	8791894
New Mexico	62399.363636	2059179
	avg	**sum**
STNAME		
New York	312550.032258	19378102
North Carolina	95354.830000	9535483
North Dakota	12690.396226	672591
Ohio	131096.636364	11536504
Oklahoma	48718.844156	3751351
Oregon	106418.722222	3831074
Pennsylvania	189587.746269	12702379
Rhode Island	210513.400000	1052567
South Carolina	100551.391304	4625364
South Dakota	12336.060606	814180
Tennessee	66801.105263	6346105
Texas	98998.271654	25145561
Utah	95306.379310	2763885
Vermont	44695.785714	625741
Virginia	60111.293233	7994802
Washington	172424.102564	6724540
West Virginia	33690.800000	1852994
Wisconsin	78985.916667	5686986
Wyoming	24505.478261	563626

In [35]:

```
(df.set_index('STNAME').groupby(level=0)['POPESTIMATE2010','POPESTIMATE2011']
    .agg({'avg': np.average, 'sum': np.sum}))
```

Out[35]:

	avg		sum	
	POPESTIMATE2010	POPESTIMATE2011	POPESTIMATE2010	POPESTI
STNAME				
Alabama	71420.313433	71658.328358	4785161	4801108
Alaska	24621.413793	24921.379310	714021	722720
Arizona	427213.866667	431248.800000	6408208	6468732
Arkansas	38965.253333	39180.506667	2922394	2938538
California	643691.017241	650000.586207	37334079	3770034
Colorado	78878.968750	79991.875000	5048254	5119480
Connecticut	447464.625000	448719.875000	3579717	3589759
Delaware	299930.333333	302638.666667	899791	907916
District of Columbia	605126.000000	620472.000000	605126	620472
Florida	281341.641791	285157.208955	18849890	19105533
Georgia	61090.905660	61712.452830	9713454	9812280
Hawaii	272796.000000	275645.400000	1363980	1378227
Idaho	35704.227273	36003.045455	1570986	1584134
Illinois	125894.598039	126096.882353	12841249	12861882
Indiana	70549.891304	70835.271739	6490590	6516845
Iowa	30815.090909	30963.525253	3050694	3065389
Kansas	27226.895238	27332.542857	2858824	2869917
Kentucky	36232.808333	36399.016667	4347937	4367882
Louisiana	71014.859375	71490.328125	4544951	4575381
Maine	82980.937500	83016.062500	1327695	1328257
Maryland	241183.708333	243507.125000	5788409	5844171
Massachusetts	468931.142857	472271.214286	6565036	6611797
Michigan	119004.445783	118995.048193	9877369	9876589
Minnesota	61044.862069	61472.632184	5310903	5348119
Mississippi	36223.365854	36317.060976	2970316	2977999
Missouri	52139.582609	52265.973913	5996052	6010587
Montana	17690.053571	17816.892857	990643	997746
Nebraska	19677.688172	19810.569892	1830025	1842383

Nevada	159025.882353	159930.529412	2703440	2718819

	avg		sum	
	POPESTIMATE2010	**POPESTIMATE2011**	**POPESTIMATE2010**	**POPESTI**
STNAME				
New Hampshire	131670.800000	131834.400000	1316708	1318344
New Jersey	419232.428571	421092.095238	8803881	8842934
New Mexico	62567.909091	62976.545455	2064741	2078226
New York	312950.322581	314890.354839	19402920	1952320
North Carolina	95589.790000	96510.250000	9558979	9651025
North Dakota	12726.981132	12930.679245	674530	685326
Ohio	131145.068182	131198.204545	11540766	1154544
Oklahoma	48825.922078	49176.961039	3759596	3786626
Oregon	106610.333333	107458.583333	3837972	3868509
Pennsylvania	189731.552239	190226.895522	12712014	1274520
Rhode Island	210643.800000	210371.200000	1053219	1051856
South Carolina	100780.304348	101581.152174	4635894	4672733
South Dakota	12368.166667	12489.227273	816299	824289
Tennessee	66911.421053	67351.663158	6356585	6398408
Texas	99387.255906	101001.826772	25244363	2565446
Utah	95704.344828	97118.620690	2775426	2816440
Vermont	44713.142857	44763.357143	625984	626687
Virginia	60344.263158	60983.330827	8025787	8110783
Washington	172898.974359	174954.589744	6743060	6823229
West Virginia	33713.181818	33726.327273	1854225	1854948
Wisconsin	79030.611111	79301.666667	5690204	5709720
Wyoming	24544.173913	24685.565217	564516	567768

In [39]:

```
(df.set_index('STNAME').groupby(level = 0)['POPESTIMATE2010','POPESTIMATE2011']
    .agg({'POPESTIMATE2010': np.average, 'POPESTIMATE2011': np.sum}))
```

Out[39]:

	POPESTIMATE2010	POPESTIMATE2011
STNAME		
Alabama	71420.313433	4801108
Alaska	24621.413793	722720
Arizona	427213.866667	6468732
Arkansas	38965.253333	2938538
California	643691.017241	37700034
Colorado	78878.968750	5119480
Connecticut	447464.625000	3589759
Delaware	299930.333333	907916
District of Columbia	605126.000000	620472
Florida	281341.641791	19105533
Georgia	61090.905660	9812280
Hawaii	272796.000000	1378227
Idaho	35704.227273	1584134
Illinois	125894.598039	12861882
Indiana	70549.891304	6516845
Iowa	30815.090909	3065389
Kansas	27226.895238	2869917
Kentucky	36232.808333	4367882
Louisiana	71014.859375	4575381
Maine	82980.937500	1328257
Maryland	241183.708333	5844171
Massachusetts	468931.142857	6611797
Michigan	119004.445783	9876589
Minnesota	61044.862069	5348119
Mississippi	36223.365854	2977999
Missouri	52139.582609	6010587
Montana	17690.053571	997746
Nebraska	19677.688172	1842383
Nevada	159025.882353	2718819
New Hampshire	131670.800000	1318344

New Jersey	419232.428571	8842934
New Mexico	62567.909091	2078226
	POPESTIMATE2010	**POPESTIMATE2011**
STNAME		
New York	312950.322581	19523202
North Carolina	95589.790000	9651025
North Dakota	12726.981132	685326
Ohio	131145.068182	11545442
Oklahoma	48825.922078	3786626
Oregon	106610.333333	3868509
Pennsylvania	189731.552239	12745202
Rhode Island	210643.800000	1051856
South Carolina	100780.304348	4672733
South Dakota	12368.166667	824289
Tennessee	66911.421053	6398408
Texas	99387.255906	25654464
Utah	95704.344828	2816440
Vermont	44713.142857	626687
Virginia	60344.263158	8110783
Washington	172898.974359	6823229
West Virginia	33713.181818	1854948
Wisconsin	79030.611111	5709720
Wyoming	24544.173913	567768

In [37]:

```
df.groupby?
```

Scales

In [40]:

```
df = pd.DataFrame(['A+', 'A', 'A-', 'B+', 'B', 'B-', 'C+', 'C', 'C-', 'D+', 'D'],
                  index=['excellent', 'excellent', 'excellent', 'good', 'good', 'good', 'ok
df.rename(columns={0: 'Grades'}, inplace=True)
df
```

Out[40]:

	Grades
excellent	A+
excellent	A
excellent	A-
good	B+
good	B
good	B-
ok	C+
ok	C
ok	C-
poor	D+
poor	D

In [41]:

```
df['Grades'].astype('category').head()
```

Out[41]:

```
excellent    A+
excellent     A
excellent    A-
good         B+
good          B
Name: Grades, dtype: category
Categories (11, object): [A, A+, A-, B, ..., C+, C-, D, D+]
```

In [42]:

```
grades = df['Grades'].astype('category',
                             categories=['D', 'D+', 'C-', 'C', 'C+', 'B-', 'B', 'B+', 'A-',
                             ordered=True)
grades.head()
```

Out[42]:

```
excellent    A+
excellent     A
excellent    A-
good         B+
good          B
Name: Grades, dtype: category
Categories (11, object): [D < D+ < C- < C ... B+ < A- < A < A+]
```

In [43]:

```
grades > 'C'
```

Out[43]:

```
excellent     True
excellent     True
excellent     True
good          True
good          True
good          True
ok            True
ok           False
ok           False
poor         False
poor         False
Name: Grades, dtype: bool
```

In [44]:

```
df = pd.read_csv('census.csv')
df = df[df['SUMLEV']==50]
df = df.set_index('STNAME').groupby(level=0)['CENSUS2010POP'].agg({'avg': np.average})
pd.cut(df['avg'],10)
```

Out[44]:

```
STNAME
Alabama                 (11706.0871, 75333.413]
Alaska                  (11706.0871, 75333.413]
Arizona                (390320.176, 453317.529]
Arkansas                (11706.0871, 75333.413]
California             (579312.234, 642309.586]
Colorado                (75333.413, 138330.766]
Connecticut            (390320.176, 453317.529]
Delaware               (264325.471, 327322.823]
District of Columbia   (579312.234, 642309.586]
Florida                (264325.471, 327322.823]
Georgia                 (11706.0871, 75333.413]
Hawaii                 (264325.471, 327322.823]
Idaho                   (11706.0871, 75333.413]
Illinois                (75333.413, 138330.766]
Indiana                 (11706.0871, 75333.413]
Iowa                    (11706.0871, 75333.413]
Kansas                  (11706.0871, 75333.413]
Kentucky                (11706.0871, 75333.413]
Louisiana               (11706.0871, 75333.413]
Maine                   (75333.413, 138330.766]
Maryland               (201328.118, 264325.471]
Massachusetts          (453317.529, 516314.881]
Michigan                (75333.413, 138330.766]
Minnesota               (11706.0871, 75333.413]
Mississippi             (11706.0871, 75333.413]
Missouri                (11706.0871, 75333.413]
Montana                 (11706.0871, 75333.413]
Nebraska                (11706.0871, 75333.413]
Nevada                 (138330.766, 201328.118]
New Hampshire           (75333.413, 138330.766]
New Jersey             (390320.176, 453317.529]
New Mexico              (11706.0871, 75333.413]
New York               (264325.471, 327322.823]
North Carolina          (75333.413, 138330.766]
North Dakota            (11706.0871, 75333.413]
Ohio                    (75333.413, 138330.766]
Oklahoma                (11706.0871, 75333.413]
Oregon                  (75333.413, 138330.766]
Pennsylvania           (138330.766, 201328.118]
Rhode Island           (201328.118, 264325.471]
South Carolina          (75333.413, 138330.766]
South Dakota            (11706.0871, 75333.413]
Tennessee               (11706.0871, 75333.413]
Texas                   (75333.413, 138330.766]
Utah                    (75333.413, 138330.766]
Vermont                 (11706.0871, 75333.413]
Virginia                (11706.0871, 75333.413]
Washington             (138330.766, 201328.118]
West Virginia           (11706.0871, 75333.413]
Wisconsin               (75333.413, 138330.766]
Wyoming                 (11706.0871, 75333.413]
Name: avg, dtype: category
```

```
Categories (10, object): [(11706.0871, 75333.413] < (75333.413, 138330.766]
 < (138330.766, 201328.118] < (201328.118, 264325.471] ... (390320.176, 4533

17.529] < (453317.529, 516314.881] < (516314.881, 579312.234] < (579312.234,
642309.586]]
```

In [45]:

```
s = pd.Series([168, 180, 174, 190, 170, 185, 179, 181, 175, 169, 182, 177, 180, 171])
pd.cut(s, 3)
# You can also add labels for the sizes [Small < Medium < Large].
pd.cut(s, 3, labels=['Small', 'Medium', 'Large'])
```

Out[45]:

```
0      Small
1     Medium
2      Small
3      Large
4      Small
5      Large
6     Medium
7     Medium
8      Small
9      Small
10    Medium
11    Medium
12    Medium
13     Small
dtype: category
Categories (3, object): [Small < Medium < Large]
```

Pivot Tables

In [46]:

```
#http://open.canada.ca/data/en/dataset/98f1a129-f628-4ce4-b24d-6f16bf24dd64
df = pd.read_csv('cars.csv')
```

In [47]:

```
df.head()
```

Out[47]:

	YEAR	Make	Model	Size	(kW)	Unnamed: 5	TYPE	CITY (kWh/100 km)	HW (kV km
0	2012	MITSUBISHI	i-MiEV	SUBCOMPACT	49	A1	B	16.9	21.
1	2012	NISSAN	LEAF	MID-SIZE	80	A1	B	19.3	23.
2	2013	FORD	FOCUS ELECTRIC	COMPACT	107	A1	B	19.0	21.
3	2013	MITSUBISHI	i-MiEV	SUBCOMPACT	49	A1	B	16.9	21.
4	2013	NISSAN	LEAF	MID-SIZE	80	A1	B	19.3	23.

In [48]:

```
df.pivot_table(values='(kW)', index='YEAR', columns='Make', aggfunc=np.mean)
```

Out[48]:

Make	BMW	CHEVROLET	FORD	KIA	MITSUBISHI	NISSAN	SMART	TESLA
YEAR								
2012	NaN	NaN	NaN	NaN	49.0	80.0	NaN	NaN
2013	NaN	NaN	107.0	NaN	49.0	80.0	35.0	280.000000
2014	NaN	104.0	107.0	NaN	49.0	80.0	35.0	268.333333
2015	125.0	104.0	107.0	81.0	49.0	80.0	35.0	320.666667
2016	125.0	104.0	107.0	81.0	49.0	80.0	35.0	409.700000

In [52]:

```
df.pivot_table(values='(kW)', index='YEAR', columns='Make', aggfunc=[np.mean,np.min], margi
```

Out[52]:

	mean								
Make	**BMW**	**CHEVROLET**	**FORD**	**KIA**	**MITSUBISHI**	**NISSAN**	**SMART**	**TESLA**	**All**
YEAR									
2012	NaN	NaN	NaN	NaN	49.0	80.0	NaN	NaN	64.5
2013	NaN	NaN	107.0	NaN	49.0	80.0	35.0	280.000000	158
2014	NaN	104.0	107.0	NaN	49.0	80.0	35.0	268.333333	135
2015	125.0	104.0	107.0	81.0	49.0	80.0	35.0	320.666667	181
2016	125.0	104.0	107.0	81.0	49.0	80.0	35.0	409.700000	252
All	125.0	104.0	107.0	81.0	49.0	80.0	35.0	345.478261	190

Date Functionality in Pandas

In [1]:

```
import pandas as pd
import numpy as np
```

Timestamp

In [2]:

```
pd.Timestamp('9/1/2016 10:05AM')
```

Out[2]:

```
Timestamp('2016-09-01 10:05:00')
```

Period

In [3]:

```
pd.Period('1/2016')
```

Out[3]:

```
Period('2016-01', 'M')
```

In [4]:

```
pd.Period('3/5/2016')
```

Out[4]:

```
Period('2016-03-05', 'D')
```

DatetimeIndex

In [5]:

```
t1 = pd.Series(list('abc'), [pd.Timestamp('2016-09-01'), pd.Timestamp('2016-09-02'), pd.Tim
t1
```

Out[5]:

```
2016-09-01    a
2016-09-02    b
2016-09-03    c
dtype: object
```

In [6]:

```
type(t1.index)
```

Out[6]:

```
pandas.tseries.index.DatetimeIndex
```

PeriodIndex

In [7]:

```
t2 = pd.Series(list('def'), [pd.Period('2016-09'), pd.Period('2016-10'), pd.Period('2016-11
t2
```

Out[7]:

```
2016-09    d
2016-10    e
2016-11    f
Freq: M, dtype: object
```

In [8]:

```
type(t2.index)
```

Out[8]:

```
pandas.tseries.period.PeriodIndex
```

Converting to Datetime

In [9]:

```
d1 = ['2 June 2013', 'Aug 29, 2014', '2015-06-26', '7/12/16']
ts3 = pd.DataFrame(np.random.randint(10, 100, (4,2)), index=d1, columns=list('ab'))
ts3
```

Out[9]:

	a	b
2 June 2013	72	44
Aug 29, 2014	43	63
2015-06-26	94	70
7/12/16	66	48

In [10]:

```
ts3.index = pd.to_datetime(ts3.index)
ts3
```

Out[10]:

	a	b
2013-06-02	72	44
2014-08-29	43	63
2015-06-26	94	70
2016-07-12	66	48

In [11]:

```
pd.to_datetime('4.7.12', dayfirst=True)
```

Out[11]:

```
Timestamp('2012-07-04 00:00:00')
```

Timedeltas

In [12]:

```
pd.Timestamp('9/3/2016')-pd.Timestamp('9/1/2016')
```

Out[12]:

```
Timedelta('2 days 00:00:00')
```

In [13]:

```
pd.Timestamp('9/2/2016 8:10AM') + pd.Timedelta('12D 3H')
```

Out[13]:

```
Timestamp('2016-09-14 11:10:00')
```

Working with Dates in a Dataframe

In [21]:

```
dates = pd.date_range('10-01-2016', periods = 9, freq = '2W-SUN')
dates
```

Out[21]:

```
DatetimeIndex(['2016-10-02', '2016-10-16', '2016-10-30', '2016-11-13',
               '2016-11-27', '2016-12-11', '2016-12-25', '2017-01-08',
               '2017-01-22'],
              dtype='datetime64[ns]', freq='2W-SUN')
```

In [26]:

```
df = pd.DataFrame({'Count 1': 100 + np.random.randint(-5, 10, 9).cumsum(),
                   'Count 2': 120 + np.random.randint(-5, 10, 9)}, index=dates)
df
```

Out[26]:

	Count 1	Count 2
2016-10-02	100	129
2016-10-16	106	121
2016-10-30	113	129
2016-11-13	108	129
2016-11-27	107	118
2016-12-11	102	119
2016-12-25	110	116
2017-01-08	118	115
2017-01-22	119	118

In [27]:

```
df.index.weekday_name
```

Out[27]:

```
array(['Sunday', 'Sunday', 'Sunday', 'Sunday', 'Sunday', 'Sunday',
       'Sunday', 'Sunday', 'Sunday'], dtype=object)
```