



MERGING DATAFRAMES WITH PANDAS

# Reading multiple data files



# Tools for pandas data import

- `pd.read_csv()` for CSV files
  - `dataframe = pd.read_csv(filepath)`
  - dozens of optional input parameters
- Other data import tools:
  - `pd.read_excel()`
  - `pd.read_html()`
  - `pd.read_json()`

# Loading separate files

```
In [1]: import pandas as pd
```

```
In [2]: dataframe0 = pd.read_csv('sales-jan-2015.csv')
```

```
In [3]: dataframe1 = pd.read_csv('sales-feb-2015.csv')
```

# Using a loop

```
In [4]: filenames = ['sales-jan-2015.csv', 'sales-feb-2015.csv']
```

```
In [5]: dataframes = []
```

```
In [6]: for f in filenames:  
...:     dataframes.append(pd.read_csv(f))
```



# Using a comprehension

```
In [7]: filenames = ['sales-jan-2015.csv', 'sales-feb-2015.csv']
```

```
In [8]: dataframes = [pd.read_csv(f) for f in filenames]
```



# Using glob

```
In [9]: from glob import glob
```

```
In [10]: filenames = glob('sales*.csv')
```

```
In [11]: dataframes = [pd.read_csv(f) for f in filenames]
```



MERGING DATAFRAMES WITH PANDAS

**Let's practice!**



MERGING DATAFRAMES WITH PANDAS

# Reindexing DataFrames





# “Indexes” vs. “Indices”

- *indices*: many *index labels* within *Index data structures*
- *indexes*: many pandas *Index data structures*

indexes

indices

	NOC	count	Host NOC	Grand Total	fraction	change
Edition						
1896	GRE	52	GRE	151	0.344371	NaN
1900	FRA	185	FRA	512	0.361328	198.002486
1904	USA	394	USA	470	0.838298	199.651245
1908	GBR	347	GBR	804	0.431592	134.489218
1912	SWE	173	SWE	885	0.195480	71.896226
1920	BEL	188	BEL	1298	0.144838	54.757887
1924	FRA	122	FRA	884	0.138009	2.046362
1928	NED	65	NED	710	0.091549	36.315243
1932	USA	181	USA	615	0.294309	3.739184
1936	GER	210	GER	875	0.240000	24.108011
1948	GBR	56	GBR	814	0.068796	-3.635059
1952	FIN	40	FIN	889	0.044994	0.121662

NOC	AFG	AHO	ALG	ANZ	ARG	ARM	...	VEN	VIE	YUG	ZAM	ZIM	ZZX
Edition													
1896	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
1900	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	33.561198
1904	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	-22.642384
1908	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	0.000000
1912	NaN	NaN	NaN	-26.092774	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	0.000000
1920	NaN	NaN	NaN	0.000000	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	0.000000
1924	NaN	NaN	NaN	0.000000	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	0.000000
1928	NaN	NaN	NaN	0.000000	131.101152	NaN	...	NaN	NaN	323.521127	NaN	NaN	0.000000
1932	NaN	NaN	NaN	0.000000	-25.794206	NaN	...	NaN	NaN	0.000000	NaN	NaN	0.000000
1936	NaN	NaN	NaN	0.000000	-10.271982	NaN	...	NaN	NaN	-29.357594	NaN	NaN	0.000000
1948	NaN	NaN	NaN	0.000000	-4.601500	NaN	...	NaN	NaN	47.596769	NaN	NaN	0.000000
1952	NaN	NaN	NaN	0.000000	-10.508545	NaN	...	NaN	NaN	34.043608	NaN	NaN	0.000000



# Importing weather data

```
In [1]: import pandas as pd
```

```
In [2]: w_mean = pd.read_csv('quarterly_mean_temp.csv', index_col='Month')
```

```
In [3]: w_max = pd.read_csv('quarterly_max_temp.csv', index_col='Month')
```



# Examining the data

```
In [4]: print(w_mean)
      Mean TemperatureF
```

Month	
Apr	61.956044
Jan	32.133333
Jul	68.934783
Oct	43.434783

```
In [5]: print(w_max)
      Max TemperatureF
```

Month	
Jan	68
Apr	89
Jul	91
Oct	84



# The DataFrame indexes

```
In [6]: print(w_mean.index)
Index(['Apr', 'Jan', 'Jul', 'Oct'], dtype='object', name='Month')

In [7]: print(w_max.index)
Index(['Jan', 'Apr', 'Jul', 'Oct'], dtype='object', name='Month')

In [8]: print(type(w_mean.index))
<class 'pandas.indexes.base.Index'>
```



# Using `.reindex()`

```
In [9]: ordered = ['Jan', 'Apr', 'Jul', 'Oct']
```

```
In [10]: w_mean2 = w_mean.reindex(ordered)
```

```
In [11]: print(w_mean2)
          Mean TemperatureF
```

Month	
Jan	32.133333
Apr	61.956044
Jul	68.934783
Oct	43.434783



# Using `.sort_index()`

```
In [12]: w_mean2.sort_index()
```

```
Out[12]:
```

	Mean TemperatureF
Month	
Apr	61.956044
Jan	32.133333
Jul	68.934783
Oct	43.434783



# Reindex from a DataFrame Index

```
In [13]: w_mean.reindex(w_max.index)
```

```
Out[13]:
```

	Mean TemperatureF
Month	
Jan	32.133333
Apr	61.956044
Jul	68.934783
Oct	43.434783





# Reindexing with missing labels

```
In [14]: w_mean3 = w_mean.reindex(['Jan', 'Apr', 'Dec'])
```

```
In [15]: print(w_mean3)
          Mean TemperatureF
Month
Jan          32.133333
Apr          61.956044
Dec           NaN
```





# Reindex from a DataFrame Index

```
In [16]: w_max.reindex(w_mean3.index)
```

```
Out[16]:
```

	Max TemperatureF
Month	
Jan	68.0
Apr	89.0
Dec	NaN

```
In [17]: w_max.reindex(w_mean3.index).dropna()
```

```
Out[17]:
```

	Max TemperatureF
Month	
Jan	68.0
Apr	89.0



# Order matters

```
In [18]: w_max.reindex(w_mean.index)
```

```
Out[18]:
```

Max TemperatureF

Month

Apr 89

Jan 68

Jul 91

Oct 84

```
In [19]: w_mean.reindex(w_max.index)
```

```
Out[19]:
```

Mean TemperatureF

Month

Jan 32.133333

Apr 61.956044

Jul 68.934783

Oct 43.434783



MERGING DATAFRAMES WITH PANDAS

**Let's practice!**



MERGING DATAFRAMES WITH PANDAS

# **Arithmetic with Series & DataFrames**



# Loading weather data

```
In [1]: import pandas as pd
```

```
In [2]: weather = pd.read_csv('pittsburgh2013.csv',  
    ....:                      index_col='Date', parse_dates=True)
```

```
In [3]: weather.loc['2013-7-1':'2013-7-7', 'PrecipitationIn']
```

```
Out[3]:
```

Date

2013-07-01	0.18
2013-07-02	0.14
2013-07-03	0.00
2013-07-04	0.25
2013-07-05	0.02
2013-07-06	0.06
2013-07-07	0.10

Name: PrecipitationIn, dtype: float64



# Scalar multiplication

```
In [4]: weather.loc['2013-07-01':'2013-07-07', 'PrecipitationIn'] * 2.54
```

```
Out[4]:
```

```
Date
```

2013-07-01	0.4572
2013-07-02	0.3556
2013-07-03	0.0000
2013-07-04	0.6350
2013-07-05	0.0508
2013-07-06	0.1524
2013-07-07	0.2540

```
Name: PrecipitationIn, dtype: float64
```



# Absolute temperature range

```
In [5]: week1_range = weather.loc['2013-07-01':'2013-07-07',  
    ....:                        ['Min TemperatureF', 'Max TemperatureF']]
```

```
In [6]: print(week1_range)
```

	Min TemperatureF	Max TemperatureF
Date		
2013-07-01	66	79
2013-07-02	66	84
2013-07-03	71	86
2013-07-04	70	86
2013-07-05	69	86
2013-07-06	70	89
2013-07-07	70	77



# Average temperature

```
In [7]: week1_mean = weather.loc['2013-07-01':'2013-07-07',  
    ....:                        'Mean TemperatureF']
```

```
In [8]: print(week1_mean)
```

Date

2013-07-01	72
2013-07-02	74
2013-07-03	78
2013-07-04	77
2013-07-05	76
2013-07-06	78
2013-07-07	72

Name: Mean TemperatureF, dtype: int64





# Relative temperature range

```
In [9]: week1_range / week1_mean
```

```
RuntimeWarning: Cannot compare type 'Timestamp' with type 'str', sort order is  
undefined for incomparable objects
```

```
    return this.join(other, how=how, return_indexers=return_indexers)
```

```
Out[9]:
```

```
                2013-07-01 00:00:00    2013-07-02 00:00:00    2013-07-03 00:00:00    \  
Date  
2013-07-01                NaN                NaN                NaN  
2013-07-02                NaN                NaN                NaN  
2013-07-03                NaN                NaN                NaN  
2013-07-04                NaN                NaN                NaN  
2013-07-05                NaN                NaN                NaN  
2013-07-06                NaN                NaN                NaN  
2013-07-07                NaN                NaN                NaN  
  
                2013-07-04 00:00:00    2013-07-05 00:00:00    2013-07-06 00:00:00    \  
Date  
2013-07-01                NaN                NaN                NaN  
... ..
```



# Relative temperature range

```
In [10]: week1_range.divide(week1_mean, axis='rows')  
Out[10]:
```

Date	Min TemperatureF	Max TemperatureF
2013-07-01	0.916667	1.097222
2013-07-02	0.891892	1.135135
2013-07-03	0.910256	1.102564
2013-07-04	0.909091	1.116883
2013-07-05	0.907895	1.131579
2013-07-06	0.897436	1.141026
2013-07-07	0.972222	1.069444



# Percentage changes

```
In [11]: week1_mean.pct_change() * 100
```

```
Out[11]:
```

```
Date
```

```
2013-07-01      NaN
```

```
2013-07-02      2.777778
```

```
2013-07-03      5.405405
```

```
2013-07-04     -1.282051
```

```
2013-07-05     -1.298701
```

```
2013-07-06      2.631579
```

```
2013-07-07     -7.692308
```

```
Name: Mean TemperatureF, dtype: float64
```



# Bronze Olympic medals

```
In [12]: bronze = pd.read_csv('bronze_top5.csv', index_col=0)
```

```
In [13]: print(bronze)
```

	Total
Country	
United States	1052.0
Soviet Union	584.0
United Kingdom	505.0
France	475.0
Germany	454.0



# Silver Olympic medals

```
In [14]: silver = pd.read_csv('silver_top5.csv', index_col=0)
```

```
In [15]: print(silver)
```

	Total
Country	
United States	1195.0
Soviet Union	627.0
United Kingdom	591.0
France	461.0
Italy	394.0



# Gold Olympic medals

```
In [16]: gold = pd.read_csv('gold_top5.csv', index_col=0)
```

```
In [17]: print(gold)
```

Country	Total
United States	2088.0
Soviet Union	838.0
United Kingdom	498.0
Italy	460.0
Germany	407.0



# Adding bronze, silver

```
In [18]: bronze + silver
Out[18]:
Country
France          936.0
Germany         NaN
Italy           NaN
Soviet Union    1211.0
United Kingdom  1096.0
United States   2247.0
Name: Total, dtype: float64
```



# Adding bronze, silver

```
In [19]: bronze + silver
```

```
Out[19]:
```

```
Country
```

```
France          936.0
```

```
Germany         NaN
```

```
Italy           NaN
```

```
Soviet Union    1211.0
```

```
United Kingdom  1096.0
```

```
United States   2247.0
```

```
Name: Total, dtype: float64
```

```
In [22]: print(bronze['United States'])
```

```
1052.0
```

```
In [23]: print(silver['United States'])
```

```
1195.0
```





# Using the .add() method

```
In [21]: bronze.add(silver)
```

```
Out[21]:
```

```
Country
```

```
France          936.0
```

```
Germany         NaN
```

```
Italy           NaN
```

```
Soviet Union    1211.0
```

```
United Kingdom  1096.0
```

```
United States   2247.0
```

```
Name: Total, dtype: float64
```



# Using a fill\_value

```
In [22]: bronze.add(silver, fill_value=0)
```

```
Out[22]:
```

```
Country
```

```
France          936.0
```

```
Germany         454.0
```

```
Italy           394.0
```

```
Soviet Union    1211.0
```

```
United Kingdom  1096.0
```

```
United States   2247.0
```

```
Name: Total, dtype: float64
```



# Adding bronze, silver, gold

```
In [23]: bronze + silver + gold
```

```
Out[23]:
```

```
Country
```

France	NaN
Germany	NaN
Italy	NaN

Soviet Union	2049.0
--------------	--------

United Kingdom	1594.0
----------------	--------

United States	4335.0
---------------	--------

```
Name: Total, dtype: float64
```



# Chaining .add()

```
In [24]: bronze.add(silver, fill_value=0).add(gold, fill_value=0)
```

```
Out[24]:
```

```
Country
```

```
France          936.0
```

```
Germany         861.0
```

```
Italy           854.0
```

```
Soviet Union    2049.0
```

```
United Kingdom  1594.0
```

```
United States   4335.0
```

```
Name: Total, dtype: float64
```



MERGING DATAFRAMES WITH PANDAS

**Let's practice!**