



Real Python


Lesson #03

Combining Data with Pandas

September 2020



Introduction to Pandas
Exploring Data with Pandas
Data Cleaning Basics
Data Aggregation
Combining Data with Pandas
Transforming Data with Pandas

- 
- Combining Dataframes with the `concat()`
 - Joining Dataframes with the `merge()`



Update from repository

```
git clone https://github.com/ivanovitchm/datascience2020.6
```

Or

```
git pull
```



Dataset

^

745

World Happiness Report

Happiness scored according to economic production, social support, etc.



Sustainable Development Solutions Network • updated 2 years ago (Version 2)

[Data](#)[Kernels \(421\)](#)[Discussion \(6\)](#)[Activity](#)[Download \(29 KB\)](#)[New Kernel](#)

CC0: Public Domain



economics, social sciences, emotion

Data Sources



2015.csv

158 x 12



2016.csv

157 x 13



2017.csv

155 x 12

It's very common in practice to work with more than one data set at a time.

Introduction to data set

Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family
Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951
Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223
Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058
Norway	Western Europe	4	7.522	0.0388	1.459	1.33095
Canada	North America	5	7.427	0.03553	1.32629	1.32261

Df1

	Column 1	Column 2
0	A	B

Df2

	Column 1	Column 2
0	C	D

Axis = 0

Concatenated

	Column 1	Column 2
0	A	B
0	C	D

Since we imported pandas as "pd",
we use this naming convention
to access the concat function.

Pass the dataframes you want to
combine into the function as a list.

`pd.concat([df1, df2])`

Df1

	Column 1
0	A
1	C

Axis = 1

Df2

	Column 2
0	B
1	D

Concatenated

	Column 1	Column 2
0	A	B
1	C	D

	Country	Happiness Score	Year
0	Switzerland	7.587	2015
1	Iceland	7.561	2015
2	Denmark	7.527	2015

	Country	Happiness Score	Year
0	Denmark	7.526	2016
1	Switzerland	7.509	2016
2	Iceland	7.501	2016

```
pd.concat([head_2015, head_2016], axis=0)
```

	Country	Happiness Score	Year
0	Switzerland	7.587	2015
1	Iceland	7.561	2015
2	Denmark	7.527	2015
0	Denmark	7.526	2016
1	Switzerland	7.509	2016
2	Iceland	7.501	2016

	Country	Happiness Score	Year
0	Switzerland	7.587	2015
1	Iceland	7.561	2015
2	Denmark	7.527	2015

	Country	Happiness Score	Year
0	Denmark	7.526	2016
1	Switzerland	7.509	2016
2	Iceland	7.501	2016

```
pd.concat([head_2015, head_2016], axis=1)
```

	Country	Happiness Score	Year	Country	Happiness Score	Year
0	Switzerland	7.587	2015	Denmark	7.526	2016
1	Iceland	7.561	2015	Switzerland	7.509	2016
2	Denmark	7.527	2015	Iceland	7.501	2016

	Year	Country	Happiness Score	Standard Error
0	2015	Switzerland	7.587	0.03411
1	2015	Iceland	7.561	0.04884
2	2015	Denmark	7.527	0.03328
3	2015	Norway	7.522	0.03880

	Country	Happiness Score	Year
0	Denmark	7.526	2016
1	Switzerland	7.509	2016
2	Iceland	7.501	2016

```
pd.concat([head_2015, head_2016], axis=0)
```

	Country	Happiness Score	Standard Error	Year
0	Switzerland	7.587	0.03411	2015
1	Iceland	7.561	0.04884	2015
2	Denmark	7.527	0.03328	2015
3	Norway	7.522	0.03880	2015
0	Denmark	7.526	NaN	2016
1	Switzerland	7.509	NaN	2016
2	Iceland	7.501	NaN	2016

	Year	Country	Happiness Score	Standard Error
0	2015	Switzerland	7.587	0.03411
1	2015	Iceland	7.561	0.04884
2	2015	Denmark	7.527	0.03328
3	2015	Norway	7.522	0.03880

	Country	Happiness Score	Year
0	Denmark	7.526	2016
1	Switzerland	7.509	2016
2	Iceland	7.501	2016

```
pd.concat([head_2015, head_2016], axis=0, ignore_index=True)
```

	Country	Happiness Score	Standard Error	Year
0	Switzerland	7.587	0.03411	2015
1	Iceland	7.561	0.04884	2015
2	Denmark	7.527	0.03328	2015
3	Norway	7.522	0.03880	2015
4	Denmark	7.526	NaN	2016
5	Switzerland	7.509	NaN	2016
6	Iceland	7.501	NaN	2016

	Year	Country	Happiness Score	Standard Error
0	2015	Switzerland	7.587	0.03411
1	2015	Iceland	7.561	0.04884
2	2015	Denmark	7.527	0.03328
3	2015	Norway	7.522	0.03880

	Country	Happiness Score	Year
0	Denmark	7.526	2016
1	Switzerland	7.509	2016
2	Iceland	7.501	2016

```
pd.concat([head_2015,head_2016],axis=1)
```

	Year	Country	Happiness Score	Standard Error	Country	Happiness Score	Year
0	2015	Switzerland	7.587	0.03411	Denmark	7.526	2016.0
1	2015	Iceland	7.561	0.04884	Switzerland	7.509	2016.0
2	2015	Denmark	7.527	0.03328	Iceland	7.501	2016.0
3	2015	Norway	7.522	0.03880	NaN	NaN	NaN

Lesson #03 - Part #01 Combining Data With Pandas.ipynb

Up to section 1.4



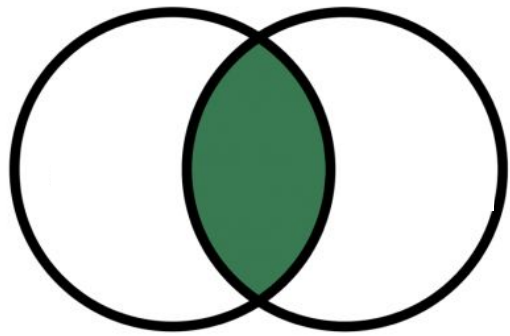
Python Pandas



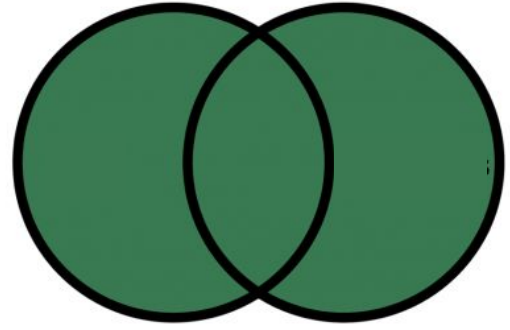
merge

- Merge can execute **high performance** database-style joins
- Only combines dataframes horizontally (**axis=1**)
 - Index
 - Column
- Can only combine **two dataframes at a time**
- However, it can be valuable when we need to combine very **large dataframes** quickly
- It provides more flexibility in terms of how data can be combined (**outer, inner, left, right**)

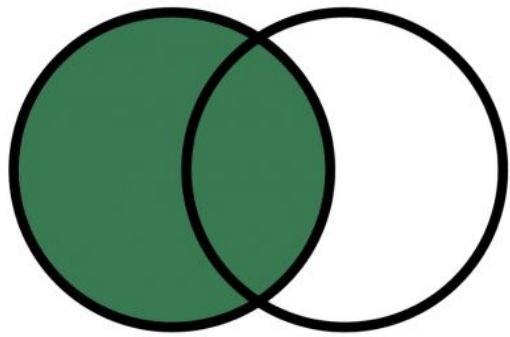
INNER



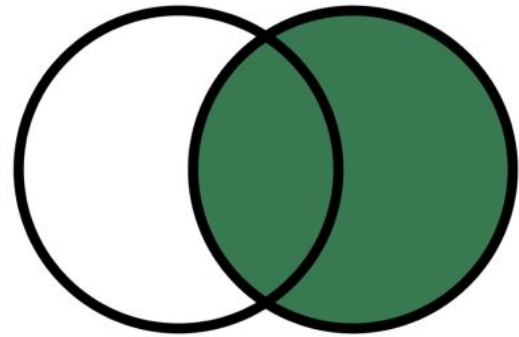
OUTER



LEFT



RIGHT



	Country	Happiness Rank	Year
2	Denmark	3	2015
3	Norway	4	2015
4	Canada	5	2015

	Country	Happiness Rank	Year
2	Iceland	3	2016
3	Norway	4	2016
4	Finland	5	2016

```
pd.merge(left=three_2015, right=three_2016, on="Country")
```

	Country	Happiness Rank_x	Year_x	Happiness Rank_y	Year_y
0	Norway	4	2015	4	2016

	Country	Happiness Rank	Year
2	Denmark	3	2015
3	Norway	4	2015
4	Canada	5	2015

	Country	Happiness Rank	Year
2	Iceland	3	2016
3	Norway	4	2016
4	Finland	5	2016

```
pd.merge(left=three_2015, right=three_2016, on='Country', how="left")
```

	Country	Happiness Rank_x	Year_x	Happiness Rank_y	Year_y
0	Denmark	3	2015	NaN	NaN
1	Norway	4	2015	4.0	2016.0
2	Canada	5	2015	NaN	NaN

	Country	Happiness Rank	Year
2	Denmark	3	2015
3	Norway	4	2015
4	Canada	5	2015

	Country	Happiness Rank	Year
2	Iceland	3	2016
3	Norway	4	2016
4	Finland	5	2016

```
pd.merge(left=three_2015, right=three_2016,
        how='left', on='Country', suffixes=('_2015', '_2016'))
```

	Country	Happiness Rank_2015	Year_2015	Happiness Rank_2016	Year_2016
0	Denmark	3	2015	NaN	NaN
1	Norway	4	2015	4.0	2016.0
2	Canada	5	2015	NaN	NaN

Join on index with merge function

Index



	Country
0	Switzerland
1	Iceland
2	Denmark

Index



	Country	Happiness Rank	Year
2	Denmark	3	2015



2	Iceland	3	2016
---	---------	---	------

3	Norway	4	2015
---	--------	---	------



3	Norway	4	2016
---	--------	---	------

4	Canada	5	2015
---	--------	---	------



4	Finland	5	2016
---	---------	---	------

	Country	Happiness Rank	Year
2	Denmark	3	2015
3	Norway	4	2015
4	Canada	5	2015
5	Finland	6	2015

	Country	Happiness Rank	Year
2	Iceland	3	2016
3	Norway	4	2016
4	Finland	5	2016

```
pd.merge(left = four_2015, right = three_2016,
         left_index = True,
         right_index = True,
         suffixes = ('_2015', '_2016'))
```

	Country_2015	Happiness Rank_2015	Year_2015	Country_2016	Happiness Rank_2016	Year_2016
2	Denmark	3	2015	Iceland	3	2016
3	Norway	4	2015	Norway	4	2016
4	Canada	5	2015	Finland	5	2016

	Country	Happiness Rank	Year
2	Denmark	3	2015
3	Norway	4	2015
4	Canada	5	2015
5	Finland	6	2015

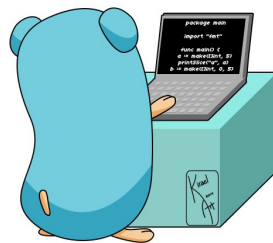
	Country	Happiness Rank	Year
2	Iceland	3	2016
3	Norway	4	2016
4	Finland	5	2016

```
pd.merge(left = four_2015, right = three_2016,
         left_index = True, right_index = True,
         how='left',
         suffixes = ('_2015', '_2016'))
```

	Country_2015	Happiness Rank_2015	Year_2015	Country_2016	Happiness Rank_2016	Year_2016
2	Denmark	3	2015	Iceland	3.0	2016.0
3	Norway	4	2015	Norway	4.0	2016.0
4	Canada	5	2015	Finland	5.0	2016.0
5	Finland	6	2015	NaN	NaN	NaN

	pd.concat()	pd.merge()
Default Join Type	Outer	Inner
Can Combine More Than Two Dataframes at a Time?	Yes	No
Can Combine Dataframes Vertically (axis=0) or Horizontally (axis=1)?	Both	Horizontally
Syntax	Concat (Vertically) concat([df1,df2,df3]) Concat (Horizontally) concat([df1,df2,df3], axis = 1)	Merge (Join on Columns) merge(left = df1, right = df2, how = 'join_type', on = 'Col') Merge (Join on Index) merge(left = df1, right = df2, how = 'join_type', left_index = True, right_index = True)

Did world happiness
increase, decrease, or
stay about the same from
2015 to 2020?



Lesson #03 Part #01 - Combining Data With Pandas.ipynb

