kaggle pandas

Getting started

to start use pandas libaray

In [2]:
```
import pandas as pd
```

DataFrame

A DataFrame is a table. It contains an array of individual entries, each of which has a certain value. Each entry corresponds to a row (or record) and a column.

We are using the pd.DataFrame() constructor to generate these DataFrame objects. The syntax for declaring a new one is a dictionary whose keys are the column names, and whose values are a list of entries. This is the standard way of constructing a new DataFrame, and the one you are most likely to encounter.

In [4]:
```
pd.DataFrame({'Yes': [50, 21], 'No': [131, 2]})
```

Out[4]:

	Yes	No
0	50	131
1	21	2

In [5]:
```
pd.DataFrame({'Bob': ['I liked it.', 'It was awful.'], 'Sue': ['Pretty good.', 'Bland.'
```

Out[5]:

	Bob	Sue
0	I liked it.	Pretty good.
1	It was awful.	Bland.

The list of row labels used in a DataFrame is known as an Index. We can assign values to it by using an index parameter in our constructor:

In [6]:
```
pd.DataFrame({'Bob': ['I liked it.', 'It was awful.'],
              'Sue': ['Pretty good.', 'Bland.']},
             index=['Product A', 'Product B'])
```

Out[6]:

	Bob	Sue
Product A	I liked it.	Pretty good.
Product B	It was awful.	Bland.

Series

is a sequence of data values, is a list. And in fact you can create one with nothing more than a list:

In [7]:
```
pd.Series([1, 2, 3, 4, 5])
```

Out[7]:
```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

Series is, in essence, a single column of a DataFrame. So you can assign row labels to the Series the same way as before, using an index parameter. However, a Series does not have a column name, it only has one overall name:

In [8]:
```
pd.Series([30, 35, 40], index=['2015 Sales', '2016 Sales', '2017 Sales'], name='Product
```

Out[8]:
```
2015 Sales    30
2016 Sales    35
2017 Sales    40
Name: Product A, dtype: int64
```

Reading data files

Data can be stored in any of a number of different forms and formats. By far the most basic of these is the humble CSV file. When you open a CSV file you get something that looks like this:

In [3]:
```
reviews = pd.read_csv('win-data.csv')
```

In [4]:
```
reviews
```

Out[4]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	region
0	0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	96	235.0	California	Napa Valley	Na

	Unnamed: 0	country	description	designation	points	price	province	region_1	region
1	1	Spain	Ripe aromas of fig, blackberry and cassis are ...	Carodorum Selección Especial Reserva	96	110.0	Northern Spain	Toro	Na
2	2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	96	90.0	California	Knights Valley	Sonor
3	3	US	This spent 20 months in 30% new French oak, an...	Reserve	96	65.0	Oregon	Willamette Valley	Willamet Vall
4	4	France	This is the top wine from La Bégude, named aft...	La Brûlade	95	66.0	Provence	Bandol	Na
...	...	...	...	...	...	...	...	...	
150925	150925	Italy	Many people feel Fiano represents southern Ita...	NaN	91	20.0	Southern Italy	Fiano di Avellino	Na
150926	150926	France	Offers an intriguing nose with ginger, lime an...	Cuvée Prestige	91	27.0	Champagne	Champagne	Na
150927	150927	Italy	This classic example comes from a cru vineyard...	Terre di Dora	91	20.0	Southern Italy	Fiano di Avellino	Na
150928	150928	France	A perfect salmon shade, with scents of peaches...	Grand Brut Rosé	90	52.0	Champagne	Champagne	Na
150929	150929	Italy	More Pinot Grigios should taste like this. A r...	NaN	90	15.0	Northeastern Italy	Alto Adige	Na

150930 rows × 11 columns

```
In [22]: data.head()
```

Out[22]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	region_2	vari
0	0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	96	235.0	California	Napa Valley	Napa	Caber Sauvig
1	1	Spain	Ripe aromas of fig, blackberry and cassis are ...	Carodorum Selección Especial Reserva	96	110.0	Northern Spain	Toro	NaN	Tinta T
2	2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	96	90.0	California	Knights Valley	Sonoma	Sauvig Bl
3	3	US	This spent 20 months in 30% new French oak, an...	Reserve	96	65.0	Oregon	Willamette Valley	Willamette Valley	Pinot
4	4	France	This is the top wine from La Bégude, named aft...	La Brûlade	95	66.0	Provence	Bandol	NaN	Prove red bl

Indexing, Selecting & Assigning

Selecting specific values of a pandas DataFrame or Series to work on is an implicit step in almost any data operation.

Hence to access the country property of data we can use:

```
In [23]: reviews.country
```

```
Out[23]: 0             US
         1          Spain
```

```
2             US
3             US
4         France
           ...
150925     Italy
150926    France
150927     Italy
150928    France
150929     Italy
Name: country, Length: 150930, dtype: object
```

for columns in a DataFrame: we can access its values using the indexing ([ ]) operator.

In [24]:
```
reviews['country']
```

Out[24]:
```
0             US
1          Spain
2             US
3             US
4         France
           ...
150925     Italy
150926    France
150927     Italy
150928    France
150929     Italy
Name: country, Length: 150930, dtype: object
```

to drill down to a single specific value, we need only use the indexing operator [ ] once more:

In [26]:
```
reviews['country'][0]
```

Out[26]:
```
'US'
```

Indexing

Index-based selection

Pandas indexing works in one of two paradigms. The first is index-based selection: selecting data based on its numerical position in the data. iloc follows this paradigm.

In [27]:
```
reviews.iloc[0]
```

Out[27]:
```
Unnamed: 0                                                     0
country                                                       US
description    This tremendous 100% varietal wine hails from ...
designation                                    Martha's Vineyard
points                                                        96
price                                                      235.0
province                                              California
region_1                                             Napa Valley
region_2                                                    Napa
variety                                       Cabernet Sauvignon
winery                                                     Heitz
Name: 0, dtype: object
```

In [28]:
```
reviews.iloc[:, 0]
```

Out[28]:
```
0              0
1              1
2              2
3              3
4              4
          ...
150925    150925
150926    150926
150927    150927
150928    150928
150929    150929
Name: Unnamed: 0, Length: 150930, dtype: int64
```

In [29]:
```
reviews.iloc[:3, 0]
```

Out[29]:
```
0    0
1    1
2    2
Name: Unnamed: 0, dtype: int64
```

In [30]:
```
reviews.iloc[1:3, 0]
```

Out[30]:
```
1    1
2    2
Name: Unnamed: 0, dtype: int64
```

In [31]:
```
reviews.iloc[[0, 1, 2], 0]
```

Out[31]:
```
0    0
1    1
2    2
Name: Unnamed: 0, dtype: int64
```

Finally, it's worth knowing that negative numbers can be used in selection. This will start counting forwards from the end of the values. So for example here are the last five elements of the dataset.

In [32]:
```
reviews.iloc[-5:]
```

Out[32]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	region_2
150925	150925	Italy	Many people feel Fiano represents southern Ita...	NaN	91	20.0	Southern Italy	Fiano di Avellino	NaN
150926	150926	France	Offers an intriguing nose with ginger, lime an...	Cuvée Prestige	91	27.0	Champagne	Champagne	NaN

	Unnamed: 0	country	description	designation	points	price	province	region_1	region_2
150927	150927	Italy	This classic example comes from a cru vineyard...	Terre di Dora	91	20.0	Southern Italy	Fiano di Avellino	NaN
150928	150928	France	A perfect salmon shade, with scents of peaches...	Grand Brut Rosé	90	52.0	Champagne	Champagne	NaN
150929	150929	Italy	More Pinot Grigios should taste like this. A r...	NaN	90	15.0	Northeastern Italy	Alto Adige	NaN

Label-based selection

he second paradigm for attribute selection is the one followed by the loc operator: label-based selection. In this paradigm, it's the data index value, not its position, which matters.

In [33]:
```
reviews.loc[0, 'country']
```

Out[33]:
```
'US'
```

loc is conceptually simpler than loc because it ignores the dataset's indices. When we use iloc we treat the dataset like a big matrix (a list of lists), one that we have to index into by position. loc, by contrast, uses the information in the indices to do its work. Since your dataset usually has meaningful indices, it's usually easier to do things using loc instead. For example, here's one operation that's much easier using loc:

In [36]:
```
reviews.loc[:, ['country', 'description', 'points']]
```

Out[36]:

	country	description	points
0	US	This tremendous 100% varietal wine hails from ...	96
1	Spain	Ripe aromas of fig, blackberry and cassis are ...	96
2	US	Mac Watson honors the memory of a wine once ma...	96
3	US	This spent 20 months in 30% new French oak, an...	96
4	France	This is the top wine from La Bégude, named aft...	95
...	...	...	...
150925	Italy	Many people feel Fiano represents southern Ita...	91

	country	description	points
150926	France	Offers an intriguing nose with ginger, lime an...	91
150927	Italy	This classic example comes from a cru vineyard...	91
150928	France	A perfect salmon shade, with scents of peaches...	90
150929	Italy	More Pinot Grigios should taste like this. A r...	90

150930 rows × 3 columns

In [37]:
```
reviews.columns
```

Out[37]:
```
Index(['Unnamed: 0', 'country', 'description', 'designation', 'points',
       'price', 'province', 'region_1', 'region_2', 'variety', 'winery'],
      dtype='object')
```

In [38]:
```
reviews.set_index("province")
```

Out[38]:

	Unnamed: 0	country	description	designation	points	price	region_1	region_2	
province									
California	0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	96	235.0	Napa Valley	Napa	C Sau
Northern Spain	1	Spain	Ripe aromas of fig, blackberry and cassis are ...	Carodorum Selección Especial Reserva	96	110.0	Toro	NaN	T
California	2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	96	90.0	Knights Valley	Sonoma	Sau
Oregon	3	US	This spent 20 months in 30% new French oak, an...	Reserve	96	65.0	Willamette Valley	Willamette Valley	Pir
Provence	4	France	This is the top wine from La Bégude, named aft...	La Brûlade	95	66.0	Bandol	NaN	Pı re
...	...	...	...	...	...	...	...	...	

	Unnamed: 0	country	description	designation	points	price	region_1	region_2	
province									
Southern Italy	150925	Italy	Many people feel Fiano represents southern Ita...	NaN	91	20.0	Fiano di Avellino	NaN	
Champagne	150926	France	Offers an intriguing nose with ginger, lime an...	Cuvée Prestige	91	27.0	Champagne	NaN	Char
Southern Italy	150927	Italy	This classic example comes from a cru vineyard...	Terre di Dora	91	20.0	Fiano di Avellino	NaN	
Champagne	150928	France	A perfect salmon shade, with scents of peaches...	Grand Brut Rosé	90	52.0	Champagne	NaN	Char
Northeastern Italy	150929	Italy	More Pinot Grigios should taste like this. A r...	NaN	90	15.0	Alto Adige	NaN	Pino

150930 rows × 10 columns

Conditional selection

So far we've been indexing various strides of data, using structural properties of the DataFrame itself. To do interesting things with the data, however, we often need to ask questions based on conditions.

In [39]:
```
reviews.country == 'Italy'
```

Out[39]:
```
0         False
1         False
2         False
3         False
4         False
          ...
150925     True
150926    False
150927     True
150928    False
```

```
150929     True
Name: country, Length: 150930, dtype: bool
```

In [40]:
```
reviews.loc[reviews.country == 'Italy']
```

Out[40]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	regic
10	10	Italy	Elegance, complexity and structure come togeth...	Ronco della Chiesa	95	80.0	Northeastern Italy	Collio	I
32	32	Italy	Underbrush, scorched earth, menthol and plum s...	Vigna Piaggia	90	NaN	Tuscany	Brunello di Montalcino	I
35	35	Italy	Forest floor, tilled soil, mature berry and a ...	Riserva	90	135.0	Tuscany	Brunello di Montalcino	I
37	37	Italy	Aromas of forest floor, violet, red berry and ...	NaN	90	29.0	Tuscany	Vino Nobile di Montepulciano	I
38	38	Italy	This has a charming nose that boasts rose, vio...	NaN	90	23.0	Tuscany	Chianti Classico	I
...	...	...	...	...	...	...	...	...	
150920	150920	Italy	Rich and mature aromas of smoke, earth and her...	Brut Riserva	91	19.0	Northeastern Italy	Trento	I
150922	150922	Italy	Made by 30-ish Roberta Borghese high above Man...	Superiore	91	NaN	Northeastern Italy	Colli Orientali del Friuli	I
150925	150925	Italy	Many people feel Fiano represents southern Ita...	NaN	91	20.0	Southern Italy	Fiano di Avellino	I

	Unnamed: 0	country	description	designation	points	price	province	region_1	regic
150927	150927	Italy	This classic example comes from a cru vineyard...	Terre di Dora	91	20.0	Southern Italy	Fiano di Avellino	I
150929	150929	Italy	More Pinot Grigios should taste like this. A r...	NaN	90	15.0	Northeastern Italy	Alto Adige	I

23478 rows × 11 columns

In [41]:

```
reviews.loc[(reviews.country == 'Italy') & (reviews.points >= 90)]
```

Out[41]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	regic
10	10	Italy	Elegance, complexity and structure come togeth...	Ronco della Chiesa	95	80.0	Northeastern Italy	Collio	I
32	32	Italy	Underbrush, scorched earth, menthol and plum s...	Vigna Piaggia	90	NaN	Tuscany	Brunello di Montalcino	I
35	35	Italy	Forest floor, tilled soil, mature berry and a ...	Riserva	90	135.0	Tuscany	Brunello di Montalcino	I
37	37	Italy	Aromas of forest floor, violet, red berry and ...	NaN	90	29.0	Tuscany	Vino Nobile di Montepulciano	I
38	38	Italy	This has a charming nose that boasts rose, vio...	NaN	90	23.0	Tuscany	Chianti Classico	I
...	...	...	...	...	...	...	...	...	

	Unnamed: 0	country	description	designation	points	price	province	region_1	regic
150920	150920	Italy	Rich and mature aromas of smoke, earth and her...	Brut Riserva	91	19.0	Northeastern Italy	Trento	I
150922	150922	Italy	Made by 30-ish Roberta Borghese high above Man...	Superiore	91	NaN	Northeastern Italy	Colli Orientali del Friuli	I
150925	150925	Italy	Many people feel Fiano represents southern Ita...	NaN	91	20.0	Southern Italy	Fiano di Avellino	I
150927	150927	Italy	This classic example comes from a cru vineyard...	Terre di Dora	91	20.0	Southern Italy	Fiano di Avellino	I
150929	150929	Italy	More Pinot Grigios should taste like this. A r...	NaN	90	15.0	Northeastern Italy	Alto Adige	I

7945 rows × 11 columns

```
In [42]: reviews.loc[(reviews.country == 'Italy') | (reviews.points >= 90)]
```

Out[42]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	region
0	0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	96	235.0	California	Napa Valley	Na
1	1	Spain	Ripe aromas of fig, blackberry and cassis are ...	Carodorum Selección Especial Reserva	96	110.0	Northern Spain	Toro	Na

	Unnamed: 0	country	description	designation	points	price	province	region_1	region
2	2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	96	90.0	California	Knights Valley	Sonor
3	3	US	This spent 20 months in 30% new French oak, an...	Reserve	96	65.0	Oregon	Willamette Valley	Willamet Vall
4	4	France	This is the top wine from La Bégude, named aft...	La Brûlade	95	66.0	Provence	Bandol	Na
...	...	...	...	...	...	...	...	...	
150925	150925	Italy	Many people feel Fiano represents southern Ita...	NaN	91	20.0	Southern Italy	Fiano di Avellino	Na
150926	150926	France	Offers an intriguing nose with ginger, lime an...	Cuvée Prestige	91	27.0	Champagne	Champagne	Na
150927	150927	Italy	This classic example comes from a cru vineyard...	Terre di Dora	91	20.0	Southern Italy	Fiano di Avellino	Na
150928	150928	France	A perfect salmon shade, with scents of peaches...	Grand Brut Rosé	90	52.0	Champagne	Champagne	Na
150929	150929	Italy	More Pinot Grigios should taste like this. A r...	NaN	90	15.0	Northeastern Italy	Alto Adige	Na

63743 rows × 11 columns

In [43]:

```
reviews.loc[reviews.price.notnull()]
```

Out[43]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	region
0	0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	96	235.0	California	Napa Valley	Na
1	1	Spain	Ripe aromas of fig, blackberry and cassis are ...	Carodorum Selección Especial Reserva	96	110.0	Northern Spain	Toro	Na
2	2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	96	90.0	California	Knights Valley	Sonor
3	3	US	This spent 20 months in 30% new French oak, an...	Reserve	96	65.0	Oregon	Willamette Valley	Willamet Vall
4	4	France	This is the top wine from La Bégude, named aft...	La Brûlade	95	66.0	Provence	Bandol	Na
...	...	...	...	...	...	...	...	...	
150925	150925	Italy	Many people feel Fiano represents southern Ita...	NaN	91	20.0	Southern Italy	Fiano di Avellino	Na
150926	150926	France	Offers an intriguing nose with ginger, lime an...	Cuvée Prestige	91	27.0	Champagne	Champagne	Na
150927	150927	Italy	This classic example comes from a cru vineyard...	Terre di Dora	91	20.0	Southern Italy	Fiano di Avellino	Na
150928	150928	France	A perfect salmon shade, with scents of peaches...	Grand Brut Rosé	90	52.0	Champagne	Champagne	Na

	Unnamed: 0	country	description	designation	points	price	province	region_1	region
150929	150929	Italy	More Pinot Grigios should taste like this. A r...	NaN	90	15.0	Northeastern Italy	Alto Adige	Na

137235 rows × 11 columns

with more and more Conditions, thw size of dateset will change to select only data that matches the Conditions

---

Assigning data

assigning data to a DataFrame is easy. You can assign either a constant value:

In [44]:
```
reviews['critic'] = 'everyone'
reviews['critic']
```

Out[44]:
```
0         everyone
1         everyone
2         everyone
3         everyone
4         everyone
            ...
150925    everyone
150926    everyone
150927    everyone
150928    everyone
150929    everyone
Name: critic, Length: 150930, dtype: object
```

In [46]:
```
reviews['index_backwards'] = range(len(reviews), 0, -1)
reviews['index_backwards']
```

Out[46]:
```
0         150930
1         150929
2         150928
3         150927
4         150926
           ...
150925         5
150926         4
150927         3
150928         2
150929         1
Name: index_backwards, Length: 150930, dtype: int32
```

Summary Functions and Maps

the data does not always come out of memory in the format we want it in right out of the bat. Sometimes we have to do some more work ourselves to reformat it for the task at hand. This tutorial will cover different operations we can apply to our data to get the input "just right".

Summary

This method generates a high-level summary of the attributes of the given column. It is type-aware, meaning that its output changes based on the data type of the input. The output above only makes sense for numerical data; for string data here's what we get:

In [6]:
```
reviews.describe()
```

Out[6]:

	Unnamed: 0	points	price
count	150930.000000	150930.000000	137235.000000
mean	75464.500000	87.888418	33.131482
std	43569.882402	3.222392	36.322536
min	0.000000	80.000000	4.000000
25%	37732.250000	86.000000	16.000000
50%	75464.500000	88.000000	24.000000
75%	113196.750000	90.000000	40.000000
max	150929.000000	100.000000	2300.000000

In [9]:
```
reviews.points.mean()
```

Out[9]:
```
87.8884184721394
```

To see a list of unique values we can use the unique() function:

In [11]:
```
reviews.country.unique()
```

Out[11]:
```
array(['US', 'Spain', 'France', 'Italy', 'New Zealand', 'Bulgaria',
       'Argentina', 'Australia', 'Portugal', 'Israel', 'South Africa',
       'Greece', 'Chile', 'Morocco', 'Romania', 'Germany', 'Canada',
       'Moldova', 'Hungary', 'Austria', 'Croatia', 'Slovenia', nan,
       'India', 'Turkey', 'Macedonia', 'Lebanon', 'Serbia', 'Uruguay',
       'Switzerland', 'Albania', 'Bosnia and Herzegovina', 'Brazil',
       'Cyprus', 'Lithuania', 'Japan', 'China', 'South Korea', 'Ukraine',
       'England', 'Mexico', 'Georgia', 'Montenegro', 'Luxembourg',
       'Slovakia', 'Czech Republic', 'Egypt', 'Tunisia', 'US-France'],
      dtype=object)
```

To see a list of unique values and how often they occur in the dataset, we can use the value_counts() method:

In [12]:

```
reviews.country.value_counts()
```

```
Out[12]: US                        62397
Italy                     23478
France                    21098
Spain                      8268
Chile                      5816
Argentina                  5631
Portugal                   5322
Australia                  4957
New Zealand                3320
Austria                    3057
Germany                    2452
South Africa               2258
Greece                      884
Israel                      630
Hungary                     231
Canada                      196
Romania                     139
Slovenia                     94
Uruguay                      92
Croatia                      89
Bulgaria                     77
Moldova                      71
Mexico                       63
Turkey                       52
Georgia                      43
Lebanon                      37
Cyprus                       31
Brazil                       25
Macedonia                    16
Serbia                       14
Morocco                      12
England                       9
Luxembourg                    9
Lithuania                     8
India                         8
Czech Republic                6
Ukraine                       5
Switzerland                   4
South Korea                   4
Bosnia and Herzegovina        4
Slovakia                      3
Egypt                         3
China                         3
Albania                       2
Tunisia                       2
Montenegro                    2
Japan                         2
US-France                     1
Name: country, dtype: int64
```

Maps

A map is a term, borrowed from mathematics, for a function that takes one set of values and "maps" them to another set of values. In data science we often have a need for creating new representations from existing data, or for transforming data from the format it is in now to the format that we want it to be in later. Maps are what handle this work, making them extremely important for getting your work done!

The function you pass to map() should expect a single value from the Series (a point value, in the above example), and return a transformed version of that value. map() returns a new Series where all the values have been transformed by your function.

In [13]:
```python
mean = reviews.points.mean()
reviews.points.map(lambda p: p - mean)
```

Out[13]:
```
0         8.111582
1         8.111582
2         8.111582
3         8.111582
4         7.111582
            ...
150925    3.111582
150926    3.111582
150927    3.111582
150928    2.111582
150929    2.111582
Name: points, Length: 150930, dtype: float64
```

apply() is the equivalent method if we want to transform a whole DataFrame by calling a custom method on each row.

In [15]:
```python
def remean_points(row):
    row.points = row.points - mean
    return row

reviews.apply(remean_points, axis='columns')
```

Out[15]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	regi
0	0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	8.111582	235.0	California	Napa Valley	N
1	1	Spain	Ripe aromas of fig, blackberry and cassis are ...	Carodorum Selección Especial Reserva	8.111582	110.0	Northern Spain	Toro	
2	2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	8.111582	90.0	California	Knights Valley	Son

	Unnamed: 0	country	description	designation	points	price	province	region_1	regi
3	3	US	This spent 20 months in 30% new French oak, an...	Reserve	8.111582	65.0	Oregon	Willamette Valley	Willam V
4	4	France	This is the top wine from La Bégude, named aft...	La Brûlade	7.111582	66.0	Provence	Bandol	
...	...	...	...	...	...	...	...	...	
150925	150925	Italy	Many people feel Fiano represents southern Ita...	NaN	3.111582	20.0	Southern Italy	Fiano di Avellino	
150926	150926	France	Offers an intriguing nose with ginger, lime an...	Cuvée Prestige	3.111582	27.0	Champagne	Champagne	
150927	150927	Italy	This classic example comes from a cru vineyard...	Terre di Dora	3.111582	20.0	Southern Italy	Fiano di Avellino	
150928	150928	France	A perfect salmon shade, with scents of peaches...	Grand Brut Rosé	2.111582	52.0	Champagne	Champagne	
150929	150929	Italy	More Pinot Grigios should taste like this. A r...	NaN	2.111582	15.0	Northeastern Italy	Alto Adige	

150930 rows × 11 columns

Note that map( ) and apply( ) return new, transformed Series and DataFrames, respectively. They don't modify the original data they're called on. If we look at the first row of reviews, we can see that it still has its original points value.

In [16]:

```
reviews.head(1)
```

Out[16]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	region_2	variety
0	0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	96	235.0	California	Napa Valley	Napa	Cabernet Sauvignon

Grouping and Sorting

we want to group our data, and then do something specific to the group the data is in.

$groupby()$ operation. We'll also cover some additional topics, such as more complex ways to index your DataFrames, along with how to sort your data.

In [17]:
```
reviews.groupby('points').points.count()
```

Out[17]:
```
points
80        898
81       1502
82       4041
83       6048
84      10708
85      12411
86      15573
87      20747
88      17871
89      12921
90      15973
91      10536
92       9241
93       6017
94       3462
95       1716
96        695
97        365
98        131
99         50
100        24
Name: points, dtype: int64
```

In [18]:
```
reviews.groupby('points').price.min()
```

Out[18]:
```
points
80       5.0
81       5.0
82       5.0
83       4.0
84       4.0
85       4.0
86       4.0
87       6.0
```

```
88      6.0
89      7.0
90      5.0
91      8.0
92     11.0
93     12.0
94     15.0
95     20.0
96     20.0
97     42.0
98     50.0
99     65.0
100    65.0
Name: price, dtype: float64
```

=> ***you can also group by more than one column. For an example, here's how we would pick out the best wine by country and province***

In [19]:

```
reviews.groupby(['country', 'province']).apply(lambda df: df.loc[df.points.idxmax()])
```

Out[19]:

		Unnamed: 0	**country**	**description**	**designation**	**points**	**price**	**province**	**reg**
country	**province**								
Albania	**Mirditë**	4642	Albania	This garnet-colored wine made from 100% Kallme...	NaN	88	20.0	Mirditë	
Argentina	**Mendoza Province**	65331	Argentina	If the color doesn't tell the full story, the ...	Nicasia Vineyard	97	120.0	Mendoza Province	Me
	Other	10619	Argentina	Take note, this could be the best wine Colomé ...	Reserva	95	90.0	Other	
Australia	**Australia Other**	68251	Australia	This big wine presents a sophisticated bouquet...	Yattarna	92	65.0	Australia Other	E Au
	New South Wales	54205	Australia	This wine's deep brassy color suggests honey, ...	Noble One Botrytis	93	32.0	New South Wales	
...	**...**	...	...	...	...	...	...	...	
Uruguay	**Juanico**	3160	Uruguay	This mature Bordeaux-style blend is earthy on ...	Preludio Barrel Select Lote N 77	90	45.0	Juanico	

country	province	Unnamed: 0	country	description	designation	points	price	province	reg
	Montevideo	3164	Uruguay	Bouza ranks as one of Uruguay's top wineries, ...	Monte Vide Eu Tannat-Merlot-Tempranillo	90	57.0	Montevideo	
	Progreso	6541	Uruguay	Blackberry and plum aromas come with wood-smok...	RPF	89	23.0	Progreso	
	San Jose	70157	Uruguay	While this ranks as one of the best Uruguayan ...	El Preciado Premier Gran Reserva	89	60.0	San Jose	
	Uruguay	132482	Uruguay	They call it Special Barrel, and one sniff tel...	Special Barrel	89	50.0	Uruguay	

455 rows × 11 columns

$agg()$***, which lets you run a bunch of different functions on your DataFrame simultaneously. For example, we can generate a simple statistical summary of the dataset as follows:***

In [24]:
```
reviews.groupby(['country']).price.agg([len, min, max]).head(10)
```

Out[24]:

country	len	min	max
Albania	2.0	20.0	20.0
Argentina	5631.0	4.0	250.0
Australia	4957.0	5.0	850.0
Austria	3057.0	8.0	1100.0
Bosnia and Herzegovina	4.0	12.0	13.0
Brazil	25.0	11.0	35.0
Bulgaria	77.0	7.0	28.0
Canada	196.0	12.0	145.0
Chile	5816.0	5.0	400.0
China	3.0	7.0	27.0

Multi-indexes

A multi-index differs from a regular index in that it has multiple levels. For example:

In [9]:
```
countries_reviewed = reviews.groupby(['country', 'province']).description.agg([len])
countries_reviewed
```

Out[9]:

		len
country	**province**	
Albania	**Mirditë**	2
Argentina	**Mendoza Province**	4742
	Other	889
Australia	**Australia Other**	553
	New South Wales	246
...	**...**	...
Uruguay	**Juanico**	19
	Montevideo	3
	Progreso	5
	San Jose	15
	Uruguay	18

455 rows × 1 columns

In [27]:
```
type(reviews.index) ## RangeIndex
```

Out[27]:
```
pandas.core.indexes.range.RangeIndex
```

In [26]:
```
mi = countries_reviewed.index ## MultiIndex
type(mi)
```

Out[26]:
```
pandas.core.indexes.multi.MultiIndex
```

for converting back to a regular index

In [30]:
```
countries_reviewed.reset_index()
```

Out[30]:

	country	province	len
0	Albania	Mirditë	2
1	Argentina	Mendoza Province	4742
2	Argentina	Other	889

	country	province	len
3	Australia	Australia Other	553
4	Australia	New South Wales	246
...	...	...	...
450	Uruguay	Juanico	19
451	Uruguay	Montevideo	3
452	Uruguay	Progreso	5
453	Uruguay	San Jose	15
454	Uruguay	Uruguay	18

455 rows × 3 columns

Sorting

To get data in the order want it in we can sort it ourselves. The sort_values() method is handy for this.

In [10]:
```
countries_reviewed = countries_reviewed.reset_index()
countries_reviewed.sort_values(by='len')
```

Out[10]:

	country	province	len
154	Greece	Central Greece	1
207	Greece	Zitsa	1
115	Cyprus	Pafos	1
362	Slovenia	Slovenska Istra	1
213	Hungary	Pannon	1
...	...	...	...
407	Spain	Northern Spain	4892
122	France	Bordeaux	6111
242	Italy	Tuscany	7281
442	US	Washington	9750
422	US	California	44508

455 rows × 3 columns

To sort by index values, use the companion method sort_index(). This method has the same arguments and default order:

In [33]:
```
countries_reviewed.sort_index()
```

Out[33]:

	country	province	len
0	Albania	Mirditë	2
1	Argentina	Mendoza Province	4742
2	Argentina	Other	889
3	Australia	Australia Other	553
4	Australia	New South Wales	246
...	...	...	...
450	Uruguay	Juanico	19
451	Uruguay	Montevideo	3
452	Uruguay	Progreso	5
453	Uruguay	San Jose	15
454	Uruguay	Uruguay	18

455 rows × 3 columns

know that you can sort by more than one column at a time:

In [34]:
```
countries_reviewed.sort_values(by=['country', 'len'])
```

Out[34]:

	country	province	len
0	Albania	Mirditë	2
2	Argentina	Other	889
1	Argentina	Mendoza Province	4742
5	Australia	Queensland	3
7	Australia	Tasmania	47
...	...	...	...
448	Uruguay	Colonia	6
453	Uruguay	San Jose	15
454	Uruguay	Uruguay	18
447	Uruguay	Canelones	19
450	Uruguay	Juanico	19

455 rows × 3 columns

Dtypes

The data type for a column in a DataFrame or a Series is known as the dtype

Data types tell us something about how pandas is storing the data internally. float64 means that it's using a 64-bit floating point number; int64 means a similarly sized integer instead, and so on.

In [5]:
```
reviews.dtypes
```

Out[5]:
```
Unnamed: 0       int64
country         object
description     object
designation     object
points           int64
price          float64
province        object
region_1        object
region_2        object
variety         object
winery          object
dtype: object
```

In [4]:
```
reviews.price.dtype
```

Out[4]:
```
dtype('float64')
```

One peculiarity to keep in mind (and on display very clearly here) is that columns consisting entirely of strings do not get their own type; they are instead given the object type.

In [7]:
```
reviews.country.dtype
```

Out[7]:
```
dtype('O')
```

Missing data

Entries missing values are given the value NaN, short for "Not a Number". For technical reasons these NaN values are always of the float64 dtype.

In [5]:
```
reviews[pd.isnull(reviews.country)]
```

Out[5]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	region_2	va
1133	1133	NaN	Delicate white flowers and a spin of lemon pee...	Askitikos	90	17.0	NaN	NaN	NaN	Assy
1440	1440	NaN	A blend of 60% Syrah, 30% Cabernet Sauvignon a...	Shah	90	30.0	NaN	NaN	NaN	E

	Unnamed: 0	country	description	designation	points	price	province	region_1	region_2	va
68226	68226	NaN	From first sniff to last, the nose never makes...	Piedra Feliz	81	15.0	NaN	NaN	NaN	
113016	113016	NaN	From first sniff to last, the nose never makes...	Piedra Feliz	81	15.0	NaN	NaN	NaN	
135696	135696	NaN	From first sniff to last, the nose never makes...	Piedra Feliz	81	15.0	NaN	NaN	NaN	

In [6]:

```
reviews[pd.notnull(reviews.country)]
```

Out[6]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	region
0	0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	96	235.0	California	Napa Valley	Na
1	1	Spain	Ripe aromas of fig, blackberry and cassis are ...	Carodorum Selección Especial Reserva	96	110.0	Northern Spain	Toro	Na
2	2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	96	90.0	California	Knights Valley	Sonor
3	3	US	This spent 20 months in 30% new French oak, an...	Reserve	96	65.0	Oregon	Willamette Valley	Willamet Vall
4	4	France	This is the top wine from La Bégude, named aft...	La Brûlade	95	66.0	Provence	Bandol	Na

	Unnamed: 0	country	description	designation	points	price	province	region_1	region
...	...	...	...	...	...	...	...	...	
150925	150925	Italy	Many people feel Fiano represents southern Ita...	NaN	91	20.0	Southern Italy	Fiano di Avellino	Na
150926	150926	France	Offers an intriguing nose with ginger, lime an...	Cuvée Prestige	91	27.0	Champagne	Champagne	Na
150927	150927	Italy	This classic example comes from a cru vineyard...	Terre di Dora	91	20.0	Southern Italy	Fiano di Avellino	Na
150928	150928	France	A perfect salmon shade, with scents of peaches...	Grand Brut Rosé	90	52.0	Champagne	Champagne	Na
150929	150929	Italy	More Pinot Grigios should taste like this. A r...	NaN	90	15.0	Northeastern Italy	Alto Adige	Na

150925 rows × 11 columns

In [9]:

```
reviews.isnull().sum()
```

Out[9]:

```
Unnamed: 0         0
country            5
description        0
designation    45735
points             0
price          13695
province           5
region_1       25060
region_2       89977
variety            0
winery             0
dtype: int64
```

In [10]:

```
reviews.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150930 entries, 0 to 150929
Data columns (total 11 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   Unnamed: 0   150930 non-null  int64  
 1   country      150925 non-null  object 
 2   description  150930 non-null  object 
 3   designation  105195 non-null  object 
 4   points       150930 non-null  int64  
 5   price        137235 non-null  float64
 6   province     150925 non-null  object 
 7   region_1     125870 non-null  object 
 8   region_2     60953 non-null   object 
 9   variety      150930 non-null  object 
 10  winery       150930 non-null  object 
dtypes: float64(1), int64(2), object(8)
memory usage: 12.7+ MB
```

Replacing missing values is a common operation. Pandas provides a really handy method for this problem: fillna(). fillna() provides a few different strategies for mitigating such data. For example, we can simply replace each NaN with an "Unknown":

In [11]:
```
reviews.region_2.fillna("Unknown")
```

Out[11]:
```
0                       Napa
1                    Unknown
2                     Sonoma
3          Willamette Valley
4                    Unknown
                 ...        
150925               Unknown
150926               Unknown
150927               Unknown
150928               Unknown
150929               Unknown
Name: region_2, Length: 150930, dtype: object
```

Alternatively, we may have a non-null value that we would like to replace. For example, suppose that since this dataset was published, reviewer Kerin O'Keefe has changed her country from US to United States. One way to reflect this in the dataset is using the replace() method:

In [15]:
```
reviews.country.replace("US", "United States")
```

Out[15]:
```
0         United States
1                 Spain
2         United States
3         United States
4                France
              ...      
150925            Italy
150926           France
150927            Italy
150928           France
150929            Italy
Name: country, Length: 150930, dtype: object
```

There are many ways to deal with NULL values. But this course was satisfied with these methods above

Renaming and Combining

Oftentimes data will come to us with column names, index names, or other naming conventions that we are not satisfied with. In that case, you'll learn how to use pandas functions to change the names of the offending entries to something better.

You'll also explore how to combine data from multiple DataFrames and/or Series.

Renaming

The first function we'll introduce here is rename(), which lets you change index names and/or column names. For example, to change the points column in our dataset to score, we would do:

In [17]:
```
reviews.columns
```

Out[17]:
```
Index(['Unnamed: 0', 'country', 'description', 'designation', 'points',
       'price', 'province', 'region_1', 'region_2', 'variety', 'winery'],
      dtype='object')
```

In [24]:
```
reviews.drop('Unnamed: 0', inplace=True, axis=1)
```

In [28]:
```
reviews.rename(columns={'points': 'score'})
```

Out[28]:

	country	description	designation	score	price	province	region_1	region_2	varie
0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	96	235.0	California	Napa Valley	Napa	Cabern Sauvign
1	Spain	Ripe aromas of fig, blackberry and cassis are ...	Carodorum Selección Especial Reserva	96	110.0	Northern Spain	Toro	NaN	Tinta (To
2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	96	90.0	California	Knights Valley	Sonoma	Sauvign Bla
3	US	This spent 20 months in 30% new French oak, an...	Reserve	96	65.0	Oregon	Willamette Valley	Willamette Valley	Pinot N

	country	description	designation	score	price	province	region_1	region_2	varie
4	France	This is the top wine from La Bégude, named aft...	La Brûlade	95	66.0	Provence	Bandol	NaN	Proven red bler
...	...	...	...	...	...	...	...	...	
150925	Italy	Many people feel Fiano represents southern Ita...	NaN	91	20.0	Southern Italy	Fiano di Avellino	NaN	Whi Bler
150926	France	Offers an intriguing nose with ginger, lime an...	Cuvée Prestige	91	27.0	Champagne	Champagne	NaN	Champagı Bler
150927	Italy	This classic example comes from a cru vineyard...	Terre di Dora	91	20.0	Southern Italy	Fiano di Avellino	NaN	Whi Bler
150928	France	A perfect salmon shade, with scents of peaches...	Grand Brut Rosé	90	52.0	Champagne	Champagne	NaN	Champagı Bler
150929	Italy	More Pinot Grigios should taste like this. A r...	NaN	90	15.0	Northeastern Italy	Alto Adige	NaN	Pinot Grig

150930 rows × 10 columns

You'll probably rename columns very often, but rename index values very rarely. For that, set_index() is usually more convenient.

Both the row index and the column index can have their own name attribute. The complimentary rename_axis() method may be used to change these names. For example:

In [22]:
```
reviews.rename_axis("wines", axis='rows').rename_axis("fields", axis='columns')
```

Out[22]:

fields	**Unnamed: 0**	**country**	**description**	**designation**	**points**	**price**	**province**	**region_1**	**region**
wines									

fields	Unnamed: 0	country	description	designation	points	price	province	region_1	region
wines									
0	0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	96	235.0	California	Napa Valley	Na
1	1	Spain	Ripe aromas of fig, blackberry and cassis are ...	Carodorum Selección Especial Reserva	96	110.0	Northern Spain	Toro	Na
2	2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	96	90.0	California	Knights Valley	Sonor
3	3	US	This spent 20 months in 30% new French oak, an...	Reserve	96	65.0	Oregon	Willamette Valley	Willamet Vall
4	4	France	This is the top wine from La Bégude, named aft...	La Brûlade	95	66.0	Provence	Bandol	Na
...	...	...	...	...	...	...	...	...	
150925	150925	Italy	Many people feel Fiano represents southern Ita...	NaN	91	20.0	Southern Italy	Fiano di Avellino	Na
150926	150926	France	Offers an intriguing nose with ginger, lime an...	Cuvée Prestige	91	27.0	Champagne	Champagne	Na
150927	150927	Italy	This classic example comes from a cru vineyard...	Terre di Dora	91	20.0	Southern Italy	Fiano di Avellino	Na

fields	Unnamed: 0	country	description	designation	points	price	province	region_1	region
wines									
150928	150928	France	A perfect salmon shade, with scents of peaches...	Grand Brut Rosé	90	52.0	Champagne	Champagne	Na
150929	150929	Italy	More Pinot Grigios should taste like this. A r...	NaN	90	15.0	Northeastern Italy	Alto Adige	Na

150930 rows × 11 columns

In [25]:
```
reviews
```

Out[25]:

	country	description	designation	points	price	province	region_1	region_2	vari
0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	96	235.0	California	Napa Valley	Napa	Caber Sauvigı
1	Spain	Ripe aromas of fig, blackberry and cassis are ...	Carodorum Selección Especial Reserva	96	110.0	Northern Spain	Toro	NaN	Tinta T
2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	96	90.0	California	Knights Valley	Sonoma	Sauvigı Bl
3	US	This spent 20 months in 30% new French oak, an...	Reserve	96	65.0	Oregon	Willamette Valley	Willamette Valley	Pinot N
4	France	This is the top wine from La Bégude, named aft...	La Brûlade	95	66.0	Provence	Bandol	NaN	Prove red bl
...	...	...	...	...	...	...	...	...	

	country	description	designation	points	price	province	region_1	region_2	vari
150925	Italy	Many people feel Fiano represents southern Ita...	NaN	91	20.0	Southern Italy	Fiano di Avellino	NaN	Wl Bl
150926	France	Offers an intriguing nose with ginger, lime an...	Cuvée Prestige	91	27.0	Champagne	Champagne	NaN	Champag Bl
150927	Italy	This classic example comes from a cru vineyard...	Terre di Dora	91	20.0	Southern Italy	Fiano di Avellino	NaN	Wl Bl
150928	France	A perfect salmon shade, with scents of peaches...	Grand Brut Rosé	90	52.0	Champagne	Champagne	NaN	Champag Bl
150929	Italy	More Pinot Grigios should taste like this. A r...	NaN	90	15.0	Northeastern Italy	Alto Adige	NaN	Pinot Gri

150930 rows × 10 columns

Combining

When performing operations on a dataset, we will sometimes need to combine different DataFrames and/or Series in non-trivial ways. Pandas has three core methods for doing this. In order of increasing complexity, these are $concat()$, $join()$, and $merge()$.

The simplest combining method is $concat()$. Given a list of elements, this function will smush those elements together along an axis. This is useful when we have data in different DataFrame or Series objects but having the same fields (columns).

In [37]:

```
df1 = pd.DataFrame({'Yes': [10,20,30,40,50], 'No': [60,70,80,90,100]})
```

In [38]:

```
df2 = pd.DataFrame({'Yes':[60,70,80,90,100], 'No':[10,20,30,40,50]})
```

In [39]:

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

Out[39]:

	Yes	No
0	10	60
1	20	70
2	30	80
3	40	90
4	50	100
0	60	10
1	70	20
2	80	30
3	90	40
4	100	50

In [45]:
```
df1['maybe'] = [15,25,35,45,55]
df1['sure'] = [100,100,100,100,100]
```

In [46]:
```
df1
```

Out[46]:

	Yes	No	maybe	sure
0	10	60	15	100
1	20	70	25	100
2	30	80	35	100
3	40	90	45	100
4	50	100	55	100

In [47]:
```
df2['maybe'] = [65,75,85,95,105]
df2['sure'] = [100,100,100,100,100]
```

In [48]:
```
df2
```

Out[48]:

	Yes	No	maybe	sure
0	60	10	65	100
1	70	20	75	100
2	80	30	85	100
3	90	40	95	100
4	100	50	105	100

The middlemost combiner in terms of complexity is join(). join() lets you combine different DataFrame objects which have an index in common.

In [56]:

```
left = df1.set_index(['Yes'])
right = df2.set_index(['Yes'])

left.join(left, lsuffix='df1', rsuffix='df2')
```

Out[56]:

	Nodf1	maybedf1	suredf1	Nodf2	maybedf2	suredf2
Yes						
10	60	15	100	60	15	100
20	70	25	100	70	25	100
30	80	35	100	80	35	100
40	90	45	100	90	45	100
50	100	55	100	100	55	100

Congratulations!