

# **Pandas**

**Data analysis with Pandas**

Kunal Khurana

2023-12-29

# Table of contents

Creating, Reading and Writing . . . . .	2
DataFrame and Series . . . . .	2
Writing data files . . . . .	4
Reading data files . . . . .	4
Indexing, Selecting and Assigning . . . . .	4
Indexing . . . . .	6
Manipulating the index . . . . .	8
Assigning data . . . . .	10
Renaming and Combining . . . . .	11
Summary Functions and Maps . . . . .	14
Maps . . . . .	16
Grouping and Sorting . . . . .	19
Multi-indexes . . . . .	22
Sorting . . . . .	24
Data Types and Missing Data . . . . .	25

## Creating, Reading and Writing

### DataFrame and Series

```
import pandas as pd
```

2 core objects- - DataFrame - array of individual entries (contains row and column)

keys = 'column names', values = list of entries

rows = **Index**

- Series- sequence of data values

don't have any column name

**row names** defined by **index** parameter aswell

```
#DataFrame_integer
pd.DataFrame({'Yes' : [390, 233], 'No' : [1,23]})
```

	Yes	No
0	390	1
1	233	23

```
# DataFrame_Strings
pd.DataFrame({'Suzaine': ['I liked chocolate', 'Lets have some fun'],
              'Marie': ['butterscotch worked fine', 'wow, its raining']},
              index = ['topic_1', 'topic_2'])
```

	Suzaine	Marie
topic_1	I liked chocolate	butterscotch worked fine
topic_2	Lets have some fun	wow, its raining

```
# series
pd.Series([1, 2, 3],
          index= ['2014_sales', '2015_sales', '2016_sales'],
          name = 'Product A')
```

```
2014_sales    1
2015_sales    2
2016_sales    3
Name: Product A, dtype: int64
```

```
# example
Dinner = pd.Series(['4 cups', '1 cup', '2 large', '1 can'],
                   index = ['Flour', 'Milk', 'Eggs', 'Spam'],
                   name = 'Dinner')
print(Dinner)
```

```
Flour    4 cups
Milk     1 cup
Eggs     2 large
Spam     1 can
Name: Dinner, dtype: object
```

## Writing data files

```
Dinner.to_csv("Dinner.csv")
```

## Reading data files

```
reactions = pd.read_csv('Reactions.csv')
print(reactions.shape)
```

(25553, 5)

```
print(reactions.head())
```

	Unnamed: 0	Content ID \
0	0	97522e57-d9ab-4bd6-97bf-c24d952602d2
1	1	97522e57-d9ab-4bd6-97bf-c24d952602d2
2	2	97522e57-d9ab-4bd6-97bf-c24d952602d2
3	3	97522e57-d9ab-4bd6-97bf-c24d952602d2
4	4	97522e57-d9ab-4bd6-97bf-c24d952602d2

	User ID	Type	Datetime
0	NaN	NaN	2021-04-22 15:17:15
1	5d454588-283d-459d-915d-c48a2cb4c27f	disgust	2020-11-07 09:43:50
2	92b87fa5-f271-43e0-af66-84fac21052e6	dislike	2021-06-17 12:22:51
3	163daa38-8b77-48c9-9af6-37a6c1447ac2	scared	2021-04-18 05:13:58
4	34e8add9-0206-47fd-a501-037b994650a2	disgust	2021-01-06 19:13:01

## Indexing, Selecting and Assigning

```
data = pd.read_csv("winemag-data-130k-v2.csv")
pd.set_option('display.max_rows', 5)
print(data.head())
```

	Unnamed: 0	country	description \
0	0	Italy	Aromas include tropical fruit, broom, brimston...

```

1      1  Portugal  This is ripe and fruity, a wine that is smooth...
2      2      US   Tart and snappy, the flavors of lime flesh and...
3      3      US   Pineapple rind, lemon pith and orange blossom ...
4      4      US   Much like the regular bottling from 2012, this...

```

```

                                designation  points  price      province \
0                                Vulkà Bianco      87    NaN  Sicily & Sardinia
1                                Avidagos          87   15.0        Douro
2                                NaN              87   14.0        Oregon
3                                Reserve Late Harvest      87   13.0      Michigan
4  Vintner's Reserve Wild Child Block      87   65.0        Oregon

```

```

                                region_1      region_2      taster_name \
0                                Etna              NaN      Kerin O'Keefe
1                                NaN              NaN      Roger Voss
2  Willamette Valley  Willamette Valley      Paul Gregutt
3  Lake Michigan Shore              NaN  Alexander Peartree
4  Willamette Valley  Willamette Valley      Paul Gregutt

```

```

taster_twitter_handle      title \
0  @kerinokeefe      Nicosia 2013 Vulkà Bianco (Etna)
1  @vossroger      Quinta dos Avidagos 2011 Avidagos Red (Douro)
2  @paulgwine      Rainstorm 2013 Pinot Gris (Willamette Valley)
3      NaN  St. Julian 2013 Reserve Late Harvest Riesling ...
4  @paulgwine      Sweet Cheeks 2012 Vintner's Reserve Wild Child...

```

```

                                variety      winery
0  White Blend      Nicosia
1  Portuguese Red  Quinta dos Avidagos
2  Pinot Gris      Rainstorm
3  Riesling      St. Julian
4  Pinot Noir      Sweet Cheeks

```

```
print(data.columns)
```

```

Index(['Unnamed: 0', 'country', 'description', 'designation', 'points',
      'price', 'province', 'region_1', 'region_2', 'taster_name',
      'taster_twitter_handle', 'title', 'variety', 'winery'],
      dtype='object')

```

```
print(data.country)
```

```
0          Italy
1          Portugal
...
129969      France
129970      France
Name: country, Length: 129971, dtype: object
```

```
print(data['country']) #handles reserved characters
```

```
0          Italy
1          Portugal
...
129969      France
129970      France
Name: country, Length: 129971, dtype: object
```

```
print(data['country'][4])
```

US

## Indexing

index based or numerical position based (.iloc operator used)

- python's std. library approach (0:10 selects 0, 1, ...9)

label based or value based (.loc operator used)

-indexes inclusively. So 0:10 will select entries 0,...,10

```
# selecting first row
data.iloc[0]
```

```

Unnamed: 0          0
country             Italy
...
variety             White Blend
winery              Nicosia
Name: 0, Length: 14, dtype: object

```

```
data.iloc[:3, 1]
```

```

0      Italy
1  Portugal
2        US
Name: country, dtype: object

```

```
data.iloc[-5:] #selecting last 5 rows, plus all columns
```

	Unnamed: 0	country	description	designation
129966	129966	Germany	Notes of honeysuckle and cantaloupe sweeten th...	Brauneberger Juffer-S
129967	129967	US	Citation is given as much as a decade of bottl...	NaN
129968	129968	France	Well-drained gravel soil gives this wine its c...	Kritt
129969	129969	France	A dry style of Pinot Gris, this is crisp with ...	NaN
129970	129970	France	Big, rich and off-dry, this is powered by inte...	Lieu-dit Harth Cuvée

```
data.loc[:, ['taster_name', 'variety', 'winery']]
```

	taster_name	variety	winery
0	Kerin O'Keefe	White Blend	Nicosia
1	Roger Voss	Portuguese Red	Quinta dos Avidagos
...	...	...	...
129969	Roger Voss	Pinot Gris	Domaine Marcel Deiss
129970	Roger Voss	Gewürztraminer	Domaine Schoffit

## Manipulating the index

```
data.set_index('title') #now first column is title
```

	Unnamed: 0	country	des
title			
Nicosia 2013 Vulkà Bianco (Etna)	0	Italy	Ar
Quinta dos Avidagos 2011 Avidagos Red (Douro)	1	Portugal	Th
...	...	...	...
Domaine Marcel Deiss 2012 Pinot Gris (Alsace)	129969	France	A
Domaine Schoffit 2012 Lieu-dit Harth Cuvée Caroline Gewurztraminer (Alsace)	129970	France	Bi

```
# conditional selection
# selects data with US in columns names for countries
data.loc[data.country == 'US']
```

	Unnamed: 0	country	description	designation
2	2	US	Tart and snappy, the flavors of lime flesh and...	NaN
3	3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest
...	...	...	...	...
129952	129952	US	This Zinfandel from the eastern section of Nap...	NaN
129967	129967	US	Citation is given as much as a decade of bottl...	NaN

```
# selecting particular rows
indices = [1, 2, 3, 5, 8]
sample_rows = data.loc[indices]
print(sample_rows)
```

	Unnamed: 0	country	description	\			
1	1	Portugal	This is ripe and fruity, a wine that is smooth...				
2	2	US	Tart and snappy, the flavors of lime flesh and...				
3	3	US	Pineapple rind, lemon pith and orange blossom ...				
5	5	Spain	Blackberry and raspberry aromas show a typical...				
8	8	Germany	Savory dried thyme notes accent sunnier flavor...				
		designation	points	price	province	region_1	\
1		Avidagos	50	15.0	Douro	NaN	



```

2           NaN      50   14.0           Oregon   Willamette Valley
3 Reserve Late Harvest      50   13.0           Michigan Lake Michigan Shore
5           Ars In Vitro      50   15.0 Northern Spain           Navarra
8           Shine      50   12.0           Rheinhessen           NaN

```

```

           region_2      taster_name taster_twitter_handle \
1           NaN      Roger Voss      @vossroger
2 Willamette Valley      Paul Gregutt      @paulgwine
3           NaN Alexander Peartree      NaN
5           NaN Michael Schachner      @wineschach
8           NaN Anna Lee C. Iijima      NaN

```

```

           title           variety \
1 Quinta dos Avidagos 2011 Avidagos Red (Douro) Portuguese Red
2 Rainstorm 2013 Pinot Gris (Willamette Valley) Pinot Gris
3 St. Julian 2013 Reserve Late Harvest Riesling ... Riesling
5 Tandem 2011 Ars In Vitro Tempranillo-Merlot (N... Tempranillo-Merlot
8 Heinz Eifel 2013 Shine Gewürztraminer (Rheinhe... Gewürztraminer

```

```

           winery
1 Quinta dos Avidagos
2 Rainstorm
3 St. Julian
5 Tandem
8 Heinz Eifel

```

```

# selecting costly wines from US
data.loc[(data.country == 'US') & (data.price >= 75)]

```

	Unnamed: 0	country	description	designation
60	60	US	Syrupy and dense, this wine is jammy in plum a...	Estate
73	73	US	Juicy plum, raspberry and pencil lead lead the...	Bella Vetta Vineyard
...	...	...	...	...
129919	129919	US	This ripe, rich, almost decadently thick wine ...	Reserve
129967	129967	US	Citation is given as much as a decade of bottl...	NaN

```

# wines from Australia and New Zealand
data.loc[
    (data.country.isin(['Australia', 'New Zealand']))
]

```

	Unnamed: 0	country	description	designation
77	77	Australia	This medium-bodied Chardonnay features aromas ...	Made With Org
83	83	Australia	Pale copper in hue, this wine exudes passion f...	Jester Sangioves
...	...	...	...	...
129956	129956	New Zealand	The blend is 44% Merlot, 33% Cabernet Sauvigno...	Gimblett Grave
129958	129958	New Zealand	This blend of Cabernet Sauvignon-Merlot and Ca...	Irongate

```
# selecting rows and columns
columns = ['price', 'region_1', 'region_2']
rows = [1, 10, 100]
df = data.loc[rows, columns]
print(df)
```

	price	region_1	region_2
1	15.0	NaN	NaN
10	19.0	Napa Valley	Napa
100	18.0	Finger Lakes	Finger Lakes

```
# selecting notnull values
data.loc[data.price.notnull()]
```

	Unnamed: 0	country	description	designation
1	1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos
2	2	US	Tart and snappy, the flavors of lime flesh and...	NaN
...	...	...	...	...
129969	129969	France	A dry style of Pinot Gris, this is crisp with ...	NaN
129970	129970	France	Big, rich and off-dry, this is powered by inte...	Lieu-dit Harth Cuvée Car

## Assigning data

```
data['points'] = 50
print(data['points'])
```

0	50
1	50

```

..
129969    50
129970    50
Name: points, Length: 129971, dtype: int64

```

## Renaming and Combining

```

# renaming columns
print(data.rename(columns={'points' : 'score'}))

```

```

      Unnamed: 0    country \
0              0      Italy
1              1  Portugal
...          ...      ...
129969      129969    France
129970      129970    France

```

```

                                description \
0      Aromas include tropical fruit, broom, brimston...
1      This is ripe and fruity, a wine that is smooth...
...
129969  A dry style of Pinot Gris, this is crisp with ...
129970  Big, rich and off-dry, this is powered by inte...

```

```

                                designation  score  price    province \
0                                Vulkà Bianco    87    NaN  Sicily & Sardinia
1                                Avidagos      87   15.0         Douro
...
129969                                NaN     90   32.0         Alsace
129970  Lieu-dit Harth Cuvée Caroline     90   21.0         Alsace

```

```

      region_1 region_2    taster_name taster_twitter_handle \
0          Etna     NaN  Kerin O'Keefe      @kerinokeefe
1          NaN     NaN   Roger Voss      @vossroger
...
129969    Alsace     NaN   Roger Voss      @vossroger
129970    Alsace     NaN   Roger Voss      @vossroger

```

```

                                title    variety \
0                                Nicosia 2013 Vulkà Bianco  (Etna)    White Blend

```

```

1          Quinta dos Avidagos 2011 Avidagos Red (Douro) Portuguese Red
...
129969      Domaine Marcel Deiss 2012 Pinot Gris (Alsace)      Pinot Gris
129970      Domaine Schoffit 2012 Lieu-dit Harth Cuvée Car... Gewürztraminer

```

```

          winery
0          Nicosia
1          Quinta dos Avidagos
...
129969      Domaine Marcel Deiss
129970      Domaine Schoffit

```

[129971 rows x 14 columns]

```

# renaming indexes
print(data.rename(index={0:'first_entry', 1: 'second_entry'}))

```

```

          Unnamed: 0    country \
first_entry          0      Italy
second_entry         1    Portugal
...
129969          129969      France
129970          129970      France

```

```

          description \
first_entry  Aromas include tropical fruit, broom, brimston...
second_entry This is ripe and fruity, a wine that is smooth...
...
129969      A dry style of Pinot Gris, this is crisp with ...
129970      Big, rich and off-dry, this is powered by inte...

```

```

          designation  points  price          province \
first_entry          Vulkà Bianco      87      NaN  Sicily & Sardinia
second_entry          Avidagos      87    15.0          Douro
...
129969          NaN      90    32.0          Alsace
129970      Lieu-dit Harth Cuvée Caroline      90    21.0          Alsace

```

```

          region_1 region_2    taster_name taster_twitter_handle \
first_entry      Etna      NaN    Kerin O'Keefe      @kerinokeefe
second_entry      NaN      NaN      Roger Voss      @vossroger

```

```

...
129969      Alsace      NaN      Roger Voss      @vossroger
129970      Alsace      NaN      Roger Voss      @vossroger

```

```

                                title \
first_entry      Nicosia 2013 Vulkà Bianco (Etna)
second_entry     Quinta dos Avidagos 2011 Avidagos Red (Douro)
...
129969      Domaine Marcel Deiss 2012 Pinot Gris (Alsace)
129970      Domaine Schoffit 2012 Lieu-dit Harth Cuvée Car...

```

```

                                variety      winery
first_entry      White Blend      Nicosia
second_entry     Portuguese Red    Quinta dos Avidagos
...
129969      Pinot Gris      Domaine Marcel Deiss
129970      Gewürztraminer      Domaine Schoffit

```

[129971 rows x 14 columns]

```

# renaming axis
data.rename_axis ("wines", axis = 'rows').rename_axis('fields', axis = 'columns')

```

fields	Unnamed: 0	country	description	designation
wines				
0	0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco
1	1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos
...	...	...	...	...
129969	129969	France	A dry style of Pinot Gris, this is crisp with ...	NaN
129970	129970	France	Big, rich and off-dry, this is powered by inte...	Lieu-dit Harth Cuvée C

```

# combining with concat(), join(), and merge()
file1 = 'CAvideos.csv'
CAdat = pd.read_csv(file1)
CAdat

```

	video_id	trending_date	title	channel_title
0	n1WpP7iowLc	17.14.11	Eminem - Walk On Water (Audio) ft. Beyoncé	EminemVEVO
1	0dBlkQ4Mz1M	17.14.11	PLUSH - Bad Unboxing Fan Mail	iDubbbzTV

	video_id	trending_date	title	channel_title
...	...	...	...	...
40879	lbMKLzQ4cNQ	18.14.06	Trump Advisor Grovels To Trudeau	The Young Turks
40880	POTgw38-m58	18.14.06	2018.06.13	

```
file2 = 'FRvideos.csv'
FRdata = pd.read_csv(file2)
FRdata
```

	video_id	trending_date	title	channel_title
0	Ro6eob0LrCY	17.14.11	Malika LePen : Femme de Gauche - Trailer	Le Raptor I
1	Yo84eqYwP98	17.14.11	LA PIRE PARTIE ft Le Rire Jaune, Pierre Croce,...	Le Labo
...	...	...	...	...
40722	NlxE_QQMRzg	18.14.06	, 192 / Pomegranate seed / Nra...	PanArmenia
40723	_LgKglnqlc	18.14.06	Mandoubé ak Koor Gui 2018 Episode 28	Yesdakar

```
# joining
left = CAdat.set_index(['title', 'trending_date'])
right = FRdata.set_index(['title', 'trending_date'])

left.join(right, lsuffix= '_CAN', rsuffix = '_FR')
```

	title	trending_date	video_id
	!! THIS VIDEO IS NOTHING BUT PAIN !!   Getting Over It - Part 7	18.04.01	PNn8s
	#1 Fortnite World Rank - 2,323 Solo Wins!	18.09.03	DvPW
...	...	...	...
	BREAKING NEWS Raja Live all Slot Channels Welcome	18.07.05	Wt9G
	Active Shooter at YouTube Headquarters - LIVE BREAKING NEWS COVERAGE	18.04.04	Az72j

## Summary Functions and Maps

```
# some of the summary functions include- describe, mean, unique, value_counts
print(data.columns)
```

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

```
print(data.points.describe())
```

```
count      129971.000000
mean         88.447138
...
75%         91.000000
max         100.000000
Name: points, Length: 8, dtype: float64
```

```
# to see the list of unique values
print(data.taster_name.unique)
```

```
<bound method Series.unique of 0          Kerin O'Keefe
1          Roger Voss
...
129969      Roger Voss
129970      Roger Voss
Name: taster_name, Length: 129971, dtype: object>
```

```
print(data.taster_name.value_counts)
```

```
<bound method IndexOpsMixin.value_counts of 0          Kerin O'Keefe
1          Roger Voss
...
129969      Roger Voss
129970      Roger Voss
Name: taster_name, Length: 129971, dtype: object>
```

```
# best_bargain_wine- wine with the highest points-to-price ratio
bargain_idx = (data.points / data.price).idxmax()
bargain_wine = data.loc[bargain_idx, 'title']
print(bargain_wine)
```

```
Bandit NV Merlot (California)
```

## Maps

- takes one set of values and ‘maps’ them to another set of values
- example usage - **remean** the scores of wines received to 0
- use **apply** if you wish to call custom method on each row

```
review_points_mean = data.points.mean()
data.points.map(lambda p:p - review_points_mean)
```

```
0          -1.447138
1          -1.447138
...
129969      1.552862
129970      1.552862
Name: points, Length: 129971, dtype: float64
```

```
data_points_mean = data.points.mean()
data.points.map(lambda p:p - data_points_mean)
```

```
0          -1.447138
1          -1.447138
...
129969      1.552862
129970      1.552862
Name: points, Length: 129971, dtype: float64
```

```
# create descriptor_counts from description for 'tropical' and 'fruity'
n_tropical = data.description.map(lambda desc:'tropical' in desc).sum()
# desc signifies description
n_fruity = data.description.map(lambda desc:'fruity' in desc).sum()
descriptor_counts = pd.Series([n_tropical, n_fruity], index= ['tropical', 'fruity'])
print(descriptor_counts)
```

```
tropical    3607
fruity      9090
dtype: int64
```



### simplify with star ratings

- 95 and above = 3 stars
- between 85 and 95 = 2 stars
- less than 85 = 1 star
- plus, any wines from Canada should get 3 stars

```
print(data.columns)
```

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

```
# categorizing using map for points  
cat = data.points.map(lambda  
    p: 'three_stars' if p >= 95  
    else 'two stars' if p >= 85  
    else 'one star')
```

```
#count  
star_rating = cat.value_counts()  
  
print(star_rating)
```

```
points  
two stars    115125  
one star     12430  
three_stars   2416  
Name: count, dtype: int64
```

```
# categorizing using apply for points and Country  
cat2 = data.apply(lambda row:  
    'three stars' if (row['points'] >= 95 or row['country'] == 'Canada')  
    else 'two stars' if (row['points'] >= 85)  
    else 'one star', axis = 1)  
star_rating2 = cat2.value_counts()  
print(star_rating2)
```

```

two stars      114877
one star       12421
three stars    2673
Name: count, dtype: int64

```

```

# simple way without mapping
def stars(row):
    if row.country == 'Canada':
        return 3
    elif row.points >= 95:
        return 3
    elif row.points >= 85:
        return 2
    else:
        return 1

star_ratings = data.apply(stars, axis = 'columns')
print(star_ratings)

```

```

0      2
1      2
..
129969  2
129970  2
Length: 129971, dtype: int64

```

```

def data_points(row):
    row.points = row.points - data_points_mean
    return row

data.apply(data_points, axis = 'columns')

```

	Unnamed: 0	country	description	designation
0	0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco
1	1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos
...	...	...	...	...
129969	129969	France	A dry style of Pinot Gris, this is crisp with ...	NaN
129970	129970	France	Big, rich and off-dry, this is powered by inte...	Lieu-dit Harth Cuvée C

```
data.head(1)
```

	Unnamed: 0	country	description	designation	points	price
0	0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN

- operation (below) between a lot of values on the **left-hand** side  $>$  and a single value on the **right-hand** side (the mean value).

```
data_points_mean = data.points.mean()
data.points - data_points_mean
```

```
0      -1.447138
1      -1.447138
...
129969    1.552862
129970    1.552862
Name: points, Length: 129971, dtype: float64
```

```
data.country + "-" + data.region_1
```

```
0      Italy-Etna
1           NaN
...
129969  France-Alsace
129970  France-Alsace
Length: 129971, dtype: object
```

## Grouping and Sorting

use `groupby` to group data

`apply()` method can fetch us the data that matches the group

```
# groupwise analysis
data.groupby('points').points.count()
```

```
points
80      397
81      692
...
99       33
100      19
Name: points, Length: 21, dtype: int64
```

```
# ascending or descending order
data.groupby('points').price.min()
```

```
points
80      5.0
81      5.0
...
99     44.0
100    80.0
Name: price, Length: 21, dtype: float64
```

```
#grouping in countries and sorting
data.groupby(['country', 'province']).apply(lambda df:df.loc[df.points.idxmax()])
```

		Unnamed: 0	country	description
country	province			
Argentina	Mendoza Province	82754	Argentina	If the color doesn't tell the full story, the ...
	Other	78303	Argentina	Take note, this could be the best wine Colomé ...
...	...	...	...	...
Uruguay	San Jose	39898	Uruguay	Baked, sweet, heavy aromas turn earthy with ti...
	Uruguay	39361	Uruguay	Cherry and berry aromas are ripe, healthy and ...

```
help(pd.Series.idxmax)
```

Help on function idxmax in module pandas.core.series:

```
idxmax(self, axis: 'Axis' = 0, skipna: 'bool' = True, *args, **kwargs) -> 'Hashable'
    Return the row label of the maximum value.
```

If multiple values equal the maximum, the first row label with that

value is returned.

#### Parameters

-----

axis : {0 or 'index'}

Unused. Parameter needed for compatibility with DataFrame.

skipna : bool, default True

Exclude NA/null values. If the entire Series is NA, the result will be NA.

\*args, \*\*kwargs

Additional arguments and keywords have no effect but might be accepted for compatibility with NumPy.

#### Returns

-----

##### Index

Label of the maximum value.

#### Raises

-----

##### ValueError

If the Series is empty.

#### See Also

-----

numpy.argmax : Return indices of the maximum values along the given axis.

DataFrame.idxmax : Return index of first occurrence of maximum over requested axis.

Series.idxmin : Return index \*label\* of the first occurrence of minimum of values.

#### Notes

-----

This method is the Series version of ``ndarray.argmax``. This method returns the label of the maximum, while ``ndarray.argmax`` returns the position. To get the position, use ``series.values.argmax()``.

#### Examples

-----

```
>>> s = pd.Series(data=[1, None, 4, 3, 4],
...               index=['A', 'B', 'C', 'D', 'E'])
>>> s
```

```
A    1.0
B    NaN
C    4.0
D    3.0
E    4.0
dtype: float64
```

```
>>> s.idxmax()
'C'
```

If `skipna` is False and there is an NA value in the data, the function returns ``nan``.

```
>>> s.idxmax(skipna=False)
nan
```

```
data.groupby(['country']).price.agg([len, 'min', 'max'])
```

	len	min	max
country			
Argentina	3800	4.0	230.0
Armenia	2	14.0	15.0
...	...	...	...
Ukraine	14	6.0	13.0
Uruguay	109	10.0	130.0

## Multi-indexes

can help to convert to regular index

```
countries_reviewed = data.groupby(['country', 'province']).description.agg([len])
print(countries_reviewed)
```

		len
country	province	
Argentina	Mendoza Province	3264
	Other	536
...	...	...
Uruguay	San Jose	3

[425 rows x 1 columns]

```
mi = countries_reviewed.index
type(mi)
```

pandas.core.indexes.multi.MultiIndex

```
countries_reviewed.reset_index()
```

	country	province	len
0	Argentina	Mendoza Province	3264
1	Argentina	Other	536
...	...	...	...
423	Uruguay	San Jose	3
424	Uruguay	Uruguay	24

```
# create a series of price and points. sort values by price (ascending)
rating = data.groupby('price')['points'].max().sort_index()
print(rating)
```

```
price
4.0      86
5.0      87
..
2500.0   96
3300.0   88
Name: points, Length: 390, dtype: int64
```

```
df = data.groupby('variety').price.agg('max', 'min')
print(df)
```

```
variety
Abouriou      75.0
Agiorgitiko    66.0
```

```

...
Çalkarasi      19.0
Žilavka        15.0
Name: price, Length: 707, dtype: float64

```

## Sorting

```

#ascending by default
countries_reviewed = countries_reviewed.reset_index()
countries_reviewed.sort_values(by= 'len')

```

	country	province	len
179	Greece	Muscat of Kefallonian	1
192	Greece	Sterea Ellada	1
...	...	...	...
415	US	Washington	8639
392	US	California	36247

```

# descending
countries_reviewed.sort_values(by= 'len', ascending= False)

```

	country	province	len
392	US	California	36247
415	US	Washington	8639
...	...	...	...
63	Chile	Coelemu	1
149	Greece	Beotia	1

```

# sorting index_values
countries_reviewed.sort_index()

```

	country	province	len
0	Argentina	Mendoza Province	3264
1	Argentina	Other	536
...	...	...	...
423	Uruguay	San Jose	3



	country	province	len
424	Uruguay	Uruguay	24

```
# sorting more than one column
countries_reviewed.sort_values(by=['country', 'len'])
```

	country	province	len
1	Argentina	Other	536
0	Argentina	Mendoza Province	3264
...	...	...	...
424	Uruguay	Uruguay	24
419	Uruguay	Canelones	43

## Data Types and Missing Data

missing values are given the value NaN - 'Not a Number'- float64 dtype

```
# find the data type
data.price.dtype
```

```
dtype('float64')
```

```
# for every column
print(data.dtypes)
```

```
Unnamed: 0      int64
country         object
...
variety         object
winery          object
Length: 14, dtype: object
```

```
# transform data type
data.points.astype('float64')
```

```

0          87.0
1          87.0
...
129969     90.0
129970     90.0
Name: points, Length: 129971, dtype: float64

```

```

# finding values in country by NaN
data[pd.isnull(data.country)]

```

	Unnamed: 0	country	description	designation	points
913	913	NaN	Amber in color, this wine has aromas of peach ...	Asureti Valley	87
3131	3131	NaN	Soft, fruity and juicy, this is a pleasant, si...	Partager	83
...	...	...	...	...	...
129590	129590	NaN	A blend of 60% Syrah, 30% Cabernet Sauvignon a...	Shah	90
129900	129900	NaN	This wine offers a delightful bouquet of black...	NaN	91

```

# replacing missing values
data.country.fillna('Unknown')

```

```

0          Italy
1          Portugal
...
129969     France
129970     France
Name: country, Length: 129971, dtype: object

```

```

# replacing ('what?','bywhat?')
data.price.replace('NaN', '@Unknown')

```

```

0          NaN
1          15.0
...
129969     32.0
129970     21.0
Name: price, Length: 129971, dtype: float64

```

```
# missing price values and count them
data.price.isnull().sum()
```

8996

```
# arrange region_1 in ascending order of values
data.region_1.fillna('Unkown').value_counts().sort_values(ascending= False)
```

```
region_1
Unkown      21247
Napa Valley   4480
...
Geelong        1
Paestum        1
Name: count, Length: 1230, dtype: int64
```