<div id='noscript'> Jupyter Notebook requires JavaScript.<br> Please enable it to proceed. </div>

[Jupyter Notebook](http://localhost:8888/tree?token=889103a6f67953001eff9af7d52abb91420826df69628386)

Data Analysis with winemag-data Last Checkpoint: 12 minutes ago (unsaved changes) Logout

Menu

Python 3

Trusted

[File](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[New NotebookToggle Dropdown](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Python 3](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Open...](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Make a Copy...](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Save as...](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Rename...](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Save and Checkpoint](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Revert to CheckpointToggle Dropdown](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Thursday, August 15, 2019 4:29 PM](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Print Preview](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Download asToggle Dropdown](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[asciidoc (.asciidoc)](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[html (.html)](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[latex (.tex)](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[markdown (.md)](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[notebook (.ipynb)](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[pdf (.pdf)](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[rst (.rst)](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Python (.py)](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[slides (.slides.html)](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Deploy as](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Trusted Notebook](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Close and Halt](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Edit](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Cut Cells](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Copy Cells](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Paste Cells Above](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Paste Cells Below](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Paste Cells & Replace](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Delete Cells](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Undo Delete Cells](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Split Cell](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Merge Cell Above](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Merge Cell Below](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Move Cell Up](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Move Cell Down](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Edit Notebook Metadata](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Find and Replace](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Cut Cell Attachments](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Copy Cell Attachments](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Paste Cell Attachments](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Insert Image](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[View](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Toggle Header](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Toggle Toolbar](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Toggle Line Numbers](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Cell Toolbar](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[None](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Edit Metadata](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Raw Cell Format](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Slideshow](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Attachments](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Tags](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Insert](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Insert Cell Above](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Insert Cell Below](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Cell](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Run Cells](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Run Cells and Select Below](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Run Cells and Insert Below](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Run All](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Run All Above](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Run All Below](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Cell Type](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Code](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Markdown](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Raw NBConvert](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Current Outputs](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Toggle](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Toggle Scrolling](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Clear](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[All Output](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Toggle](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Toggle Scrolling](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Clear](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Kernel](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Interrupt](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Restart](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Restart & Clear Output](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Restart & Run All](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Reconnect](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Shutdown](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Change kernel](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Python 3](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Help](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[User Interface Tour](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Keyboard Shortcuts](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Edit Keyboard Shortcuts](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

[Notebook Help](http://nbviewer.jupyter.org/github/ipython/ipython/blob/3.x/examples/Notebook/Index.ipynb)

[Markdown](https://help.github.com/articles/markdown-basics/)

[Python Reference](https://docs.python.org/3.7?v=20190815090205)

[IPython Reference](https://ipython.org/documentation.html?v=20190815090205)

[NumPy Reference](https://docs.scipy.org/doc/numpy/reference/?v=20190815090205)

[SciPy Reference](https://docs.scipy.org/doc/scipy/reference/?v=20190815090205)

[Matplotlib Reference](https://matplotlib.org/contents.html?v=20190815090205)

[SymPy Reference](http://docs.sympy.org/latest/index.html?v=20190815090205)

[pandas Reference](https://pandas.pydata.org/pandas-docs/stable/?v=20190815090205)

[About](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)

Run



In [96]:



# import modules which r the libriary

import numpy as np

import pandas as pd

import seaborn as sns

import bokeh as bk

from bokeh.io import output\_notebook, show

output\_notebook()

BokehJS 1.2.0 successfully loaded.

. . .

In [97]:



%matplotlib inline

# inline matplotlib (keep charts in this nb(notebook))

import matplotlib.pyplot as plt

. . .

In [3]:



x

import warnings

warnings.filterwarnings('ignore')

#This was a warning on the KDE plot for 2D topo mapping (topographical mapping)

# in simple words we r ignorning any warning

. . .

In [6]:



x

# A list of column names to be displayed

column\_names = [ "Index", "Country", "Description", "Designation", "Points", "Prices", "Provinces",

"Region\_1", "Region\_2", "Variety", "Winery"]

. . .

In [7]:



#Reading the CSV file using the col\_names list in the name parameters:

reviews\_df = pd.read\_csv("Desktop/winemag-data\_first150k.csv", sep=",", header=1, names = column\_names, index\_col=0)

. . .

In [44]:



#Dataframe shape

shape = reviews\_df.shape

print("The shape of the dataframe (rows, columns): \t {}".format (shape))

The shape of the dataframe (rows, columns): (137229, 10)

. . .

In [9]:



#Display the first ten records:

reviews\_df.head(10)

Out[9]:

|  | Country | Description | Designation | Points | Prices | Provinces | Region\_1 | Region\_2 | Variety | Winery |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index |  |  |  |  |  |  |  |  |  |  |
| 1 | Spain | Ripe aromas of fig, blackberry and cassis are ... | Carodorum Selección Especial Reserva | 96 | 110.0 | Northern Spain | Toro | NaN | Tinta de Toro | Bodega Carmen Rodríguez |
| 2 | US | Mac Watson honors the memory of a wine once ma... | Special Selected Late Harvest | 96 | 90.0 | California | Knights Valley | Sonoma | Sauvignon Blanc | Macauley |
| 3 | US | This spent 20 months in 30% new French oak, an... | Reserve | 96 | 65.0 | Oregon | Willamette Valley | Willamette Valley | Pinot Noir | Ponzi |
| 4 | France | This is the top wine from La Bégude, named aft... | La Brûlade | 95 | 66.0 | Provence | Bandol | NaN | Provence red blend | Domaine de la Bégude |
| 5 | Spain | Deep, dense and pure from the opening bell, th... | Numanthia | 95 | 73.0 | Northern Spain | Toro | NaN | Tinta de Toro | Numanthia |
| 6 | Spain | Slightly gritty black-fruit aromas include a s... | San Román | 95 | 65.0 | Northern Spain | Toro | NaN | Tinta de Toro | Maurodos |
| 7 | Spain | Lush cedary black-fruit aromas are luxe and of... | Carodorum Único Crianza | 95 | 110.0 | Northern Spain | Toro | NaN | Tinta de Toro | Bodega Carmen Rodríguez |
| 8 | US | This re-named vineyard was formerly bottled as... | Silice | 95 | 65.0 | Oregon | Chehalem Mountains | Willamette Valley | Pinot Noir | Bergström |
| 9 | US | The producer sources from two blocks of the vi... | Gap's Crown Vineyard | 95 | 60.0 | California | Sonoma Coast | Sonoma | Pinot Noir | Blue Farm |
| 10 | Italy | Elegance, complexity and structure come togeth... | Ronco della Chiesa | 95 | 80.0 | Northeastern Italy | Collio | NaN | Friulano | Borgo del Tiglio |

. . .

In [10]:



reviews\_df.info()# To obtain information on the datatype and the memory

<class 'pandas.core.frame.DataFrame'>

Int64Index: 150929 entries, 1 to 150929

Data columns (total 10 columns):

Country 150924 non-null object

Description 150929 non-null object

Designation 105194 non-null object

Points 150929 non-null int64

Prices 137234 non-null float64

Provinces 150924 non-null object

Region\_1 125869 non-null object

Region\_2 60952 non-null object

Variety 150929 non-null object

Winery 150929 non-null object

dtypes: float64(1), int64(1), object(8)

memory usage: 12.7+ MB

. . .

In [11]:



# Display the datatypes of the reviews\_df Dataframe object(Displaying the datatypes for the various columns in the set)

reviews\_df.dtypes

Out[11]:

Country object

Description object

Designation object

Points int64

Prices float64

Provinces object

Region\_1 object

Region\_2 object

Variety object

Winery object

dtype: object

. . .

In [12]:



# Boolean - True if column has (missing) null values, False if no null values:

reviews\_df.isnull().any()

Out[12]:

Country True

Description False

Designation True

Points False

Prices True

Provinces True

Region\_1 True

Region\_2 True

Variety False

Winery False

dtype: bool

. . .

In [13]:



# The sum of null values per column:

reviews\_df.isnull().sum()

Out[13]:

Country 5

Description 0

Designation 45735

Points 0

Prices 13695

Provinces 5

Region\_1 25060

Region\_2 89977

Variety 0

Winery 0

dtype: int64

. . .

In [14]:



reviews\_df[reviews\_df.Country.isnull()]

Out[14]:

|  | Country | Description | Designation | Points | Prices | Provinces | Region\_1 | Region\_2 | Variety | Winery |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index |  |  |  |  |  |  |  |  |  |  |
| 1133 | NaN | Delicate white flowers and a spin of lemon pee... | Askitikos | 90 | 17.0 | NaN | NaN | NaN | Assyrtiko | Tsililis |
| 1440 | NaN | A blend of 60% Syrah, 30% Cabernet Sauvignon a... | Shah | 90 | 30.0 | NaN | NaN | NaN | Red Blend | Büyülübağ |
| 68226 | NaN | From first sniff to last, the nose never makes... | Piedra Feliz | 81 | 15.0 | NaN | NaN | NaN | Pinot Noir | Chilcas |
| 113016 | NaN | From first sniff to last, the nose never makes... | Piedra Feliz | 81 | 15.0 | NaN | NaN | NaN | Pinot Noir | Chilcas |
| 135696 | NaN | From first sniff to last, the nose never makes... | Piedra Feliz | 81 | 15.0 | NaN | NaN | NaN | Pinot Noir | Chilcas |

. . .

In [15]:



# confirm dataframe shape

reviews\_df.shape

Out[15]:

(150929, 10)

. . .

In [16]:



#Drop the records with null values for "Country", update the reviews\_df dataframe:

reviews\_df = reviews\_df.dropna (how="any", subset=["Country"])

. . .

In [17]:



#Confirm null values have been dropped:

reviews\_df.shape

Out[17]:

(150924, 10)

. . .

In [18]:



# Display confirmation of no records with "Country" null values:

reviews\_df[reviews\_df.Country.isnull()]

Out[18]:

|  | Country | Description | Designation | Points | Prices | Provinces | Region\_1 | Region\_2 | Variety | Winery |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index |  |  |  |  |  |  |  |  |  |  |

. . .

In [19]:



# checking if country still has five missen values after dropping them

reviews\_df.isnull().sum()

Out[19]:

Country 0

Description 0

Designation 45735

Points 0

Prices 13695

Provinces 0

Region\_1 25055

Region\_2 89972

Variety 0

Winery 0

dtype: int64

. . .

In [20]:



# Calculate the percentage of remaining records if missing prices are dropped:

total\_records = len(reviews\_df)

missing\_prices = reviews\_df["Prices"].isnull().sum()

missing\_ratio = 100-((missing\_prices / total\_records)\*100)

print ("Missing prices:{}".format (missing\_prices)+ "\n" +

"Total records:{}".format (total\_records) + "\n" +

"Percentage remaining: {:.2f} %".format (missing\_ratio))

Missing prices:13695

Total records:150924

Percentage remaining: 90.93 %

. . .

In [21]:



# Confirm dataframe shape

reviews\_df.shape

Out[21]:

(150924, 10)

. . .

In [22]:



# Drop the records with null values for "Prices", update the reviews\_df dataframe:

reviews\_df = reviews\_df.dropna (how="any", subset=["Prices"])

​

. . .

In [23]:



# Confirm the null values have been dropped:

reviews\_df.shape

Out[23]:

(137229, 10)

. . .

In [24]:



# Display confirmation of no records with "Prices" null values:

reviews\_df[reviews\_df.Prices.isnull()]

Out[24]:

|  | Country | Description | Designation | Points | Prices | Provinces | Region\_1 | Region\_2 | Variety | Winery |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index |  |  |  |  |  |  |  |  |  |  |

. . .

In [25]:



# How many rows hove zero null values in any column?

complete = reviews\_df[reviews\_df.isnull ().any (axis=1)].shape

print("Records without null values:\n (rows, columns) \t {}".format (complete))

Records without null values:

(rows, columns) (97989, 10)

. . .

In [26]:



# calculate the percentage of fully complete rows i the dataset:

complete\_rows = int (complete[0])

total\_rows = len(reviews\_df)

. . .

In [27]:



# Calculations

complete\_ratio = ((complete\_rows / total\_rows) \* 100)

​

print("Total complete records: \t\t{}".format(complete\_rows) +

"\nTotal records: \t\t\t\t{}".format(total\_rows) +

"\nComplete ratio: \t\t\t{:.2f}%".format(complete\_ratio))

Total complete records: 97989

Total records: 137229

Complete ratio: 71.41%

. . .

In [ ]:



​

. . .

In [28]:



# Country i.e USA

reviews\_df["Country"].describe()

Out[28]:

count 137229

unique 46

top US

freq 62138

Name: Country, dtype: object

. . .

In [29]:



# Designation

reviews\_df["Designation"].describe()

Out[29]:

count 94918

unique 28342

top Reserve

freq 2704

Name: Designation, dtype: object

. . .

In [30]:



# Points

reviews\_df["Points"].describe()

Out[30]:

count 137229.000000

mean 87.787975

std 3.221507

min 80.000000

25% 86.000000

50% 88.000000

75% 90.000000

max 100.000000

Name: Points, dtype: float64

. . .

In [31]:



# Prices

reviews\_df["Prices"].describe()

Out[31]:

count 137229.000000

mean 33.130548

std 36.319116

min 4.000000

25% 16.000000

50% 24.000000

75% 40.000000

max 2300.000000

Name: Prices, dtype: float64

. . .

In [32]:



# Provinces

reviews\_df["Provinces"].describe()

Out[32]:

count 137229

unique 446

top California

freq 44355

Name: Provinces, dtype: object

. . .

In [33]:



#Standard Deviation of the DataFrame

reviews\_df.std()

Out[33]:

Points 3.221507

Prices 36.319116

dtype: float64

. . .

In [34]:



# Total amount of Countries featured in review:

total\_countries = reviews\_df["Country"].unique()

print("Number of Countries reviewed: {}".format (len(total\_countries)))

Number of Countries reviewed: 46

. . .

In [35]:



# The unique countries in reviews:

unique\_countries = reviews\_df["Country"].unique ()

print("Total amount of unique countries: {}".format(len(unique\_countries)))

print(unique\_countries)

Total amount of unique countries: 46

['Spain' 'US' 'France' 'Italy' 'New Zealand' 'Bulgaria' 'Argentina'

'Australia' 'Portugal' 'Israel' 'South Africa' 'Greece' 'Chile' 'Morocco'

'Romania' 'Germany' 'Canada' 'Moldova' 'Hungary' 'Austria' 'Croatia'

'Slovenia' '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'

'US-France']

. . .

In [36]:



# Total wine reviews per country (Top 25)

review\_totals\_by\_country = reviews\_df["Country"].value\_counts().head(25)

print("Total amount of Reviews submitted for each country: \n")

print(review\_totals\_by\_country)

Total amount of Reviews submitted for each country:

US 62138

Italy 18784

France 14785

Spain 8160

Chile 5766

Argentina 5587

Australia 4894

Portugal 4176

New Zealand 3070

Austria 2483

Germany 2347

South Africa 2237

Greece 872

Israel 610

Hungary 230

Canada 194

Romania 139

Uruguay 85

Croatia 83

Slovenia 81

Bulgaria 77

Moldova 71

Mexico 63

Turkey 50

Georgia 43

Name: Country, dtype: int64

. . .

In [37]:



# Group country by the most common variety reviewed (first ten).

reviews\_df.groupby("Country").Variety.max().head(10)

Out[37]:

Country

Albania Kallmet

Argentina White Blend

Australia Zinfandel

Austria Zweigelt

Bosnia and Herzegovina Vranec

Brazil Tannat

Bulgaria White Blend

Canada White Blend

Chile White Blend

China White Blend

Name: Variety, dtype: object

. . .

In [38]:



# Group points per country -count, min, max points:

reviews\_df.groupby(['Country']).Points.agg([len, min, max])

Out[38]:

|  | len | min | max |
| --- | --- | --- | --- |
| Country |  |  |  |
| Albania | 2 | 88 | 88 |
| Argentina | 5587 | 80 | 97 |
| Australia | 4894 | 80 | 100 |
| Austria | 2483 | 81 | 98 |
| Bosnia and Herzegovina | 4 | 83 | 88 |
| Brazil | 25 | 81 | 88 |
| Bulgaria | 77 | 80 | 90 |
| Canada | 194 | 82 | 93 |
| Chile | 5766 | 80 | 95 |
| China | 3 | 82 | 82 |
| Croatia | 83 | 81 | 91 |
| Cyprus | 31 | 80 | 89 |
| Czech Republic | 6 | 85 | 87 |
| England | 8 | 91 | 94 |
| France | 14785 | 80 | 100 |
| Georgia | 43 | 81 | 92 |
| Germany | 2347 | 80 | 97 |
| Greece | 872 | 80 | 92 |
| Hungary | 230 | 80 | 96 |
| India | 8 | 82 | 91 |
| Israel | 610 | 80 | 93 |
| Italy | 18784 | 80 | 100 |
| Japan | 2 | 85 | 85 |
| Lebanon | 37 | 81 | 91 |
| Lithuania | 8 | 84 | 85 |
| Luxembourg | 9 | 86 | 88 |
| Macedonia | 16 | 81 | 89 |
| Mexico | 63 | 80 | 92 |
| Moldova | 71 | 81 | 90 |
| Montenegro | 2 | 82 | 82 |
| Morocco | 12 | 82 | 93 |
| New Zealand | 3070 | 80 | 94 |
| Portugal | 4176 | 80 | 99 |
| Romania | 139 | 80 | 92 |
| Serbia | 14 | 86 | 89 |
| Slovakia | 3 | 82 | 87 |
| Slovenia | 81 | 82 | 92 |
| South Africa | 2237 | 80 | 95 |
| South Korea | 4 | 81 | 82 |
| Spain | 8160 | 80 | 98 |
| Switzerland | 4 | 83 | 90 |
| Turkey | 50 | 84 | 92 |
| US | 62138 | 80 | 100 |
| US-France | 1 | 88 | 88 |
| Ukraine | 5 | 83 | 86 |
| Uruguay | 85 | 80 | 90 |

. . .

In [39]:



# Sum Reviews per Country + Region

reviews\_df.Country + " - " + reviews\_df.Region\_1

​

Out[39]:

Index

1 Spain - Toro

2 US - Knights Valley

3 US - Willamette Valley

4 France - Bandol

5 Spain - Toro

6 Spain - Toro

7 Spain - Toro

8 US - Chehalem Mountains

9 US - Sonoma Coast

10 Italy - Collio

11 US - Ribbon Ridge

12 US - Dundee Hills

13 France - Madiran

14 US - Dundee Hills

15 US - Willamette Valley

16 US - Diamond Mountain District

17 Spain - Ribera del Duero

18 France - Cahors

19 US - Sonoma Coast

20 US - Napa Valley

21 Spain - Rioja

22 Spain - Toro

23 US - Edna Valley

24 US - Santa Cruz Mountains

25 NaN

26 US - Willamette Valley

27 US - Willamette Valley

28 US - Santa Lucia Highlands

29 US - Walla Walla Valley (WA)

30 NaN

...

150899 NaN

150900 NaN

150901 NaN

150902 NaN

150903 NaN

150904 NaN

150905 NaN

150906 France - Vosne-Romanée

150907 France - Vosne-Romanée

150908 France - Vosne-Romanée

150909 France - Nuits-St.-Georges

150910 France - Chambolle-Musigny

150911 France - Nuits-St.-Georges

150912 France - Chambolle-Musigny

150913 France - Châteauneuf-du-Pape

150914 US - Anderson Valley

150915 US - North Coast

150916 US - Napa Valley

150917 France - Champagne

150918 France - Champagne

150919 France - Champagne

150920 Italy - Trento

150921 France - Champagne

150923 France - Champagne

150924 France - Champagne

150925 Italy - Fiano di Avellino

150926 France - Champagne

150927 Italy - Fiano di Avellino

150928 France - Champagne

150929 Italy - Alto Adige

Length: 137229, dtype: object

. . .

In [43]:



# The most expensive wine

max\_price = reviews\_df['Prices'].max()

print("The most expensive wine in all the reviews cost $ {:.2f}".format (max\_price))

The most expensive wine in all the reviews cost $ 2300.00

. . .

In [41]:



# reviews\_df.groupby ("Variety").Price.max()

reviews\_df[reviews\_df['Prices'] == 2300]

Out[41]:

|  | Country | Description | Designation | Points | Prices | Provinces | Region\_1 | Region\_2 | Variety | Winery |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index |  |  |  |  |  |  |  |  |  |  |
| 34920 | France | A big, powerful wine that sums up the richness... | NaN | 99 | 2300.0 | Bordeaux | Pauillac | NaN | Bordeaux-style Red Blend | Château Latour |

. . .

In [ ]:



x

# The average wine cost

avg\_price = reviews\_df['Prices'].mean()

print("The average bottle in the review cost $ {:.2f} USD".format (avg\_price))

. . .

In [ ]:



# The cheapest wine cost

min\_price = reviews\_df['Prices'].min()

print("The cheapest wine in the review cost $ {:.2f} USD".format (min\_price))

. . .

In [ ]:



# The country with the cheapest wine

reviews\_df[reviews\_df['Prices'] == 4.0]

. . .

In [45]:



reviews\_df.Points.describe() # calling previous computation on count, meean etc

Out[45]:

count 137229.000000

mean 87.787975

std 3.221507

min 80.000000

25% 86.000000

50% 88.000000

75% 90.000000

max 100.000000

Name: Points, dtype: float64

. . .

In [47]:



# Sum of reviews per variety (Top 25)

reviews\_df["Variety"].value\_counts().head(25)

Out[47]:

Chardonnay 13775

Pinot Noir 13625

Cabernet Sauvignon 12670

Red Blend 9377

Sauvignon Blanc 6054

Syrah 5667

Riesling 5212

Merlot 4987

Bordeaux-style Red Blend 4545

Zinfandel 3794

Malbec 3085

Sangiovese 2879

White Blend 2554

Tempranillo 2525

Rosé 2461

Shiraz 1945

Sparkling Blend 1820

Portuguese Red 1812

Nebbiolo 1529

Rhône-style Red Blend 1455

Cabernet Franc 1310

Corvina, Rondinella, Molinara 1292

Pinot Gris 1275

Pinot Grigio 1270

Viognier 1255

Name: Variety, dtype: int64

. . .

In [49]:



x

# Number of unique wine varieties in the reviews:

unique\_wines = reviews\_df["Variety"].unique()

print("Unique wine Varieties in the reviews: \n{}".format(len(unique\_wines)))

Unique wine Varieties in the reviews:

619

. . .

In [52]:



# Top 3 countries

reviews\_df["Variety"].value\_counts().head(3)

Out[52]:

Chardonnay 13775

Pinot Noir 13625

Cabernet Sauvignon 12670

Name: Variety, dtype: int64

. . .

In [53]:



# Groupby country by variety (first ten only)

reviews\_df.groupby("Country").Variety.max().head(10)

Out[53]:

Country

Albania Kallmet

Argentina White Blend

Australia Zinfandel

Austria Zweigelt

Bosnia and Herzegovina Vranec

Brazil Tannat

Bulgaria White Blend

Canada White Blend

Chile White Blend

China White Blend

Name: Variety, dtype: object

. . .

In [56]:



x

%%time

# Plotting points and Price grouped by Variety (only ten visible)

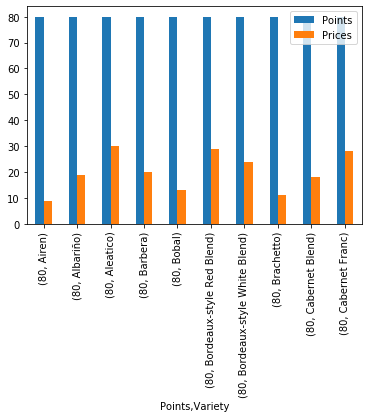
​

reviews\_df.groupby(['Points', 'Variety']).apply(lambda reviews\_df: reviews\_df.loc[reviews\_df.Points.argmax()]).head(10).plot.bar()

Wall time: 1.39 s

Out[56]:

<matplotlib.axes.\_subplots.AxesSubplot at 0xe869f36780>



. . .

In [60]:



%%time

​

# Group point and price values by country and variety

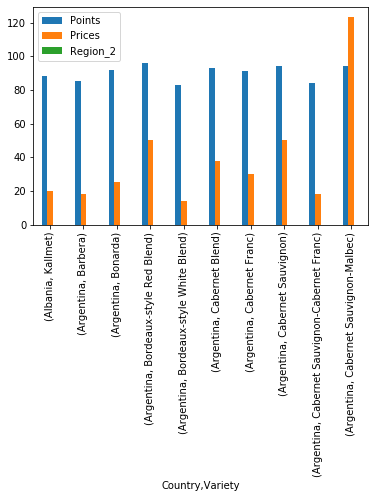
​

reviews\_df.groupby(['Country', 'Variety']).apply(lambda reviews\_df: reviews\_df.loc[reviews\_df.Points.argmax()]).head(10).plot.bar()

Wall time: 594 ms

Out[60]:

<matplotlib.axes.\_subplots.AxesSubplot at 0xe8622bfef0>



. . .

In [61]:



# Get wine from Argentina~!

#Here are the Top Ten Wines sorted by points~!

​

Argentina\_top\_ten = reviews\_df[reviews\_df['Country'].str.match('Argentina')]

Argentina\_top\_ten.sort\_values("Points", ascending=False).head(10)

Out[61]:

|  | Country | Description | Designation | Points | Prices | Provinces | Region\_1 | Region\_2 | Variety | Winery |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index |  |  |  |  |  |  |  |  |  |  |
| 116691 | Argentina | If the color doesn't tell the full story, the ... | Nicasia Vineyard | 97 | 120.0 | Mendoza Province | Mendoza | NaN | Malbec | Bodega Catena Zapata |
| 121311 | Argentina | If the color doesn't tell the full story, the ... | Nicasia Vineyard | 97 | 120.0 | Mendoza Province | Mendoza | NaN | Malbec | Bodega Catena Zapata |
| 65331 | Argentina | If the color doesn't tell the full story, the ... | Nicasia Vineyard | 97 | 120.0 | Mendoza Province | Mendoza | NaN | Malbec | Bodega Catena Zapata |
| 107804 | Argentina | This blend from the Nicasia and Adrianna viney... | Argentino | 96 | 123.0 | Mendoza Province | Mendoza | NaN | Malbec | Bodega Catena Zapata |
| 63766 | Argentina | Features intoxicating pastry, blueberry, cola ... | Nicasia Vineyard | 96 | 123.0 | Mendoza Province | Mendoza | NaN | Malbec | Bodega Catena Zapata |
| 107806 | Argentina | Features intoxicating pastry, blueberry, cola ... | Nicasia Vineyard | 96 | 123.0 | Mendoza Province | Mendoza | NaN | Malbec | Bodega Catena Zapata |
| 63764 | Argentina | This blend from the Nicasia and Adrianna viney... | Argentino | 96 | 123.0 | Mendoza Province | Mendoza | NaN | Malbec | Bodega Catena Zapata |
| 83001 | Argentina | This wine takes you on a trip to an outer wine... | Gran Corte Las Divas Vineyard | 96 | 50.0 | Mendoza Province | Tupungato | NaN | Bordeaux-style Red Blend | Riglos |
| 130756 | Argentina | Features intoxicating pastry, blueberry, cola ... | Nicasia Vineyard | 96 | 123.0 | Mendoza Province | Mendoza | NaN | Malbec | Bodega Catena Zapata |
| 130754 | Argentina | This blend from the Nicasia and Adrianna viney... | Argentino | 96 | 123.0 | Mendoza Province | Mendoza | NaN | Malbec | Bodega Catena Zapata |

. . .

In [ ]:



# create a new dataframe for the perfect scores records:

perfect\_scores\_df = reviews\_df[reviews\_df['Points'] == 100]

perfect\_scores\_df

. . .

In [63]:



# How many records have a score of 100?

perfect\_scores\_df.shape

Out[63]:

(24, 10)

. . .

In [64]:



# Count how many perfect scores per country:

perfect\_scores\_df["Country"].value\_counts()

Out[64]:

US 10

Italy 7

France 4

Australia 3

Name: Country, dtype: int64

. . .

In [66]:



# Analyze the range in price for perfect score wines by country:

perfect\_scores\_df.groupby(['Country']).Prices.agg([len, min, max])

Out[66]:

|  | len | min | max |
| --- | --- | --- | --- |
| Country |  |  |  |
| Australia | 3.0 | 300.0 | 300.0 |
| France | 4.0 | 848.0 | 1400.0 |
| Italy | 7.0 | 195.0 | 460.0 |
| US | 10.0 | 65.0 | 245.0 |

. . .

In [67]:



#seaborn representation

%config InlineBackend.figure\_format='retina'

sns.set() #Revert to matplotlib defaults

plt.rcParams['figure.figsize'] = (9, 6)

plt.rcParams['axes.labelpad'] = 10

sns.set\_style("darkgrid")

#sns.set\_palette("Reds")

#sns.set\_context("Poster", font\_scale=1.0)

. . .

In [69]:



x

%%time

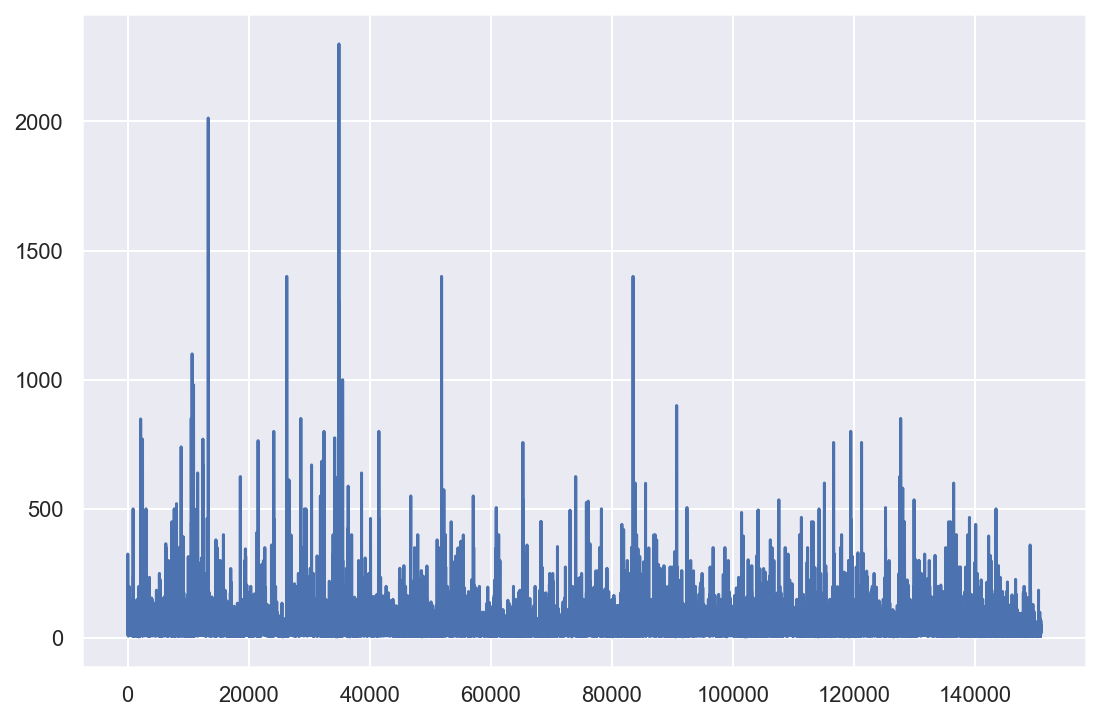
#Matplotlib to show random order by price

plt.plot(reviews\_df.index, reviews\_df["Prices"])

Wall time: 31.3 ms

Out[69]:

[<matplotlib.lines.Line2D at 0xe868a9d240>]



. . .

In [70]:



x

fig, ax = plt.subplots(figsize=(14, 6))

​

reviews\_df.groupby(reviews\_df['Country']) ['Prices'].mean().plot.bar()

mean\_price = reviews\_df["Prices"].mean(0)

reviews\_df["Prices"].fillna(mean\_price)

​

# create labels

ax.set\_xlabel('Country')

ax.set\_ylabel('Mean Price')

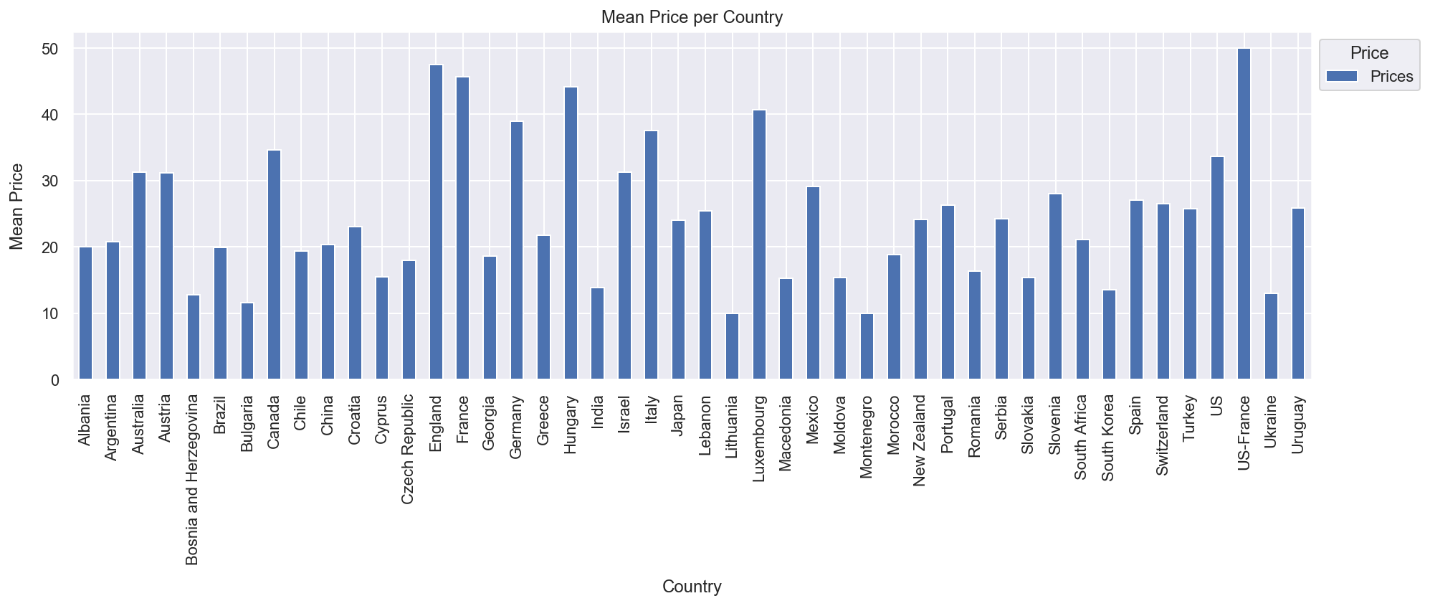
# sets the plot title

ax.set\_title('Mean Price per Country')

ax.legend(bbox\_to\_anchor=(1, 1), loc=2, title='Price')

​

fig.tight\_layout()



. . .

In [72]:



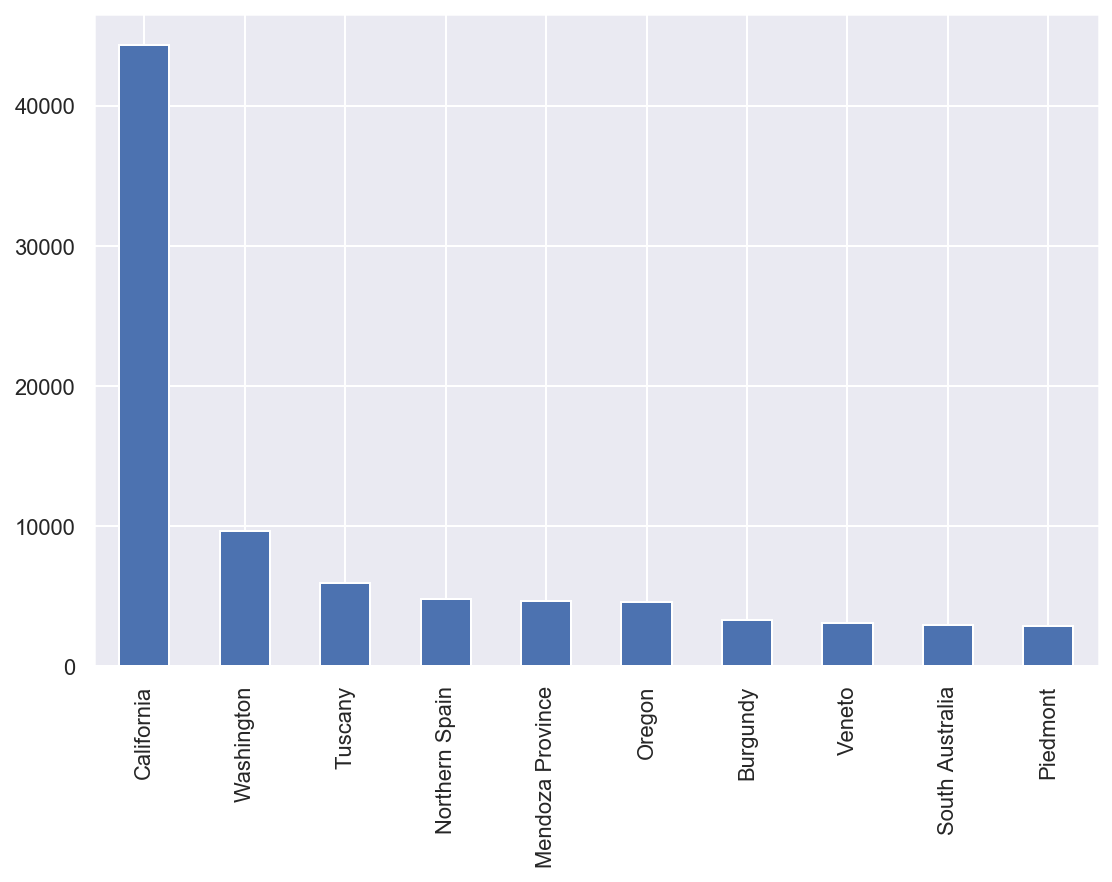
%%time

reviews\_df['Provinces'].value\_counts().head(10).plot.bar()

Wall time: 46.9 ms

Out[72]:

<matplotlib.axes.\_subplots.AxesSubplot at 0xe869e90438>



. . .

In [73]:



top\_province = reviews\_df['Provinces'].value\_counts().head(1)

print("The province with the most reviews is {}".format(top\_province))

The province with the most reviews is California 44355

Name: Provinces, dtype: int64

. . .

In [74]:



%%time

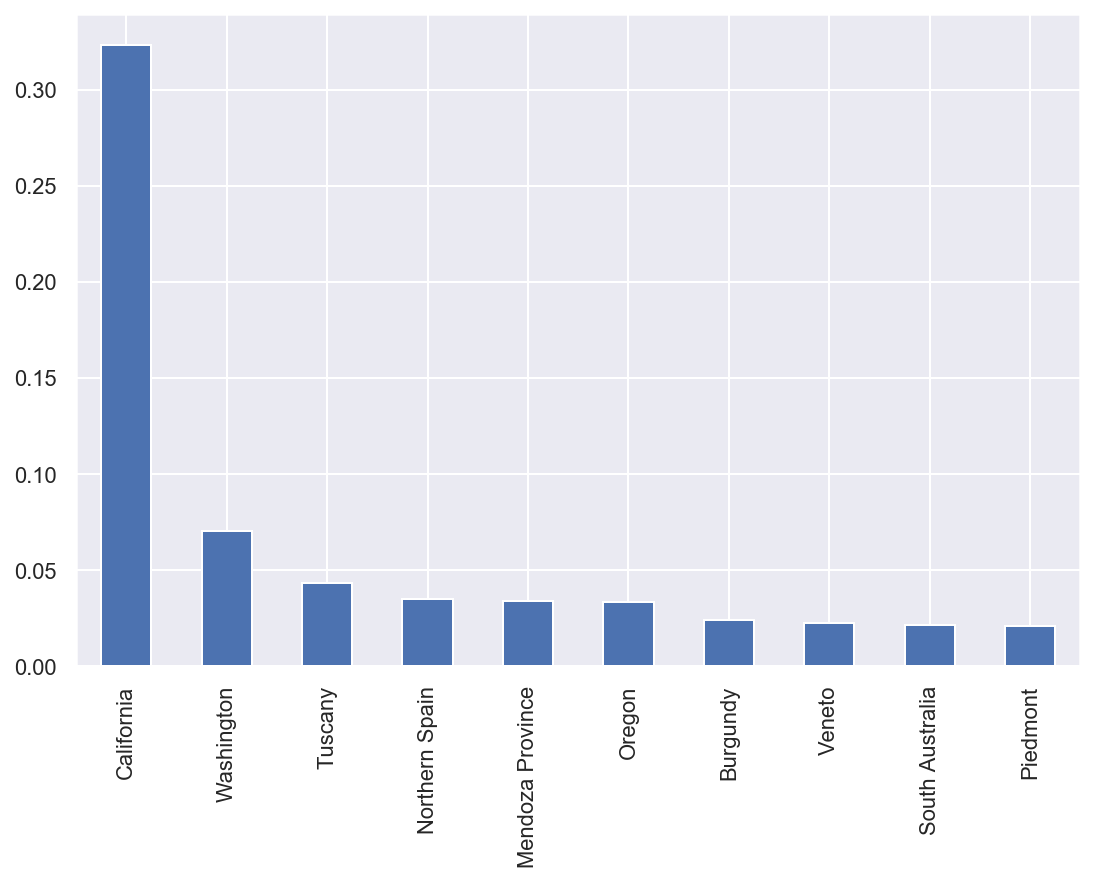
# Bar plot - percentage of reviews per top ten province

(reviews\_df['Provinces'].value\_counts().head(10) / len(reviews\_df)).plot.bar()

Wall time: 46.9 ms

Out[74]:

<matplotlib.axes.\_subplots.AxesSubplot at 0xe86cbc9ba8>



. . .

In [79]:



%%time

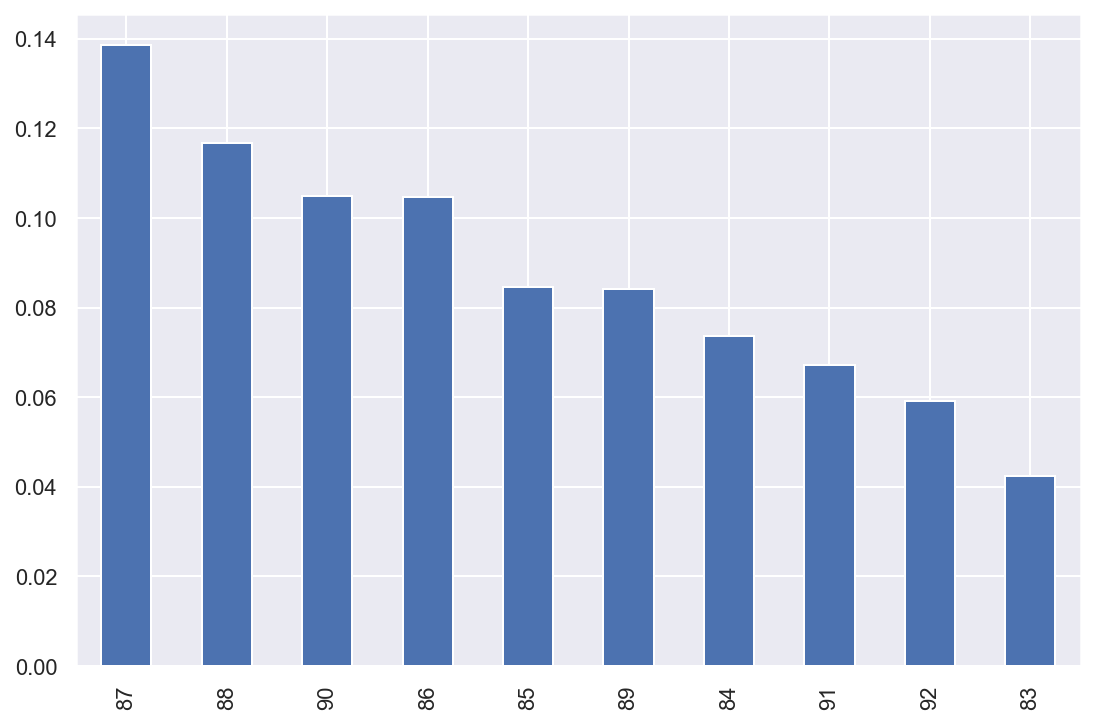
# Bar plot - Percentage of reviews per top ten provinces

(reviews\_df['Points'].value\_counts().head(10) / len(reviews\_df)).plot.bar()

Wall time: 31.3 ms

Out[79]:

<matplotlib.axes.\_subplots.AxesSubplot at 0xe86e14b550>



. . .

In [81]:



%%time

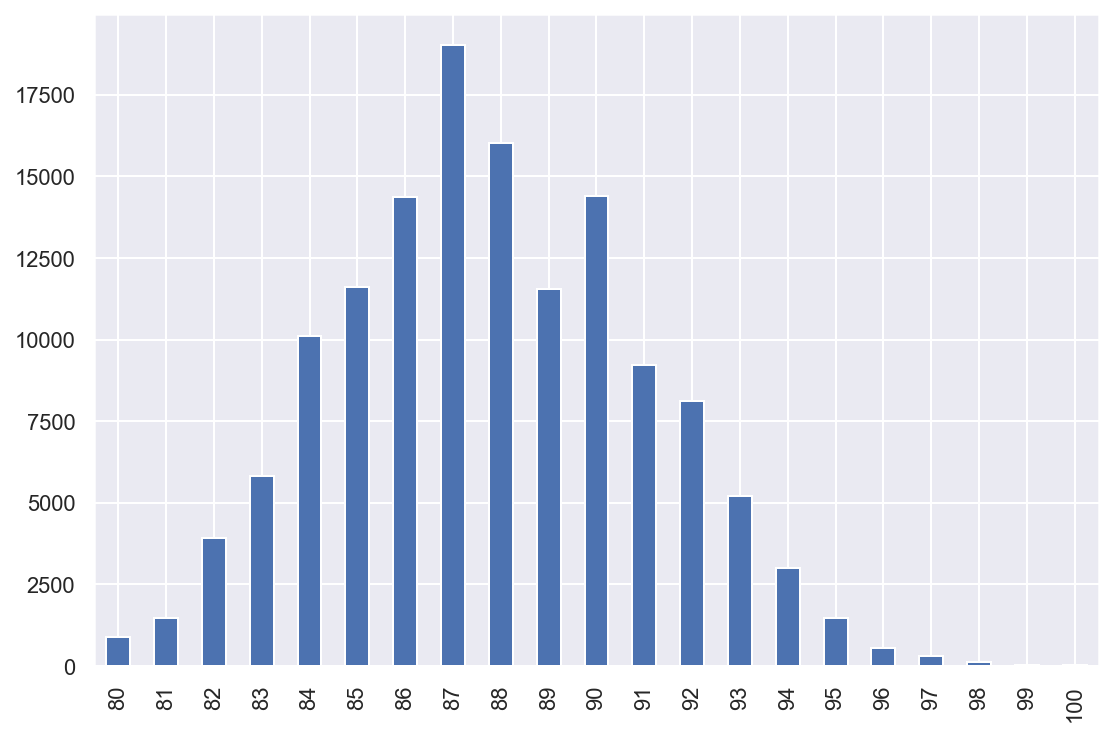
# Bar plot - points/reviews

reviews\_df['Points'].value\_counts().sort\_index().plot.bar()

Wall time: 46.9 ms

Out[81]:

<matplotlib.axes.\_subplots.AxesSubplot at 0xe86e1c56a0>



. . .

In [82]:



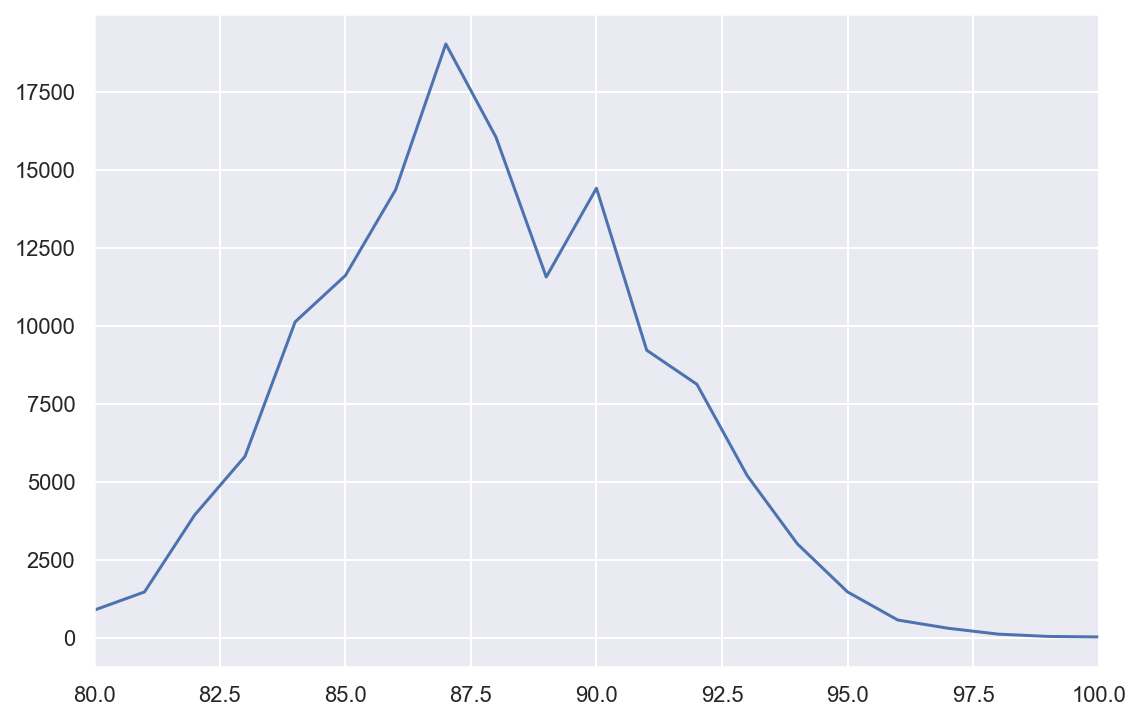
x

# Line plot of points/reviews

reviews\_df['Points'].value\_counts().sort\_index().plot.line()

Out[82]:

<matplotlib.axes.\_subplots.AxesSubplot at 0xe86e4b8860>



. . .

In [83]:



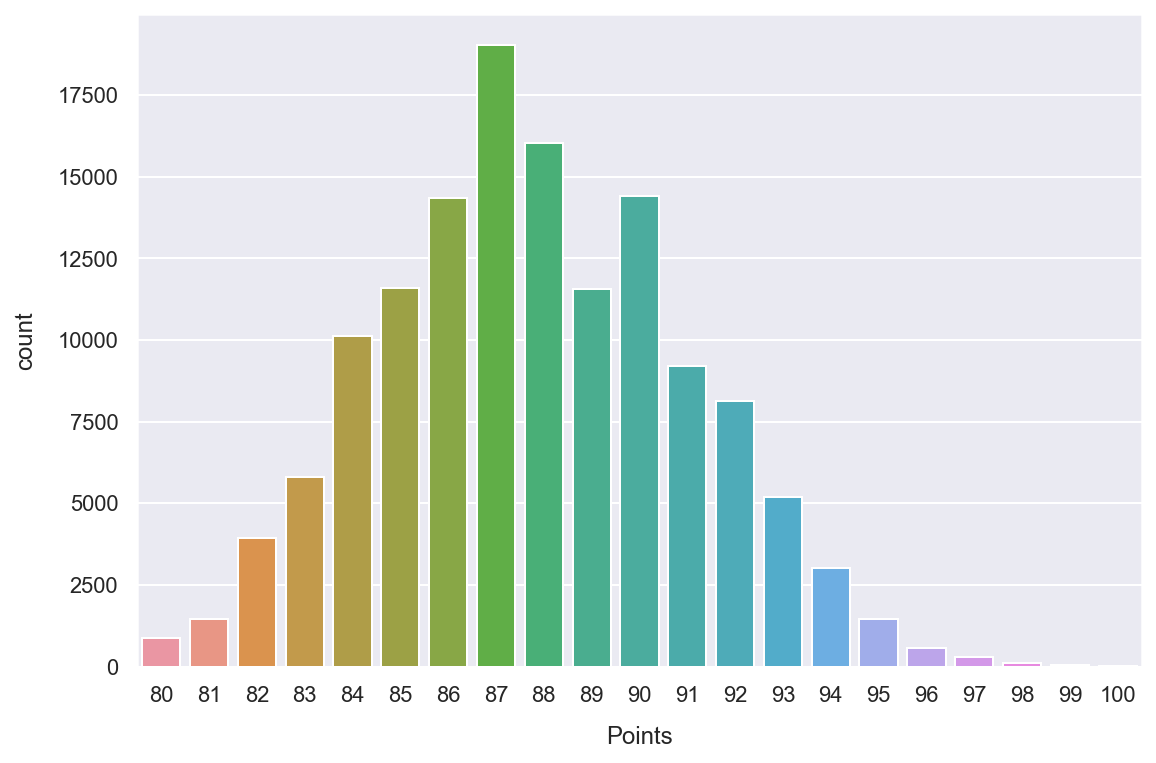
x

# countplot - Distribution of points and how many reviews

sns.countplot(reviews\_df['Points'])

Out[83]:

<matplotlib.axes.\_subplots.AxesSubplot at 0xe86e52c940>



. . .

In [88]:



%%time

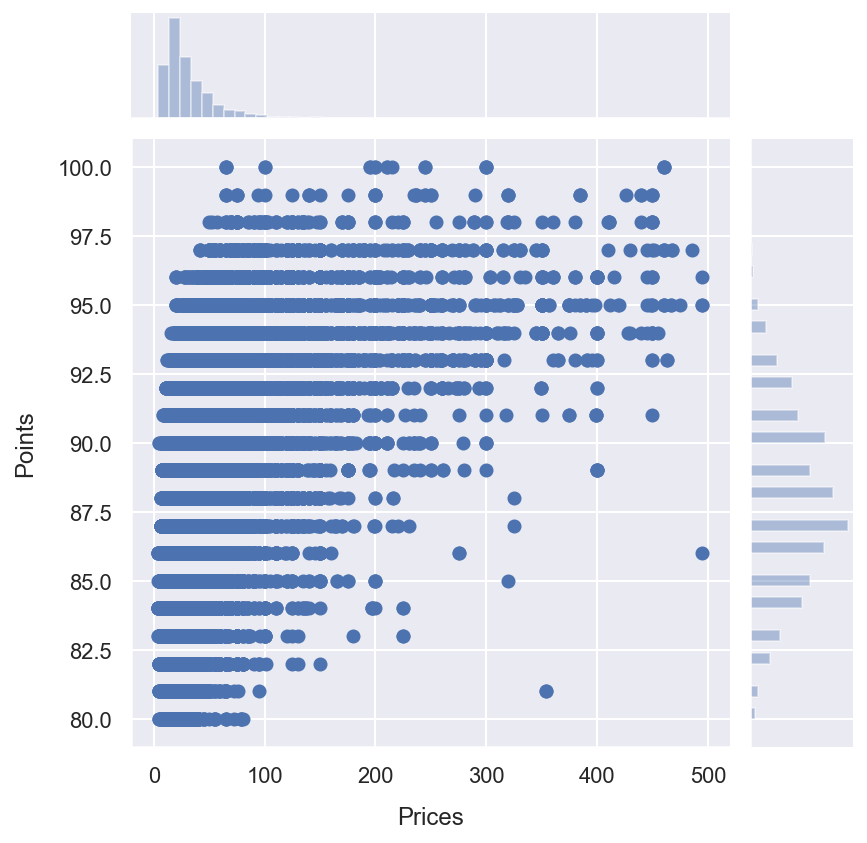
# Jointplot

sns.jointplot(x='Prices', y='Points', data=reviews\_df[reviews\_df['Prices'] < 500])

Wall time: 578 ms

Out[88]:

<seaborn.axisgrid.JointGrid at 0xe873d6f630>



. . .

In [90]:

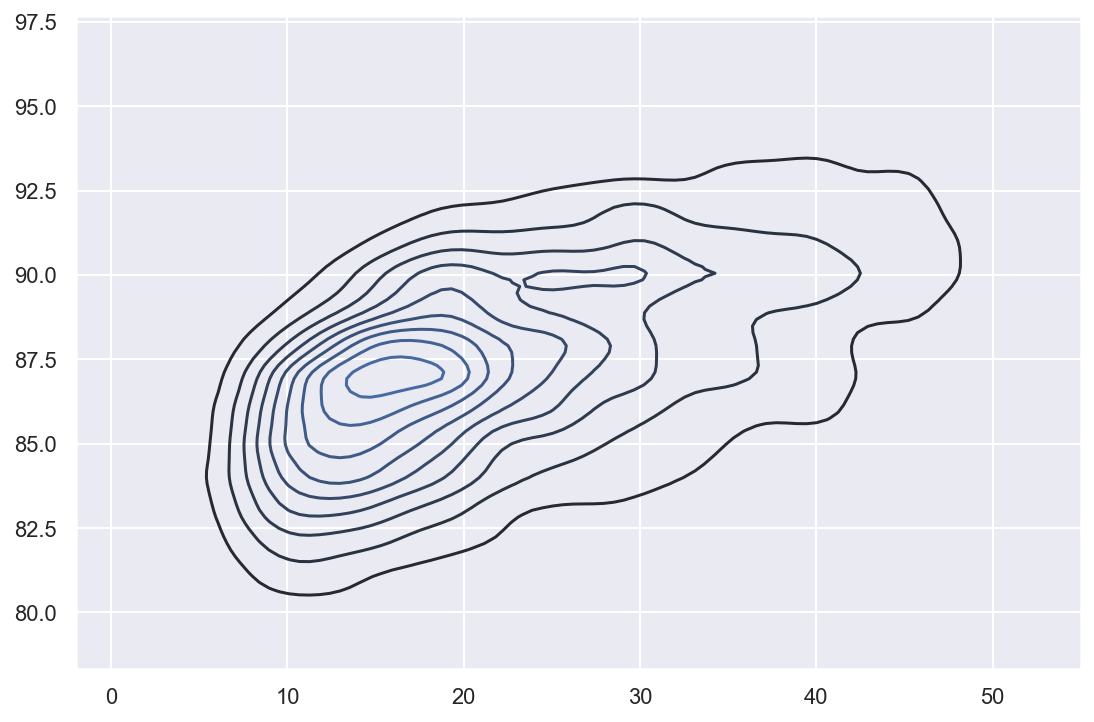


# plotted price and point value KDE chart

sns.kdeplot(reviews\_df[reviews\_df['Prices'] < 50].loc[:, ['Prices', 'Points']].dropna().sample(5000))

Out[90]:

<matplotlib.axes.\_subplots.AxesSubplot at 0xe873ebd9e8>



. . .



%%time

# Hex grid

​

sns.jointplot(x='Prices', y='Points', data=reviews\_df[reviews\_df['Prices'] < 100], kind=

            'hex',

                    gridsize=20)

Type Markdown and LaTeX: 𝛼2α2

In [94]:



x

%%time

# Hex grid

​

sns.jointplot(x='Prices', y='Points', data=reviews\_df[reviews\_df['Prices'] < 50], kind=

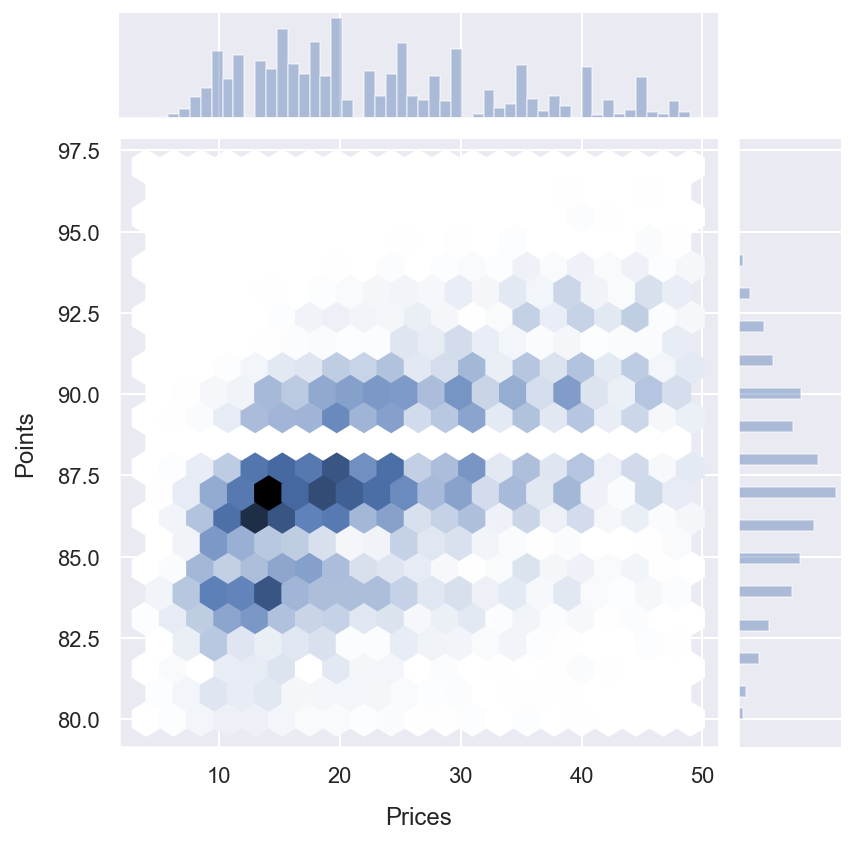
'hex',

gridsize=20)

Wall time: 375 ms

Out[94]:

<seaborn.axisgrid.JointGrid at 0xe875384048>



. . .

In [ ]:



​

. . .

[CloseExpandOpen in PagerClose](http://localhost:8888/notebooks/Data%20Analysis%20with%20winemag-data.ipynb)