In [1]:
```
import pandas as pd
pd.set_option("display.max_rows",5)
db = pd.read_csv("../Data/winemag-data_first150k.csv", index_col=0)
```

In [2]:
```
db.loc[db.country == "US"]
```

Out[2]:

	country	description	designation	points	price	province	region_1	region_2
0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	96	235.0	California	Napa Valley	Napa
2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	96	90.0	California	Knights Valley	Sonoma
...	...	...	...	...	...	...	...	...
150915	US	Decades ago, Beringer's then-winemaker Myron N...	Nightingale	93	30.0	California	North Coast	North Coast
150916	US	An impressive wine that presents a full bouque...	J. Schram	93	65.0	California	Napa Valley	Napa

62397 rows × 10 columns

In [3]:
```
db.loc[[2,4,5]]
```

Out[3]:

	country	description	designation	points	price	province	region_1	region_2	va
2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	96	90.0	California	Knights Valley	Sonoma	Sauv
4	France	This is the top wine from La Bégude, named aft...	La Brûlade	95	66.0	Provence	Bandol	NaN	Pro red
5	Spain	Deep, dense and pure from the opening bell, th...	Numanthia	95	73.0	Northern Spain	Toro	NaN	Tir

In [4]:
```
db.loc[db.country == "France"].points.median()
```

Out[4]:
```
89.0
```

What countries are represented in the dataset?

Use ' unique() '

In [5]:
```
db.country.unique()
```

Out[5]:
```
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)
```

How often does each country appear in the dataset?

Use ' value_counts() '

In [6]:
```
db.country.value_counts()
```

Out[6]:
```
country
US           62397
Italy        23478
             ...
Japan            2
US-France        1
Name: count, Length: 48, dtype: int64
```

Centered Price

In [7]:
```
db.price - db.price.mean()
```

Out[7]:
```
0         201.868518
1          76.868518
              ...
150928     18.868518
150929    -18.131482
Name: price, Length: 150930, dtype: float64
```

Sort by the quality

In [8]:
```
db.iloc[(db.points / db.price).sort_values(ascending= False).index[0]]
```

Out[8]:
```
country                                                    US
description    There's a lot going on in this Merlot, which i...
                                        ...
variety                                               Merlot
winery                                                Bandit
Name: 25645, Length: 10, dtype: object
```

2 Pd Series

There are only so many words you can use when describing a bottle of wine. Is a wine more likely to be "tropical" or "fruity"? Create a Series descriptor_counts counting how many times each of these two words appears in the description column in the dataset. (For simplicity, let's ignore the capitalized versions of these words.)

In [9]:
```
description_counts = pd.Series([db.description.map(lambda r : "tropical" in r).s
description_counts
```

Out[9]:
```
tropical    4135
fruity      8669
dtype: int64
```

3 Apply

We'd like to host these wine reviews on our website, but a rating system ranging from 80 to 100 points is too hard to understand - we'd like to translate them into simple star

ratings. A score of 95 or higher counts as 3 stars, a score of at least 85 but less than 95 is 2 stars. Any other score is 1 star.

Also, the Canadian Vintners Association bought a lot of ads on the site, so any wines from Canada should automatically get 3 stars, regardless of points.

Create a ser `star_ratings` ngs with the number of stars corresponding to each review in the dataset.

In [10]:
```
def give_star(r):
    if r.country == "Canada":
        return 3
    elif r.points >= 95:
        return 3
    elif r.points >= 85:
        return 2
    else:
        return 1

star_ratings = db.apply(give_star, axis = "columns")
```

Grouping and Sorting

In [11]:
```
db = pd.read_csv("../Data/winemag-data-130k-v2.csv")
db.head()
```

Out[11]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	
0	0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	
1	1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	
2	2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	W
3	3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	
4	4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	W

1

Who are the most common wine reviewers in the dataset? Create a Series whose index is the taster_twitter_handle category from the dataset, and whose values count how many reviews each person wrote.

In [12]:
```
reviews_written = db.groupby("taster_twitter_handle").points.count()
reviews_written
```

Out[12]:
```
taster_twitter_handle
@AnneInVino        3685
@JoeCz             5147
                   ...
@winewchristina       6
@worldwineguys     1005
Name: points, Length: 15, dtype: int64
```

In [13]:
```
db.groupby("price").points.max().sort_values()
```

Out[13]:
```
price
4.0       86
5.0       87
         ...
460.0    100
150.0    100
Name: points, Length: 390, dtype: int64
```

Agg

What are the minimum and maximum prices for each variety of wine? Create a DataFrame whose index is the variety category from the dataset and whose values are the min and max values thereof.

In [14]:
```
price_extremes = db.groupby("variety").points.agg(["max","min"])
price_extremes
```

Out[14]:

variety	max	min
Abouriou	91	85
Agiorgitiko	92	83
...	...	...
Çalkarası	87	86
Žilavka	88	88

707 rows × 2 columns

In [15]:
```
price_extremes.sort_values(by = ["min","max"], ascending = False).iloc[2:19]
```

Out[15]:

variety	max	min
Tinta del Pais	96	94
Riesling-Chardonnay	94	94
...	...	...
Pignolo	92	92
Sauvignon Blanc-Assyrtiko	92	92

17 rows × 2 columns

In [16]:
```
db_mean_ratings = db.groupby("taster_name").points.mean()
```

In [17]:
```
db_mean_ratings.describe()
```

Out[17]:
```
count    19.000000
mean     88.233026
           ...
75%      88.975256
max      90.562551
Name: points, Length: 8, dtype: float64
```

What combination of countries and varieties are most common? Create a Series whose index is a MultiIndexof {country, variety} pairs. For example, a pinot noir produced in the US should map to {"US", "Pinot Noir"}. Sort the values in the Series in descending order based on wine count.

In [18]:
```
country_variety_counts = db.groupby(['country', 'variety']).points.mean().sort_v
country_variety_counts
```

Out[18]:
```
country    variety
Australia  Cabernet-Shiraz    96.0
Spain      Tinta del Pais     95.0
                              ...
Mexico     Cinsault           80.0
Peru       Sparkling Blend    80.0
Name: points, Length: 1612, dtype: float64
```

Data Types and Missing Values

In [19]:
```
db = pd.read_csv("../Data/winemag-data-130k-v2.csv")
```

Get the data type

In [20]:
```
db.points.dtype
```

Out[20]:
```
dtype('int64')
```

Create a Series from entries in the `points` column, but convert the entries to strings.

In [21]:
```
str_points = db.points.astype(str)
```

In [22]:
```
str_points.dtype
```

Out[22]:
```
dtype('O')
```

In [23]:
```
str_points
```

Out[23]:
```
0         87
1         87
          ..
129969    90
129970    90
Name: points, Length: 129971, dtype: object
```

NULL

Sometimes the price column is `null`. How many reviews in the dataset are missing a price?

In [24]:
```
pd.isnull(db.price).sum()
```

Out[24]: 8996

In [25]:
```
db.price.isnull().sum()
```

Out[25]: 8996

Fill NULL

What are the most common wine-producing regions? Create a Series counting the number of times each value occurs in the `region_1` field. This field is often missing data, so replace missing values with `Unknown`. Sort in descending order.int64

In [26]:
```
db.region_1.fillna("Unknown").value_counts()
```

Out[26]:
```
region_1
Unknown                    21247
Napa Valley                 4480
                           ...
Vin Santo di Carmignano        1
Paestum                        1
Name: count, Length: 1230, dtype: int64
```

Rename

In [34]:
```
db = pd.read_csv("../Data/winemag-data-130k-v2.csv",index_col = 0)
rename = db.rename(columns = {"region_1":"region", "region_2":"locale"})
rename.head()
```

Out[34]:

	country	description	designation	points	price	province	region	locale	ta
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	P
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	
4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	Willamette Valley	P

In [35]:
```
db.head()
```

Out[35]:

	country	description	designation	points	price	province	region_1	region_2	ta
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	P
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	
4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	Willamette Valley	P

Reindexed

In [36]:
```
reindexed = db.rename_axis("wines", axis = "rows")
```

In [37]:
```
reindexed
```

Out[37]:

wines	country	description	designation	points	price	province	region_1	region_2
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN
...	...	...	...	...	...	...	...	...
129969	France	A dry style of Pinot Gris, this is crisp with ...	NaN	90	32.0	Alsace	Alsace	NaN
129970	France	Big, rich and off-dry, this is powered by inte...	Lieu-dit Harth Cuvée Caroline	90	21.0	Alsace	Alsace	NaN

129971 rows × 13 columns

Merge

CONCAT

In [40]:
```
gaming_products = pd.read_csv("../Data/top-things/top-things/reddits/g/gaming.cs
gaming_products['subreddit'] = "r/gaming"
movie_products = pd.read_csv("../Data/top-things/top-things/reddits/m/movies.csv
movie_products['subreddit'] = "r/movies"
```

In [42]:
```
combined_products = pd.concat([gaming_products, movie_products])
combined_products
```

Out[42]:

	name	category	amazon_link	total_mentions	subred
0	BOOMco Halo Covenant Needler Blaster	Toys & Games	https://www.amazon.com/BOOMco-Halo-Covenant-Ne...	4.0	
1	Raspberry PI 3 Model B 1.2GHz 64-bit quad-core...	Electronics	https://www.amazon.com/Raspberry-Model-A1-2GHz...	19.0	
...	...	...	...	...	
301	Apocalypto [Blu-ray]	Movies & TV	https://www.amazon.com/Apocalypto-Blu-ray-Rudy...	1.0	
302	Cinelinx: A Card Game for People Who Love Movi...	Toys & Games	https://www.amazon.com/Cinelinx-Card-Game-Peop...	1.0	

796 rows × 6 columns

JOIN

In [44]:
```
powerlifting_meets = pd.read_csv("../Data/meets.csv")
powerlifting_competitors = pd.read_csv("../Data/openpowerlifting.csv")
```

In [46]:
```
powerlifting_competitors.head()
```

Out[46]:

	MeetID	Name	Sex	Equipment	Age	Division	BodyweightKg	WeightClassKg	Sq
0	0	Angie Belk Terry	F	Wraps	47.0	Mst 45-49	59.60	60	
1	0	Dawn Bogart	F	Single-ply	42.0	Mst 40-44	58.51	60	
2	0	Dawn Bogart	F	Single-ply	42.0	Open Senior	58.51	60	
3	0	Dawn Bogart	F	Raw	42.0	Open Senior	58.51	60	
4	0	Destiny Dula	F	Raw	18.0	Teen 18-19	63.68	67.5	

In [47]:
```
powerlifting_meets.head()
```

Out[47]:

	MeetID	MeetPath	Federation	Date	MeetCountry	MeetState	MeetTown	M
0	0	365strong/1601	365Strong	2016-10-29	USA	NC	Charlotte	2 P
1	1	365strong/1602	365Strong	2016-11-19	USA	MO	Ozark	Th P
2	2	365strong/1603	365Strong	2016-07-09	USA	NC	Charlotte	
3	3	365strong/1604	365Strong	2016-06-11	USA	SC	Rock Hill	Ca
4	4	365strong/1605	365Strong	2016-04-10	USA	SC	Rock Hill	E

In [48]:
```
powerlifting_combined = powerlifting_meets.set_index("MeetID").join(powerlifting
```

In [49]:
```
powerlifting_combined
```

Out[49]:

MeetID	MeetPath	Federation	Date	MeetCountry	MeetState	MeetTown	MeetI
0	365strong/1601	365Strong	2016-10-29	USA	NC	Charlotte	2016 & S Na Power
0	365strong/1601	365Strong	2016-10-29	USA	NC	Charlotte	2016 & S Na Power
...	...	...	...	...	...	...	
8481	xpc/2017-finals	XPC	2017-03-03	USA	OH	Columbus	201
8481	xpc/2017-finals	XPC	2017-03-03	USA	OH	Columbus	201

386414 rows × 23 columns