Project 10 -- SEYI OGUNMODEDE

INSTRUCTOR: Dr. Ward

- Help with figuring out how to write a function.
- Help how to analyse through visualisation

TA'S HELP: Jeong, Ik Su

- Help with figuring out on how to plot my graphs

Question 1

In [1]:
```
# Bar chart can be stacked or grouped.
# This involves three varibles to be compared together.
# It must have discrete value to enable ploting
# Colour coding (hue) good for distinguishing data from various groups
# Histogram shows the distribution of a set of Data
# To draw a histogram, the data are grouped into bins or intervals
# Visualisation provides insight that canot be appreciated
# by any other approach to learning from data.
# A large amount of quantitative information can be packed into a small region.
```

Markdown notes and sentences and analysis written here.

Question 2

In [1]:
```
import pandas as pd
```

In [2]:
```
pd.read_csv("/anvil/projects/tdm/data/beer/beers.csv")
```

Out[2]:

	id	name	brewery_id	state	country	style	availability	abv	notes	retire
0	202522	Olde Cogitator	2199	CA	US	English Oatmeal Stout	Rotating	7.3	No notes at this time.	
1	82352	Konrads Stout Russian Imperial Stout	18604	NaN	NO	Russian Imperial Stout	Rotating	10.4	No notes at this time.	
2	214879	Scottish Right	44306	IN	US	Scottish Ale	Year-round	4.0	No notes at this time.	
3	320009	MegaMeow Imperial Stout	4378	WA	US	American Imperial Stout	Winter	8.7	Every time this year	
4	246438	Peaches-N-Cream	44617	PA	US	American Cream Ale	Rotating	5.1	No notes at this time.	
...	...	...	...	...	...	...	...	...	...	
358868	267703	Collective Project: Gose	32763	ON	CA	Leipzig Gose	Limited (brewed once)	5.0	Our Gose is an unfiltered wheat beer made with...	
358869	300013	Tripel	50238	NaN	BE	Belgian Tripel	Year-round	8.0	No notes at this time.	
358870	187618	RIPTA	34665	RI	US	Belgian Tripel	Rotating	9.5	No notes at this time.	
358871	283124	Rumble Fish	29238	MI	US	American Imperial IPA	Rotating	8.3	No notes at this time.	
358872	267484	Joie De Vivre	25032	CA	US	Belgian Saison	Limited (brewed once)	6.6	No notes at this time.	

358873 rows × 10 columns

In [3]:
```
mybeer=pd.read_csv("/anvil/projects/tdm/data/beer/beers.csv")
```

In [5]:
```
mybeer.head()
```

Out[5]:

	id	name	brewery_id	state	country	style	availability	abv	notes	retired
0	202522	Olde Cogitator	2199	CA	US	English Oatmeal Stout	Rotating	7.3	No notes at this time.	f
1	82352	Konrads Stout Russian Imperial Stout	18604	NaN	NO	Russian Imperial Stout	Rotating	10.4	No notes at this time.	f
2	214879	Scottish Right	44306	IN	US	Scottish Ale	Year-round	4.0	No notes at this time.	t
3	320009	MegaMeow Imperial Stout	4378	WA	US	American Imperial Stout	Winter	8.7	Every time this year	f
4	246438	Peaches-N-Cream	44617	PA	US	American Cream Ale	Rotating	5.1	No notes at this time.	f

In [6]:
```
mybeer.tail()
```

Out[6]:

	id	name	brewery_id	state	country	style	availability	abv	notes	retired
358868	267703	Collective Project: Gose	32763	ON	CA	Leipzig Gose	Limited (brewed once)	5.0	Our Gose is an unfiltered wheat beer made with...	t
358869	300013	Tripel	50238	NaN	BE	Belgian Tripel	Year-round	8.0	No notes at this time.	f
358870	187618	RIPTA	34665	RI	US	Belgian Tripel	Rotating	9.5	No notes at this time.	f
358871	283124	Rumble Fish	29238	MI	US	American Imperial IPA	Rotating	8.3	No notes at this time.	f
358872	267484	Joie De Vivre	25032	CA	US	Belgian Saison	Limited (brewed once)	6.6	No notes at this time.	t

In [7]:
```
mybeer.columns
```

Out[7]:
```
Index(['id', 'name', 'brewery_id', 'state', 'country', 'style', 'availability',
       'abv', 'notes', 'retired'],
      dtype='object')
```

In [8]:
```
# 1. considering the country, style and their respictive occurrence using bar chart

# 2. looking at the retired column to the abv column using box plot. Also, Abv to avai

# 3. Using group by to relate availability to those that have retired using Stacked ba
```

Markdown notes and sentences and analysis written here.

Question 3

In [9]:
```
import pandas as pd
```

In [10]:
```
pd.read_csv("/anvil/projects/tdm/data/beer/beers.csv")
```

Out[10]:

	id	name	brewery_id	state	country	style	availability	abv	notes	retire
0	202522	Olde Cogitator	2199	CA	US	English Oatmeal Stout	Rotating	7.3	No notes at this time.	
1	82352	Konrads Stout Russian Imperial Stout	18604	NaN	NO	Russian Imperial Stout	Rotating	10.4	No notes at this time.	
2	214879	Scottish Right	44306	IN	US	Scottish Ale	Year-round	4.0	No notes at this time.	
3	320009	MegaMeow Imperial Stout	4378	WA	US	American Imperial Stout	Winter	8.7	Every time this year	
4	246438	Peaches-N-Cream	44617	PA	US	American Cream Ale	Rotating	5.1	No notes at this time.	
...	...	...	...	...	...	...	...	...	...	
358868	267703	Collective Project: Gose	32763	ON	CA	Leipzig Gose	Limited (brewed once)	5.0	Our Gose is an unfiltered wheat beer made with...	
358869	300013	Tripel	50238	NaN	BE	Belgian Tripel	Year-round	8.0	No notes at this time.	
358870	187618	RIPTA	34665	RI	US	Belgian Tripel	Rotating	9.5	No notes at this time.	
358871	283124	Rumble Fish	29238	MI	US	American Imperial IPA	Rotating	8.3	No notes at this time.	
358872	267484	Joie De Vivre	25032	CA	US	Belgian Saison	Limited (brewed once)	6.6	No notes at this time.	

358873 rows × 10 columns

In [11]:
```
mydf=pd.read_csv("/anvil/projects/tdm/data/beer/beers.csv")
```

In [12]:
```
mydf.head()
```

Out[12]:

	id	name	brewery_id	state	country	style	availability	abv	notes	retired
0	202522	Olde Cogitator	2199	CA	US	English Oatmeal Stout	Rotating	7.3	No notes at this time.	f
1	82352	Konrads Stout Russian Imperial Stout	18604	NaN	NO	Russian Imperial Stout	Rotating	10.4	No notes at this time.	f
2	214879	Scottish Right	44306	IN	US	Scottish Ale	Year-round	4.0	No notes at this time.	t
3	320009	MegaMeow Imperial Stout	4378	WA	US	American Imperial Stout	Winter	8.7	Every time this year	f
4	246438	Peaches-N-Cream	44617	PA	US	American Cream Ale	Rotating	5.1	No notes at this time.	f

In [13]:
```
# To rename my column country to country code

mydf.rename(columns={"country":"country_code"})
```

Out[13]:

	id	name	brewery_id	state	country_code	style	availability	abv	notes
0	202522	Olde Cogitator	2199	CA	US	English Oatmeal Stout	Rotating	7.3	No notes at this time.
1	82352	Konrads Stout Russian Imperial Stout	18604	NaN	NO	Russian Imperial Stout	Rotating	10.4	No notes at this time.
2	214879	Scottish Right	44306	IN	US	Scottish Ale	Year-round	4.0	No notes at this time.
3	320009	MegaMeow Imperial Stout	4378	WA	US	American Imperial Stout	Winter	8.7	Every time this year
4	246438	Peaches-N-Cream	44617	PA	US	American Cream Ale	Rotating	5.1	No notes at this time.
...	...	...	...	...	...	...	...	...	...
358868	267703	Collective Project: Gose	32763	ON	CA	Leipzig Gose	Limited (brewed once)	5.0	Our Gose is an unfiltered wheat beer made with...
358869	300013	Tripel	50238	NaN	BE	Belgian Tripel	Year-round	8.0	No notes at this time.
358870	187618	RIPTA	34665	RI	US	Belgian Tripel	Rotating	9.5	No notes at this time.
358871	283124	Rumble Fish	29238	MI	US	American Imperial IPA	Rotating	8.3	No notes at this time.
358872	267484	Joie De Vivre	25032	CA	US	Belgian Saison	Limited (brewed once)	6.6	No notes at this time.

358873 rows × 10 columns

In [14]:
```
mydf=mydf.rename(columns={"country":"country_code"})
```

In [15]:
```
import pycountry
```

In [16]:
```
mydf.head()
```

Out[16]:

	id	name	brewery_id	state	country_code	style	availability	abv	notes	retired
0	202522	Olde Cogitator	2199	CA	US	English Oatmeal Stout	Rotating	7.3	No notes at this time.	f
1	82352	Konrads Stout Russian Imperial Stout	18604	NaN	NO	Russian Imperial Stout	Rotating	10.4	No notes at this time.	f
2	214879	Scottish Right	44306	IN	US	Scottish Ale	Year-round	4.0	No notes at this time.	t
3	320009	MegaMeow Imperial Stout	4378	WA	US	American Imperial Stout	Winter	8.7	Every time this year	f
4	246438	Peaches-N-Cream	44617	PA	US	American Cream Ale	Rotating	5.1	No notes at this time.	f

In [17]:
```
# Extract the fisrt 5 conutry code

mydf.head()["country_code"]
```

Out[17]:
```
0    US
1    NO
2    US
3    US
4    US
Name: country_code, dtype: object
```

In [18]:
```
mydf.head(20)["country_code"]
```

Out[18]:
```
0     US
1     NO
2     US
3     US
4     US
5     JP
6     US
7     IT
8     US
9     CA
10    AU
11    US
12    US
13    CA
14    US
15    US
16    US
17    US
18    SE
19    US
Name: country_code, dtype: object
```

In [19]:
```
# To get the countries full names
# Diverse info will pop up


pycountry.countries.get(alpha_2="US")
```

Out[19]:
```
Country(alpha_2='US', alpha_3='USA', flag='US', name='United States', numeric='840', official_name='United States of America')
```

In [20]:
```
# we can just get the name alone

pycountry.countries.get(alpha_2="US").name
```

Out[20]:
```
'United States'
```

In [21]:
```
pycountry.countries.get(alpha_2="NO").name
```

Out[21]:
```
'Norway'
```

In [22]:
```
pycountry.countries.get(alpha_2="JP").name
```

Out[22]:
```
'Japan'
```

In [23]:
```
pycountry.countries.get(alpha_2="IT").name
```

Out[23]:
```
'Italy'
```

In [24]:
```
# Then we can go through all of the countries in the head of the country code column
# To pull out what those codes are

[mycountry for mycountry in mydf.head()["country_code"]]
```

Out[24]:
```
['US', 'NO', 'US', 'US', 'US']
```

In [25]:
```
#Then instead of printing the contry itself,
# we can just use the techniques from above
```

```
# That is the first five full country names

[pycountry.countries.get(alpha_2=mycountry).name for mycountry in mydf.head()["country
```

Out[25]:
```
['United States', 'Norway', 'United States', 'United States', 'United States']
```

In [26]:
```
# let us look for the first 30 country full names
# you can do it for number 50, 100,200

[pycountry.countries.get(alpha_2=mycountry).name for mycountry in mydf.head(30)["count
```

Out[26]:
```
['United States',
 'Norway',
 'United States',
 'United States',
 'United States',
 'Japan',
 'United States',
 'Italy',
 'United States',
 'Canada',
 'Australia',
 'United States',
 'United States',
 'Canada',
 'United States',
 'United States',
 'United States',
 'United States',
 'Sweden',
 'United States',
 'United States',
 'Puerto Rico',
 'Poland',
 'United States',
 'United States',
 'Costa Rica',
 'United States',
 'United States',
 'United States',
 'United States']
```

In [27]:
```
# But if you do it for 201 there will be an error

[pycountry.countries.get(alpha_2=mycountry).name for mycountry in mydf.head(201)["cour
```

```
---------------------------------------------------------------------------
LookupError                               Traceback (most recent call last)
Input In [27], in <cell line: 3>()
      1 # But if you do it for 201 there will be an error
----> 3 [pycountry.countries.get(alpha_2=mycountry).name for mycountry in mydf.head(2
01)["country_code"]]

Input In [27], in <listcomp>(.0)
      1 # But if you do it for 201 there will be an error
----> 3 [pycountry.countries.get(alpha_2=mycountry).name for mycountry in mydf.head(2
01)["country_code"]]

File /usr/local/lib/python3.10/site-packages/pycountry/db.py:39, in lazy_load.<locals
>.load_if_needed(self, *args, **kw)
     37     with self._load_lock:
     38         self._load()
---> 39 return f(self, *args, **kw)

File /usr/local/lib/python3.10/site-packages/pycountry/db.py:111, in Database.get(sel
f, **kw)
    109 field, value = kw.popitem()
    110 if not isinstance(value, str):
--> 111     raise LookupError()
    112 # Normalize for case-insensitivity
    113 value = value.lower()

LookupError:
```

In [28]:

```
# Let us look at our first 201 DataFrame
# for 201st row there is an NAN on the country code
# Then we need to go in and get rid of the NAN in just the country code
# You don"t want to get rid of them together because,
# contry that are not US we have an NAN in their state

mydf.head(201)
```

Out[28]:

	id	name	brewery_id	state	country_code	style	availability	abv	notes	retired
0	202522	Olde Cogitator	2199	CA	US	English Oatmeal Stout	Rotating	7.3	No notes at this time.	f
1	82352	Konrads Stout Russian Imperial Stout	18604	NaN	NO	Russian Imperial Stout	Rotating	10.4	No notes at this time.	f
2	214879	Scottish Right	44306	IN	US	Scottish Ale	Year-round	4.0	No notes at this time.	t
3	320009	MegaMeow Imperial Stout	4378	WA	US	American Imperial Stout	Winter	8.7	Every time this year	f
4	246438	Peaches-N-Cream	44617	PA	US	American Cream Ale	Rotating	5.1	No notes at this time.	f
...	...	...	...	...	...	...	...	...	...	...
196	201533	Dark Cloud	30249	FL	US	American Brown Ale	Limited (brewed once)	4.7	No notes at this time.	t
197	104273	Badetiss	31326	MN	US	American Blonde Ale	Rotating	3.5	No notes at this time.	f
198	251232	Altbier	36639	GB2	GB	German Altbier	Rotating	5.2	No notes at this time.	f
199	42880	Barley's Bombshell Blonde	1513	OH	US	German Helles	Spring	4.5	No notes at this time.	f
200	145165	Wasatch Switch	28908	NaN	NaN	American Wild Ale	Limited (brewed once)	8.0	No notes at this time.	t

In [29]:
```
# But the NAN in the country code we want to drop them
# And keep track of the numbers we are dropping

mydf.shape

# There are 358873 rows all together.
```

Out[29]:
```
(358873, 10)
```

In [30]:
```
# look at the country code columns to see how many are NAs
#There are 154 that are NAs
# we are only going to loose 154 entries if we drop the ones that are NAs

sum(mydf["country_code"].isna())
```

Out[30]:
```
154
```

In [31]:
```
# Make a new DataFrame where we throw away those that were Nas
# The syntax code mydf["country_code"].isna() means finding the ones that were NAs
# ~mydf["country_code"].isna() means let get those that are not NAs
# the output are for those that are not NAs

mynewdf = mydf[~mydf["country_code"].isna()]
```

In [32]:
```
# The shape of our new DataFrame
# It is almost thesame as the shape of our old DataFrame

mynewdf.shape
```

Out[32]:
```
(358719, 10)
```

In [33]:
```
# we will go back to our code above
# we will now use our new DataFrame
# it gives me all of the country names for all of the rows in my new data frame

[pycountry.countries.get(alpha_2=mycountry).name for mycountry in mynewdf["country_coc
```

Out[33]:
```
['United States',
 'Norway',
 'United States',
 'United States',
 'United States',
 'Japan',
 'United States',
 'Italy',
 'United States',
 'Canada',
 'Australia',
 'United States',
 'United States',
 'Canada',
 'United States',
 'United States',
 'United States',
 'United States',
 'Sweden',
 'United States',
 'United States',
 'Puerto Rico',
 'Poland',
 'United States',
 'United States',
 'Costa Rica',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'Poland',
 'United States',
 'Germany',
 'United States',
 'United States',
 'Belgium',
 'United States',
 'United States',
 'Australia',
 'United States',
 'Canada',
 'United States',
 'United States',
 'United States',
 'Brazil',
 'Canada',
 'United States',
 'United States',
 'United Kingdom',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'Spain',
 'Russian Federation',
 'Canada',
 'United States',
 'United States',
```

```
'United States',
'Belgium',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Sweden',
'United States',
'United States',
'United States',
'United States',
'Spain',
'United States',
'United States',
'United States',
'United States',
'Ireland',
'United States',
'Belgium',
'United States',
'United States',
'Canada',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Canada',
'Denmark',
'United States',
'Denmark',
'Finland',
'United States',
'Canada',
'United States',
'United States',
'United Kingdom',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
```

```
'United States',
'United States',
'United States',
'United States',
'Canada',
'United States',
'United States',
'United States',
'Netherlands',
'United States',
'United States',
'Australia',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Canada',
'United Kingdom',
'United States',
'United States',
'Poland',
'United States',
'Austria',
'United States',
'United Kingdom',
'Sweden',
'Ireland',
'United States',
'United States',
'United States',
'United States',
'Netherlands',
'United States',
'Sweden',
'United States',
'United States',
'India',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United Kingdom',
'Germany',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Canada',
'United Kingdom',
'United States',
'Canada',
'United States',
'United States',
'France',
```

```
'United States',
'Canada',
'United States',
'Estonia',
'United States',
'United States',
'Italy',
'United States',
'United States',
'United States',
'Japan',
'United States',
'Belgium',
'United States',
'United States',
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'United States',
'Canada',
'United States',
'Poland',
'Brazil',
'Denmark',
'Canada',
'United States',
'United States',
'United States',
'United States',
'Mexico',
'United States',
'United States',
'United States',
'United States',
'United States',
'Canada',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Armenia',
'United States',
'Germany',
'Poland',
'United States',
'Canada',
'Canada',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
```

```
'United States',
'Sweden',
'United States',
'United States',
'Germany',
'United States',
'United States',
'United States',
'United States',
'Portugal',
'Romania',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Ireland',
'United Kingdom',
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'Italy',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Argentina',
'United States',
'United States',
'Japan',
'United States',
'United States',
'Denmark',
'United States',
'Netherlands',
'United States',
'United Kingdom',
```

```
'Germany',
'United States',
'United States',
'Sweden',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Netherlands',
'United States',
'Germany',
'United States',
'United States',
'Czechia',
'United States',
'United States',
'United States',
'United States',
'Spain',
'Latvia',
'Canada',
'New Zealand',
'United States',
'United States',
'Germany',
'United States',
'United States',
'United States',
'United States',
'Canada',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Canada',
'United States',
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'United States',
'United States',
'Belgium',
```

```
'Canada',
'United States',
'United States',
'Denmark',
'United States',
'United States',
'Sweden',
'Canada',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Russian Federation',
'Czechia',
'Canada',
'United States',
'United States',
'Paraguay',
'United States',
'United States',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Thailand',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Canada',
'Argentina',
'Czechia',
'United States',
'Denmark',
'Germany',
'United States',
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'United States',
'Finland',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Canada',
'United States',
'Norway',
```

```
'United States',
'United States',
'Canada',
'Canada',
'United States',
'United States',
'Canada',
'United States',
'Belgium',
'United States',
'United States',
'United States',
'Canada',
'United States',
'United States',
'Italy',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Sweden',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Germany',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'United States',
'United States',
'Sweden',
'United States',
'United States',
'Belgium',
'United States',
```

```
'Canada',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'Netherlands',
'United States',
'United Kingdom',
'United States',
'Ireland',
'Denmark',
'Ireland',
'United States',
'United States',
'United States',
'Chile',
'United States',
'United States',
'United States',
'United States',
'France',
'United States',
'United States',
'United States',
'United States',
'Switzerland',
'United States',
'United States',
'United States',
'New Zealand',
'United States',
'United States',
'United States',
'United States',
'United States',
'United Kingdom',
'Germany',
'United States',
'United States',
'United States',
'United States',
'United States',
'Belgium',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Denmark',
'United States',
'United States',
'United States',
```

```
'United States',
'United States',
'Ukraine',
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Ireland',
'United States',
'Russian Federation',
'United States',
'United States',
'United States',
'United States',
'Finland',
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'China',
'United States',
'United States',
'United States',
'Belgium',
'United States',
'United States',
'United States',
'United States',
'Germany',
'Canada',
'United States',
'Czechia',
'United States',
'United States',
'Canada',
'United States',
'Canada',
'United States',
'Poland',
'United States',
'Kazakhstan',
'Mexico',
'Australia',
'United States',
'United States',
'United Kingdom',
```

```
'United States',
'France',
'United States',
'Russian Federation',
'United States',
'Belgium',
'United States',
'Canada',
'United States',
'United States',
'United States',
'France',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Sweden',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Belgium',
'Argentina',
'United States',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'Romania',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Denmark',
'Australia',
'United States',
'Lithuania',
'Hungary',
'United States',
'United States',
```

```
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'United States',
'United States',
'Belgium',
'United States',
'United States',
'Belgium',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Germany',
'United States',
'United States',
'Canada',
'United States',
'United States',
'United States',
'United States',
'Germany',
'United States',
'United States',
'United States',
'United States',
'Canada',
'United States',
'Austria',
'Canada',
'United States',
'Sweden',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Germany',
'Poland',
'United States',
'Canada',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Canada',
'Italy',
'Italy',
'United States',
'United States',
```

```
'United States',
'United States',
'United States',
'Finland',
'Germany',
'United States',
'United States',
'United Kingdom',
'Germany',
'United States',
'United States',
'Germany',
'United States',
'United States',
'Australia',
'United States',
'United States',
'United States',
'United States',
'United States',
'New Zealand',
'Germany',
'United States',
'United States',
'United States',
'Serbia',
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'United States',
'Romania',
'United States',
'Finland',
'United States',
'United States',
'New Zealand',
'Canada',
'United States',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Australia',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Germany',
```

```
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Canada',
'United States',
'United Kingdom',
'Canada',
'United States',
'New Zealand',
'United States',
'United States',
'United States',
'United States',
'Australia',
'United States',
'United States',
'United States',
'United States',
'Germany',
'Italy',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'Canada',
'United States',
'United States',
'United States',
'Netherlands',
'Denmark',
'Canada',
'United States',
'United States',
'Canada',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
```

```
'United States',
'United States',
'Poland',
'Poland',
'United States',
'United States',
'Canada',
'United States',
'Switzerland',
'United States',
'United States',
'Spain',
'United States',
'United States',
'United States',
'United States',
'United States',
'Mexico',
'Canada',
'United States',
'United States',
'Canada',
'United States',
'United States',
'Ukraine',
'United States',
'United States',
'United States',
'Spain',
'Italy',
'United States',
'United States',
'United States',
'United Kingdom',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Netherlands',
'Netherlands',
'United States',
'Poland',
'Canada',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'North Macedonia',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
```

```
'United States',
'United States',
'United States',
'United States',
'United States',
'Poland',
'United States',
'United States',
'United States',
'United States',
'United States',
'Brazil',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Canada',
'Russian Federation',
'United States',
'United States',
'Germany',
'United States',
'United States',
'United States',
'Indonesia',
'United States',
'Japan',
'United States',
'Sweden',
'United Kingdom',
'United States',
'Italy',
'United Kingdom',
'United States',
'Italy',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'Germany',
'Hong Kong',
'United States',
'Canada',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
'United States',
```

```
 'United States',
 'United States',
 'United States',
 'United States',
 'Belgium',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United Kingdom',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'Australia',
 'Philippines',
 'United States',
 'Sweden',
 'United States',
 'United States',
 'United States',
 'United States',
 'Germany',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 ...]
```

In [34]:
```
# I will take my new DataFrame and add a column call contry_name

mynewdf["country_name"]=[pycountry.countries.get(alpha_2=mycountry).name for mycountry
```

```
/tmp/ipykernel_15/1704559076.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  mynewdf["country_name"]=[pycountry.countries.get(alpha_2=mycountry).name for mycoun
try in mynewdf["country_code"]]
```

In [35]:
```
mynewdf.head(20)
```

Out[35]:

	id	name	brewery_id	state	country_code	style	availability	abv	notes	retir
0	202522	Olde Cogitator	2199	CA	US	English Oatmeal Stout	Rotating	7.3	No notes at this time.	
1	82352	Konrads Stout Russian Imperial Stout	18604	NaN	NO	Russian Imperial Stout	Rotating	10.4	No notes at this time.	
2	214879	Scottish Right	44306	IN	US	Scottish Ale	Year-round	4.0	No notes at this time.	
3	320009	MegaMeow Imperial Stout	4378	WA	US	American Imperial Stout	Winter	8.7	Every time this year	
4	246438	Peaches-N-Cream	44617	PA	US	American Cream Ale	Rotating	5.1	No notes at this time.	
5	8036	World Burp Beer 2002	3469	NaN	JP	Japanese Rice Lager	Limited (brewed once)	5.5	No notes at this time.	
6	108605	Icon Sender	22598	CA	US	American Lager	Year-round	5.6	No notes at this time.	
7	345382	Divina IPA	45567	NaN	IT	American IPA	Rotating	6.5	No notes at this time.	
8	255286	Light Of The Ozarks	11203	AR	US	American Lager	Rotating	4.3	No notes at this time.	
9	29556	Warrior's Bock	8203	SK	CA	German Bock	Rotating	7.5	No notes at this time.	
10	106840	Firehouse Coffee Stout	25251	NaN	AU	English Stout	Limited (brewed once)	5.0	No notes at this time.	
11	253234	Coffee Infused Bitter	46862	WA	US	English Bitter	Rotating	3.5	No notes at this time.	
12	15271	Belgian Style Wit	1345	WA	US	Belgian Witbier	Year-round	4.5	No notes at	

	id	name	brewery_id	state	country_code	style	availability	abv	notes	retir
									this time.	
13	69212	Réserve No. 3	18796	QC	CA	American Strong Ale	Limited (brewed once)	11.8	No notes at this time.	
14	181442	Southern Saints India Wheat Ale	33659	NC	US	American Pale Wheat Ale	Spring	6.9	No notes at this time.	
15	351259	Champagne Toast	34416	WA	US	Berliner Weisse	Rotating	4.0	No notes at this time.	
16	303023	The Sky Is High (wet Hopped Pale Ale)	46616	NJ	US	American Pale Ale (APA)	Limited (brewed once)	5.2	This hazy pale ale is brewed with fresh Cascad...	
17	265827	Addison	30870	PA	US	American Pale Ale (APA)	Rotating	5.5	No notes at this time.	
18	219798	Ocean Barley Wine	16420	NaN	SE	British Barleywine	Year-round	9.8	No notes at this time.	
19	110318	Schwartz Hop	22598	CA	US	American Black Ale	Limited (brewed once)	6.0	No notes at this time.	

In [ ]:

Markdown notes and sentences and analysis written here.

Question 4

In [36]:
```
mybeer['country'].value_counts()
```

Out[36]:
```
US    265461
CA     21522
GB     13923
DE      6888
BE      5253
      ...
PY         1
LY         1
YT         1
GQ         1
NE         1
Name: country, Length: 193, dtype: int64
```

In [7]:
```
mybeerct=mybeer['country'].value_counts()
```

In [8]:
```
# reset_index() help in generating an indexing for my counts
#Then i created a column for counts

mybeerct=mybeerct.reset_index()

mybeerct.columns=['country', 'counts']
```

In [39]:
```
mybeerct
```

Out[39]:

	country	counts
0	US	265461
1	CA	21522
2	GB	13923
3	DE	6888
4	BE	5253
...	...	...
188	PY	1
189	LY	1
190	YT	1
191	GQ	1
192	NE	1

193 rows × 2 columns

In [5]:
```
mybeerct=mybeer['country'].value_counts()
```

In [9]:
```
mybeerct.loc[mybeerct["counts"]>1000]
```

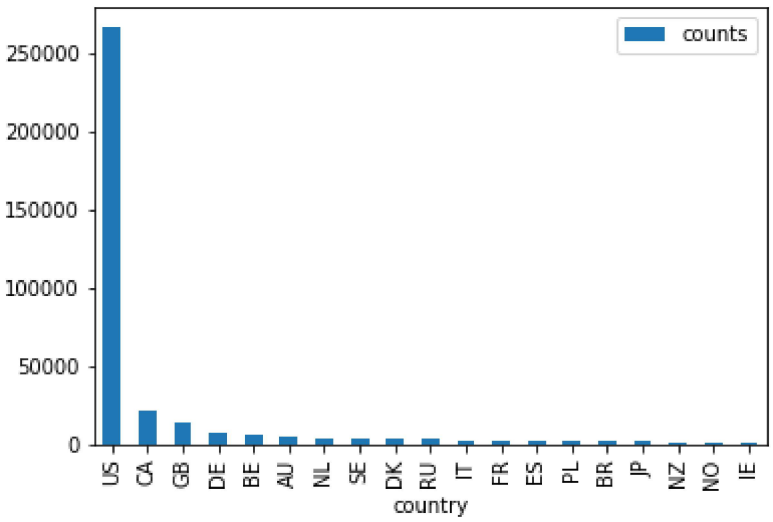
Out[9]:

	country	counts
0	US	265461
1	CA	21522
2	GB	13923
3	DE	6888
4	BE	5253
5	AU	4084
6	NL	3480
7	SE	3229
8	DK	3102
9	RU	2839
10	IT	2589
11	FR	2526
12	ES	2160
13	PL	1812
14	BR	1735
15	JP	1710
16	NZ	1219
17	NO	1112
18	IE	1056

In [10]:
```
jkbeer=mybeerct.loc[mybeerct["counts"]>1000]
```

In [11]:
```
jkbeer.plot(kind="bar",x="country",y="counts")
```

Out[11]:
```
<AxesSubplot:xlabel='country'>
```



In [27]:
```
mybeer['style'].value_counts()
```

Out[27]:
```
American IPA                          44719
American Pale Ale (APA)               22159
American Imperial IPA                 18338
Belgian Saison                        18167
American Wild Ale                     12972
                                      ...
Japanese Happoshu                       123
Finnish Sahti                           123
Bière de Champagne / Bière Brut         116
Belgian Faro                             32
Wild/Sour Beers                           4
Name: style, Length: 112, dtype: int64
```

In [28]:
```
mybeercl=mybeer['style'].value_counts()
```

In [29]:
```
mybeercl=mybeercl.reset_index()

mybeercl.columns=['style', 'counts']
```

In [30]:
```
mybeercl
```

Out[30]:

	style	counts
0	American IPA	44719
1	American Pale Ale (APA)	22159
2	American Imperial IPA	18338
3	Belgian Saison	18167
4	American Wild Ale	12972
...	...	...
107	Japanese Happoshu	123
108	Finnish Sahti	123
109	Bière de Champagne / Bière Brut	116
110	Belgian Faro	32
111	Wild/Sour Beers	4

112 rows × 2 columns

In [31]:
```
mybeercl=mybeer['style'].value_counts()
```

In [4]:
```
mybeer.groupby(['style']).mean()
```

```
/tmp/ipykernel_15/2690924485.py:1: FutureWarning: The default value of numeric_only i
n DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will default
to False. Either specify numeric_only or select only columns which should be valid fo
r the function.
  mybeer.groupby(['style']).mean()
```

Out[4]:

style	id	brewery_id	abv
American Adjunct Lager	106053.641052	11207.942238	4.947657
American Amber / Red Ale	159233.104842	23530.601764	5.936179
American Amber / Red Lager	161839.856655	19335.509386	5.469577
American Barleywine	165034.387255	19376.992341	10.823768
American Black Ale	178272.578331	25690.162142	7.021434
...	...	...	...
Smoke Beer	178389.670688	24502.041164	6.196947
Smoke Porter	141196.919565	20847.330435	6.480275
Vienna Lager	169981.006382	21058.134014	5.168732
Wild/Sour Beers	367931.250000	38088.000000	5.600000
Winter Warmer	142307.618817	18992.982258	7.185325

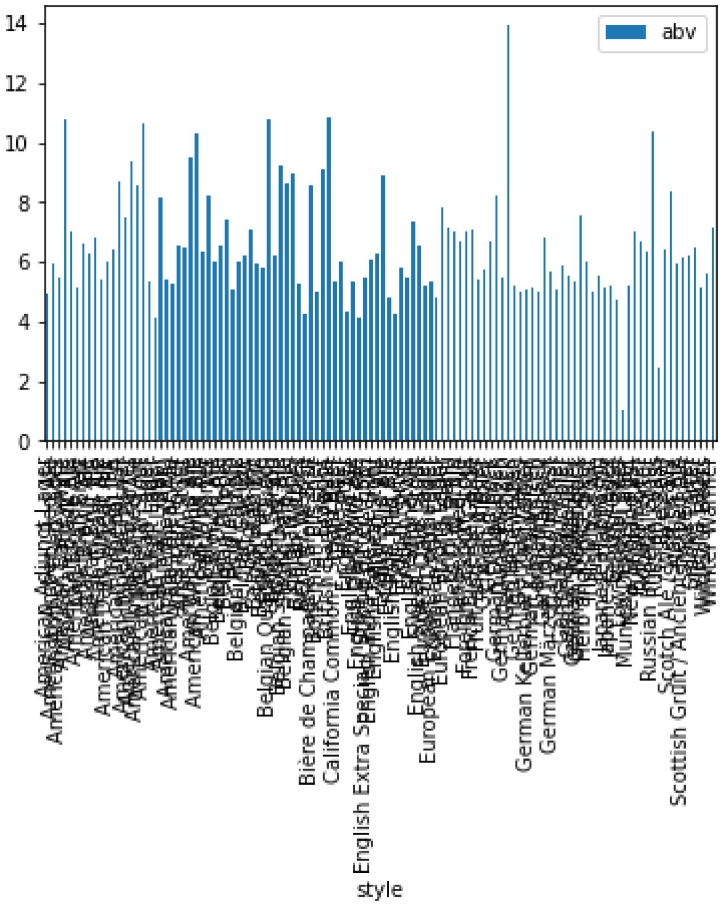
112 rows × 3 columns

In [5]:
```
dfmybeer=mybeer.groupby(['style']).mean().reset_index()
```

```
/tmp/ipykernel_15/3371066041.py:1: FutureWarning: The default value of numeric_only i
n DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will default
to False. Either specify numeric_only or select only columns which should be valid fo
r the function.
  dfmybeer=mybeer.groupby(['style']).mean().reset_index()
```

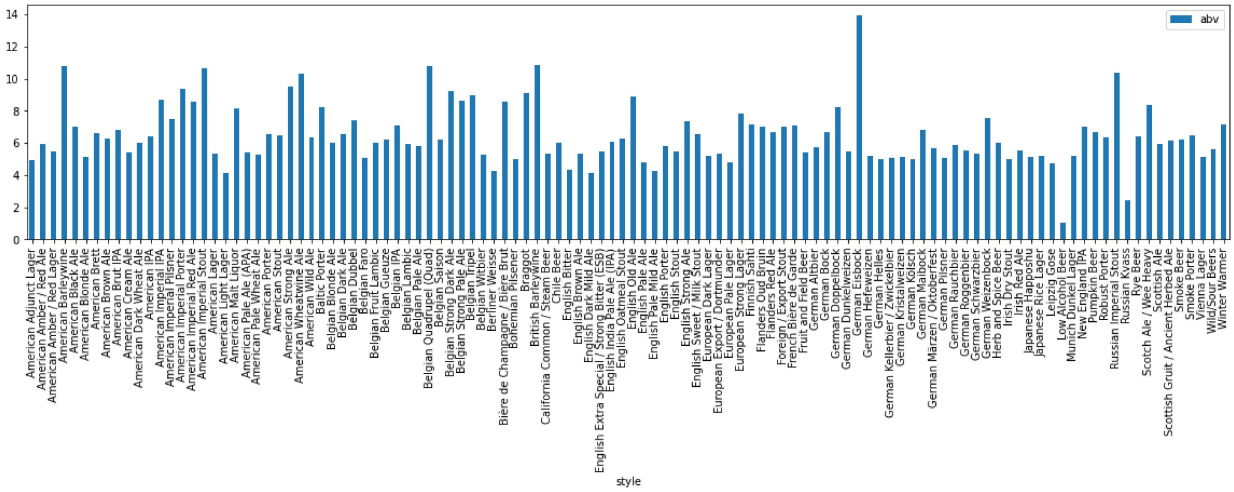
In [6]:
```
dfmybeer.plot(kind="bar", x="style", y = "abv")
```

Out[6]:
```
<AxesSubplot:xlabel='style'>
```



In [7]:
```
dfmybeer.plot(kind="bar", x="style", y = "abv", figsize=(20,4))
```

Out[7]:
```
<AxesSubplot:xlabel='style'>
```



In [8]:
```
dfmybeer.loc[dfmybeer["abv"]>8]
```

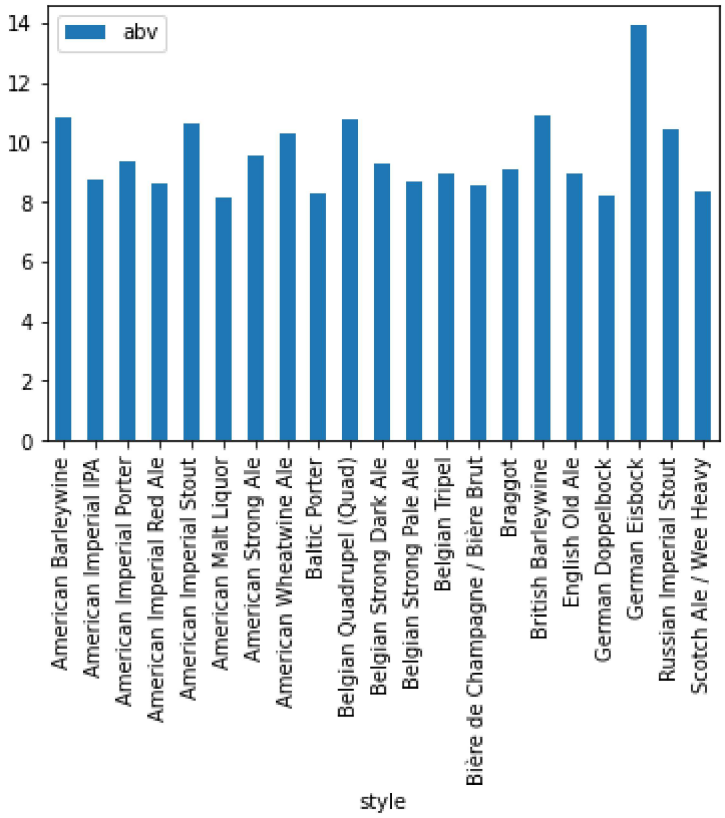
Out[8]:

	style	id	brewery_id	abv
3	American Barleywine	165034.387255	19376.992341	10.823768
12	American Imperial IPA	231055.131421	29226.761588	8.727126
14	American Imperial Porter	182064.348639	19935.899660	9.382038
15	American Imperial Red Ale	163523.823834	20881.173575	8.595798
16	American Imperial Stout	227737.525045	26512.370304	10.633764
19	American Malt Liquor	144047.011905	11878.426190	8.161108
24	American Strong Ale	150417.813472	17890.623754	9.540009
25	American Wheatwine Ale	171898.196486	19540.068690	10.315044
27	Baltic Porter	182527.409542	22079.649318	8.261075
37	Belgian Quadrupel (Quad)	195783.018300	23695.383117	10.772052
39	Belgian Strong Dark Ale	152655.935147	19697.497950	9.252459
40	Belgian Strong Pale Ale	152368.559351	19725.481366	8.663456
41	Belgian Tripel	174187.807296	23166.024055	8.960565
44	Bière de Champagne / Bière Brut	198264.801724	21023.508621	8.560714
46	Braggot	189321.484694	24341.732143	9.107830
47	British Barleywine	168215.782835	20049.894018	10.869646
56	English Old Ale	156217.654503	18078.552795	8.934646
75	German Doppelbock	147574.072368	17048.903340	8.216321
77	German Eisbock	146649.795238	11641.304762	13.928919
101	Russian Imperial Stout	183918.521238	23819.275644	10.407635
104	Scotch Ale / Wee Heavy	163101.528889	20882.779259	8.374074

In [9]:
```
ftmybeer=dfmybeer.loc[dfmybeer["abv"]>8]
```

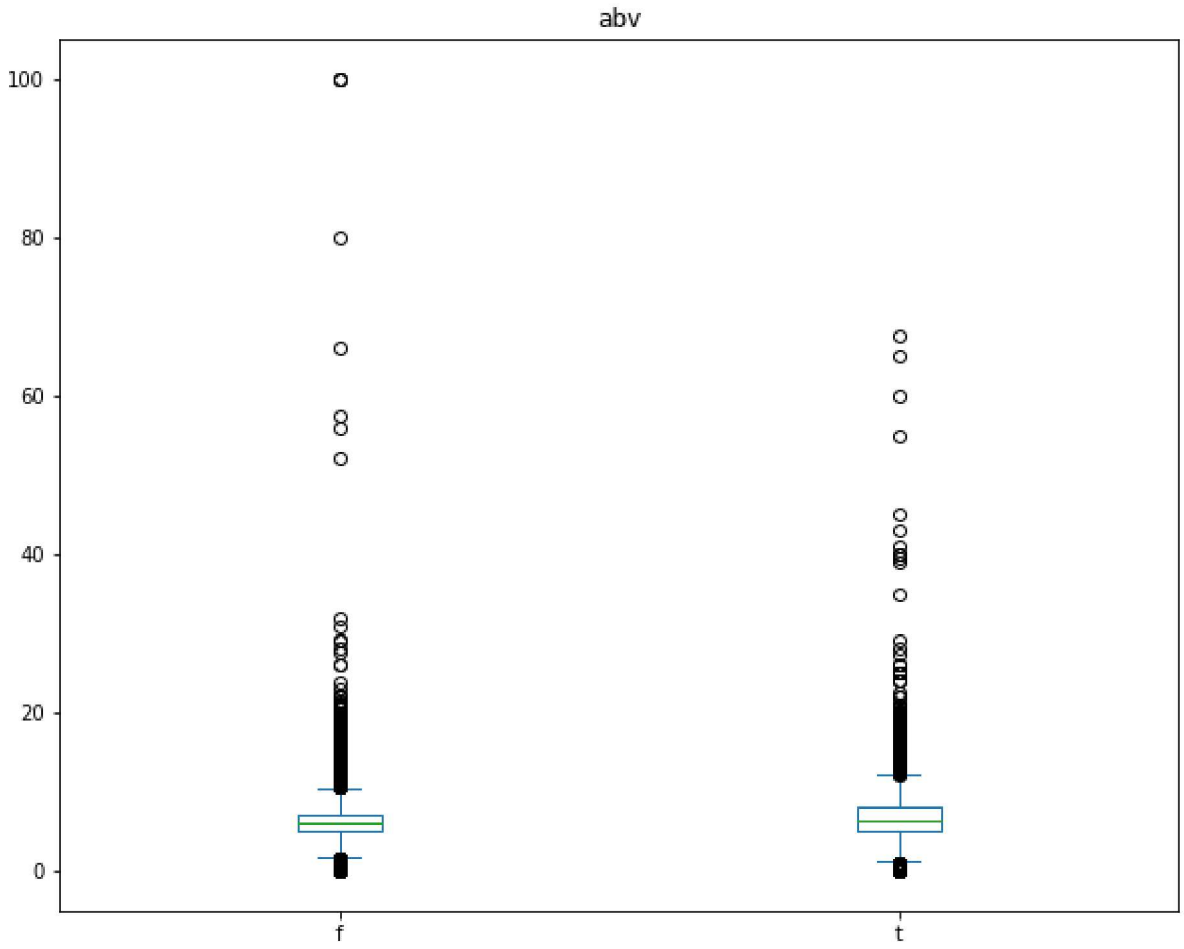
In [10]:
```
ftmybeer.plot(kind="bar", x="style", y = "abv")
```

Out[10]:
```
<AxesSubplot:xlabel='style'>
```



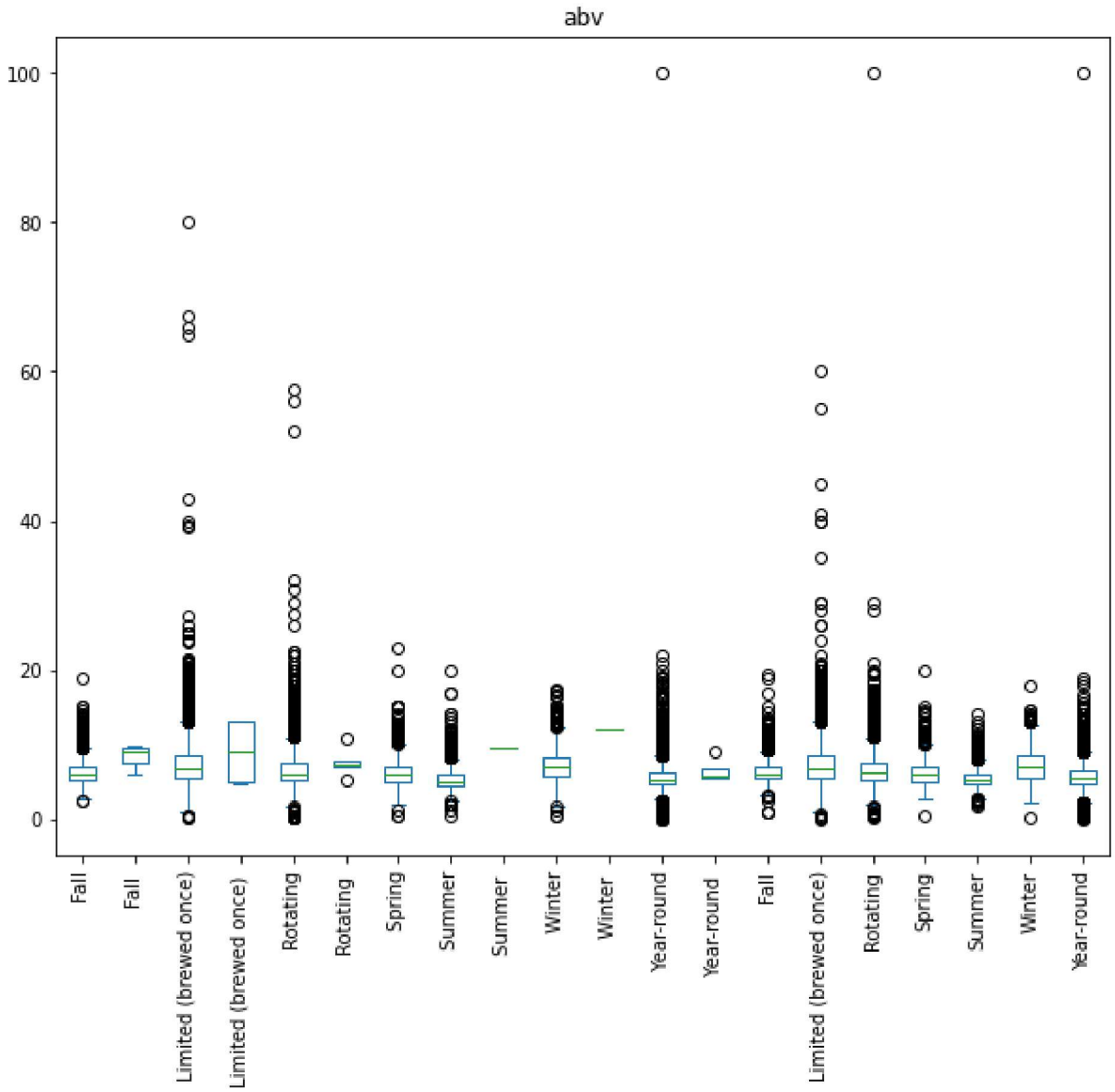
In [11]:
```
mybeer.plot.box(column="abv", by="retired", figsize=(10, 8))
```

Out[11]:
```
abv    AxesSubplot(0.125,0.125;0.775x0.755)
dtype: object
```



In [12]:
```
mybeer.plot.box(column="abv", by='availability', figsize=(10, 8), rot=90)
```

Out[12]:
```
abv    AxesSubplot(0.125,0.125;0.775x0.755)
dtype: object
```



In [13]:
```
mybeer.groupby(['availability','retired'])
```

Out[13]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7f6c1a26e6b0>
```

In [14]:
```
mybeer.groupby(['availability','retired']).size()
```

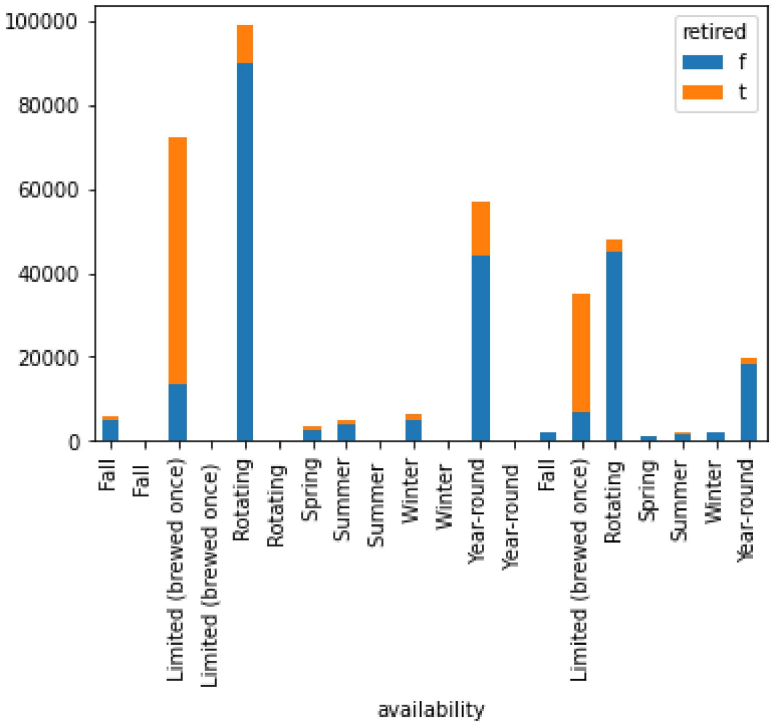
Out[14]:

availability	retired	
Fall	f	4783
	t	1125
Fall	f	2
	t	1
Limited (brewed once)	f	13423
	t	58624
Limited (brewed once)	f	1
	t	8
Rotating	f	89941
	t	8835
Rotating	f	6
Spring	f	2402
	t	947
Summer	f	3973
	t	1163
Summer	f	1
Winter	f	4772
	t	1655
Winter	f	1
Year-round	f	44004
	t	12796
Year-round	f	2
	t	2
Fall	f	2073
	t	237
Limited (brewed once)	f	6662
	t	28136
Rotating	f	45243
	t	2759
Spring	f	1114
	t	172
Summer	f	1780
	t	193
Winter	f	2036
	t	257
Year-round	f	18261
	t	1483

dtype: int64

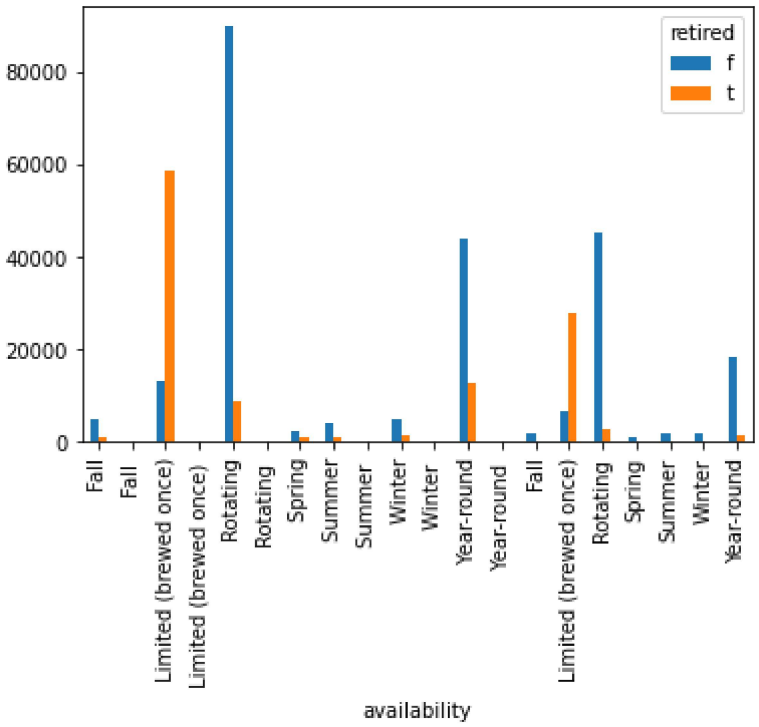
In [15]:
```
mybeer.groupby(['availability','retired']).size().unstack().plot(kind='bar', stacked=T
```

Out[15]:
```
<AxesSubplot:xlabel='availability'>
```



In [16]:
```
mybeer.groupby(['availability','retired']).size().unstack().plot(kind='bar', stacked=
```

Out[16]:
```
<AxesSubplot:xlabel='availability'>
```



Markdown notes and sentences and analysis written here.

Question 5

In [ ]:
```
# code here
```

Markdown notes and sentences and analysis written here.

Pledge

By submitting this work I hereby pledge that this is my own, personal work. I've acknowledged in the designated place at the top of this file all sources that I used to complete said work, including but not limited to: online resources, books, and electronic communications. I've noted all collaboration with fellow students and/or TA's. I did not copy or plagiarize another's work.

> As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do. Accountable together – We are Purdue.