BeerDataScienceProject

-presented by

Shilpa Kewalaramani

Table Of Contents

Introduction & Objective

The Beer Challenge Analysis Project dataset has the data related to beer reviews which contains information regarding beers on the basis of the reviews collected by the users. The dataset has fields like Beer id, Beer_Style, Beer Appearance, Beer Aroma and many such columns.

There are Total 5,28,870 rows and 13 columns

Objective

The main objective behind this analysis project is to answer some questions after taking the insights from the dataset provided

Exploring more about Columns

Details of ColumnS

Visualizing Numerical Columns

Visualizing Categorical Columns

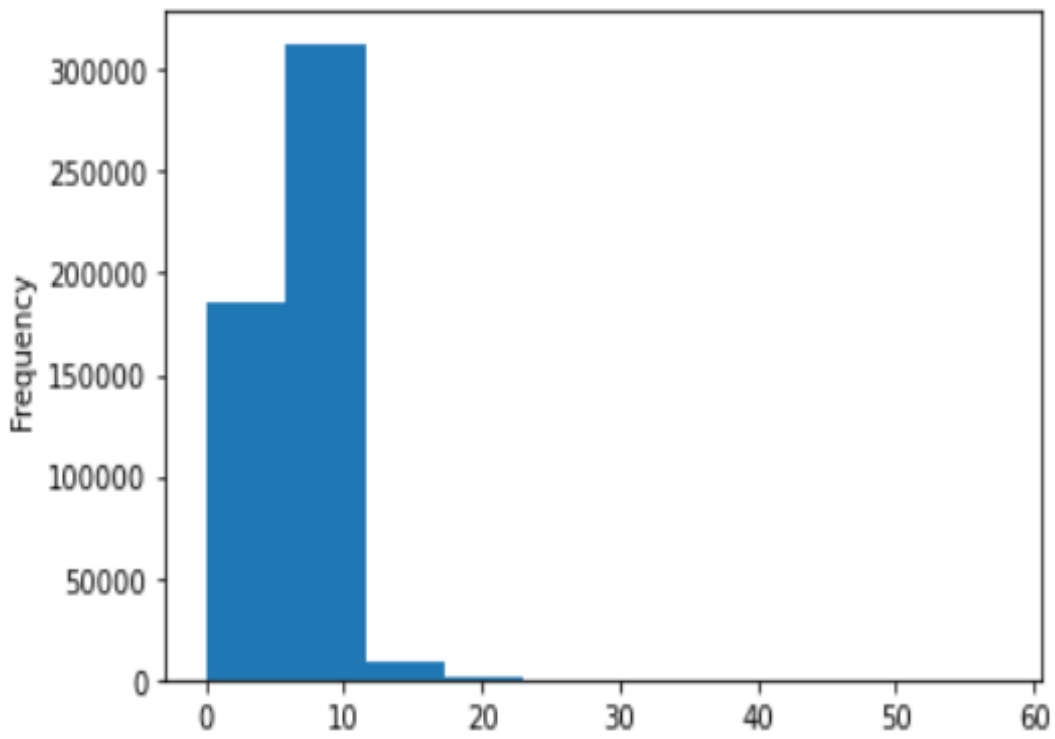
Details of Columns

- beer_ABV means = Here ABV means Alcohol By Volume , beer_ABV means the percent of alcohol present in beer
- beer_beerId = Its the id assigned to different beers
- beer_brewerId = It is the id of a place where beer is made commercially.
- beer_name = It is the name of the beer
- beer_style = Beer styles differentiates & categorise beers by colour, flavour, strength, ingredients, production
- review_appearance = The beer_appearance means the beer looks in color
- review_palette = Rating based on how the beer interacts with the palate
- review_overall = Overall review points given by the user
- review_taste = Rating based on the taste of beer
- review_profileName = Name of the person who reviewed the beer
- review_aroma = Rating based on the smell of the review
- review_text = Reviews in text written by the user
- review_time = Timestamp when the review was recorded

Visualizing Numerical Columns

beer_ABV

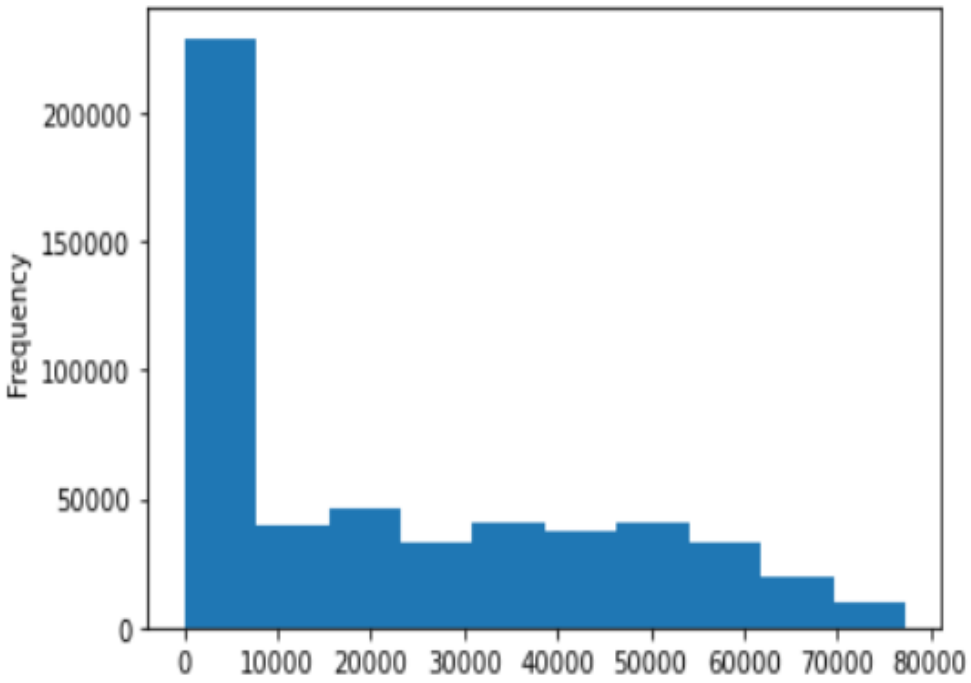
- There are 283 different beer_ABV recorded,
- Minimum Beer_ABV recorded is 0.01 and
- maximum recorded is 57.7,
- maximum Beer_ABV lies between the range of 0 to 10



beer_beerId

- There are 20,200 unique beer_beerId
- Minimum beer_beerId is 3
- Maximum beer_beerIdis 77310
- Maximum beer_beerId ranges between 0 to 10,000

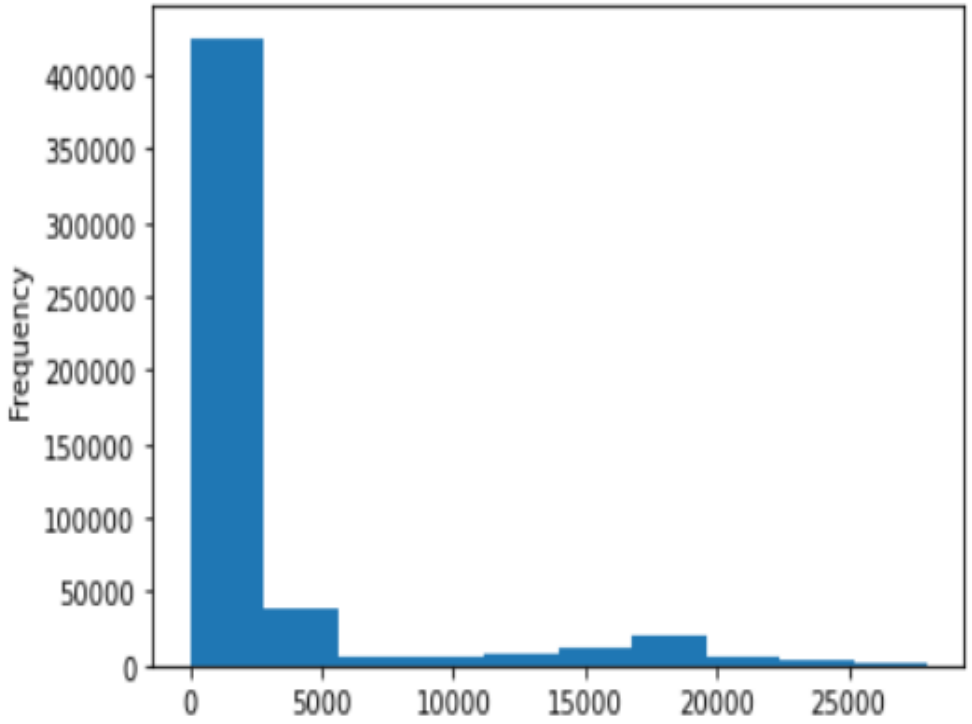
Visualizing Numerical Columns



beer_brewerld

- Minimum beer_brewerld is 1
- Maximum beer_brewerld s 27980
- Maximum beer_brewerld ranges between 0 to 5000
- there are 1803 unique beer_brewerld

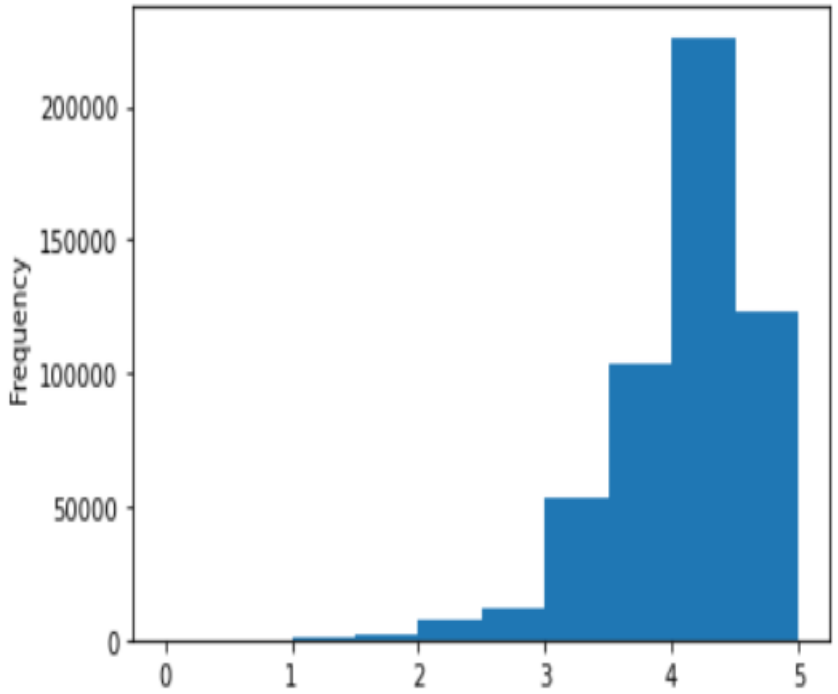
Visualizing Numerical Columns



Visualizing Numerical Columns

beer_appearance

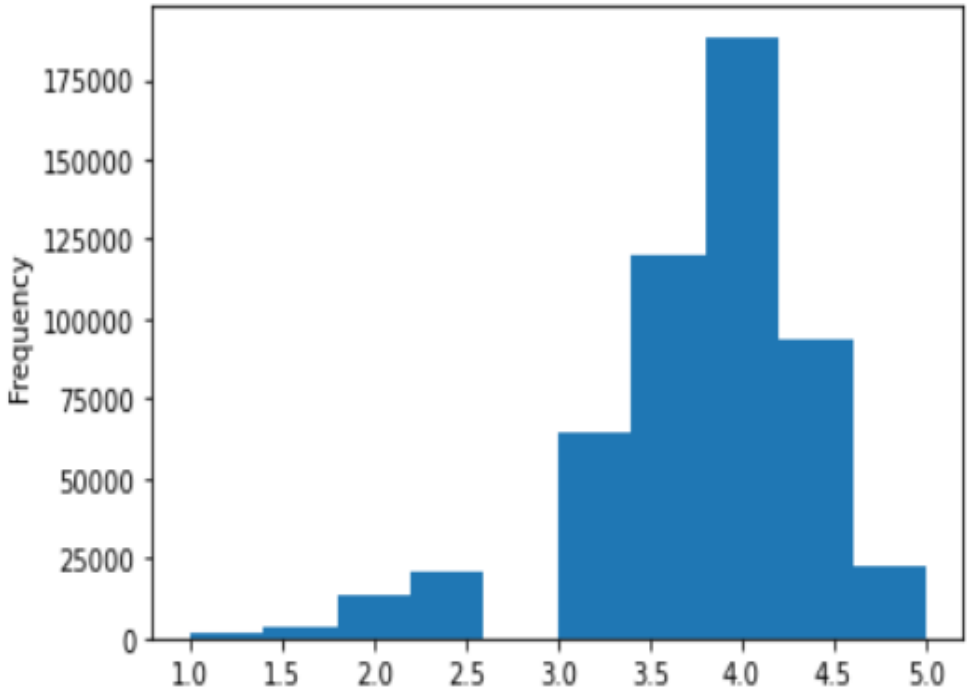
- Minimum review_appearance is 0.0
- Maximum review_appearance is 5.0
- Maximum review_appearance ranges between 4 to 5



review_palette

- * Minimum review_palette is 1.0
- * Maximum review_palette is 5.0
- * Maximum review_palette ranges between 3.5 to 4.0

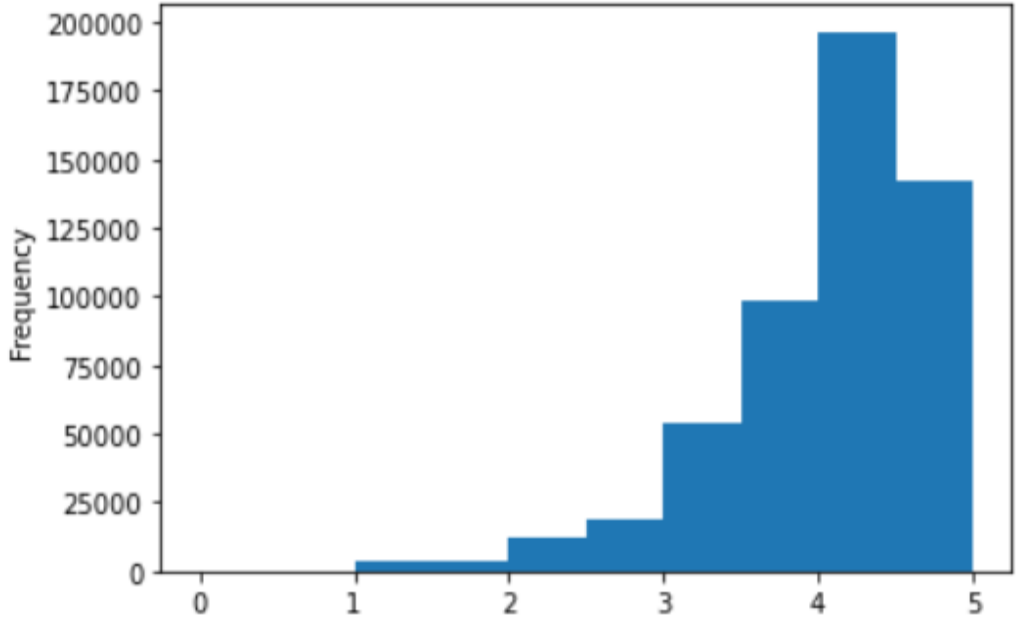
Visualizing Numerical Columns



Visualizing Numerical Columns

review_overall

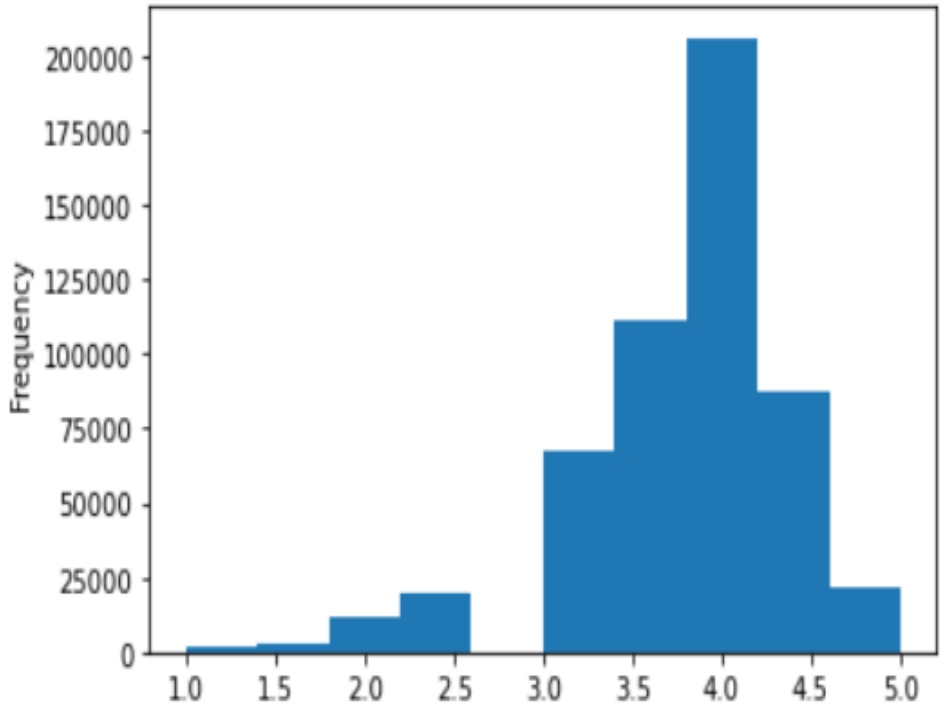
- Minimum review_overall is 0.0
- Maximum review_overall is 5.0
- Maximum review_overall ranges between 4 to 5



Visualizing Numerical Columns

review_taste

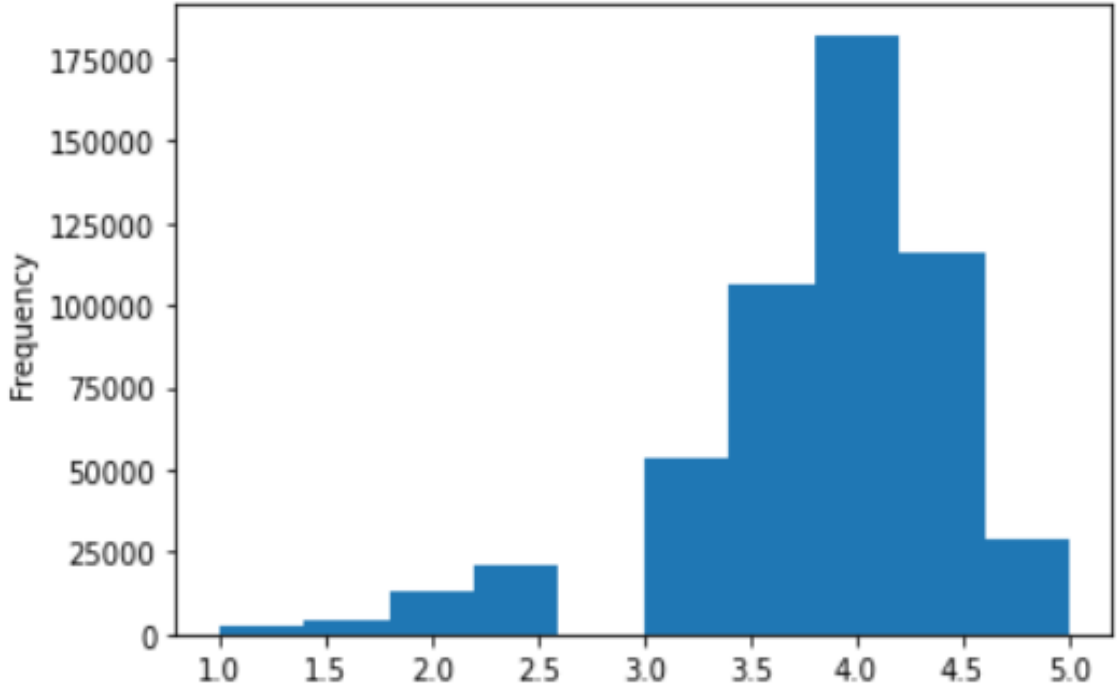
- Minimum review_taste is 1.0
- Maximum review_taste s 5.0
- Maximum review_taste ranges between 3.75 to 4 or 4.25



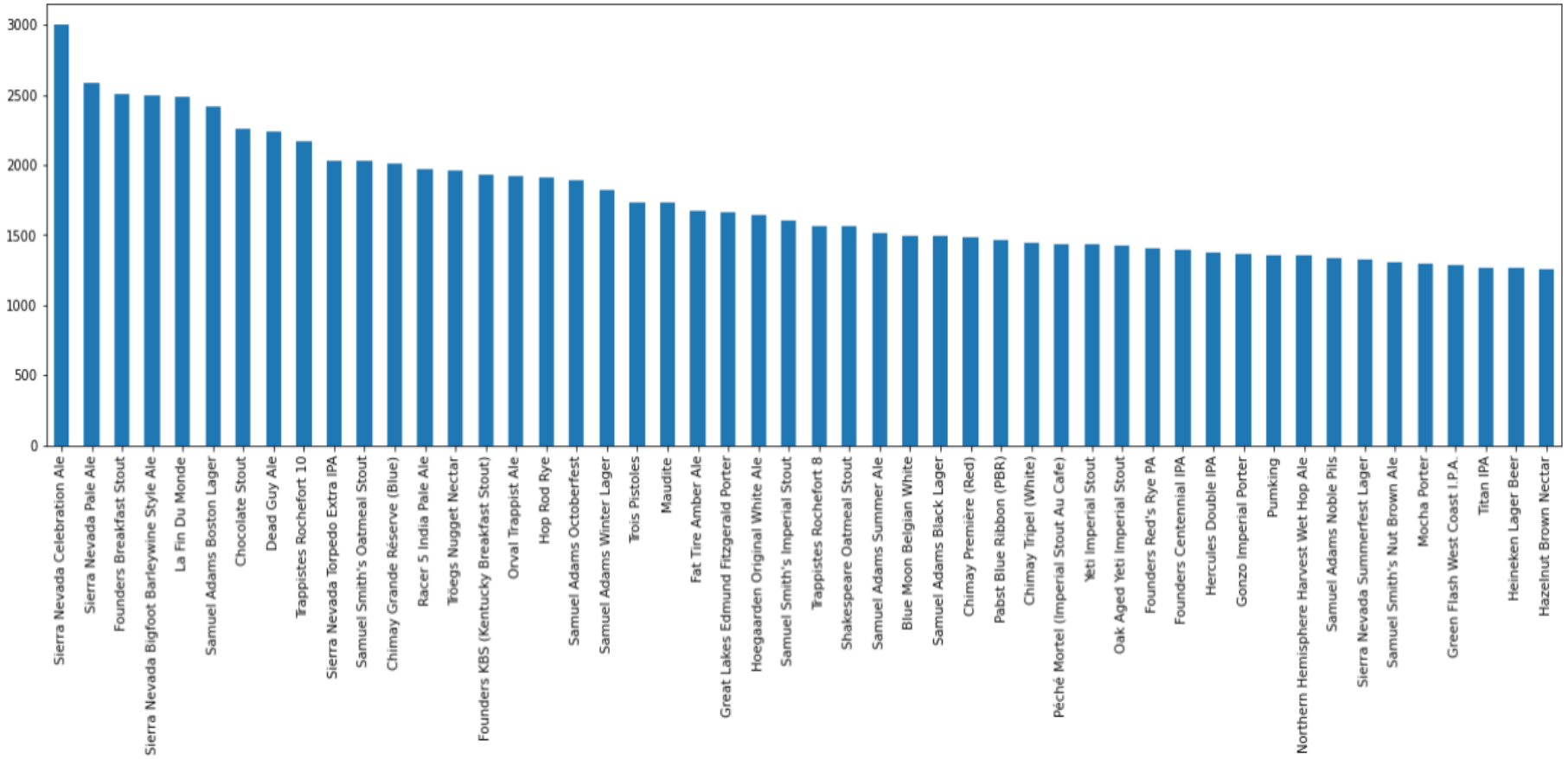
Visualizing Numerical Columns

review_aroma

- Minimum review_aroma is 1.0
- Maximum review_aroma s 5.0
- Maximum review_aroma ranges between 3.75 to 4 or 4.25

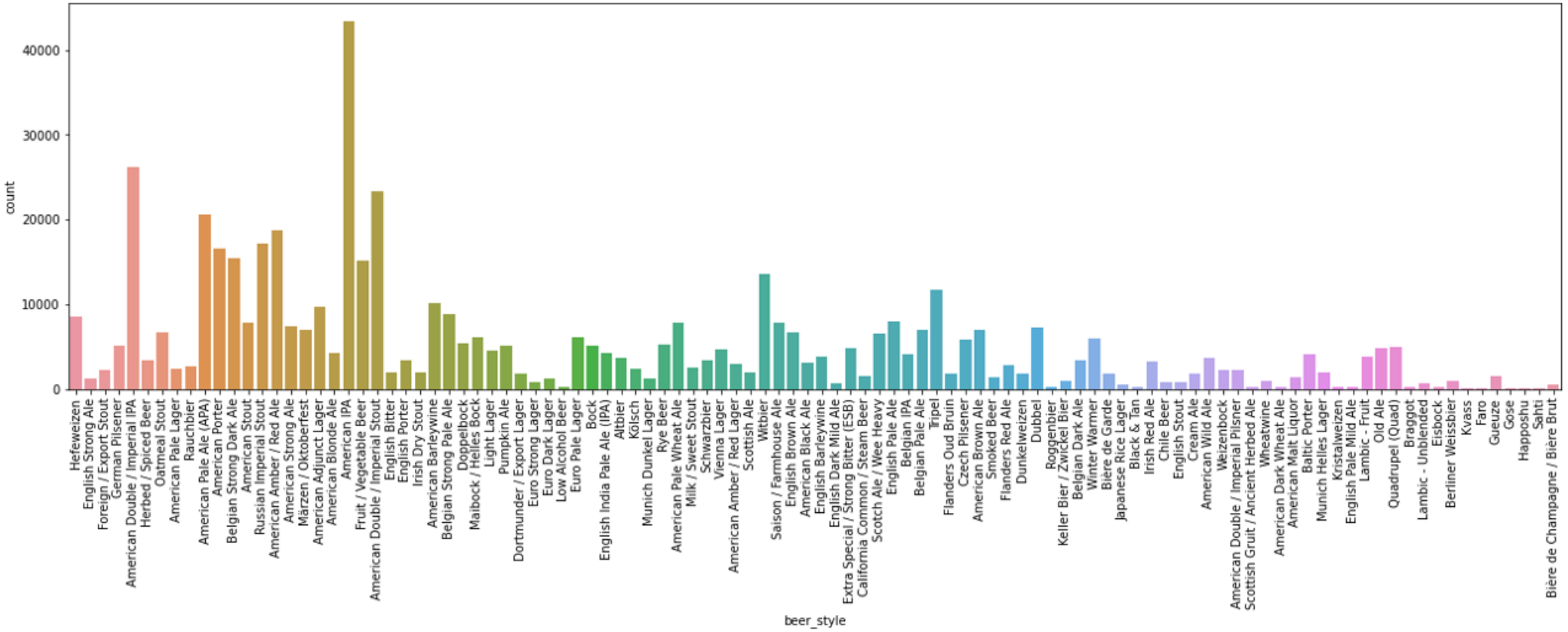


Visualizing Categorical Columns



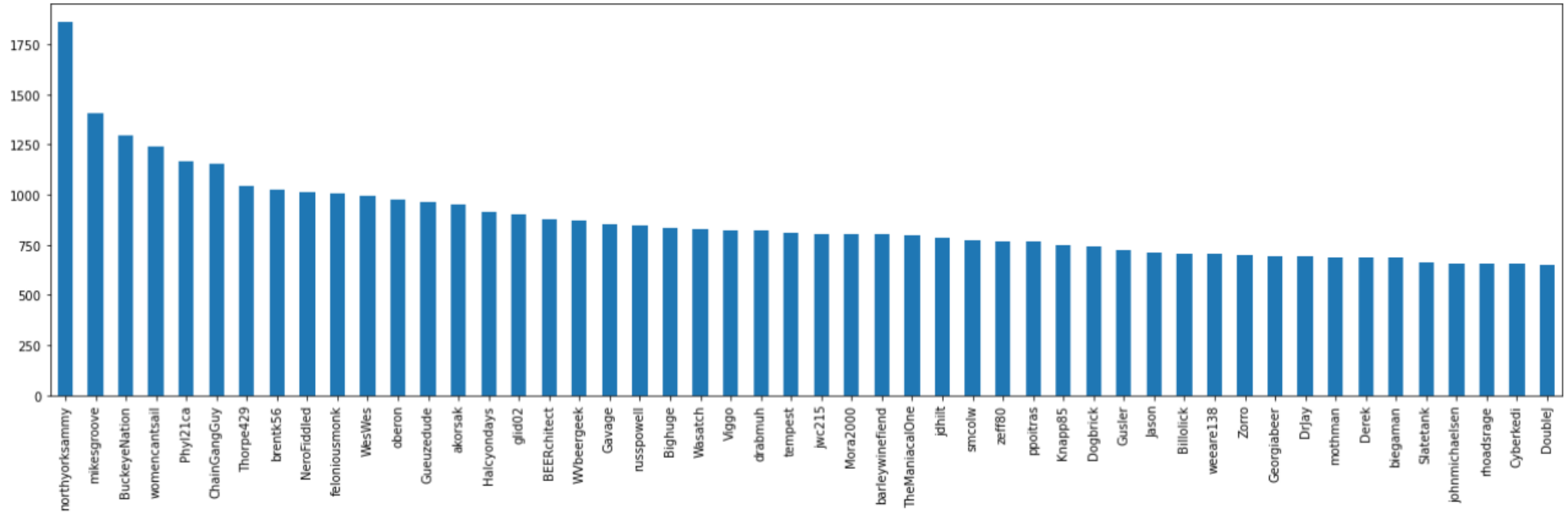
- There are total 18339 different names of beer observed in the dataset
- Maximum beer consumed by user is Sierra Nevada Celebration Ale

Visualizing Categorical Columns



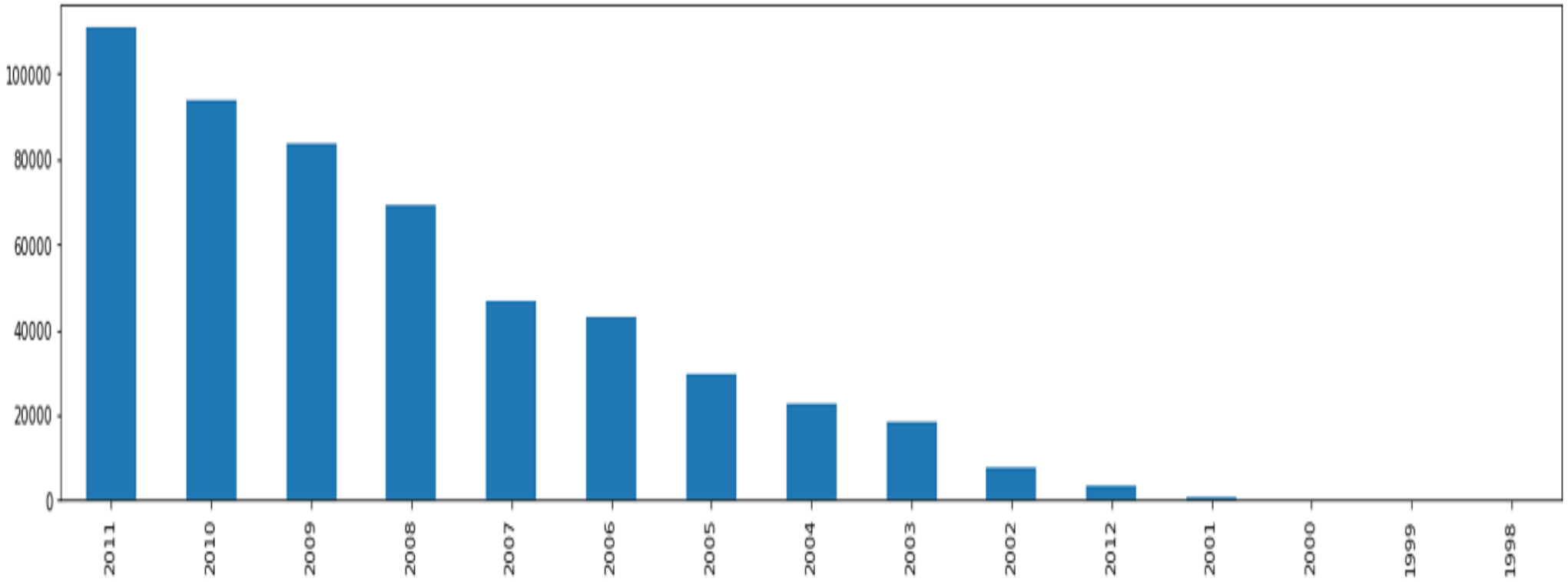
- There are total 104 different styles of beer observed in the dataset
- Maximum beer style consumed by users are 'American IPA '

Visualizing Categorical Columns



- There are total 22,800 consumers of beer who reviewed the beer observed in the dataset
- Maximum beer reviews are given 'northyorksammy'

Visualizing Year Column



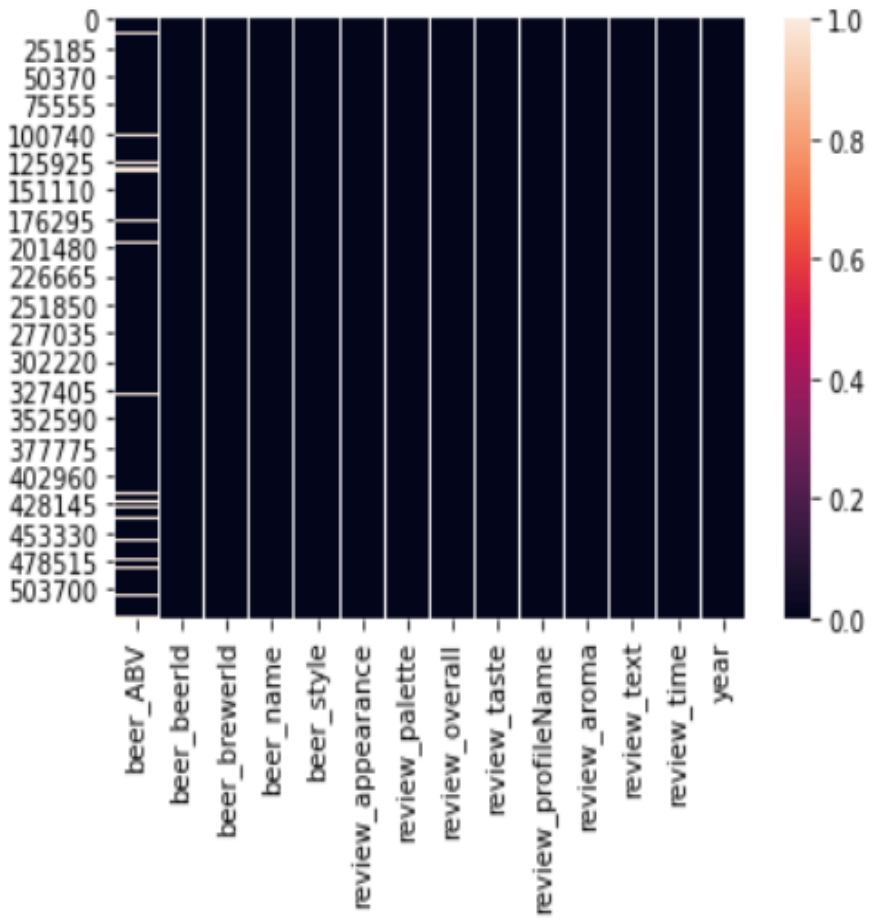
- Maximum reviews are collected during the year 2011,
- from 1998 the reviews are gradually increasing

Null Values

- White Horizontal lines in the figure shows the null values
- 3.83 % of data was NULL
- beer_ABV columns has 20280 Null values
- review_profileName has 115 Null values
- review_text has 119 Null values

- Null Values were replaced with mean and mode.

Visualizing Null Values present in dataset



Statistical Description

	beer_ABV	beer_beerId	beer_brewerId	beer_name	beer_style	review_appearance	review_palette	review_overall	review_taste	review_profileName	review_aroma	review_text	review_time	year
count	528870.000000	528870.000000	528870.000000	528870	528870	528870.000000	528870.000000	528870.000000	528870.000000	528870	528870.000000	528870	5.288700e+05	528870.000000
unique	NaN	NaN	NaN	18339	104	NaN	NaN	NaN	NaN	22801	NaN	528372	NaN	NaN
top	NaN	NaN	NaN	Sierra Nevada Celebration Ale	American IPA	NaN	NaN	NaN	NaN	northyorksammy	NaN	<bound method Series.mode of 0 A lot o...	NaN	NaN
freq	NaN	NaN	NaN	3000	43369	NaN	NaN	NaN	NaN	1858	NaN	119	NaN	NaN
mean	7.017442	22098.466016	2598.423429	NaN	NaN	3.864522	3.758926	3.833197	3.765993	NaN	3.817350	NaN	1.224885e+09	2008.308306
std	2.161781	22158.284352	5281.805350	NaN	NaN	0.604010	0.685335	0.709962	0.669018	NaN	0.718903	NaN	7.605600e+07	2.409979
min	0.010000	3.000000	1.000000	NaN	NaN	0.000000	1.000000	0.000000	1.000000	NaN	1.000000	NaN	8.843904e+08	1998.000000
25%	5.300000	1745.000000	132.000000	NaN	NaN	3.500000	3.500000	3.500000	3.500000	NaN	3.500000	NaN	1.174613e+09	2007.000000
50%	6.500000	14368.000000	394.000000	NaN	NaN	4.000000	4.000000	4.000000	4.000000	NaN	4.000000	NaN	1.240366e+09	2009.000000
75%	8.500000	40528.000000	1475.000000	NaN	NaN	4.000000	4.000000	4.500000	4.000000	NaN	4.500000	NaN	1.288560e+09	2010.000000
max	57.700000	77310.000000	27980.000000	NaN	NaN	5.000000	5.000000	5.000000	5.000000	NaN	5.000000	NaN	1.326277e+09	2012.000000

Statistical Description Continued

* The columns that are showing NAN are categorical columns.

* It can be observed that in beer_name Sierra Nevada Celebration Ale is on top with a frequency of 3000

* in beer_style American IPA is on top with frequecny of 43369

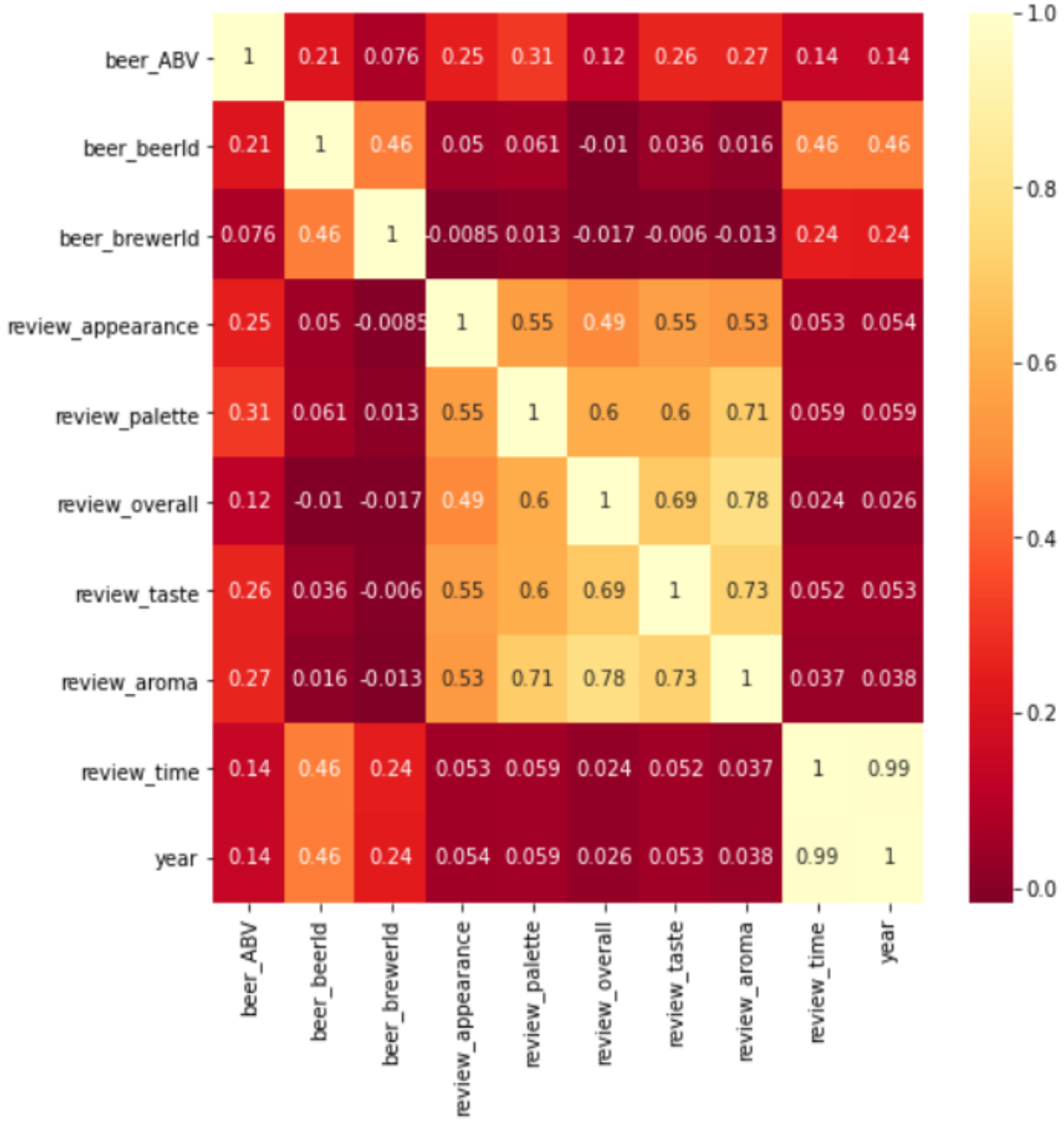
* In review_profileName northyorksammy is on top with frequency 1858

* The difference between the 75% percentile and maximum of beer_ABV is remarkable, it means outliers are present

* The difference between 75% percentile and maximum of beer_brewerId is also more but as it is Id number so it can be in that range

Statistical Description Correlation between columns

- review taste and review aroma are highly correlated with each other
- review taste and review overall are also highly correlated with each other



Questions And Answers

1. Rank top 3 Breweries which produce the strongest beers?

Here strongest beers indicates the strong presence of alcohol in the beer. so we have to find out top 3 beer_brewerId using mean of beer_ABV

So Top Three breweries are , Brewerid no 6513, 736 and 24215.

```
temp_data = df.groupby(['beer_brewerId'])['beer_ABV'].mean()
temp_df = pd.DataFrame(temp_data.reset_index())

temp_df.columns = ['beer_brewerId','beer_abv_mean']

Top_3_Breweris = temp_df.sort_values(by=['beer_abv_mean'], ascending=False).head(3)

Top_3_Breweris
```

	beer_brewerId	beer_abv_mean
784	6513	19.228824
175	736	13.395655
1644	24215	12.466667

2. Which year did beers enjoy the highest ratings?

```
temp_df= df.groupby('beer_beerId').agg({'review_overall': np.mean,  'review_aroma': np.mean,'review_appearance': np.mean,
                                        'review_palette': np.mean,  'review_taste': np.mean, 'beer_ABV': np.mean})
high_reviews = pd.DataFrame(temp_df.reset_index()).sort_values(['review_overall', 'review_aroma','review_appearance','review_pale
High_ratings = pd.merge(df[['beer_beerId','year']],high_reviews[:1],on='beer_beerId')
High_ratings
```

	beer_beerId	year	review_overall	review_aroma	review_appearance	review_palette	review_taste	beer_ABV
0	1734	2002	5.0	5.0	5.0	5.0	5.0	10.0

to find out highest ratings of beer we need to groupby beer_id and we also need to take highest counts of all other factors like review appearance, review_aroma, review_overall, review_palettte, and review_taste

<u>so the year that encountered highest rating was 2002</u>

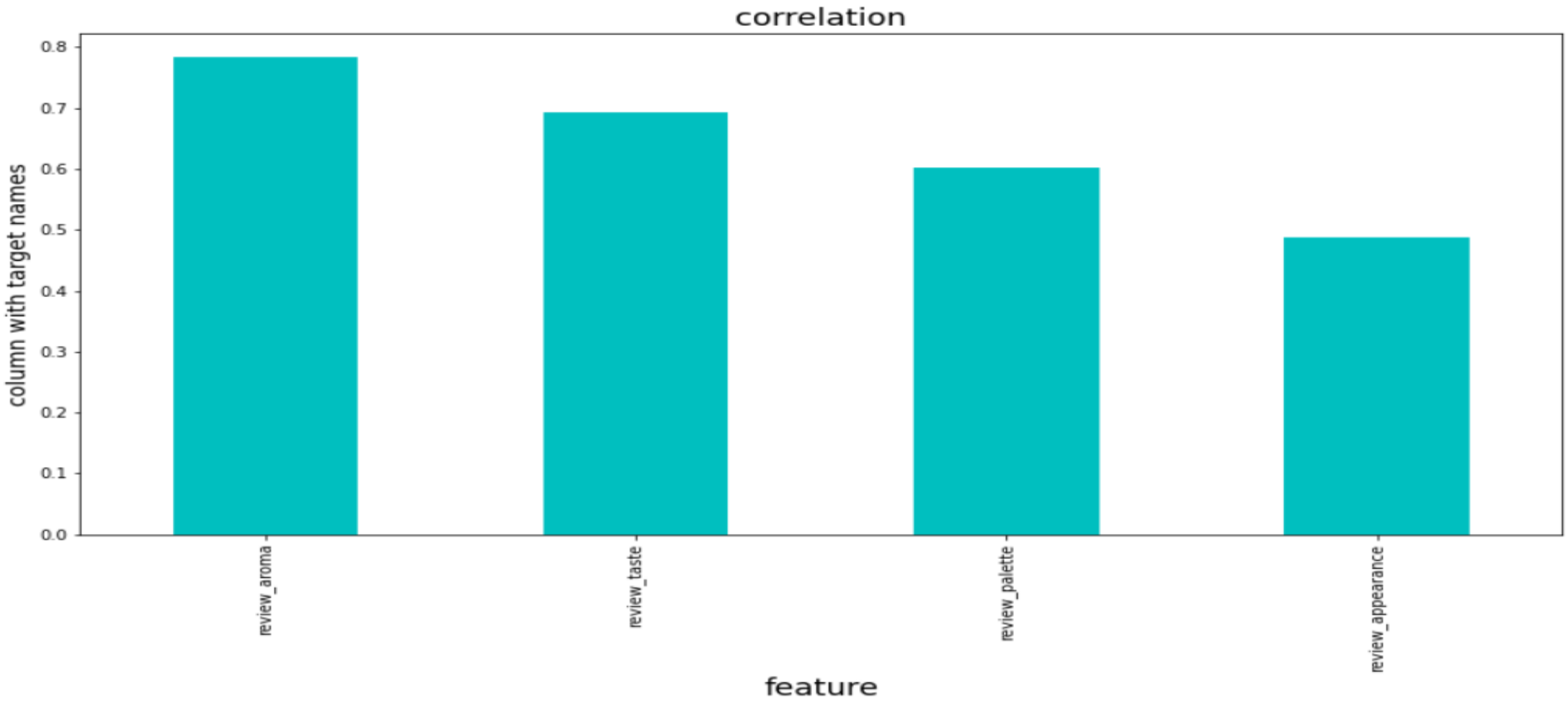
3. Based on the user's ratings which factors are important among taste, aroma, appearance, and palette?

To find the important factor, all the factors need to be considered along with overall_review and the column showing highest correlation with review_overall can be considered as importan

```
columns= df[['review_taste','review_aroma','review_appearance','review_palette','review_overall']]
columns.head()
```

	review_taste	review_aroma	review_appearance	review_palette	review_overall
0	1.5	1.5	2.5	2.0	1.5
1	3.0	3.0	3.0	2.5	3.0
2	3.0	3.0	3.0	2.5	3.0
3	2.5	3.0	3.5	3.0	3.0
4	4.0	4.5	4.0	4.5	4.0

3. Based on the user's ratings which factors are important among taste, aroma, appearance, and palette?



3. Based on the user's ratings which factors are important among taste, aroma, appearance, and palette?

From the bar graph it can be considered that important factors are¶

1) review_aroma

2)review_taste

3)review_palette

4)review_appearance

4. If you were to recommend 3 beers to your friends based on this data which ones will you recommend?¶

Finding top 3 beers to recommend to friend is similar like finding the year with highest rating, here I will use groupby function on beer_name and highest count (mean) on all factors like beer_ABV', 'beer_beerId', 'beer_brewerId', 'beer_name', 'beer_style','review_appearance', 'review_palette', 'review_overall', 'review_taste','review_aroma',

4. If you were to recommend 3 beers to your friends based on this data which ones will you recommend?¶

```
beers= df.groupby('beer_name').agg({'review_overall': np.mean,'review_aroma': np.mean,'review_appearance': np.mean,
                                    'review_palette': np.mean,'review_taste': np.mean,'beer_ABV': np.mean})
top_beers = pd.DataFrame(beers.reset_index()).sort_values(['review_overall', 'review_aroma','review_appearance','review_palette',
top_3_beers_names = pd.merge(top_beers[:3],df[['beer_name','beer_beerId','beer_style']],on='beer_name')
top_3_beers_names
```

	beer_name	review_overall	review_aroma	review_appearance	review_palette	review_taste	beer_ABV	beer_beerId	beer_style
0	Edsten Triple-Wit	5.0	5.0	5.0	5.0	5.0	10.0	1734	Belgian Strong Pale Ale
1	Old Gander Barley Wine	5.0	5.0	5.0	5.0	5.0	9.5	29423	American Barleywine
2	Rogue Black Brutal	5.0	5.0	5.0	5.0	5.0	9.0	45944	Schwarzbier

Top 3 beer names to recommend to friends are

1. **Edsten Triple-Wit**
2. **Old Gander Barley Wine**
3. **Rogue Black Brutal**

5. Which Beer style seems to be the favorite based on reviews written by users?

For finding the favourite beer style this we need to preprocess, train and do sentiment analysis of reviews considering beer style as target and review_text as feature and finally using groupby function on polarity_Score of beer style we will find top 10 Beer styles

```
# Group by 'beer_style' and calculate mean of polarity score
df.groupby('beer_style')['polarity_score'].mean().sort_values(ascending=False)[0:10]
```

```
beer_style
Quadrupel (Quad)                    0.862545
Braggot                             0.860895
Flanders Red Ale                    0.852030
Eisbock                             0.851972
Dortmunder / Export Lager           0.850105
American Double / Imperial Stout    0.845400
Wheatwine                           0.839985
Kvass                               0.837569
Old Ale                             0.837084
Belgian Strong Dark Ale             0.835026
Name: polarity_score, dtype: float64
```

5. Which Beer style seems to be the favorite based on reviews written by users?

On the basis of written reviews following Beer Styles seems to be favourite

1) Quadrupel (Quad)
2)Braggot
3)Flanders Red Ale
4)Eisbock
5)Dortmunder / Export Lager
6)American Double / Imperial Stout
7)Wheatwine
8)Kvass
9) Old Ale
10) Belgian Strong Dark Ale

6. How does written review compare to overall review score for the beer styles?

we will find it using mean of polarity_score and mean of review_overall on beer_style with groupby function

```
reviews = df.groupby('beer_style').agg({'polarity_score': np.mean,'review_overall': np.mean})
score= pd.DataFrame(reviews.reset_index()).sort_values(['polarity_score','review_overall'], ascending=[False,False])
score.head()
```

	beer_style	polarity_score	review_overall
86	Quadrupel (Quad)	0.862545	4.049250
32	Braggot	0.860895	3.645729
58	Flanders Red Ale	0.852030	3.962561
41	Eisbock	0.851972	4.079487
38	Dortmunder / Export Lager	0.850105	4.051962

7. How do find similar beer drinkers by using written reviews only?

By using polarity_score we can find the beer drinkers with similar written reviews, like the review_profilename who has same polarity_score simply means their reviews are similar

In [46]:
```
reviews = df.groupby('review_profileName').agg({'polarity_score': np.mean,'review_overall': np.mean})
score= pd.DataFrame(reviews.reset_index()).sort_values(['polarity_score','review_overall'], ascending=[False,False])
score.head()
```

Out[46]:

	review_profileName	polarity_score	review_overall
605	B0bD0bbselbock	0.9986	4.5
16620	layapandora	0.9984	3.0
7975	Stimwizzle	0.9981	5.0
7494	Scottiv	0.9980	4.5
8123	SynergyZ	0.9978	3.5

In [47]:
```
print(df[df['polarity_score']==0.99].shape)
df[df['polarity_score']==0.99]
```

```
(474, 15)
```

Out[47]:

	beer_ABV	beer_beerId	beer_brewerId	beer_name	beer_style	review_appearance	review_palette	review_overall	review_taste	review_profileName	re
841	5.4	10785	1075	Ashland Amber	American Amber / Red Ale	4.5	4.0	4.0	4.0	Slatetank	
1623	5.5	13289	1454	Oktoberfest	Märzen / Oktoberfest	4.5	3.5	5.0	4.0	RustyShackleford	
2307	5.6	44844	13614	St. Anna Festbier	Märzen / Oktoberfest	3.5	4.0	4.0	4.0	TKempe	

7. How do find similar beer drinkers by using written reviews only?

Here I have reloaded the original database to see the original reviews of ProfileNames with similar Polarity_score

to see the similarity among review_text you can go through the following original reviews whose review texts are similar as they both are very happy to taste the beer

```
In [49]: df1.loc[841,'review_text']

Out[49]: "I picked this can up at Al's of Hampden, poured chilled from the can into a tulip. A - a ruby colored ale, excellent clarity
w/ 3fingers of light beige cap w/ mix of bubbles sized large and tiny. The lace clings very well and retention is good. The col
or reminds me of cranberry juice which is attractive to look at in the light S - A sweet toasted grainy odor w/ light fruitines
s and berry-like yeast ester w/ a mild hop aroma and slight caramel notes w/ gentle vegetal aroma M - a moderately carbonated b
rew w/ mild bitterness and gentle sweetness. The texture has light toasted and spice aspects w/ dry finishing medium body overa
ll T - the flavor is relatively biscuity upfront and has subtle pine notes from the hops w/ light creamy caramel malt taste. Th
ere is a gently fruity element from the yeast which allows the toasted grain note to carry over and seems to be the focus from
midpoint on. the toast or roasted notes are the focus w/ mild herbal and the fruity quality adding contrast. The spice tinges m
ellow and a very light citrus in the hops comes out when warmer w/ another layer of flavor unfolding. The contrast makes the am
ber very balanced but seems to lean toward malt ever so slightly w/ strong biscuit taste until the semi-dry finish D - A solid
beer in every way, very good representation of the style w/ satisfying drinkability. I would drink this with barbecue or meatlo
af. The drinkability is good and I would look for this again I Can guarntee that (wink, wink)"

In [50]: df1.loc[1623,'review_text']

Out[50]: "Stopped in for dinner for my wife's Birthday. Tried this on tap with my Stuffed Sirloin and Jalapenos, then took a growler of
it home. Appearance: Amber, to dark amber, crystal-clear with an initial bit of off-white head that reduced to a ring. Not much
lacing. Smell: scents of malts and cookie-type bread, with hops noticeable. Taste: great malt sweetness and balanced hop profil
e. Really good, I get the feeling this is a real lager, not an Oktober-Ale that many places bring out. Really nice. Mouthfeel &
Drinkability: Great feel, if just a tad low carb from the growler fill. Not all that thin in the mouth, just right. Great Drink
ability, and real fresh. Overall: I have had many Oktoberfests this year (one of my fav styles) from Sam Adams, Hacker, Spaten,
etc. and this is great. I have to try them side-side-side to see what differences I get, but so far my favorite of the seaso
n!!! Great Job, BRBP!"
```

Steps To Run the Project

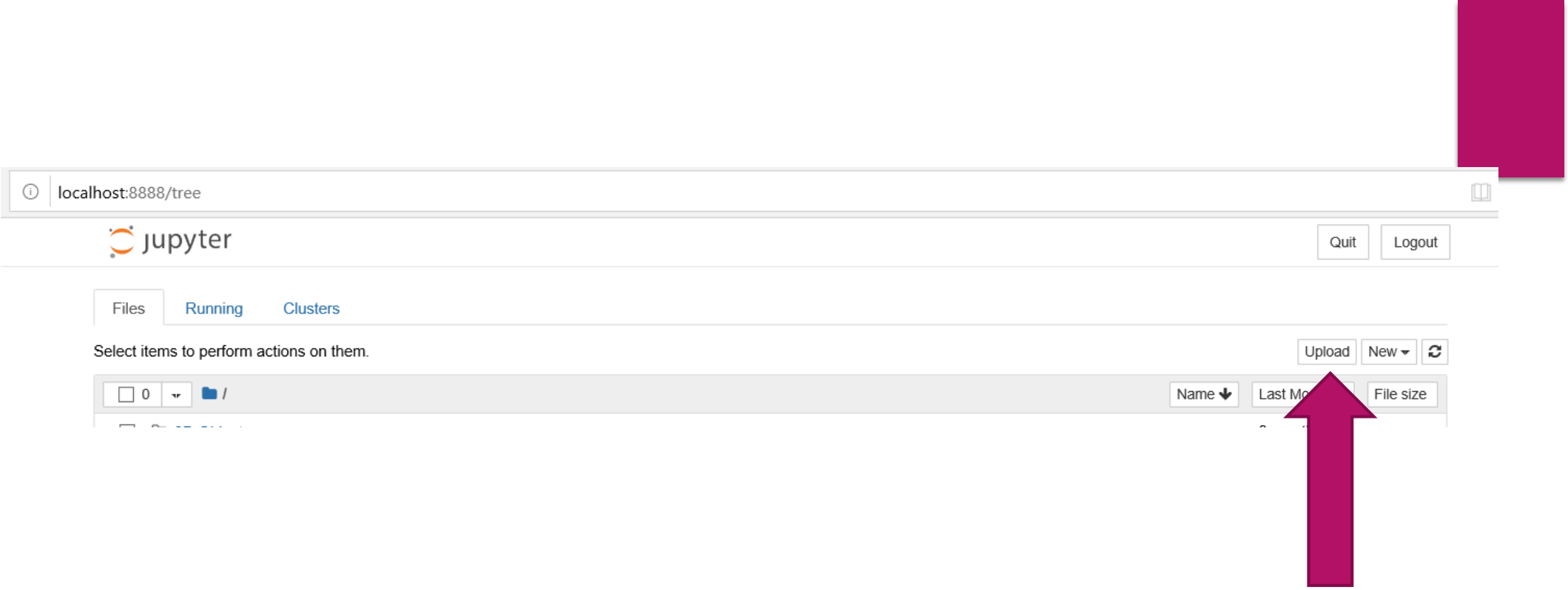
1) Anaconda Jupyter Notebook
2) Google Colab Notebook

1) Anaconda's Jupyter Notebook

1) Download the Anaconda from https://www.anaconda.com/
   i) click on Products
   ii) Click on **Anaconda Individual Edition**
   iii) Click on Download
   iv) Install it
   v) It will open a window with multiple file running options like Pycharm, Spyder,VSCODE etc and Jupyter Notebook, Install on Jupyter Notebook from it.

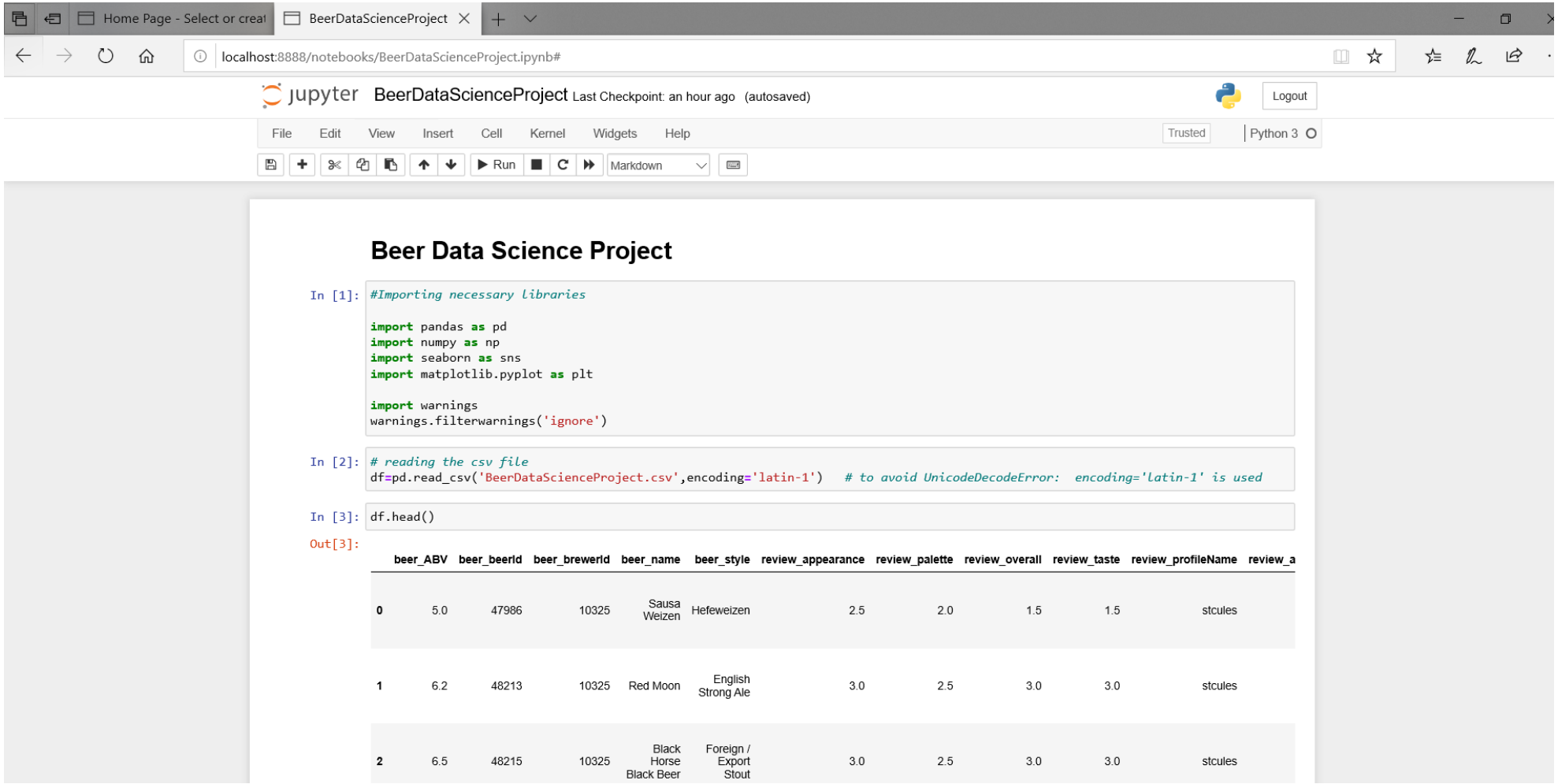
2)Download the BeerDataScience.zip folder from the github link and extract it in your local system.

3)Open the Anaconda Jupyter Notebook , click on the upload button and upload the BeerDataScienceProject.ipynb and BeerDataScienceProject.csv file from the extracted folder



4) Double click on the BeerDataScienceProject.ipynb file
it will open the new tab

5) Click on **Cell** button and select the option of **Run All** option to run the entire file



2) Google Colab Notebook

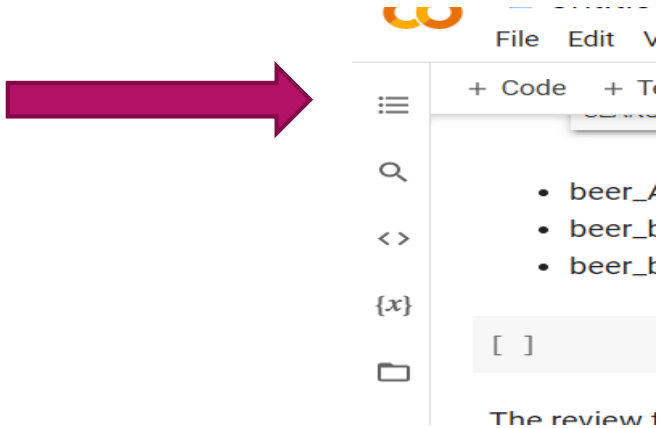
1) Copy this link and paste it in the Google Search Tab
https://colab.research.google.com/drive/1dEJiLMRaUHt7-v0RMxpra8c-OFrUJpW_?usp=sharing

You can see the entire Project already run

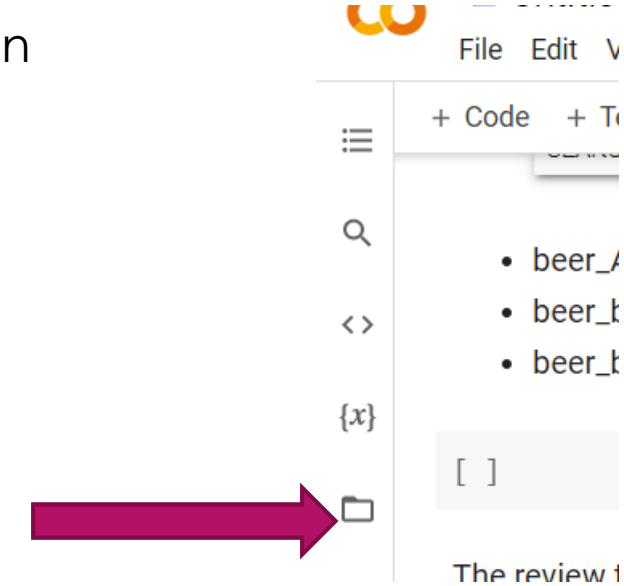
2) The file will be presented in the form of Notebook , here you have to run all the lines, So when you will start running the file it will give one warning, just click on **Run anyway**

3) Upload the CSV file in the google colab

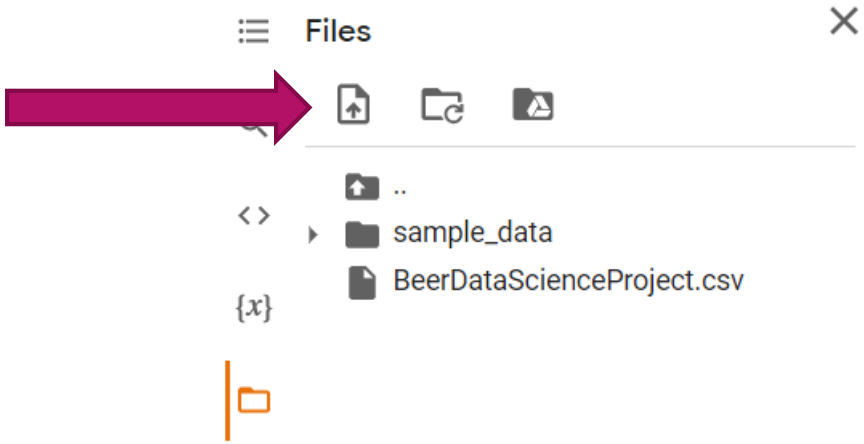
i) To upload the file click on **the table of contents** at the Top Left of screen , below **file** option adjacent to **+code** button



ii) click on **files** button



iii) click on **upload to session storage** button and upload the BeerDataScienceProject.csv file



Note : Do wait until whole file is uploaded, as the file is 395 mb long , so it will take time, otherwise code may give an error

4) Once the file is uploaded , click on **Runtime** button and then **Run All** to get all the outputs automatically

THANK YOU