Importing necessary library

```
In [ ]:  import pandas as pd
         import numpy as np
```

Turning .csv files into a Dataframe

```
In [ ]:  beers = pd.read_csv('beers.csv')
```

Checking out the original dataset

```
In [ ]:  beers.head()
```

Checking the parameters of the original dataset so I have a reference

```
In [ ]:  beers.shape
```

```
Out[ ]:  (2410, 8)
```

Figuring out the null percentage of each column

```
In [ ]:  for col in beers.columns:
             pct_missing = np.mean(beers[col].isnull())
             print('{} - {}%'.format(col, round(pct_missing*100)))

         Unnamed: 0 - 0%
         abv - 3%
         ibu - 42%
         id - 0%
         name - 0%
         style - 0%
         brewery_id - 0%
         ounces - 0%
```

Determine the datatype of each column

```
In [ ]:  beers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2410 entries, 0 to 2409
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  2410 non-null   int64
 1   abv         2348 non-null   float64
 2   ibu         1405 non-null   float64
 3   id          2410 non-null   int64
 4   name        2410 non-null   object
 5   style       2405 non-null   object
 6   brewery_id  2410 non-null   int64
 7   ounces      2410 non-null   float64
dtypes: float64(3), int64(3), object(2)
memory usage: 150.8+ KB
```

Create a new dataset with only the style and IBU columns

In [ ]:
```python
style = beers[['style', 'ibu']]
```

sort new dataset by style alphabetically

In [ ]:
```python
style = style.sort_values('style')
```

Create new dataset for each beer style

In [ ]:
```python
a1 = style.iloc[0:2,0:2]
```

In [ ]:
```python
a2 = style.iloc[2:15,0:2]
```

In [ ]:
```python
a3 = style.iloc[15:33,0:2]
```

In [ ]:
```python
a4 = style.iloc[33:166,0:2]
```

In [ ]:
```python
a5 = style.iloc[166:195,0:2]
```

In [ ]:
```python
a6 = style.iloc[195:198,0:2]
```

In [ ]:
```python
a7 = style.iloc[198:234,0:2]
```

In [ ]:
```python
a8 = style.iloc[234:342,0:2]
```

In [ ]:
```python
a9 = style.iloc[342:412,0:2]
```

In [ ]:
```python
a10 = style.iloc[412:419,0:2]
```

In [ ]:
```python
a11 = style.iloc[419:524,0:2]
```

In [ ]:
```python
a12 = style.iloc[524:526, 0:2]
```

```python
a13 = style.iloc[526:535,0:2]
```

```python
a14 = style.iloc[535:959,0:2]
```

```python
a15 = style.iloc[959:962,0:2]
```

```python
a16 = style.iloc[962,0:2]
```

```python
a17 = style.iloc[963:1208,0:2]
```

```python
a18 = style.iloc[1208:1247,0:2]
```

```python
a19 = style.iloc[1247:1344,0:2]
```

```python
a20 = style.iloc[1344:1369,0:2]
```

```python
a21 = style.iloc[1369:1437,0:2]
```

```python
a22 = style.iloc[1437:1476,0:2]
```

```python
a23 = style.iloc[1476:1490,0:2]
```

```python
a24 = style.iloc[1490:1501,0:2]
```

```python
a25 = style.iloc[1501:1507,0:2]
```

```python
a26 = style.iloc[1507:1513,0:2]
```

```python
a27 = style.iloc[1513:1524,0:2]
```

```python
a28 = style.iloc[1524:1542,0:2]
```

```python
a29 = style.iloc[1542:1566,0:2]
```

```python
a30 = style.iloc[1566:1579,0:2]
```

```python
a31 = style.iloc[1579:1590,0:2]
```

```python
a32 = style.iloc[1590:1597,0:2]
```

```python
a33 = style.iloc[1597:1604,0:2]
```

```python
a34 = style.iloc[1604,0:2]
```

```python
a35 = style.iloc[1605:1611,0:2]
```

```python
a36 = style.iloc[1611:1614,0:2]
```

```python
a37 = style.iloc[1614:1651,0:2]
```

```python
a38 = style.iloc[1651:1680,0:2]
```

```python
a39 = style.iloc[1680:1708,0:2]
```

```python
a40 = style.iloc[1708:1715,0:2]
```

```python
a41 = style.iloc[1715:1721,0:2]
```

```python
a42 = style.iloc[1721:1726,0:2]
```

```python
a43 = style.iloc[1726:1730,0:2]
```

```python
a44 = style.iloc[1730:1733,0:2]
```

```python
a45 = style.iloc[1733:1736,0:2]
```

```python
a46 = style.iloc[1736:1754,0:2]
```

```python
a47 = style.iloc[1754:1760,0:2]
```

```python
a48 = style.iloc[1760:1773,0:2]
```

```python
a49 = style.iloc[1773:1785,0:2]
```

```python
a50 = style.iloc[1785:1788,0:2]
```

```python
a51 = style.iloc[1788:1790,0:2]
```

```python
a52 = style.iloc[1790:1794,0:2]
```

```python
a53 = style.iloc[1794:1801,0:2]
```

```python
a54 = style.iloc[1801:1821,0:2]
```

```python
a55 = style.iloc[1821,0:2]
```

```python
a56 = style.iloc[1822,0:2]
```

```python
a57 = style.iloc[1823:1829,0:2]
```

```python
a58 = style.iloc[1829:1878,0:2]
```

```python
a59 = style.iloc[1878:1914,0:2]
```

```python
a60 = style.iloc[1914:1924,0:2]
```

```python
a61 = style.iloc[1924,0:2]
```

```python
a62 = style.iloc[1925:1965,0:2]
```

```python
a63 = style.iloc[1965:1974,0:2]
```

```python
a64 = style.iloc[1974:1979,0:2]
```

```python
a65 = style.iloc[1979:1991,0:2]
```

```python
a66 = style.iloc[1991:1994,0:2]
```

```python
a67 = style.iloc[1994,0:2]
```

```python
a68 = style.iloc[1995:2037,0:2]
```

```python
a69 = style.iloc[2037:2049,0:2]
```

```python
a70 = style.iloc[2049,0:2]
```

```python
a71 = style.iloc[2050:2055,0:2]
```

```python
a72 = style.iloc[2055:2060,0:2]
```

```python
a73 = style.iloc[2060:2070,0:2]
```

```python
a74 = style.iloc[2070:2074,0:2]
```

```python
a75 = style.iloc[2074:2094,0:2]
```

```python
a76 = style.iloc[2094:2124,0:2]
```

```python
a77 = style.iloc[2124:2142,0:2]
```

```python
a78 = style.iloc[2142:2144,0:2]
```

```python
a79 = style.iloc[2144,0:2]
```

```python
a80 = style.iloc[2145:2168,0:2]
```

```python
a81 = style.iloc[2168:2172,0:2]
```

```python
a82 = style.iloc[2172:2175,0:2]
```

```python
a83 = style.iloc[2175:2177,0:2]
```

```python
a84 = style.iloc[2177:2179,0:2]
```

```
In [ ]: a85 = style.iloc[2179:2190,0:2]
```

```
In [ ]: a86 = style.iloc[2190:2208,0:2]
```

```
In [ ]: a87 = style.iloc[2208:2260,0:2]
```

```
In [ ]: a88 = style.iloc[2260:2269,0:2]
```

```
In [ ]: a89 = style.iloc[2269:2284,0:2]
```

```
In [ ]: a90 = style.iloc[2284:2303,0:2]
```

```
In [ ]: a91 = style.iloc[2303:2306,0:2]
```

```
In [ ]: a92 = style.iloc[2306,0:2]
```

```
In [ ]: a93 = style.iloc[2307:2318,0:2]
```

```
In [ ]: a94 = style.iloc[2318:2338,0:2]
```

```
In [ ]: a95 = style.iloc[2338,0:2]
```

```
In [ ]: a96 = style.iloc[2339:2354,0:2]
```

```
In [ ]: a97 = style.iloc[2354:2405,0:2]
```

```
In [ ]: a98 = style.iloc[2405:2410,0:2]
```

Fill every null value with the mean IBU value for each beer style.

```
In [ ]: a1 = a1.fillna(a1.mean().astype(int))
```

```
In [ ]: a2 = a2.fillna(a2.mean().astype(int))
```

```
In [ ]: a3 = a3.fillna(a3.mean().astype(int))
```

```
In [ ]: a4 = a4.fillna(a4.mean().astype(int))
```

```
In [ ]: a5 = a5.fillna(a5.mean().astype(int))
```

```
In [ ]: a6 = a6.fillna(a6.mean().astype(int))
```

```
In [ ]: a7 = a7.fillna(a7.mean().astype(int))
```

```
In [ ]: a8 = a8.fillna(a8.mean().astype(int))
```

```
In [ ]: a9 = a9.fillna(a9.mean().astype(int))
```

```python
a10 = a10.fillna(a10.mean().astype(int))
```

```python
a11 = a11.fillna(a11.mean().astype(int))
```

```python
a12 = a12.fillna(a12.mean().astype(int))
```

```python
a13 = a13.fillna(a13.mean().astype(int))
```

```python
a14 = a14.fillna(a14.mean().astype(int))
```

```python
a15 = a15.fillna(a15.mean().astype(int))
```

```python
a17 = a17.fillna(a17.mean().astype(int))
```

```python
a18 = a18.fillna(a18.mean().astype(int))
```

```python
a19 = a19.fillna(a19.mean().astype(int))
```

```python
a20 = a20.fillna(a20.mean().astype(int))
```

```python
a21 = a21.fillna(a21.mean().astype(int))
```

```python
a22 = a22.fillna(a22.mean().astype(int))
```

```python
a23 = a23.fillna(a23.mean().astype(int))
```

```python
a24 = a24.fillna(a24.mean().astype(int))
```

```python
a25 = a25.fillna(a25.mean().astype(int))
```

```python
a26 = a26.fillna(a26.mean().astype(int))
```

```python
a27 = a27.fillna(a27.mean().astype(int))
```

```python
a28 = a28.fillna(a28.mean().astype(int))
```

```python
a29 = a29.fillna(a29.mean().astype(int))
```

```python
a30 = a30.fillna(a30.mean().astype(int))
```

```python
a31 = a31.fillna(a31.mean().astype(int))
```

```python
a32 = a32.fillna(a32.mean().astype(int))
```

```python
a33 = a33.fillna(a33.mean().astype(int))
```

```python
a35 = a35.fillna(a35.mean().astype(int))
```

```
In [ ]: a36 = a36.fillna(a36.mean().astype(int))
```

```
In [ ]: a38 = a38.fillna(a38.mean().astype(int))
```

```
In [ ]: a39 = a39.fillna(a39.mean().astype(int))
```

```
In [ ]: a40 = a40.fillna(a40.mean().astype(int))
```

```
In [ ]: a41 = a41.fillna(a41.mean().astype(int))
```

```
In [ ]: a42 = a42.fillna(a42.mean().astype(int))
```

```
In [ ]: a43 = a43.fillna(a43.mean().astype(int))
```

```
In [ ]: a44 = a44.fillna(a44.mean().astype(int))
```

```
In [ ]: a45 = a45.fillna(a45.mean().astype(int))
```

```
In [ ]: a46 = a46.fillna(a46.mean().astype(int))
```

```
In [ ]: a47 = a47.fillna(a47.mean().astype(int))
```

```
In [ ]: a48 = a48.fillna(a48.mean().astype(int))
```

```
In [ ]: a49 = a49.fillna(a49.mean().astype(int))
```

```
In [ ]: a50 = a50.fillna(a50.mean().astype(int))
```

```
In [ ]: a51 = a51.fillna(a51.mean().astype(int))
```

```
In [ ]: a52 = a52.fillna(a52.mean().astype(int))
```

```
In [ ]: a53 = a53.fillna(a53.mean().astype(int))
```

```
In [ ]: a54 = a54.fillna(a54.mean().astype(int))
```

```
In [ ]: a57 = a57.fillna(a57.mean().astype(int))
```

```
In [ ]: a58 = a58.fillna(a58.mean().astype(int))
```

```
In [ ]: a59 = a59.fillna(a59.mean().astype(int))
```

```
In [ ]: a60 = a60.fillna(a60.mean().astype(int))
```

a61 has only row that already has an IBU value

```
In [ ]: a62 = a62.fillna(a62.mean().astype(int))
```

```python
In [ ]:  a63 = a63.fillna(a63.mean().astype(int))
```

```python
In [ ]:  a64 = a64.fillna(a64.mean().astype(int))
```

```python
In [ ]:  a65 = a65.fillna(a65.mean().astype(int))
```

```python
In [ ]:  a66 = a66.fillna(a66.mean().astype(int))
```

```python
In [ ]:  a68 = a68.fillna(a68.mean().astype(int))
```

```python
In [ ]:  a69 = a69.fillna(a69.mean().astype(int))
```

```python
In [ ]:  a71 = a71.fillna(a71.mean().astype(int))
```

```python
In [ ]:  a73 = a73.fillna(a73.mean().astype(int))
```

```python
In [ ]:  a74 = a74.fillna(a74.mean().astype(int))
```

```python
In [ ]:  a75 = a75.fillna(a75.mean().astype(int))
```

```python
In [ ]:  a76 = a76.fillna(a76.mean().astype(int))
```

```python
In [ ]:  a77 = a77.fillna(a77.mean().astype(int))
```

```python
In [ ]:  a78 = a78.fillna(a78.mean().astype(int))
```

```python
In [ ]:  a80 = a80.fillna(a80.mean().astype(int))
```

```python
In [ ]:  a81 = a81.fillna(a81.mean().astype(int))
```

```python
In [ ]:  a82 = a82.fillna(a82.mean().astype(int))
```

```python
In [ ]:  a84 = a84.fillna(a84.mean().astype(int))
```

```python
In [ ]:  a85 = a85.fillna(a85.mean().astype(int))
```

```python
In [ ]:  a86 = a86.fillna(a86.mean().astype(int))
```

```python
In [ ]:  a87 = a87.fillna(a87.mean().astype(int))
```

```python
In [ ]:  a88 = a88.fillna(a88.mean().astype(int))
```

```python
In [ ]:  a89 = a89.fillna(a89.mean().astype(int))
```

```python
In [ ]:  a90 = a90.fillna(a90.mean().astype(int))
```

```python
In [ ]:  a93 = a93.fillna(a93.mean().astype(int))
```

```
In [ ]:  a94 = a94.fillna(a94.mean().astype(int))
```

```
In [ ]:  a96 = a96.fillna(a96.mean().astype(int))
```

```
In [ ]:  a97 = a97.fillna(a97.mean().astype(int))
```

A83, A92 & A95 have a null IBU with just one or two records of their unique style.

A98 is 'null' style

Combine every new individual dataset with updated null values together

```
In [ ]:  style1 = pd.concat([a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,a
```

Join the original dataset with the updated one

```
In [ ]:  ibu = beers.join(style1, lsuffix='style')
```

Verify the new dataframe has the same amount of rows as original

```
In [ ]:  ibu.shape
```

Delete unncessary columns

```
In [ ]:  ibu = ibu.drop(['Unnamed: 0','ibustyle','stylestyle'], axis=1)
```

Fill the few missing 'abv' values with the total mean of the 'abv' column

```
In [ ]:  ibu['abv'].fillna((ibu['abv'].mean()), inplace=True)
```

Round every float to just two decimal values

```
In [ ]:  ibu = ibu.round(decimals=2)
```

Turn 'abv' values to percent

```
In [ ]:  ibu['abv'] = ibu['abv'] * 100
```

Eliminate every decimal point from 'abv' column

```
In [ ]:  ibu['abv'] = ibu['abv'].round(decimals=0)
```

Convert 'abv' column to strings and add a percent sign

```
In [ ]:  ibu['abv'] = ibu['abv'].astype(str) + '%'
```

Rename columns for clarification

```
In [ ]: ibu.rename(columns = {'abv': 'Alcohol %'}, inplace=True)
```

```
In [ ]: ibu.rename(columns = {'ibu': 'Bitterness Scale (1-120)'}, inplace=True)
```

Sort columns in a more user-friendly order

```
In [ ]: ibu = ibu[['name','style','Alcohol %','Bitterness Scale (1-120)','ounces','b
```

Change these rows because due to lower case they weren't in alphabetical order

```
In [ ]: ibu.replace("oSKAr the G'Rauch", "Oskar the Grauch", inplace=True)
```

```
In [ ]: ibu.replace("the Kimmie, the Yink and the Holy Gose","The Kimmie, the Yink a
```

Sort by alphabetical name for convenience

```
In [ ]: ibu = ibu.sort_values('name')
```

Just dropping all duplicates didn't erase many because of unique 'id' values so I set specific columns to target.

```
In [ ]: ibu = ibu.drop_duplicates(subset=('name','ounces'), keep='first')
```

Checking to see what percent of null values remain in each column.

```
In [ ]: for col in ibu.columns:
            pct_missing = np.mean(ibu[col].isnull())
            print('{} - {}%'.format(col, round(pct_missing*100)))
```

Find all the rows where null values still exist

```
In [ ]: ibu[ibu.isnull().any(axis=1)]
```

Dropping everything left with null values. It would be speculative to fill in the style and IBU for beers that only have a name and ABV.

Cider, mead, and shandy are unnecessary since they're not beer.

```
In [ ]: ibu = ibu.dropna()
```

Taking a look at the final product

```
In [ ]: ibu
```