

This data analysis is on a dataset from Kaggle which is centered on data gotten from walmart stores on sales of coffee, stating the coffee types, prices, ratings, reviews, sellers and weights

First, i would import the libraries needed for this analysis

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from datetime import datetime
import dateutil.parser

pd.set_option('display.max_columns', 15)
pd.set_option('expand_frame_repr', True)

sns.set_palette('hls')
%matplotlib inline
```

Then i import my dataframe which i have named 'pers-data' for now

In [2]: *#importing the dataframe to be analysed and setting the index to the first column*

```
pers_data= pd.read_csv('coffee-listings-from-all-walmart-stores.csv', index_col=''
pers_data
```

Out[2]:

	coffee_type	rating	reviews	seller_name	thumbnail	pr
	title					
	<b>folgers</b> <b>classic</b> <b>roast</b> <b>ground</b> <b>coffee,</b> <b>40.3-</b> <b>ounce</b>	classic roast	3.8	93	walmart.com	<a href="https://i5.walmartimages.com/asr/1fbbd523-8554...">https://i5.walmartimages.com/asr/1fbbd523-8554...</a> 13
	<b>café</b> <b>bustelo,</b> <b>espresso</b> <b>style dark</b> <b>roast</b> <b>ground</b> <b>coffee,</b> <b>vacuum-</b> <b>packed</b> <b>10 oz.</b> <b>brick</b>	espresso,dark roast	4.7	914	walmart.com	<a href="https://i5.walmartimages.com/asr/99a53df0-0471...">https://i5.walmartimages.com/asr/99a53df0-0471...</a> 3
	<b>folgers</b> <b>classic</b> <b>roast</b> <b>ground</b> <b>coffee,</b> <b>medium</b> <b>roast</b> <b>coffee,</b> <b>25.9</b> <b>ounce</b> <b>canister</b>	medium roast,classic roast	4.4	740	walmart.com	<a href="https://i5.walmartimages.com/asr/e6aba325-608e...">https://i5.walmartimages.com/asr/e6aba325-608e...</a> 9
	<b>maxwell</b> <b>house</b> <b>original</b> <b>roast</b> <b>ground</b> <b>coffee,</b> <b>42.5 oz.</b> <b>canister</b>	NaN	4.8	1321	walmart.com	<a href="https://i5.walmartimages.com/asr/a5be9586-b75d...">https://i5.walmartimages.com/asr/a5be9586-b75d...</a> 9
	<b>great</b> <b>value</b> <b>classic</b> <b>roast</b> <b>medium</b> <b>ground</b> <b>coffee,</b> <b>value</b> <b>size, 48</b> <b>oz</b>	classic roast	4.7	1598	walmart.com	<a href="https://i5.walmartimages.com/asr/de42310c-4cd6...">https://i5.walmartimages.com/asr/de42310c-4cd6...</a> 9
	...	...	...	...	...	...

	coffee_type	rating	reviews	seller_name	thumbnail	pr
title						
xhao reusable capsules easy to clean easy to use 304 stainless steel stainless steel reusable capsules for coffee	NaN	0.0	0	joybuy selection	<a href="https://i5.walmartimages.com/asr/5aa587bd-9757...">https://i5.walmartimages.com/asr/5aa587bd-9757...</a>	13
cafe yaucono espresso dark roast ground coffee from puerto rico - 8.8 oz	espresso,dark roast	0.0	0	coffee & filters direct, inc	<a href="https://i5.walmartimages.com/asr/073bdc20-ad75...">https://i5.walmartimages.com/asr/073bdc20-ad75...</a>	10
starbucks dark roast coffee with 2x caffeine, k-cup coffee pods, 100% arabica, 12 ct	arabica,dark roast	4.3	5036	walmart.com	<a href="https://i5.walmartimages.com/asr/242294ac-0c7d...">https://i5.walmartimages.com/asr/242294ac-0c7d...</a>	12
starbucks holiday blend, medium roast k- cup coffee pods, 100% arabica, limited edition, 22 ct	arabica,medium roast	4.5	397	walmart.com	<a href="https://i5.walmartimages.com/asr/3a714e3e-a30e...">https://i5.walmartimages.com/asr/3a714e3e-a30e...</a>	17
super coffee vanilla latte iced coffee bottle, 12 fl oz	iced coffee,latte	4.3	2791	walmart.com	<a href="https://i5.walmartimages.com/asr/9f4beb5c-35c7...">https://i5.walmartimages.com/asr/9f4beb5c-35c7...</a>	2

1400 rows × 8 columns



The dataframe is named "pers\_data and has 1400 rows and 8 columns

```
In [3]: #checking the whole column headings of the dataframe  
pers_data.columns.to_list()
```

```
Out[3]: ['coffee_type',  
         'rating',  
         'reviews',  
         'seller_name',  
         'thumbnail',  
         'price',  
         'weight',  
         'weight_formatted_to_gramms']
```

First glance at this dataframe, i want to answer this questions and gain the following insights from it:

Firstly, i would like to see the weightiest coffee, then the best rated coffee, the most sold coffee and by whom (seller's name), the coffee with the highest reviews, the relationship between the weight and the price or the ratings and the price ?? Does the weight inform the price of the coffee and all other insights i can gain from the dataset. But these will guide my thoughts and visualization.

Okay, let's get to it

Firstly, i would start by viewing my data, checking the types of data present in the dataset, get information and description of the data frame

The dataset has 8 columns with the above column headings

In [4]: #checking the types of data each column has, to gain more understanding of the data

```
pers_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1400 entries, folgers classic roast ground coffee, 40.3-ounce to super coffee vanilla latte iced coffee bottle, 12 fl oz
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   coffee_type      1121 non-null    object  
 1   rating           1400 non-null    float64 
 2   reviews          1400 non-null    int64  
 3   seller_name      1400 non-null    object  
 4   thumbnail         1400 non-null    object  
 5   price            1400 non-null    float64 
 6   weight           1400 non-null    object  
 7   weight_formatted_to_gramms 1400 non-null    float64 
dtypes: float64(3), int64(1), object(4)
memory usage: 98.4+ KB
```

The coffee\_type column is of the object type i.e it is a string. The rating is type float(which spans across 0-5, 0 being the lowest and 5, the highest. The reviews column has the integer type which depicts the numbers of reviews per coffee type. The seller\_name column has the object type i.e the names of the sellers. The thumbnail column has the object type. The price column is a float. The weight column is an object, hence the weight is formatted to grams to ensure uniformity. The weight\_formatted\_to\_grams column is a float. Hence, we have 3 floats column, 1 integer column and 4 object columns.

In [5]: #checking the description of the numerical data and the measures of central tendency

```
pers_data.describe()
```

Out[5]:

	rating	reviews	price	weight_formatted_to_grams
<b>count</b>	1400.000000	1400.000000	1400.000000	1400.000000
<b>mean</b>	3.982643	440.853571	14.041343	621.391786
<b>std</b>	1.518037	879.351997	10.257832	369.564693
<b>min</b>	0.000000	0.000000	0.000000	0.000000
<b>25%</b>	4.300000	16.000000	7.950000	340.200000
<b>50%</b>	4.600000	136.000000	12.735000	567.000000
<b>75%</b>	4.800000	604.500000	16.990000	850.500000
<b>max</b>	5.000000	15148.000000	77.090000	2835.000000

This describes the numerical data in the data set we have 1400 observations in general with the minimum,maximum,mean,standard deviation, 25th percentile, 50th and 75th percentile for each numerical data shown above

In [7]: *#checking the number of categorical data types that we have*

```
categorical = pers_data.dtypes[pers_data.dtypes=='object'].index
pers_data[categorical].describe()
```

Out[7]:

	coffee_type	seller_name	thumbnail	weight
<b>count</b>	1121	1400	1400	1400
<b>unique</b>	120	126	979	168
<b>top</b>	medium roast	walmart.com	https://i5.walmartimages.com/asr/6c3ab95d-29a5...	not mentioned
<b>freq</b>	313	1060	32	169

The coffee\_type has 1121 counts which depicts that it has 279 missing values but the seller\_name is complete. coffee\_type has 120 unique counts while seller\_name has 126, the top seller is walmart which occurs 1060 times in the data while coffee\_type has medium roast to be the top coffee and it has a frequency of 313 occurrences. It is important to note that i ignored the thumbnail and weight column, this is because the thumbnail column has no direct influence on the data set because it is a link that only confirms the sellers address and weight is already formatted to grams thereby the column is not useful. in a bid to simplify my data for better visualization, i will drop these columns

Now, i will clean my data, starting with the missing values, the columns not needed and the zero values

In [9]: *#Finding the missing values*

```
def missing_num(x):
    return sum(x.isnull())
print('Missing values per column')

#applying our fxn by column
#check sum of missing values by column and get the columns where the sum isn't zero

print(pers_data.apply(missing_num, axis = 0).where(lambda x: x!=0).dropna())
```

```
Missing values per column
coffee_type    279.0
dtype: float64
```

The only column with missing values is the coffee\_type which has 279 missing values. i will drop them to make a more wholesome dataframe

In [11]: *#dropping the missing values*

```
new_data = pers_data.dropna()
new_data
```

Out[11]:

	coffee_type	rating	reviews	seller_name	thumbnail
title					
<b>folgers</b>					
<b>classic</b>					
<b>roast</b>					
<b>ground</b>	classic roast	3.8	93	walmart.com	<a href="https://i5.walmartimages.com/asr/1fbbd523-8554...">https://i5.walmartimages.com/asr/1fbbd523-8554...</a>
<b>coffee,</b>					
<b>40.3-</b>					
<b>ounce</b>					
<b>café</b>					
<b>bustelo,</b>					
<b>espresso</b>					
<b>style dark</b>					
<b>roast</b>					
<b>ground</b>	espresso,dark	4.7	914	walmart.com	<a href="https://i5.walmartimages.com/asr/99a53df0-0471...">https://i5.walmartimages.com/asr/99a53df0-0471...</a>
<b>coffee,</b>	roast				
<b>vacuum-</b>					
<b>packed</b>					
<b>10 oz</b>					

My new dataframe after missing values were dropped is called "new\_data". However, i need to also dropped some more arbitrary data

In [12]: *#to confirm that the missing values were really dropped*

```
new_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1121 entries, folgers classic roast ground coffee, 40.3-ounce to super c
offee vanilla latte iced coffee bottle, 12 fl oz
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   coffee_type      1121 non-null   object 
 1   rating           1121 non-null   float64
 2   reviews          1121 non-null   int64  
 3   seller_name      1121 non-null   object 
 4   thumbnail        1121 non-null   object 
 5   price            1121 non-null   float64
 6   weight           1121 non-null   object 
 7   weight_formatted_to_gramms 1121 non-null   float64
dtypes: float64(3), int64(1), object(4)
memory usage: 78.8+ KB
```

In [15]: *#selecting the columns that i need and putting them together in a dataframe*

```
cols = ['coffee_type', 'rating', 'reviews', 'seller_name', 'price', 'weight_formatted_to_gramms']
data = new_data[cols]
data
```

Out[15]:

title	coffee_type	rating	reviews	seller_name	price	weight_formatted_to_gramms
<b>folgers</b> <b>classic roast</b> <b>ground</b> <b>coffee, 40.3-</b> <b>ounce</b>	classic roast	3.8	93	walmart.com	13.92	1142.5
<b>café bustelo,</b> <b>espresso</b> <b>style dark</b> <b>roast ground</b> <b>coffee,</b> <b>vacuum-</b> <b>packed 10</b> <b>oz. brick</b>	espresso,dark roast	4.7	914	walmart.com	3.76	283.5
<b>folgers</b> <b>classic roast</b> <b>ground</b> <b>coffee,</b> <b>medium</b> <b>roast coffee,</b> <b>25.9 ounce</b> <b>canister</b>	medium roast,classic roast	4.4	740	walmart.com	9.97	734.3
<b>great value</b> <b>classic roast</b> <b>medium</b> <b>ground</b> <b>coffee, value</b> <b>size, 48 oz</b>	classic roast	4.7	1598	walmart.com	9.98	1360.8
<b>great value</b> <b>classic roast</b> <b>medium</b> <b>ground</b> <b>coffee, 30.5</b> <b>oz</b>	classic roast	4.2	263	walmart.com	7.98	864.7
...	...	...	...	...	...	...
<b>nescafe</b> <b>classico</b> <b>colombia</b> <b>medium</b> <b>roast instant</b> <b>coffee, 6 oz</b>	instant coffee,medium roast	4.7	122	walmart.com	6.28	272.2
<b>cafe</b> <b>yaucono</b> <b>espresso</b> <b>dark roast</b> <b>ground</b> <b>coffee from</b> <b>puerto rico -</b> <b>8.8 oz</b>	espresso,dark roast	0.0	0	coffee & filters direct, inc	10.99	249.5

	coffee_type	rating	reviews	seller_name	price	weight_formatted_to_gramms
title						
<b>starbucks</b> <b>dark roast</b> <b>coffee with</b> <b>2x caffeine,</b> <b>k-cup coffee</b> <b>pods, 100%</b> <b>arabica, 12</b> <b>ct</b>	arabica,dark roast	4.3	5036	walmart.com	12.98	340.2
<b>starbucks</b> <b>holiday</b> <b>blend,</b> <b>medium</b> <b>roast k-cup</b> <b>coffee pods,</b> <b>100%</b> <b>arabica,</b> <b>limited</b> <b>edition, 22 ct</b>	arabica,medium roast	4.5	397	walmart.com	17.72	623.7
<b>super coffee</b> <b>vanilla latte</b> <b>iced coffee</b> <b>bottle, 12 fl</b> <b>oz</b>	iced coffee,latte	4.3	2791	walmart.com	2.98	623.7

1121 rows × 6 columns

Having dropped 2 columns and 279 missing values, lets check the details of our data again

In [16]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
Index: 1121 entries, folgers classic roast ground coffee, 40.3-ounce to super c
offee vanilla latte iced coffee bottle, 12 fl oz
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
---  -- 
 0   coffee_type      1121 non-null   object 
 1   rating           1121 non-null   float64
 2   reviews          1121 non-null   int64  
 3   seller_name      1121 non-null   object 
 4   price            1121 non-null   float64
 5   weight_formatted_to_gramms 1121 non-null   float64
dtypes: float64(3), int64(1), object(2)
memory usage: 61.3+ KB
```

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

Out[17]:

	rating	reviews	price	weight_formatted_to_gramms
<b>count</b>	1121.000000	1121.000000	1121.000000	1121.000000
<b>mean</b>	4.143711	462.358608	14.016592	623.041302
<b>std</b>	1.302435	941.986003	10.668796	361.545221
<b>min</b>	0.000000	0.000000	0.000000	8.800000
<b>25%</b>	4.300000	20.000000	7.480000	340.200000
<b>50%</b>	4.600000	156.000000	12.580000	623.700000
<b>75%</b>	4.700000	622.000000	16.990000	850.500000
<b>max</b>	5.000000	15148.000000	77.090000	2835.000000

Duplicate data needs to be dropped so as to avoid error in the visualization

In [97]: `data1 = data.drop_duplicates()  
data1`

Out[97]:

	coffee_type	rating	reviews	seller_name	price	weight_formatted_to_gramms
title						
<b>folgers</b>						
<b>classic roast</b>						
<b>ground coffee, 40.3-ounce</b>	classic roast	3.8	93	walmart.com	13.92	1142.5
<b>café bustelo, espresso style dark roast ground coffee, vacuum-packed 10 oz. brick</b>	espresso,dark roast	4.7	914	walmart.com	3.76	283.5
<b>folgers</b>						
<b>classic roast</b>						
<b>ground coffee, medium</b>	medium					

Duplicate values have been dropped, reducing our dataframe to 761 values

since the minimum rating should be one (on a rating basis of 1-5) it means there were some coffee that were not rated, lets see them and determine if to drop them or not

```
In [98]: data1.loc[data1.rating==0]
```

title							
kingdom coffee - african coffee, medium-light roast, ground coffee, 10 oz	light roast	0.0	0	walmart.com	14.00	2	
nobletree coffee dromedaire cuvee, whole bean, medium-light roast, 5 lb. bag	light roast	0.0	0	walmart.com	49.99	2	

```
In [99]: len(data1.loc[data1.rating==0])
```

Out[99]: 58

There are about 58 observations with zero ratings and reviews, this is going to hinder the visualization of this data, so they would be dropped going forward

In [100]: *#dropped the data with zero rating*

```
suc_data = data1.drop(data1[data1.rating==0].index)
suc_data
```

Out[100]:

	coffee_type	rating	reviews	seller_name	price	weight_formatted_to_gramms
<b>title</b>						

<b>folgers classic roast ground coffee, 40.3-ounce</b>	classic roast	3.8	93	walmart.com	13.92	1142.5
<b>café bustelo, espresso style dark roast ground coffee, vacuum-packed 10 oz. brick</b>	espresso,dark roast	4.7	914	walmart.com	3.76	283.5
<b>folgers classic roast ground coffee, medium roast coffee, 25.9 ounce canister</b>	medium roast,classic roast	4.4	740	walmart.com	9.97	734.3
<b>great value classic roast medium ground coffee, value size, 48 oz</b>	classic roast	4.7	1598	walmart.com	9.98	1360.8
<b>great value classic roast medium ground coffee, 30.5 oz</b>	classic roast	4.2	263	walmart.com	7.98	864.7
...	...	...	...	...	...	...
<b>the original donut shop pumpkin caramel cheesecake latte, keurig single serve k-cup pods</b>	caramel,latte	4.1	55	walmart.com	7.92	510.3
<b>nescafe classico colombia medium roast instant coffee, 6 oz</b>	instant coffee,medium roast	4.7	122	walmart.com	6.28	272.2
<b>starbucks dark roast coffee with 2x caffeine, k-cup coffee pods, 100% arabica, 12 ct</b>	arabica,dark roast	4.3	5036	walmart.com	12.98	340.2

```
coffee_type rating reviews seller_name price weight_formatted_to_gramms
```

title						
<b>starbucks</b> <b>holiday blend,</b> <b>medium roast</b> <b>k-cup coffee</b> <b>pods, 100%</b> <b>arabica, limited</b> <b>edition, 22 ct</b>	arabica,medium roast	4.5	397	walmart.com	17.72	623.7
<b>super coffee</b> <b>vanilla latte</b> <b>iced coffee</b> <b>bottle, 12 fl oz</b>	iced coffee,latte	4.3	2791	walmart.com	2.98	623.7

703 rows × 6 columns

we have reduced the data into a more compact one for a better visualization. Although for visualization that doesn't require reviews and rating the initial data 'data1' might be used.

Now that we dropped the coffee with zero ratings and reviews,lets check the dataframe again

In [101]: `suc_data.info()`

```
<class 'pandas.core.frame.DataFrame'>
Index: 703 entries, folgers classic roast ground coffee, 40.3-ounce to super co
ffee vanilla latte iced coffee bottle, 12 fl oz
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   coffee_type      703 non-null    object 
 1   rating           703 non-null    float64
 2   reviews          703 non-null    int64  
 3   seller_name      703 non-null    object 
 4   price            703 non-null    float64
 5   weight_formatted_to_gramms 703 non-null    float64
dtypes: float64(3), int64(1), object(2)
memory usage: 38.4+ KB
```

```
In [102]: suc_data.describe()
```

Out[102]:

	rating	reviews	price	weight_formatted_to_gramms
<b>count</b>	703.000000	703.000000	703.000000	703.000000
<b>mean</b>	4.537127	549.924609	13.837553	589.038549
<b>std</b>	0.357263	1140.337577	11.146943	394.152887
<b>min</b>	1.800000	1.000000	0.000000	8.800000
<b>25%</b>	4.500000	50.000000	6.380000	340.200000
<b>50%</b>	4.600000	208.000000	11.490000	453.600000
<b>75%</b>	4.700000	601.500000	17.920000	680.400000
<b>max</b>	5.000000	15148.000000	77.090000	2835.000000

There are some observations on some coffee price as zero, i will view them and think of them as complementary coffee given free of charge to loyal customers and continue with them in the visualization

In [75]: `data.loc[data.price==0]`

Out[75]:

title	coffee_type	rating	reviews	seller_name	price	weight_formatted_
<b>dunkin'</b> <b>original</b> <b>blend coffee,</b> <b>medium</b> <b>roast coffee,</b> <b>12 oz bag</b>	medium roast	4.8	495	walmart.com	0.0	
<b>dunkin'</b> <b>original</b> <b>blend coffee,</b> <b>medium</b> <b>roast coffee,</b> <b>12 oz bag</b>	medium roast	4.8	495	walmart.com	0.0	
<b>great value</b> <b>donut shop</b> <b>100%</b> <b>arabica</b> <b>medium</b> <b>roast ground</b> <b>coffee pods,</b> <b>12 ct</b>	arabica,medium roast	4.5	673	walmart.com	0.0	
<b>folgers</b> <b>classic roast</b> <b>instant</b> <b>coffee</b> <b>crystals, 8-</b> <b>ounce jar</b>	instant coffee,classic roast	4.6	13	walmart.com	0.0	
<b>caribou</b> <b>coffee,</b> <b>mahogany,</b> <b>dark roast</b> <b>ground</b> <b>coffee, 20 oz</b> <b>bag</b>	dark roast	4.8	65	walmart.com	0.0	
<b>tim hortons</b> <b>ground</b> <b>coffee, 100%</b> <b>arabica</b> <b>medium</b> <b>roast, 32.8</b> <b>oz canister</b>	arabica,medium roast	4.8	487	walmart.com	0.0	
<b>starbucks</b> <b>breakfast</b> <b>blend,</b> <b>medium</b> <b>roast k-cup</b> <b>coffee pods,</b> <b>100%</b> <b>arabica, 22</b> <b>ct</b>	arabica,medium roast	4.6	1107	walmart.com	0.0	
<b>dunkin'</b> <b>original</b> <b>blend coffee,</b> <b>medium</b> <b>roast coffee,</b> <b>12 oz bag</b>	medium roast	4.8	495	walmart.com	0.0	

	coffee_type	rating	reviews	seller_name	price	weight_formatted_
title						
<b>great value 100% arabica colombian medium dark roast ground coffee pods, 12 ct</b>	colombian,arabica,medium dark,medium dark roas...	4.6	2097	walmart.com	0.0	
<b>great value 100% arabica breakfast blend medium roast ground coffee pods, 12 ct</b>	arabica,medium roast	4.6	1838	walmart.com	0.0	
<b>great value 100% arabica french dark roast ground coffee pods, 12 ct</b>	french,arabica,dark roast	4.6	1893	walmart.com	0.0	
<b>starbucks pike place roast, medium roast k-cup coffee pods, 100% arabica, 22 ct</b>	arabica,medium roast	4.6	2686	walmart.com	0.0	
<b>the original donut shop regular keurig single-serve k-cup pods, medium roast coffee, 24 count</b>	medium roast	4.4	2937	walmart.com	0.0	
<b>starbucks veranda blend, starbucks blonde roast k-cup coffee pods, 100% arabica, 22 ct</b>	blonde roast,arabica	4.6	1824	walmart.com	0.0	
<b>dunkin' original blend coffee, medium roast coffee, 12 oz bag</b>	medium roast	4.8	495	walmart.com	0.0	

	coffee_type	rating	reviews	seller_name	price	weight_formatted_
title						
dunkin' original blend coffee, medium roast coffee, 12 oz bag	medium roast	4.8	495	walmart.com	0.0	
green mountain coffee hazelnut decaf coffee, keurig single-serve k-cup pods, light roast, 24 count	light roast	4.6	703	walmart.com	0.0	
dunkin' original blend coffee, medium roast coffee, 12 oz bag	medium roast	4.8	495	walmart.com	0.0	
green mountain coffee roasters brew over ice vanilla caramel, single serve keurig k-cup pods, flavored iced coffee, 24 count	caramel, iced coffee	4.6	304	walmart.com	0.0	
dunkin' original blend coffee, medium roast coffee, 12 oz bag	medium roast	4.8	495	walmart.com	0.0	
green mountain coffee roasters maple pecan coffee, keurig k-cup pods, light roast, 24 count	light roast	4.7	1147	walmart.com	0.0	
peet's coffee house blend, dark roast whole bean coffee, 12 oz bag	dark roast	4.7	22	walmart.com	0.0	

	coffee_type	rating	reviews	seller_name	price	weight_formatted_
title						
dunkin' original blend coffee, medium roast coffee, 12 oz bag	medium roast	4.8	495	walmart.com	0.0	
peet's coffee luminosa breakfast blend, light roast ground coffee, 10.5 oz bag	light roast	4.5	227	walmart.com	0.0	
peet's coffee single origin colombia, dark roast ground coffee, 10.5 oz bag	dark roast	4.5	374	walmart.com	0.0	
dunkin' original blend coffee, medium roast coffee, 12 oz bag	medium roast	4.8	495	walmart.com	0.0	
dunkin' original blend coffee, medium roast coffee, 12 oz bag	medium roast	4.8	495	walmart.com	0.0	
green mountain coffee roasters brew over ice classic black, single serve keurig k-cup pods, medium roast iced coffee, 24 ct	iced coffee,medium roast	4.6	180	pnwb office products	0.0	
dunkin' original blend coffee, medium roast coffee, 12 oz bag	medium roast	4.8	495	walmart.com	0.0	
dunkin' original blend coffee, medium roast coffee, 12 oz bag	medium roast	4.8	495	walmart.com	0.0	

	coffee_type	rating	reviews	seller_name	price	weight_formatted_
title						
<b>door county coffee bourbon vanilla cream, bourbon &amp; vanilla flavored specialty arabica coffee, medium roast, ground, 10oz bag</b>	arabica,medium roast	4.9	43	door county coffee	0.0	
<b>dunkin' original blend coffee, medium roast coffee, 12 oz bag</b>	medium roast	4.8	495	walmart.com	0.0	
<b>eight o'clock, the original medium roast, ground coffee, 12 oz bag</b>	medium roast	4.9	55	shaan super store	0.0	
<b>dunkin' original blend coffee, medium roast coffee, 12 oz bag</b>	medium roast	4.8	495	walmart.com	0.0	
<b>starbucks morning joe, ground coffee, dark roast, 12 oz</b>	dark roast	4.7	2383	walmart.com	0.0	
<b>dunkin' original blend coffee, medium roast coffee, 12 oz bag</b>	medium roast	4.8	495	walmart.com	0.0	
<b>dunkin' original blend coffee, medium roast coffee, 12 oz bag</b>	medium roast	4.8	495	walmart.com	0.0	

	coffee_type	rating	reviews	seller_name	price	weight_formatted_
title						
<b>lavazza espresso deciso nespresso originalline coffee capsules, dark roast (10 count)</b>	espresso,dark roast	3.7	17	walmart.com	0.0	
<b>lavazza avvolgente nespresso coffee capsules, 10 count</b>	espresso,espresso coffee	4.8	19	walmart.com	0.0	
<b>lavazza decaffeinato ricco espresso nespresso coffee capsules, dark roast (10 count)</b>	espresso,espresso coffee,dark roast	4.4	18	myofficeinnovations	0.0	
<b>dunkin' original blend coffee, medium roast coffee, 12 oz bag</b>	medium roast	4.8	495	walmart.com	0.0	
<b>peet's coffee k-cup pods, dark roasted, house blend, 22 ct</b>	dark roast	4.6	125	walmart.com	0.0	
<b>hills bros. cappuccino white chocolate caramel medium roast instant coffee, 16 oz</b>	caramel,cappuccino,instant coffee,medium roast	4.7	247	overstock drugstore	0.0	
<b>dunkin' original blend coffee, medium roast coffee, 12 oz bag</b>	medium roast	4.8	495	walmart.com	0.0	

	coffee_type	rating	reviews	seller_name	price	weight_formatted_
title						
<b>peet's coffee</b> k-cup pods, french roast dark roast (24 count) <b>single serve</b> pods compatible with keurig brewers	french,french roast,dark roast	4.7	311	walmart.com	0.0	
<b>1850 lantern</b> glow coffee, light roast ground coffee, 12 oz	light roast	4.3	8446	bestsource officesupplies	0.0	
<b>dunkin'</b> original blend coffee, medium roast coffee, 12 oz bag	medium roast	4.8	495	walmart.com	0.0	

Renaming my data for easier reference because it looks like i have a cleaner data ready for exploration

In [103]: `final_data= suc_data`  
`final_data`

Out[103]:

	coffee_type	rating	reviews	seller_name	price	weight_formatted_to_gramms
title						
<b>folgers</b> classic roast ground coffee, 40.3- ounce	classic roast	3.8	93	walmart.com	13.92	1142.5
<b>café bustelo,</b> espresso style dark roast ground coffee, vacuum- packed 10 oz. brick	espresso,dark roast	4.7	914	walmart.com	3.76	283.5
<b>folgers</b> classic roast ground coffee,	medium					

lets check this data information and description again....for the last time perhaps?

In [88]: `final_data.info()`

```
<class 'pandas.core.frame.DataFrame'>
Index: 703 entries, folgers classic roast ground coffee, 40.3-ounce to super co
ffee vanilla latte iced coffee bottle, 12 fl oz
Data columns (total 6 columns):
 #   Column           Non-Null Count Dtype  
--- 
 0   coffee_type      703 non-null   object  
 1   rating           703 non-null   float64 
 2   reviews          703 non-null   int64   
 3   seller_name      703 non-null   object  
 4   price            703 non-null   float64 
 5   weight_formatted_to_gramms 703 non-null   float64 
dtypes: float64(3), int64(1), object(2)
memory usage: 38.4+ KB
```

In [89]: `final_data.describe()`

Out[89]:

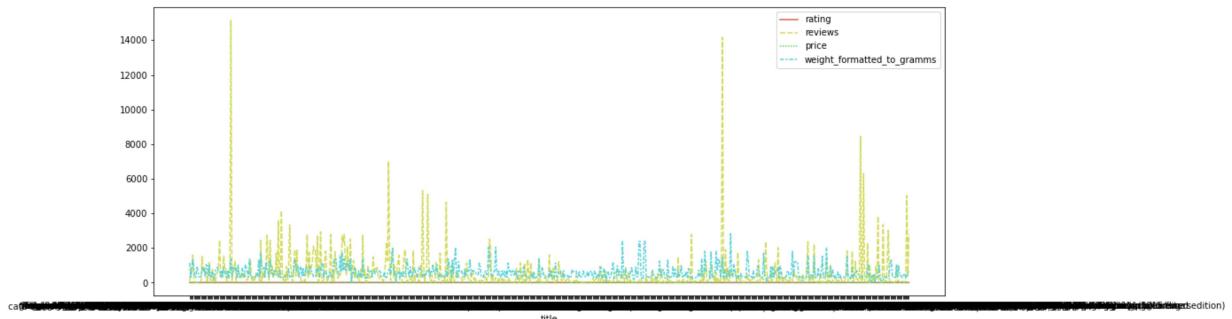
	rating	reviews	price	weight_formatted_to_gramms
<b>count</b>	703.000000	703.000000	703.000000	703.000000
<b>mean</b>	4.537127	549.924609	13.837553	589.038549
<b>std</b>	0.357263	1140.337577	11.146943	394.152887
<b>min</b>	1.800000	1.000000	0.000000	8.800000
<b>25%</b>	4.500000	50.000000	6.380000	340.200000
<b>50%</b>	4.600000	208.000000	11.490000	453.600000
<b>75%</b>	4.700000	601.500000	17.920000	680.400000
<b>max</b>	5.000000	15148.000000	77.090000	2835.000000

We can now start to explore our data and i will like to start with visualization using a line plot, to check for trends in the data

In [90]: `#lets set the figure size to be somewhat large and plot a line plot`

```
plt.figure(figsize=(16,6))
sns.lineplot(data=final_data)
```

Out[90]: <AxesSubplot:xlabel='title'>



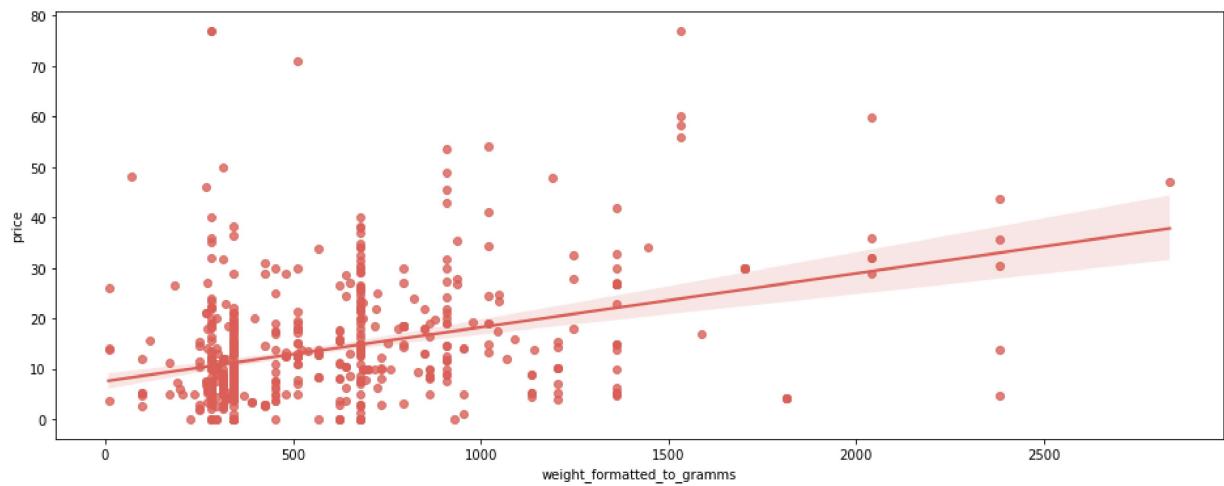
oops! there is not much insights to be gained from here, due to the large size of the data. Let me try to find the correlation between the weight and price. Remember what i said will guide my thoughts for this analysis above? Yeah, lets do this.

To find the correlation between the 'weight\_formatted\_to\_gramms' and 'price', i'd be using a scatterplot with a regression line

In [91]: *#plotting a scatter plot with a regression line for the weight and price*

```
plt.figure(figsize=(16,6))
sns.regplot(x= 'weight_formatted_to_gramms',y='price', data= final_data)
```

Out[91]: <AxesSubplot:xlabel='weight\_formatted\_to\_gramms', ylabel='price'>



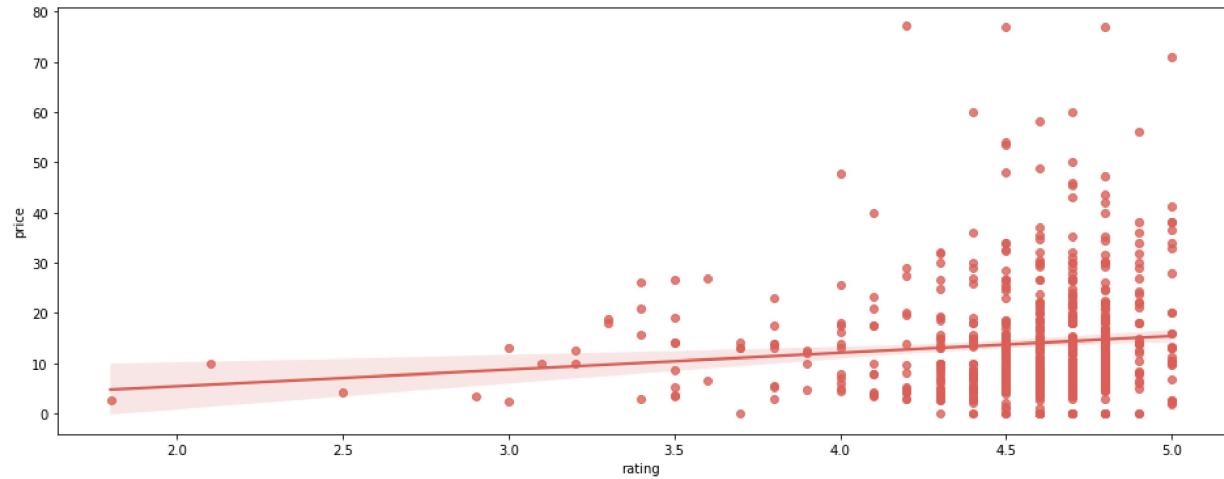
You would expect from here that the higher the weight, the higher the price (positively correlated), but this isn't so as a lot 'outliers' are seen away from the regression line which is tilting towards the fact that the variables are not really correlated. We can establish this much later in this analysis. For now let's check another variable

Now, to check if the price of the coffee affects the rating given, i'd use the scatterplot again

In [92]: *#plotting a scatter plot with regression line for the price and rating*

```
plt.figure(figsize=(16,6))
sns.regplot(x= 'rating',y='price', data= final_data)
```

Out[92]: <AxesSubplot:xlabel='rating', ylabel='price'>



Again, you would expect that the higher the price, the higher the rating, but this graph has again proven otherwise because the coffee having more ratings here have relatively lower prices with the exceptions of few ones which do not. Hence, there is nothing much to see here, lets move ahead.

Now lets extract some insights from this data and bring out some major stuff. First, we start with the rating.

Lets check the best rated

In [107]: #to extract the top best rated, On a scale of 5, a high rating should start from  
final\_data.loc[(final\_data.rating > 4)]

Out[107]:

coffee_type	rating	reviews	seller_name	price	weight_formatted_to_gramms
title					
café bustelo, espresso style dark roast ground coffee, vacuum- packed 10 oz. brick	espresso,dark roast	4.7	914	walmart.com	3.76
folgers classic roast ground coffee, medium roast coffee, 25.9 ounce canister	medium roast,classic roast	4.4	740	walmart.com	734.3

Oh! wow! we have quite a number of highly rated coffee, 650 out of 703 is a lot! Let's see who sells them. From a glance,i can see that walmart occurs a lot as a sellerin this data, i'd randomly choose to see the number of sales walmart has,just to see

In [135]: #to determine the coffee with the most number of reviews, i will check for walmart

```
final_data.loc[final_data.seller_name == 'walmart.com', :]
```

Out[135]:

	coffee_type	rating	reviews	seller_name	price	weight_formatted_to_gram
title						
great value espresso ground coffee, 10 oz	espresso	3.5	54	walmart.com	3.36	28g
chameleon organic coffee guatemala, medium roast, ground coffee, 9 oz	medium roast	4.8	54	walmart.com	14.26	120g
cafe la llave decaf espresso dark roast ground coffee, 8.8 oz	espresso,dark roast	4.4	9	walmart.com	2.88	24g
illy ground drip classico medium roast coffee, 8.8 oz	medium roast	4.8	469	walmart.com	14.52	24g
peet's coffee french roast, dark roast whole bean coffee, 10.5oz bag	french,french roast,dark roast	4.6	631	walmart.com	9.98	29g
...	...	...	...	...	...	...
nescaf clasico decaf dark roast instant coffee, 7 oz.	instant coffee,dark roast	4.7	265	walmart.com	6.32	64g

coffee_type	rating	reviews	seller_name	price	weight_formatted_to_gram
<b>title</b>					
great value classic medium roast instant coffee, 8 oz	instant coffee,medium roast	4.4	292	walmart.com	3.24
gevalia mocha latte k-cup espresso coffee pods & latte froth packets, 12 ct box	espresso,mocha,espresso coffee,latte	4.5	5315	walmart.com	13.66
1850 black gold coffee, dark roast ground coffee, 12 oz	dark roast	4.5	4165	walmart.com	8.48
folgers classic decaf ground coffee, medium roast, 25.9-ounce	medium roast	4.5	64	walmart.com	12.24

79

340

340

734

546 rows × 6 columns



Walmart alone has 546 inputs!! My instinct was correct, if only walmart can sell 546 out of 703, they are the best seller, no cap! but lets see how others did in terms of sales too....More like 'top best sellers'

In [120]: *#to check the top best sellers,i would first group by seller\_name*

```
group=final_data.groupby('seller_name').agg(len)
group
```

Out[120]:

coffee_type	rating	reviews	price	weight_formatted_to_gramms	
seller_name	101cells	1	1	1	1
<b>21wayskinnect llc</b>	1	1	1	1	1
<b>54fabulous</b>	1	1	1	1	1
<b>a s group inc</b>	1	1	1	1	1
<b>alfa retailers llc</b>	1	1	1	1	1
...	...	...	...	...	...
<b>vercingetorix llc</b>	1	1	1	1	1
<b>vitaminics inc</b>	1	1	1	1	1
<b>walmart.com</b>	546	546	546	546	546
<b>www.hadetech.com</b>	1	1	1	1	1
<b>your voice your choice inc</b>	2	2	2	2	2

96 rows × 5 columns

from this,we have quite a number of sellers having just one rating, review and even price,that means they sold just one. To further determine my top sellers, lets check those that have rating greater than 2

In [125]: `#check ratings greater than 2`

```
group2= group.loc[(group.rating > 2)]
group2
```

Out[125]:

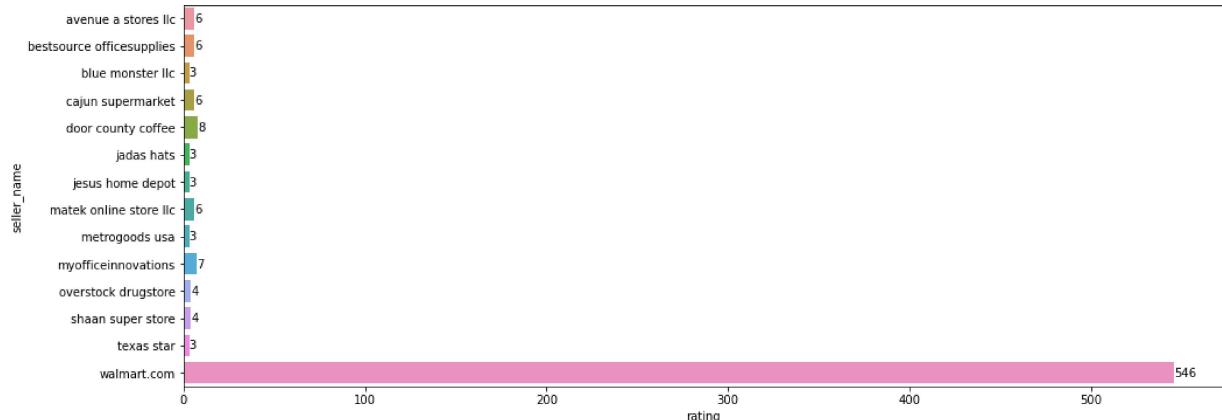
seller_name	coffee_type	rating	reviews	price	weight_formatted_to_gramms
avenue a stores llc		6	6	6	6
bestsource officesupplies		6	6	6	6
blue monster llc		3	3	3	3
cajun supermarket		6	6	6	6
door county coffee		8	8	8	8
jadas hats		3	3	3	3
jesus home depot		3	3	3	3
matek online store llc		6	6	6	6
metrogoods usa		3	3	3	3
myofficeinnovations		7	7	7	7
overstock drugstore		4	4	4	4
shaan super store		4	4	4	4
texas star		3	3	3	3
walmart.com		546	546	546	546

Okay, here they are. To aid more visualization, we would plot a barchart

In [183]: #a barplot to see the top sellers and how many they sold

```
plt.figure(figsize=(16,6))
bar= sns.barplot(x=group2['rating'], y = group2.index)
bar.bar_label(bar.containers[0])
```

Out[183]: [Text(0, 0, '6'),  
Text(0, 0, '6'),  
Text(0, 0, '3'),  
Text(0, 0, '6'),  
Text(0, 0, '8'),  
Text(0, 0, '3'),  
Text(0, 0, '3'),  
Text(0, 0, '6'),  
Text(0, 0, '3'),  
Text(0, 0, '7'),  
Text(0, 0, '4'),  
Text(0, 0, '4'),  
Text(0, 0, '3'),  
Text(0, 0, '546')]



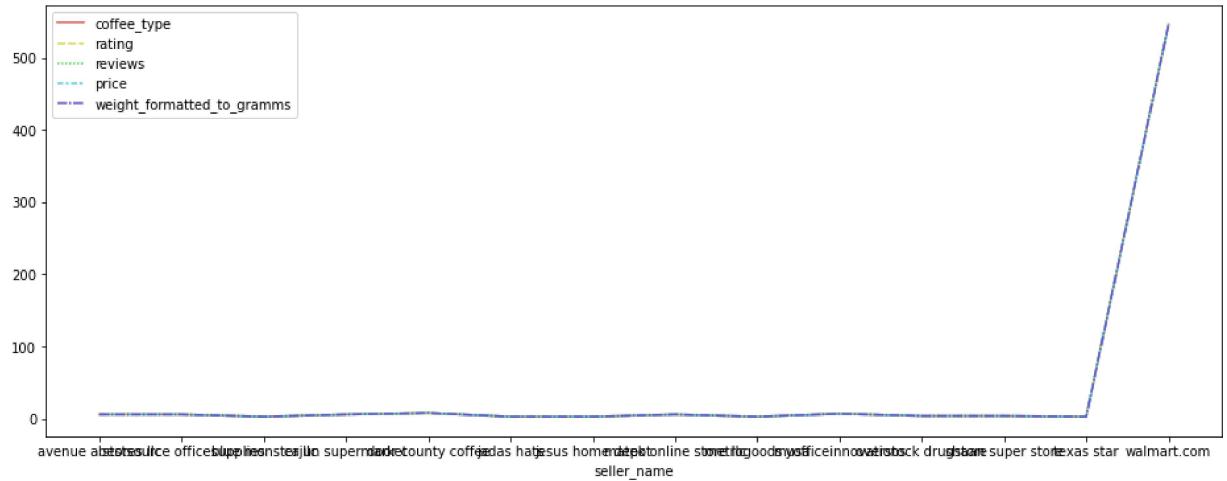
So, here we have them, the second best seller is 'door county coffee' which has just 8 sales compared to walmart with over 540 sales. It is indeed safe to say this analysis is about walmart stores (laugh)

Just to use a line plot to visualize the above insights. Walmart stores hasit

In [138]: *#plotting a Lineplotfor the best sellers*

```
plt.figure(figsize=(16,6))
sns.lineplot(data=group2)
```

Out[138]: <AxesSubplot:xlabel='seller\_name'>



Next, we will determine the best reviewed coffee. The coffee that has the highest reviews

In [144]: #to determine the best reviewed coffee, we would first group by the coffee\_type and review= final\_data.groupby('coffee\_type').reviews.agg([len, min, max])  
review

Out[144]:

		len	min	max
	coffee_type			
americano,caffe americano,medium dark,medium dark roast,dark roast		1	4	4
arabica		5	1	613
arabica,dark roast		6	2	5036
arabica,light roast		1	235	235
arabica,medium roast		44	8	6301
...		...	...	...
mocha,instant coffee		1	235	235
mocha,latte		1	505	505
mocha,light roast		2	75	432
mocha,medium roast		4	13	1484
vienna		1	373	373

119 rows × 3 columns

In [163]: #then to grouping by the length

```
group3=final_data.groupby('reviews').agg(len)
group3
```

Out[163]:

	coffee_type	rating	seller_name	price	weight_formatted_to_grams
reviews					
1	11	11		11	11
2	9	9		9	9
3	6	6		6	6
4	4	4		4	4
5	12	12		12	12
...	...	...		...	...
6301	1	1		1	1
7017	1	1		1	1
8446	1	1		1	1
14166	1	1		1	1
15148	1	1		1	1

439 rows × 5 columns

From this table '15148' is the highest review, lets check which coffee has it in the next code

In [186]: *#to check the coffee\_type with the highest review*

```
final_data.loc[final_data.reviews == 15148, :]
```

Out[186]:

	coffee_type	rating	reviews	seller_name	price	weight_formatted_to_gramms
title						
<b>folgers classic roast ground coffee, medium roast coffee, 48 oz canister</b>	medium roast,classic roast	4.9	15148	walmart.com	13.92	1360.8

This is the coffee with the highest review. coffee\_type 'medium roast, classic roast' has it

Now lets use our description of the categorical data to further establish our best seller and get the best selling coffee too

In [185]: *#to check the categorical datatype*

```
categorical12 = final_data.dtypes[final_data.dtypes=='object'].index
final_data[categorical12].describe()
```

Out[185]:

	coffee_type	seller_name
<b>count</b>	703	703
<b>unique</b>	119	96
<b>top</b>	medium roast	walmart.com
<b>freq</b>	184	546

from this table, it shows that the top coffee is 'medium roast'(occurng 184 times) and is sold by 'walmart.com' Therefore, walmart.com is our best seller and 'Medium roast' coffee is our best selling coffee

Next, we will determine the best rated coffee. The coffee that has the highest rating

In [184]: *#to determine the best rated coffee, first groupby coffee\_type and rating*

```
final_data.groupby('coffee_type').rating.agg([len, min, max])
```

Out[184]:

coffee_type	len	min	max
americano,caffè americano,medium dark,medium dark roast,dark roast	1	4.3	4.3
arabica	5	3.2	5.0
arabica,dark roast	6	4.3	5.0
arabica,light roast	1	4.7	4.7
arabica,medium roast	44	3.1	4.9
...	...	...	...
mocha,instant coffee	1	4.6	4.6
mocha,latte	1	4.3	4.3
mocha,light roast	2	4.1	4.3
mocha,medium roast	4	3.8	4.6
vienna	1	4.5	4.5

119 rows × 3 columns

In [172]: #we group again by number of occurrences of 'rating'

```
group4=final_data.groupby('rating').agg(len)
group4
```

Out[172]:

	coffee_type	reviews	seller_name	price	weight_formatted_to_gramms
rating					
1.8	1	1	1	1	1
2.1	1	1	1	1	1
2.5	1	1	1	1	1
2.9	1	1	1	1	1
3.0	2	2	2	2	2
3.1	1	1	1	1	1
3.2	2	2	2	2	2
3.3	2	2	2	2	2
3.4	4	4	4	4	4
3.5	8	8	8	8	8
3.6	2	2	2	2	2
3.7	4	4	4	4	4
3.8	8	8	8	8	8
3.9	4	4	4	4	4
4.0	12	12	12	12	12
4.1	12	12	12	12	12
4.2	16	16	16	16	16
4.3	40	40	40	40	40
4.4	48	48	48	48	48
4.5	91	91	91	91	91
4.6	139	139	139	139	139
4.7	155	155	155	155	155
4.8	92	92	92	92	92
4.9	30	30	30	30	30
5.0	27	27	27	27	27

In [167]: #the highest rating value is '5', so lets see who has it

```
final_data.loc[final_data.rating == 5, :]
```

Out[167]:

title	coffee_type	rating	reviews	seller_name	price	weight_formatted_to_g
arabica ground 100% coffee 126 pk	arabica	5.0	1	www.hadetech.com	71.00	
cafe el morro espresso dark roast caffeinated ground coffee, 8.8 oz	espresso,dark roast	5.0	2	walmart.com	1.78	
kauai coffee na pali coast k-cup coffee pods, dark roast, 24 ct	dark roast	5.0	4	walmart.com	12.94	
black rifle coffee spirit of '76, medium roast, ground coffee, 12 oz	medium roast,black rifle coffee	5.0	1	walmart.com	12.98	
black rifle coffee just black single-serve pods, medium roast, 44 ct	medium roast,black rifle coffee	5.0	1	walmart.com	27.94	
cafe aroma espresso ground coffee, dark roast caffeinated, 8.8 oz	espresso,dark roast	5.0	1	walmart.com	1.98	
black rifle coffee just black single-serve pods, medium roast, 22 ct	medium roast,black rifle coffee	5.0	2	walmart.com	15.97	
black rifle ready-to-drink coffee, espresso mocha, 11oz, can	espresso,mocha	5.0	6	walmart.com	2.68	

	coffee_type	rating	reviews	seller_name	price	weight_formatted_to_g
title						
<b>death wish coffee, organic, fair-trade, espresso roast ground, 14oz, bag</b>	espresso,espresso roast	5.0	4	walmart.com	19.97	
<b>black rifle coffee spirit of '76 single-serve pods, medium roast, 22 ct</b>	medium roast,black rifle coffee	5.0	1	walmart.com	15.97	
<b>verena street nine mile sunset ground coffee, dark roast, 32 ounces</b>	dark roast	5.0	3	walmart.com	19.99	
<b>the coffee bean &amp; tea leaf house blend light roast single serve coffee for keurig brewers, 1 box of 16 (16 total pods)</b>	light roast	5.0	1	walmart.com	13.19	
<b>gold coffee company 100% arabica morning blend ground coffee, dark roast, 10 oz</b>	arabica,dark roast	5.0	2	walmart.com	12.99	
<b>eldorado espresso brick 9 oz</b>	espresso	5.0	7	walmart.com	6.78	
<b>cameron's coffee jamaican me crazy ground coffee, light roast, 12 oz</b>	light roast	5.0	29	walmart.com	9.63	
<b>verena street cow tipper flavored ground coffee, medium roast, 12 ounces</b>	medium roast	5.0	2	walmart.com	9.99	

title	coffee_type	rating	reviews	seller_name	price	weight_formatted_to_g
<b>black rifle ready-to-drink coffee, espresso with cream, 11oz, can</b>	espresso	5.0	8	walmart.com	2.68	
<b>folgers 100% colombian coffee, medium roast ground coffee, 22.6 ounce canister</b>	colombian,medium roast	5.0	3	walmart.com	10.32	
<b>folgers half caff ground coffee, medium roast, 22.6-ounce</b>	medium roast	5.0	2	walmart.com	10.46	
<b>caf bustelo ground coffee, dark roast, 6-ounce brick</b>	dark roast	5.0	20	sams online traders	11.10	
<b>ruta maya organic coffee medium roast 2.2 pounds</b>	medium roast	5.0	1	reliable & fast	32.89	
<b>ruta maya organic coffee dark roast 2.2 lbs.</b>	dark roast	5.0	2	reliable & fast	36.44	
<b>community coffee caf special decaf medium-dark roast coffee single-serve cups 36 ct box compatible with keurig 2.0 k-cup brewers</b>	medium-dark roast,dark roast	5.0	108	pink sky store	41.18	
<b>papanicholas italian espresso 24ct</b>	italian	5.0	1	limitless capital inc	37.99	

title	coffee_type	rating	reviews	seller_name	price	weight_formatted_to_g
<b>lion coffee, vanilla macadamia flavor light roast - ground coffee, 24 ounce bag</b>	light roast	5.0	1	hawaii coffee company	33.95	
<b>caf la carreta cuban espresso coffee 10 oz brick</b>	espresso,espresso coffee	5.0	2	government discount inc	9.99	
<b>stumptown coffee roasters organic blend whole bean coffee, dark roast, 12 oz</b>	dark roast	5.0	1	beauty services pro llc	38.18	

These are the highest rated coffee. Oh wow! They are quite a number, you would expect to see the most expensive, weightiest and best selling coffee amongst them but no, this again confirms that there is no correlation between the rating, price, weight and the coffee\_type

Next is to check the most expensive coffee.

In [168]: *#lets group the dataframe by the price column*

```
group5=final_data.groupby('price').agg(len)
group5
```

Out[168]:

	coffee_type	rating	reviews	seller_name	weight_formatted_to_gramms
price					
<b>0.00</b>	29	29	29	29	29
<b>1.00</b>	2	2	2	2	2
<b>1.43</b>	1	1	1	1	1
<b>1.78</b>	1	1	1	1	1
<b>1.98</b>	2	2	2	2	2
...	...	...	...	...	...
<b>60.00</b>	1	1	1	1	1
<b>71.00</b>	1	1	1	1	1
<b>76.99</b>	1	1	1	1	1
<b>77.00</b>	1	1	1	1	1
<b>77.09</b>	1	1	1	1	1

345 rows × 5 columns

The highest price is '77.09', lets see who has it

In [174]: *#checking the details of the highest coffee price*

```
final_data.loc[final_data.price ==77.09, :]
```

Out[174]:

	coffee_type	rating	reviews	seller_name	price	weight_formatted_to_gramms
title						
<b>lavazza perfetto single-serve k-cup® pods for keurig brewer, dark roast, 10-ct boxes (pack of 6)</b>	dark roast	4.2	22	national discount centers	77.09	283.5

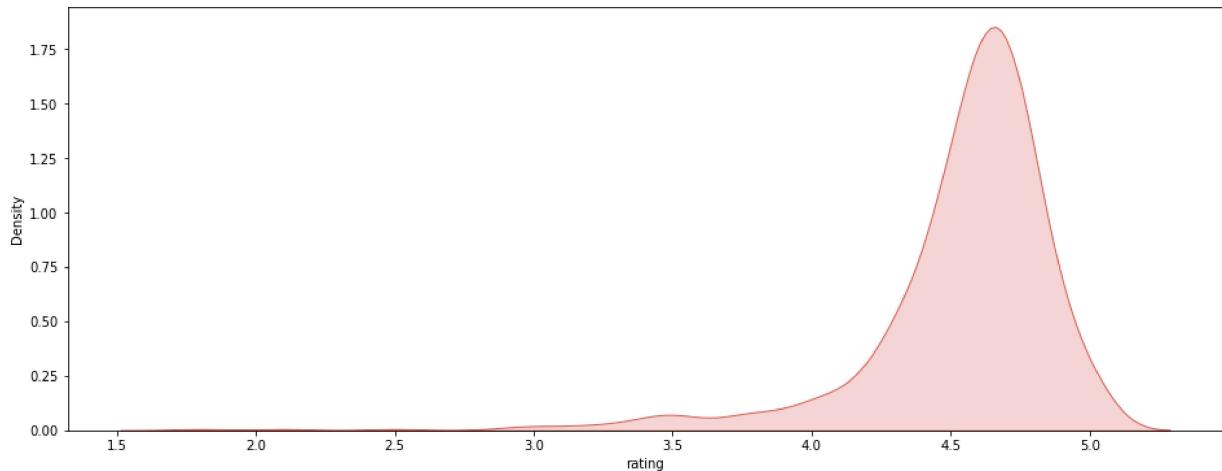
This is the most expensive coffee and it can be seen even from here that the price doesn't determine the rating and the weight of the coffee because, if it did, the most expensive coffee should be the best rated(rating of 5) and also have the most weight which is not the case. This insight is consistent with the plot of price against weight above.

Lastly, i would create some density plots to show the distributions of the ratings, weight and price independently

In [175]: *#lets show the distribution of the ratings*

```
plt.figure(figsize=(16,6))
sns.kdeplot(data=final_data['rating'], shade=True)
```

Out[175]: <AxesSubplot:xlabel='rating', ylabel='Density'>



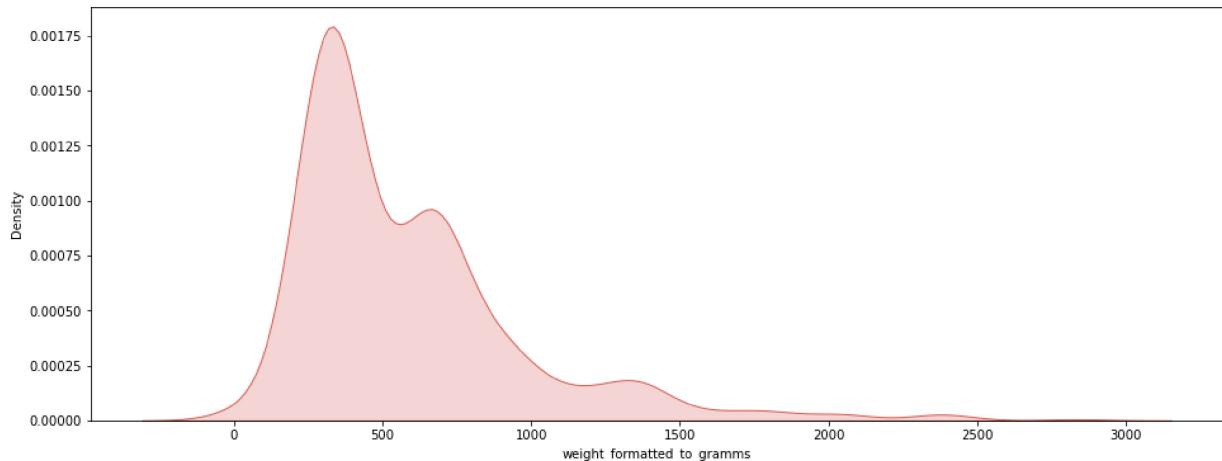
This chart depicts that the most common coffee rating is between 4.5 to 5.0. Most of the coffee were rated to be nice

In [ ]:

In [187]: *#for the weight*

```
plt.figure(figsize=(16,6))
sns.kdeplot(data=final_data['weight_formatted_to_gramms'], shade=True)
```

Out[187]: <AxesSubplot:xlabel='weight\_formatted\_to\_gramms', ylabel='Density'>

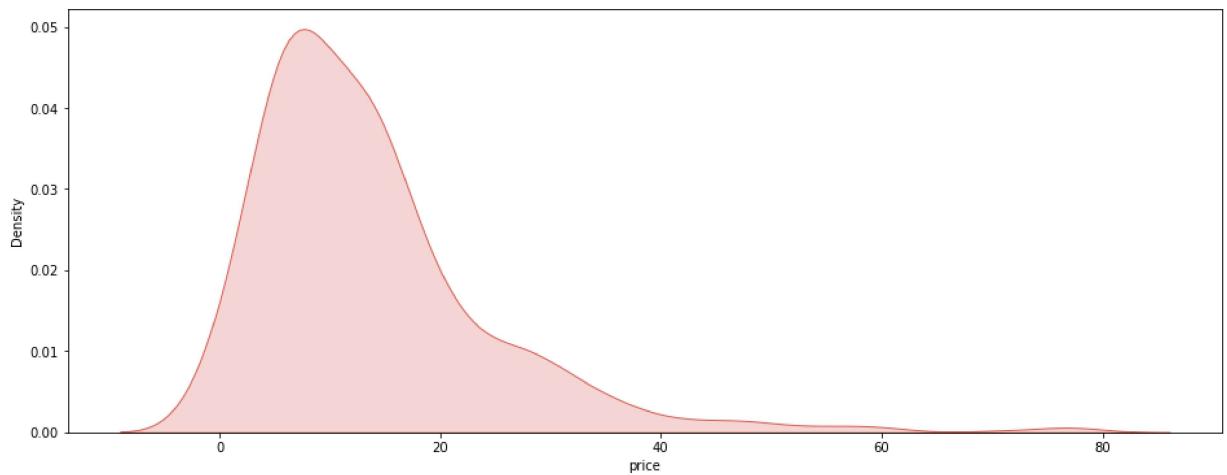


This chart shows that the most popular coffee weight is in between 250-500grams

In [188]: #for the price

```
plt.figure(figsize=(16,6))
sns.kdeplot(data=final_data['price'], shade=True)
```

Out[188]: <AxesSubplot:xlabel='price', ylabel='Density'>



This chart depicts that most of the coffee prices were within the range of 10-20

With this i have come to the end of this analysis, i hope you werent bored and you could understand my data representation and visualization. Majorly, i was able to clean the data and see the relationships between the variables, i checked for the best seller, best rated coffee, coffee with the highest review, the highest selling coffee. the most expensive coffee, the top best selling stores, the weightiest coffee and i was even able to check the most popular weights of coffee and the prices.

In [ ]: