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

In [2]:
```
df = pd.read_csv(r"C:\Users\kallzz\Desktop\Data Analytics Stuff\Data Analys
df
```

Out[2]:

	Flavor	Base Flavor	Liked	Flavor Rating	Texture Rating	Total Rating
0	Mint Chocolate Chip	Vanilla	Yes	10.0	8.0	18.0
1	Chocolate	Chocolate	Yes	8.8	7.6	16.6
2	Vanilla	Vanilla	No	4.7	5.0	9.7
3	Cookie Dough	Vanilla	Yes	6.9	6.5	13.4
4	Rocky Road	Chocolate	Yes	8.2	7.0	15.2
5	Pistachio	Vanilla	No	2.3	3.4	5.7
6	Cake Batter	Vanilla	Yes	6.5	6.0	12.5
7	Neapolitan	Vanilla	No	3.8	5.0	8.8
8	Chocolte Fudge Brownie	Chocolate	Yes	8.2	7.1	15.3

In [3]:
```
# Group By can be done on a column that has duplicated or repeated values
df.groupby("Base Flavor")
```

Out[3]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001C7A8052740
>
```

In [4]:
```
# to display the group by data we need to use any applicable method on grou
df.groupby("Base Flavor").mean()
```

```
C:\Users\kallzz\AppData\Local\Temp\ipykernel_7944\2885785111.py:2: FutureW
arning: The default value of numeric_only in DataFrameGroupBy.mean is depr
ecated. In a future version, numeric_only will default to False. Either sp
ecify numeric_only or select only columns which should be valid for the fu
nction.
  df.groupby("Base Flavor").mean()
```

Out[4]:

	Flavor Rating	Texture Rating	Total Rating
Base Flavor			
Chocolate	8.4	7.233333	15.70
Vanilla	5.7	5.650000	11.35

In [5]:
```
# Aggregations are performed only on the numeric values
df.groupby("Base Flavor").sum()
```

```
C:\Users\kallzz\AppData\Local\Temp\ipykernel_7944\2665118766.py:2: FutureW
arning: The default value of numeric_only in DataFrameGroupBy.sum is depre
cated. In a future version, numeric_only will default to False. Either spe
cify numeric_only or select only columns which should be valid for the fun
ction.
  df.groupby("Base Flavor").sum()
```

Out[5]:

Base Flavor	Flavor Rating	Texture Rating	Total Rating
Chocolate	25.2	21.7	47.1
Vanilla	34.2	33.9	68.1

In [6]:
```
# min and max can be applied on both numrical and non-numerical. String dat
df.groupby("Base Flavor").min()
```

Out[6]:

Base Flavor	Flavor	Liked	Flavor Rating	Texture Rating	Total Rating
Chocolate	Chocolate	Yes	8.2	7.0	15.2
Vanilla	Cake Batter	No	2.3	3.4	5.7

In [7]:
```
df.groupby("Base Flavor").max()
```

Out[7]:

Base Flavor	Flavor	Liked	Flavor Rating	Texture Rating	Total Rating
Chocolate	Rocky Road	Yes	8.8	7.6	16.6
Vanilla	Vanilla	Yes	10.0	8.0	18.0

In [8]:
```
df.groupby("Base Flavor").count()
```

Out[8]:

Base Flavor	Flavor	Liked	Flavor Rating	Texture Rating	Total Rating
Chocolate	3	3	3	3	3
Vanilla	6	6	6	6	6

In [9]:
```
# agg function retunrs specific aggregation values at specific columns
df.groupby("Base Flavor").agg({'Flavor Rating': ['mean', 'max', 'min', 'cou
```

Out[9]:

	Flavor Rating			
	mean	max	min	count
Base Flavor				
Chocolate	8.4	8.8	8.2	3
Vanilla	5.7	10.0	2.3	6

In [11]:
```
df.groupby("Base Flavor").agg({'Flavor Rating': ['mean', 'max', 'min', 'cou
```

Out[11]:

	Flavor Rating				Texture Rating			
	mean	max	min	count	mean	max	min	count
Base Flavor								
Chocolate	8.4	8.8	8.2	3	7.233333	7.6	7.0	3
Vanilla	5.7	10.0	2.3	6	5.650000	8.0	3.4	6

In [12]:
```
# Group by on multiple columns
df.groupby(["Base Flavor", "Liked"]).mean()
```

```
C:\Users\kallzz\AppData\Local\Temp\ipykernel_7944\2378787121.py:2: FutureW
arning: The default value of numeric_only in DataFrameGroupBy.mean is depr
ecated. In a future version, numeric_only will default to False. Either sp
ecify numeric_only or select only columns which should be valid for the fu
nction.
  df.groupby(["Base Flavor", "Liked"]).mean()
```

Out[12]:

		Flavor Rating	Texture Rating	Total Rating
Base Flavor	**Liked**			
Chocolate	**Yes**	8.4	7.233333	15.700000
Vanilla	**No**	3.6	4.466667	8.066667
	Yes	7.8	6.833333	14.633333

In [13]:
```
# describe() can be used to get the stats on numerical data
df.groupby(["Base Flavor", "Liked"]).describe()
```

Out[13]:

		Flavor Rating								Texture Rating			
		count	mean	std	min	25%	50%	75%	max	count	mean	...	75%
Base Flavor	Liked												
Chocolate	Yes	3.0	8.4	0.346410	8.2	8.20	8.2	8.50	8.8	3.0	7.233333	...	7.3
Vanilla	No	3.0	3.6	1.212436	2.3	3.05	3.8	4.25	4.7	3.0	4.466667	...	5.0
	Yes	3.0	7.8	1.915724	6.5	6.70	6.9	8.45	10.0	3.0	6.833333	...	7.2

rows × 24 columns

In [ ]:
```

```