

```
In [11]: import pandas as pd
import numpy as np

from pandas import DataFrame, Series
salesdata = pd.read_csv('/Users/chidera/Desktop/All Project Files/storesalesdata.csv')
salesdata.head()
```

Out[11]:

	Store	Day	Month	Year	Dry_Grocery	Dairy	Frozen_Food	Meat	Produce	Floral
0	S0001	1	3	2012	23401.06	5567.90	3930.28	5223.61	4542.28	238.58
1	S0001	2	3	2012	32442.93	8009.52	4712.31	6174.72	9503.45	245.29
2	S0001	3	3	2012	40433.61	9759.20	5969.03	8010.67	12793.92	282.12
3	S0001	4	3	2012	57043.39	13966.26	9008.24	13142.55	16635.77	316.72
4	S0001	5	3	2012	32894.86	7870.66	5448.35	5459.29	8358.96	198.00

```
In [19]: salesdata.tail()
salesdata['Year'].head()
```

Out[19]:

```
0    2012
1    2012
2    2012
3    2012
4    2012
Name: Year, dtype: int64
```

```
In [90]: df = pd.read_csv('/Users/chidera/Desktop/All Project Files/storesalesdata.csv')
df = pd.DataFrame(salesdata)
is_2015 = df['Year']==2015
print(is_2015.head())
```

```
0    False
1    False
2    False
3    False
4    False
Name: Year, dtype: bool
```

```
In [91]: df_2015 = df[is_2015]
print(df_2015.shape)
```

```
(30933, 13)
```

```
In [94]: print(df_2015.head())
df = pd.DataFrame(df_2015)
df.head()
```

	Store	Day	Month	Year	Dry_Grocery	Dairy	Frozen_Food	
Meat \								
669	S0001	1	1	2015	15851.32	4343.52	2659.79	3
273.70								
670	S0001	2	1	2015	24726.90	5622.35	3618.33	4
361.22								
671	S0001	3	1	2015	32463.81	7407.34	6134.92	8
888.22								
672	S0001	4	1	2015	45526.66	10239.19	8592.09	13
102.94								
673	S0001	5	1	2015	44492.21	9997.15	7215.28	12
757.61								
	Produce	Floral		Deli	Bakery	General_Merchandise		
669	2820.66	111.31		1778.42	1311.56		2870.88	
670	4050.15	52.44		1944.47	1473.97		3805.39	
671	6968.51	119.65		2774.02	2148.35		4148.57	
672	10027.17	91.87		3923.03	3138.57		5915.66	
673	8232.63	169.18		4218.18	2956.71		7570.17	

Out[94]:

	Store	Day	Month	Year	Dry_Grocery	Dairy	Frozen_Food	Meat	Produce	Flo
<b>669</b>	S0001	1	1	2015	15851.32	4343.52	2659.79	3273.70	2820.66	111.
<b>670</b>	S0001	2	1	2015	24726.90	5622.35	3618.33	4361.22	4050.15	52.
<b>671</b>	S0001	3	1	2015	32463.81	7407.34	6134.92	8888.22	6968.51	119.
<b>672</b>	S0001	4	1	2015	45526.66	10239.19	8592.09	13102.94	10027.17	91.
<b>673</b>	S0001	5	1	2015	44492.21	9997.15	7215.28	12757.61	8232.63	169.

```
In [96]: df = df.groupby(['Store']).sum()
df.head()
```

Out[96]:

	Day	Month	Year	Dry_Grocery	Dairy	Frozen_Food	Meat	Produce
Store								
S0001	5713	2370	733460	10845787.65	2423389.38	1814872.88	2531382.34	2284388.70
S0002	5713	2370	733460	7931072.94	1844188.83	1366677.52	1991807.66	1755293.15
S0003	5713	2370	733460	12741875.44	3095321.58	2087437.15	3468989.12	3792143.03
S0004	5713	2370	733460	12688533.89	2710746.38	2240847.44	3565375.84	2802811.88
S0005	5713	2370	733460	12248085.57	2957003.74	2402445.89	2834435.52	3648309.02

In [ ]:

```
In [102]: df = df.drop('Day', axis = 1)
df = df.drop('Year', axis = 1)
df.head()
```

Out[102]:

	Dry_Grocery	Dairy	Frozen_Food	Meat	Produce	Floral	Deli
Store							
S0001	10845787.65	2423389.38	1814872.88	2531382.34	2284388.70	159142.23	1023812.90
S0002	7931072.94	1844188.83	1366677.52	1991807.66	1755293.15	128935.92	689786.42
S0003	12741875.44	3095321.58	2087437.15	3468989.12	3792143.03	291259.69	1262874.27
S0004	12688533.89	2710746.38	2240847.44	3565375.84	2802811.88	170822.64	1166388.77
S0005	12248085.57	2957003.74	2402445.89	2834435.52	3648309.02	247017.36	984736.25

```
In [103]: table_2 = df[['Dry_Grocery', 'Dairy', 'Frozen_Food', 'Meat', 'Produce',
, 'Floral', 'Deli', 'Bakery', 'General_Merchandise']].apply(lambda x:
x/x.sum(), axis=1)
```

```
In [104]: print(table_2)
```

	Dry_Grocery	Dairy	Frozen_Food	Meat	Produce	Flo
Store						
S0001	0.461347	0.103084	0.077199	0.107677	0.097171	0.006
S0002	0.457528	0.106388	0.078841	0.114903	0.101259	0.007

S0003 631	0.421349	0.102356	0.069027	0.114713	0.125399	0.009
S0004 120	0.454560	0.097111	0.080277	0.127728	0.100409	0.006
S0005 877	0.440169	0.106268	0.086339	0.101863	0.131112	0.008
S0006 265	0.459835	0.098609	0.074443	0.106916	0.101782	0.008
S0007 838	0.442087	0.102135	0.089170	0.115135	0.103447	0.007
S0008 859	0.428457	0.108732	0.080321	0.106878	0.123696	0.007
S0009 288	0.442252	0.100376	0.077051	0.118908	0.103507	0.007
S0010 383	0.431445	0.114962	0.080342	0.107148	0.121629	0.010
S0011 896	0.453793	0.093984	0.071882	0.121443	0.098467	0.007
S0012 825	0.465999	0.097265	0.074357	0.114391	0.094315	0.006
S0013 912	0.445549	0.090722	0.075352	0.114048	0.092661	0.007
S0014 660	0.444373	0.097768	0.077185	0.098697	0.106849	0.010
S0015 685	0.447968	0.098766	0.077869	0.099990	0.104975	0.007
S0016 252	0.477240	0.091435	0.078064	0.110087	0.088469	0.006
S0017 390	0.438117	0.100133	0.078777	0.097699	0.133663	0.013
S0018 318	0.452848	0.098827	0.079017	0.106062	0.109022	0.009
S0019 755	0.482713	0.097466	0.077881	0.111624	0.094800	0.005
S0020 395	0.458778	0.107142	0.077872	0.108469	0.109104	0.008
S0021 951	0.432833	0.122190	0.086453	0.086198	0.148003	0.008
S0022 661	0.470503	0.096628	0.075948	0.106099	0.092153	0.008
S0023 484	0.436647	0.108652	0.084322	0.109598	0.107051	0.009
S0024 392	0.455110	0.104005	0.082141	0.104836	0.110498	0.005
S0025 304	0.462042	0.104673	0.079629	0.104944	0.111706	0.005
S0026 334	0.457669	0.105388	0.078424	0.099883	0.113665	0.008
S0027 254	0.446685	0.104030	0.077643	0.099390	0.132245	0.016

S0028 532	0.452981	0.103237	0.078961	0.119144	0.116529	0.003
S0029 220	0.441264	0.099425	0.072961	0.131545	0.099193	0.006
S0030 305	0.440521	0.109386	0.070796	0.111384	0.116304	0.008
...	...	...	...	...	...	...
S0056 107	0.476390	0.116872	0.086399	0.087432	0.117922	0.008
S0057 867	0.438124	0.097218	0.075329	0.147446	0.107270	0.005
S0058 323	0.464678	0.100375	0.079861	0.102636	0.100387	0.008
S0059 386	0.429947	0.096841	0.072498	0.117434	0.120425	0.007
S0060 507	0.449700	0.101905	0.082347	0.114625	0.100789	0.010
S0061 827	0.460530	0.106176	0.073653	0.117167	0.103018	0.005
S0062 800	0.469907	0.098878	0.079772	0.093725	0.110607	0.005
S0063 299	0.418792	0.113887	0.084998	0.105673	0.135424	0.009
S0064 574	0.436475	0.107325	0.071335	0.114659	0.122796	0.009
S0065 574	0.419622	0.106998	0.073301	0.112589	0.122428	0.011
S0066 631	0.451210	0.099513	0.082972	0.105708	0.085589	0.008
S0067 989	0.460265	0.099318	0.080956	0.122564	0.101988	0.005
S0068 386	0.440859	0.100076	0.075807	0.102962	0.122995	0.009
S0069 913	0.413319	0.106080	0.077610	0.112314	0.133252	0.007
S0070 825	0.439685	0.097528	0.076867	0.100742	0.113486	0.007
S0071 953	0.444424	0.098192	0.068710	0.134439	0.101314	0.005
S0072 559	0.439710	0.092788	0.063373	0.125564	0.108970	0.007
S0073 669	0.436023	0.097776	0.077857	0.105423	0.124461	0.008
S0074 086	0.456378	0.103004	0.078967	0.131933	0.093064	0.005
S0075 559	0.431127	0.097762	0.075503	0.159218	0.114500	0.004
S0076 701	0.438723	0.099093	0.081946	0.109776	0.115233	0.011

S0077 850	0.438946	0.103040	0.081825	0.106184	0.117936	0.010
S0078 193	0.431482	0.119661	0.088052	0.111857	0.127329	0.007
S0079 748	0.464064	0.096002	0.076860	0.120889	0.098970	0.007
S0080 712	0.476708	0.106208	0.082215	0.101295	0.082616	0.007
S0081 475	0.423280	0.102089	0.086592	0.123493	0.116666	0.008
S0082 269	0.460995	0.100760	0.075290	0.113610	0.101490	0.005
S0083 097	0.447873	0.108865	0.083793	0.109469	0.117443	0.007
S0084 377	0.460306	0.106343	0.094408	0.105936	0.100824	0.006
S0085 635	0.424240	0.112563	0.075257	0.115232	0.125251	0.010

Store	Deli	Bakery	General_Merchandise
S0001	0.043550	0.035544	0.067658
S0002	0.039792	0.029702	0.064149
S0003	0.041761	0.036108	0.079655
S0004	0.041785	0.034469	0.057542
S0005	0.035389	0.022607	0.067376
S0006	0.046309	0.035417	0.068424
S0007	0.046298	0.030176	0.063715
S0008	0.032623	0.022398	0.089036
S0009	0.044204	0.025756	0.080656
S0010	0.037099	0.038051	0.058941
S0011	0.048717	0.032441	0.071378
S0012	0.045066	0.018592	0.083191
S0013	0.046277	0.034281	0.093198
S0014	0.034927	0.020549	0.108992
S0015	0.034977	0.025106	0.102665
S0016	0.046052	0.028720	0.073681
S0017	0.040857	0.028239	0.069125
S0018	0.043024	0.034966	0.066918
S0019	0.032458	0.019593	0.077710
S0020	0.038153	0.029436	0.062651
S0021	0.036579	0.028166	0.050628
S0022	0.043142	0.034053	0.072812
S0023	0.038336	0.036189	0.069722
S0024	0.041131	0.022955	0.073932
S0025	0.042283	0.030706	0.058714
S0026	0.037658	0.030564	0.068415
S0027	0.029723	0.023370	0.070660
S0028	0.037298	0.030146	0.058172
S0029	0.042865	0.017822	0.088706

S0030	0.048072	0.030570	0.064662
...	...	...	...
S0056	0.027159	0.022477	0.057241
S0057	0.049536	0.017707	0.061503
S0058	0.047752	0.029823	0.066165
S0059	0.043467	0.028616	0.083387
S0060	0.041623	0.025869	0.072633
S0061	0.049362	0.029587	0.054679
S0062	0.040767	0.021481	0.079064
S0063	0.029934	0.029131	0.072861
S0064	0.039662	0.043817	0.054357
S0065	0.038639	0.030976	0.083873
S0066	0.039948	0.025230	0.101198
S0067	0.043480	0.024515	0.060924
S0068	0.045918	0.036083	0.065914
S0069	0.041581	0.037148	0.070784
S0070	0.037787	0.023328	0.102751
S0071	0.042982	0.021501	0.082485
S0072	0.044056	0.018201	0.099779
S0073	0.042505	0.036045	0.071240
S0074	0.037369	0.033158	0.061042
S0075	0.051568	0.016031	0.049732
S0076	0.041655	0.026107	0.075765
S0077	0.038089	0.039244	0.063885
S0078	0.036274	0.025209	0.052945
S0079	0.040397	0.022743	0.072327
S0080	0.036259	0.024245	0.082742
S0081	0.038956	0.042944	0.057506
S0082	0.048267	0.035212	0.059107
S0083	0.031885	0.025206	0.068370
S0084	0.040150	0.022119	0.063537
S0085	0.031300	0.036952	0.068570

[85 rows x 9 columns]

```
In [105]: df = pd.DataFrame(table_2)
df.head()
```

```
Out[105]:
```

	Dry_Grocery	Dairy	Frozen_Food	Meat	Produce	Floral	Deli	Bakery
Store								
S0001	0.461347	0.103084	0.077199	0.107677	0.097171	0.006769	0.043550	0.035544
S0002	0.457528	0.106388	0.078841	0.114903	0.101259	0.007438	0.039792	0.029702
S0003	0.421349	0.102356	0.069027	0.114713	0.125399	0.009631	0.041761	0.036108
S0004	0.454560	0.097111	0.080277	0.127728	0.100409	0.006120	0.041785	0.034469
S0005	0.440169	0.106268	0.086339	0.101863	0.131112	0.008877	0.035389	0.022607

```
In [107]: from sklearn.cluster import KMeans
model = KMeans (n_clusters=3)
```

```
In [108]: model.fit(df)
```

```
Out[108]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300
,
n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto
',
random_state=None, tol=0.0001, verbose=0)
```

```
In [110]: labels = model.predict(df)
```

```
In [111]: print(labels)
```

```
[0 0 1 0 1 0 1 1 2 1 0 0 2 2 2 0 1 0 0 0 1 0 1 0 0 0 1 1 2 1 2 2 1 1
0 0 2
1 0 1 0 0 1 1 2 2 2 2 1 1 0 1 0 1 0 0 1 0 1 0 0 0 1 1 1 2 0 1 1 2 2
2 1 0
1 1 1 1 0 0 1 0 1 0 1]
```

```
In [115]: import matplotlib.pyplot as plt
```

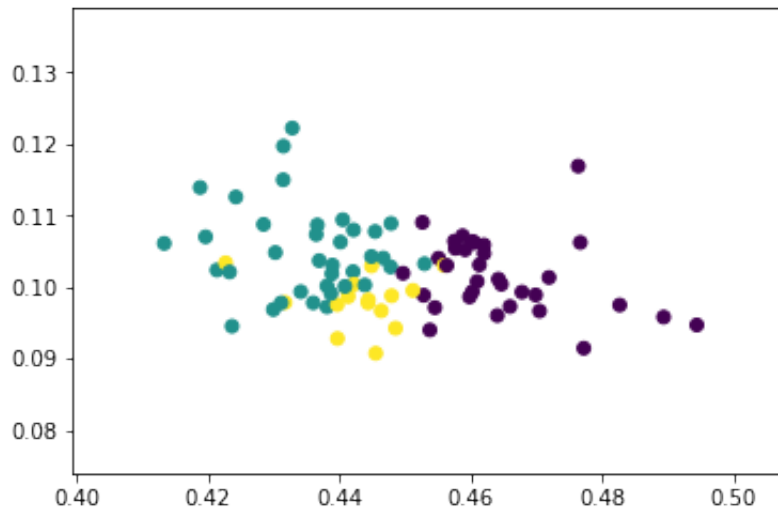
```
In [117]: xs = df.iloc[:, 0]
```

```
In [118]: ys = df.iloc[:, 1]
```



```
In [119]: plt.scatter(xs, ys, c=labels)
```

```
Out[119]: <matplotlib.collections.PathCollection at 0x1a21f0d550>
```



```
In [120]: df_cluster = pd.DataFrame(labels)
```

```
In [122]: df_cluster
```

```
Out[122]:
```

	0
0	0
1	0
2	1
3	0
4	1
5	0
6	1
7	1
8	2
9	1
10	0
11	0
12	2
13	2

14	2
15	0
16	1
17	0
18	0
19	0
20	1
21	0
22	1
23	0
24	0
25	0
26	1
27	1
28	2
29	1
...	...
55	0
56	1
57	0
58	1
59	0
60	0
61	0
62	1
63	1
64	1
65	2
66	0
67	1
68	1
69	2

```
70 2
71 2
72 1
73 0
74 1
75 1
76 1
77 1
78 0
79 0
80 1
81 0
82 1
83 0
84 1
```

85 rows × 1 columns

In [ ]:

In [ ]:

In [76]:

In [ ]:

In [78]:

In [ ]:

In [ ]:

In [81]:

In [ ]:

In [83]:

In [ ]:

In [86]:

In [ ]:

In [ ]: