

```
In [9]: crls=pd.read_csv('C:\\Users\\kama1\\OneDrive\\Desktop\\cognoriseinfotech\\cereal.csv')
```

```
In [6]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [10]: crls.head(10)
```

```
Out[10]:
```

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
0	100% Bran	N	C	70	4	1	130	10.0	5.0	6	280	25	3	1.00	0.33	68.402973
1	100% Natural Bran	Q	C	120	3	5	15	2.0	8.0	8	135	0	3	1.00	1.00	33.983679
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5	320	25	3	1.00	0.33	59.425505
3	All-Bran with Extra Fiber	K	C	50	4	0	140	14.0	8.0	0	330	25	3	1.00	0.50	93.704912
4	Almond Delight	R	C	110	2	2	200	1.0	14.0	8	-1	25	3	1.00	0.75	34.384843
5	Apple Cinnamon Cheerios	G	C	110	2	2	180	1.5	10.5	10	70	25	1	1.00	0.75	29.509541
6	Apple Jacks	K	C	110	2	0	125	1.0	11.0	14	30	25	2	1.00	1.00	33.174094
7	Basic 4	G	C	130	3	2	210	2.0	18.0	8	100	25	3	1.33	0.75	37.038562
8	Bran Chex	R	C	90	2	1	200	4.0	15.0	6	125	25	1	1.00	0.67	49.120253
9	Bran Flakes	P	C	90	3	0	210	5.0	13.0	5	190	25	3	1.00	0.67	53.313813

```
In [11]: crls.tail(10)
```

```
Out[11]:
```

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
67	Special K	K	C	110	6	0	230	1.0	16.0	3	55	25	1	1.0	1.00	53.131324
68	Strawberry Fruit Wheats	N	C	90	2	0	15	3.0	15.0	5	90	25	2	1.0	1.00	59.363993
69	Total Corn Flakes	G	C	110	2	1	200	0.0	21.0	3	35	100	3	1.0	1.00	38.839746
70	Total Raisin Bran	G	C	140	3	1	190	4.0	15.0	14	230	100	3	1.5	1.00	28.592785
71	Total Whole Grain	G	C	100	3	1	200	3.0	16.0	3	110	100	3	1.0	1.00	46.658844
72	Triples	G	C	110	2	1	250	0.0	21.0	3	60	25	3	1.0	0.75	39.106174
73	Trix	G	C	110	1	1	140	0.0	13.0	12	25	25	2	1.0	1.00	27.753301
74	Wheat Chex	R	C	100	3	1	230	3.0	17.0	3	115	25	1	1.0	0.67	49.787445
75	Wheaties	G	C	100	3	1	200	3.0	17.0	3	110	25	1	1.0	1.00	51.592193
76	Wheaties Honey Gold	G	C	110	2	1	200	1.0	16.0	8	60	25	1	1.0	0.75	36.187559

```
In [12]: crls.isna().sum()
```

```
Out[12]: name      0
mfr      0
type      0
calories  0
protein  0
fat      0
sodium   0
fiber    0
carbo    0
sugars   0
potass   0
vitamins 0
shelf    0
weight   0
cups     0
rating   0
dtype: int64
```

```
In [13]: crls['fiber'].value_counts()
```

```
Out[13]: fiber
0.0      19
1.0      16
3.0      15
2.0      10
4.0       4
5.0       4
1.5       3
10.0      1
9.0       1
14.0      1
6.0       1
2.7       1
2.5       1
Name: count, dtype: int64
```

```
In [14]: crls1=crls[['name', 'carbo', 'fiber']].groupby('name').sum().sort_values('carbo', ascending=False)
crls1
```

```
Out[14]:
```

	carbo	fiber
name		
Rice Chex	23.0	0.0
Rice Krispies	22.0	0.0
Corn Chex	22.0	0.0
Kix	21.0	0.0
Crispix	21.0	1.0
...
100% Natural Bran	8.0	2.0
All-Bran with Extra Fiber	8.0	14.0
All-Bran	7.0	9.0
100% Bran	5.0	10.0
Quaker Oatmeal	-1.0	2.7

77 rows × 2 columns

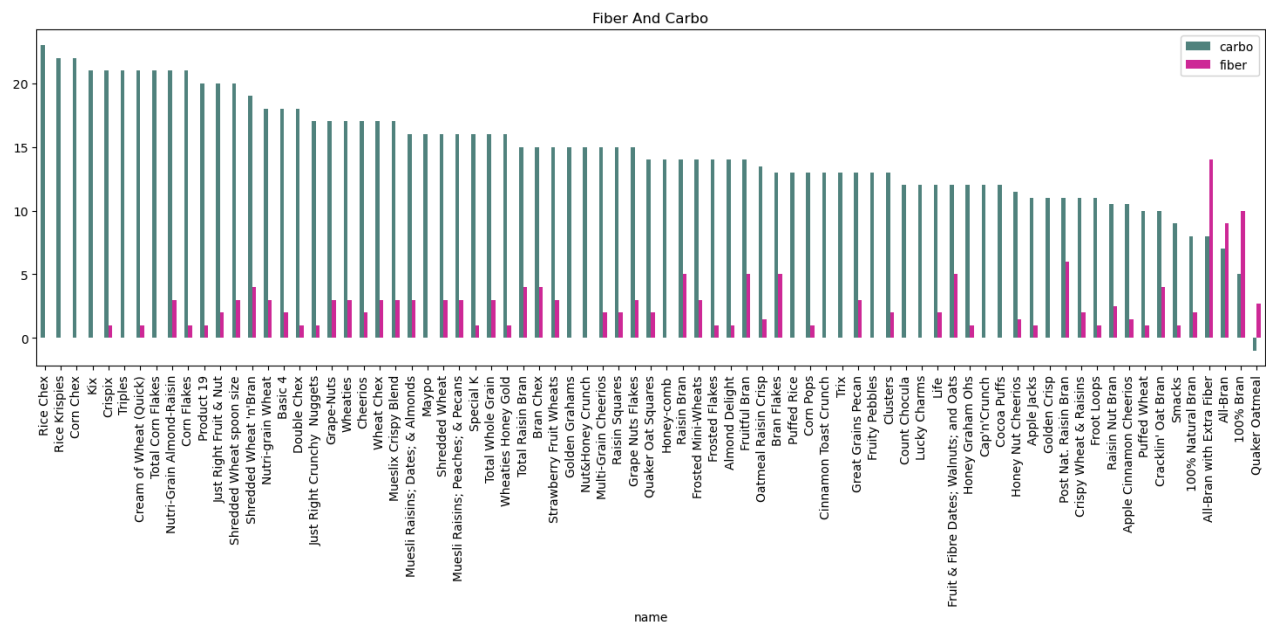
```
In [103]: crls1.head(15)
```

```
Out[103]:
```

	carbo	fiber
name		
Rice Chex	23.0	0.0
Rice Krispies	22.0	0.0
Corn Chex	22.0	0.0
Kix	21.0	0.0
Crispix	21.0	1.0
Triples	21.0	0.0
Cream of Wheat (Quick)	21.0	1.0
Total Corn Flakes	21.0	0.0
Nutri-Grain Almond-Raisin	21.0	3.0
Corn Flakes	21.0	1.0
Product 19	20.0	1.0
Just Right Fruit & Nut	20.0	2.0
Shredded Wheat spoon size	20.0	3.0
Shredded Wheat 'n'Bran	19.0	4.0
Nutri-grain Wheat	18.0	3.0

```
In [106]: fig=plt.figure()
ax0=fig.add_subplot(1,2,1)
crls1.plot(kind='bar',color=('#548580','#cf2b98'),figsize=(40,5),ax=ax0)
ax0.set_title('Fiber And Carbo')
```

Out[106]: Text(0.5, 1.0, 'Fiber And Carbo')



```
In [15]: crlsr=crls[['name','rating']].groupby('name').sum().sort_values('rating', ascending=False)
crlsr.head(30)
```

Out[15]:

	rating
name	
All-Bran with Extra Fiber	93.704912
Shredded Wheat 'n'Bran	74.472949
Shredded Wheat spoon size	72.801787
100% Bran	68.402973
Shredded Wheat	68.235885
Cream of Wheat (Quick)	64.533816
Puffed Wheat	63.005645
Puffed Rice	60.756112
Nutri-grain Wheat	59.642837
All-Bran	59.425505
Strawberry Fruit Wheats	59.363993
Frosted Mini-Wheats	58.345141
Raisin Squares	55.333142
Maypo	54.850917
Grape-Nuts	53.371007
Bran Flakes	53.313813
Special K	53.131324
Grape Nuts Flakes	52.076897
Wheaties	51.592193
Quaker Oatmeal	50.828392
Cheerios	50.764999
Wheat Chex	49.787445
Quaker Oat Squares	49.511874
Bran Chex	49.120253
Crispix	46.895644
Total Whole Grain	46.658844
Corn Flakes	45.863324
Great Grains Pecan	45.811716
Life	45.328074
Double Chex	44.330856

```
In [79]: crls2=crls[['name', 'sugars', 'vitamins']].groupby('name').sum().sort_values('sugars', ascending=False)
crls2
```

Out[79]:

	sugars	vitamins
name		
Smacks	15	25
Golden Crisp	15	25
Total Raisin Bran	14	100
Post Nat. Raisin Bran	14	25
Apple Jacks	14	25
...
Puffed Wheat	0	0
Puffed Rice	0	0
Cream of Wheat (Quick)	0	0
Shredded Wheat	0	0
Quaker Oatmeal	-1	0

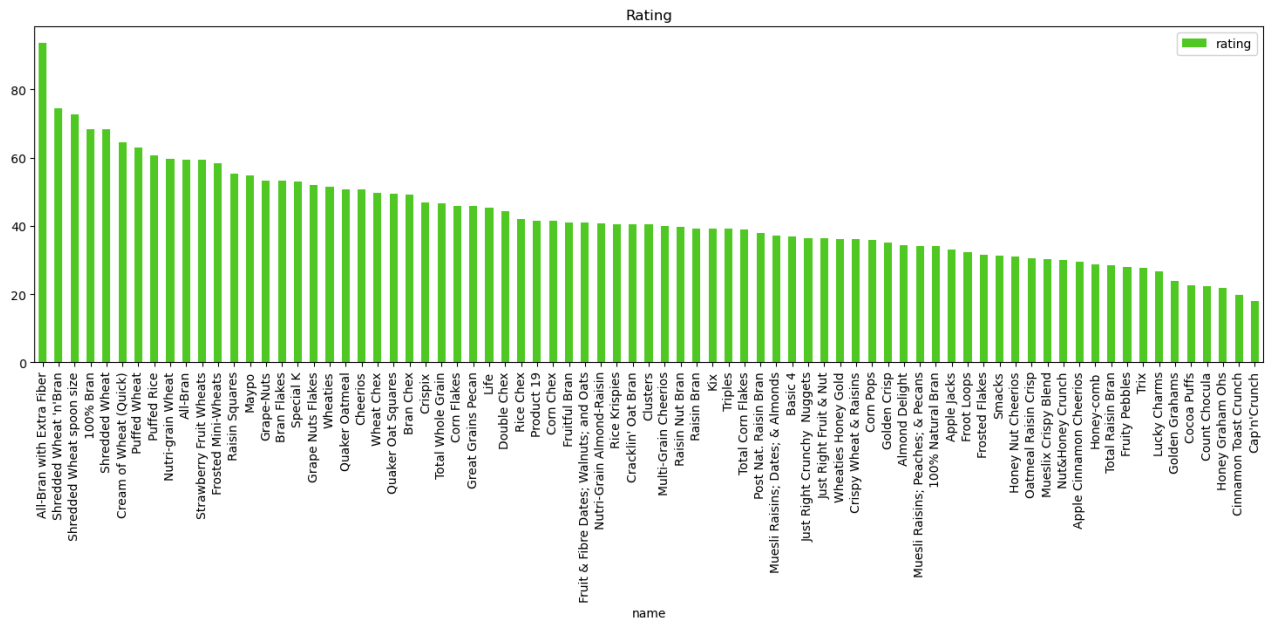
77 rows × 2 columns

```
In [84]: crls2.isna().sum()
```

Out[84]: sugars 0
vitamins 0
dtype: int64

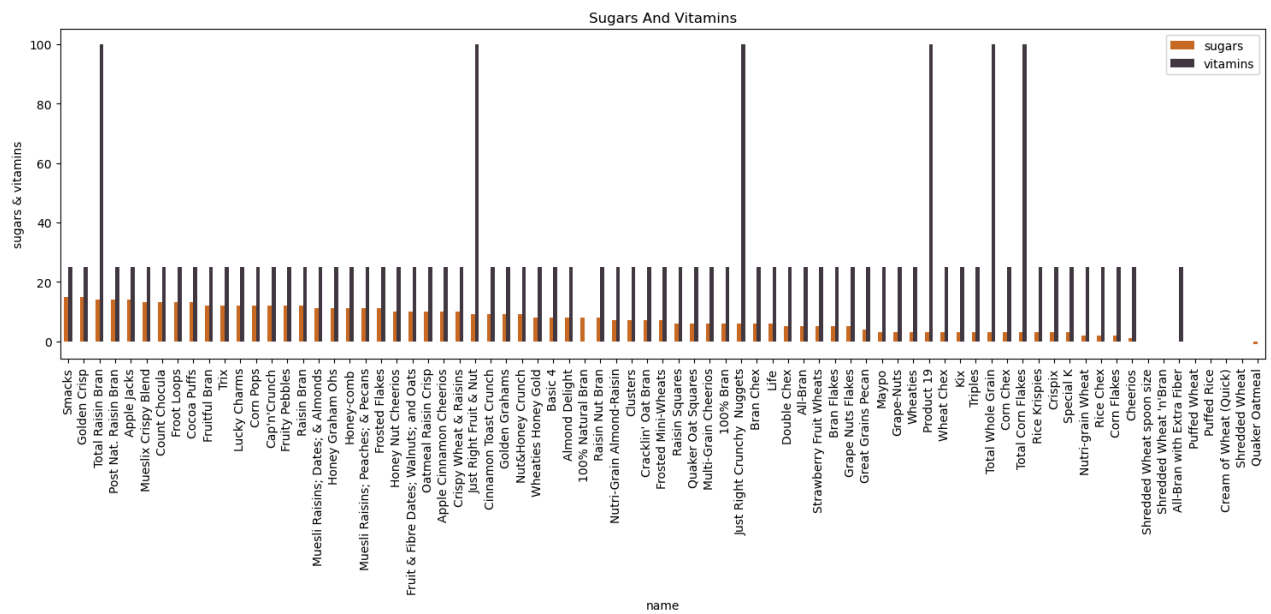
```
In [112]: fig=plt.figure()
ax0=fig.add_subplot(1,2,1)
crlsr.plot(kind='bar',color=('#53cc23'),figsize=(40,5),ax=ax0)
ax0.set_title('Rating')
```

Out[112]: Text(0.5, 1.0, 'Rating')



```
In [114]: fig=plt.figure()
ax0=fig.add_subplot(1,2,1)
crls2.plot(kind='bar',color=('#cc6c23', '#453841'),figsize=(40,5),ax=ax0)
ax0.set_title('Sugars And Vitamins')
ax0.set_ylabel('sugars & vitamins')
```

Out[114]: Text(0, 0.5, 'sugars & vitamins')

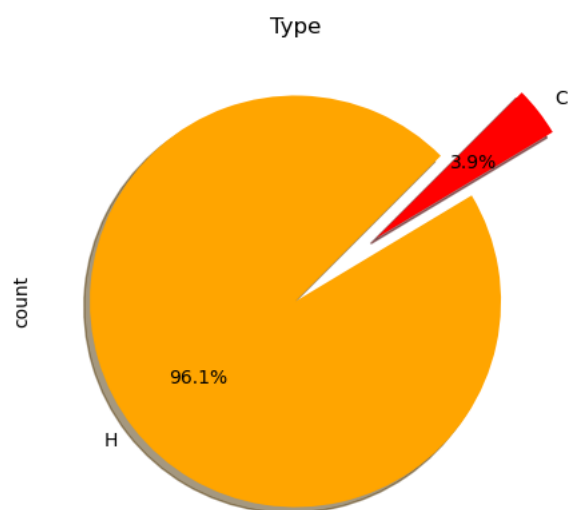


```
In [120]: crl1=cr1s['type'].value_counts()
```

Out[120]: type
C 74
H 3
Name: count, dtype: int64

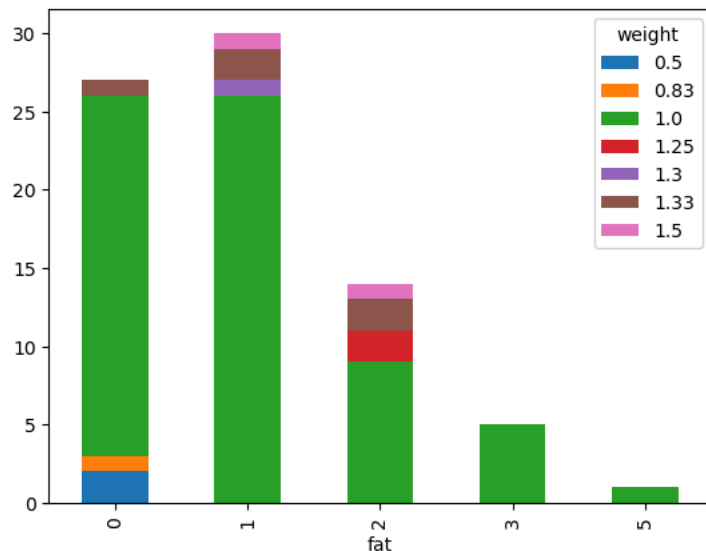
```
In [179]: t(1,2,1)
ie', colors=('orange', 'red'),figsize=(30,5),shadow=True,explode=[0.0,0.5],labels=('H', 'C'),startangle=45,autopct='%1.1f%',ax=
e')
```

Out[179]: Text(0.5, 1.0, 'Type')



```
In [16]: plt.figure(figsize=(16,5))
pd.crosstab(crls['fat'],crls['weight']).plot(kind='bar',stacked=True)
plt.show()
```

<Figure size 1600x500 with 0 Axes>



```
In [17]: crlsf=crls['name'].value_counts().head().index
crlsf
```

```
Out[17]: Index(['100% Bran', 'Nutri-Grain Almond-Raisin', 'Quaker Oat Squares',
               'Puffed Wheat', 'Puffed Rice'],
              dtype='object', name='name')
```

```
In [66]: from wordcloud import WordCloud
```

```
In [65]: !pip install wordcloud
```

Collecting wordcloud

Obtaining dependency information for wordcloud from https://files.pythonhosted.org/packages/f5/b0/247159f61c5d5d6647171bef84430b7efad4db504f0229674024f3a4f7f2/wordcloud-1.9.3-cp311-cp311-win_amd64.whl.metadata (https://files.pythonhosted.org/packages/f5/b0/247159f61c5d5d6647171bef84430b7efad4db504f0229674024f3a4f7f2/wordcloud-1.9.3-cp311-cp311-win_amd64.whl.metadata)

Downloading wordcloud-1.9.3-cp311-cp311-win_amd64.whl.metadata (3.5 kB)

Requirement already satisfied: numpy>=1.6.1 in c:\users\kamal\anaconda3\lib\site-packages (from wordcloud) (1.24.3)

Requirement already satisfied: pillow in c:\users\kamal\anaconda3\lib\site-packages (from wordcloud) (9.4.0)

Requirement already satisfied: matplotlib in c:\users\kamal\anaconda3\lib\site-packages (from wordcloud) (3.7.2)

Requirement already satisfied: contourpy>=1.0.1 in c:\users\kamal\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.0.5)

Requirement already satisfied: cycler>=0.10 in c:\users\kamal\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.11.0)

Requirement already satisfied: fonttools>=4.22.0 in c:\users\kamal\anaconda3\lib\site-packages (from matplotlib->wordcloud) (4.25.0)

Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\kamal\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.4.4)

Requirement already satisfied: packaging>=20.0 in c:\users\kamal\anaconda3\lib\site-packages (from matplotlib->wordcloud) (23.1)

Requirement already satisfied: pyparsing<3.1,>=2.3.1 in c:\users\kamal\anaconda3\lib\site-packages (from matplotlib->wordcloud) (3.0.9)

Requirement already satisfied: python-dateutil>=2.7 in c:\users\kamal\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.8.2)

Requirement already satisfied: six>=1.5 in c:\users\kamal\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)

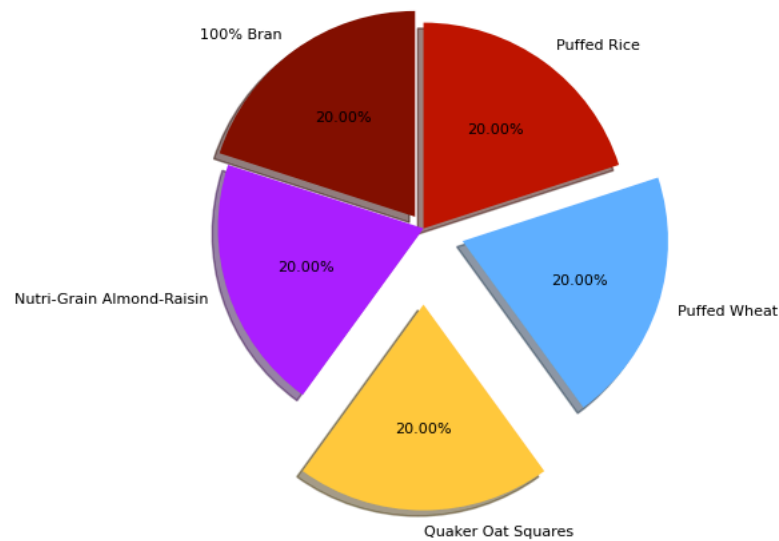
Downloading wordcloud-1.9.3-cp311-cp311-win_amd64.whl (300 kB)

```
----- 0.0/300.2 kB ? eta -:-:--
- ----- 10.2/300.2 kB ? eta -:-:--
- ----- 30.7/300.2 kB 435.7 kB/s eta 0:00:01
- ----- 92.2/300.2 kB 744.7 kB/s eta 0:00:01
- ----- 225.3/300.2 kB 1.4 MB/s eta 0:00:01
- ----- 300.2/300.2 kB 1.4 MB/s eta 0:00:00
```

Installing collected packages: wordcloud

Successfully installed wordcloud-1.9.3

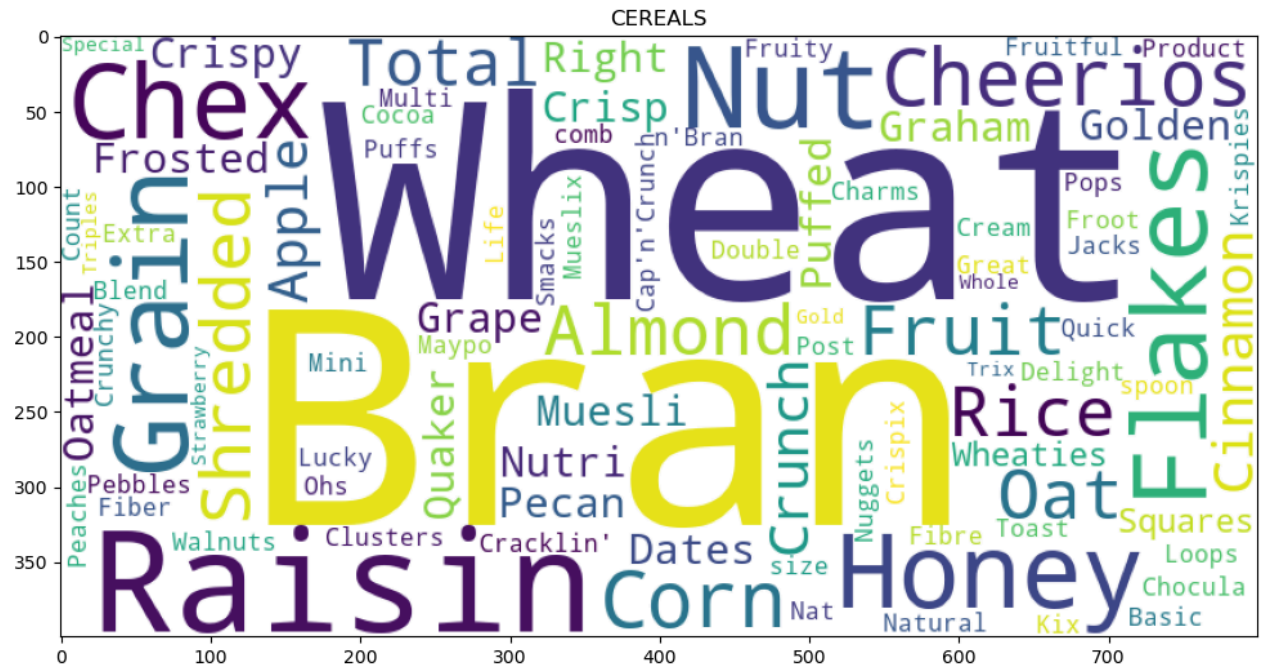
```
In [52]: plt.figure(figsize=(5,5))
d=(crls["name"].value_counts(normalize=True)*100).head()
keys=crls["name"].value_counts().head().index
colorz = ['#850F00', '#AD1FFF', '#FFC93F', '#5FB1FF', '#BF1600']
exploda=(0.07, 0.0, 0.37, 0.2, 0.0)
plt.pie(d, labels=keys, autopct='%0.2f%', explode=explode, startangle=90, colors=colorz, textprops={'fontsize':8}, shadow=True)
plt.show()
```



```
In [68]: all_reviews = ' '.join(crls['name'].dropna())

wordcloud = WordCloud(width=800, height=400, background_color='white').generate(all_reviews)

plt.figure(figsize=(12, 8))
plt.imshow(wordcloud)
plt.title('CEREALS')
plt.show()
```



In []: