Import Libraries and Content Dataset:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
con = pd.read_csv("/content/Content.csv")
con.head()
```

	Unnamed: 0	Content ID	User ID	Type	Category	URL
0	0	97522e57-d9ab-4bd6-97bf-c24d952602d2	8d3cd87d-8a31-4935-9a4f-b319bfe05f31	photo	Studying	https://socialbuzz.cdn.com/content/storage/975...
1	1	9f737e0a-3cdd-4d29-9d24-753f4e3be810	beb1f34e-7870-46d6-9fc7-2e12eb83ce43	photo	healthy eating	https://socialbuzz.cdn.com/content/storage/9f7...
2	2	230c4e4d-70c3-461d-b42c-ec09396efb3f	a5c65404-5894-4b87-82f2-d787cbee86b4	photo	healthy eating	https://socialbuzz.cdn.com/content/storage/230...

```
rea = pd.read_csv("/content/Reactions.csv")
rea.head()
```

	Unnamed: 0	Content ID	User ID	Type	Datetime
0	0	97522e57-d9ab-4bd6-97bf-c24d952602d2	NaN	NaN	2021-04-22 15:17:15
1	1	97522e57-d9ab-4bd6-97bf-c24d952602d2	5d454588-283d-459d-915d-c48a2cb4c27f	disgust	2020-11-07 09:43:50
2	2	97522e57-d9ab-4bd6-97bf-c24d952602d2	92b87fa5-f271-43e0-af66-84fac21052e6	dislike	2021-06-17 12:22:51
3	3	97522e57-d9ab-4bd6-97bf-c24d952602d2	163daa38-8b77-48c9-9af6-37a6c1447ac2	scared	2021-04-18 05:13:58
4	4	97522e57-d9ab-4bd6-97bf-c24d952602d2	34e8add9-0206-47fd-a501-037b994650a2	disgust	2021-01-06 19:13:01

```
rea_t = pd.read_csv("/content/ReactionTypes.csv")
rea_t.head()
```

	Unnamed: 0	Type	Sentiment	Score
0	0	heart	positive	60
1	1	want	positive	70
2	2	disgust	negative	0
3	3	hate	negative	5
4	4	interested	positive	30

Content Dataset:

```
con.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  1000 non-null   int64
 1   Content ID  1000 non-null   object
 2   User ID     1000 non-null   object
 3   Type        1000 non-null   object
 4   Category    1000 non-null   object
 5   URL         801 non-null    object
dtypes: int64(1), object(5)
memory usage: 47.0+ KB
```

```
con.describe()
```

	Unnamed: 0
count	1000.000000
mean	499.500000
std	288.819436
min	0.000000
25%	249.750000
50%	499.500000
75%	749.250000
max	999.000000

```
con.head()
```

	Unnamed: 0	Content ID	User ID	Type	Category	URL
0	0	97522e57-d9ab-4bd6-97bf-c24d952602d2	8d3cd87d-8a31-4935-9a4f-b319bfe05f31	photo	Studying	https://socialbuzz.cdn.com/content/storage/975...
1	1	9f737e0a-3cdd-4d29-9d24-753f4e3be810	beb1f34e-7870-46d6-9fc7-2e12eb83ce43	photo	healthy eating	https://socialbuzz.cdn.com/content/storage/9f7...
2	2	230c4e4d-70c3-461d-b42c-ec09396efb3f	a5c65404-5894-4b87-82f2-d787cbee86b4	photo	healthy eating	https://socialbuzz.cdn.com/content/storage/230...

```
con.columns
```

```
Index(['Unnamed: 0', 'Content ID', 'User ID', 'Type', 'Category', 'URL'], dtype='object')
```

```
con.dtypes
```

```
Unnamed: 0     int64
Content ID    object
User ID       object
Type          object
Category      object
URL           object
dtype: object
```

```
con.shape
```

```
(1000, 6)
```

```
df1 = con.drop('URL',axis = 1)
```

```
df1 = df1.drop('Unnamed: 0', axis = 1)
```

```
df1 = df1.drop('User ID', axis = 1)
```

```
df1.isnull().sum()
```

```
Content ID    0
Type          0
Category      0
dtype: int64
```

```
df1.Category.value_counts()
```

```
technology           71
animals              67
travel               67
culture              63
science              63
fitness              61
food                 61
healthy eating       61
cooking              60
soccer               58
tennis               58
education            57
dogs                 56
studying             55
veganism             48
public speaking      48
Fitness               5
Animals               4
Science               4
"soccer"              3
"culture"             3
Soccer                3
"dogs"                2
Education             2
Studying              2
Travel                2
Food                  2
"veganism"            1
"public speaking"     1
Public Speaking       1
"technology"          1
"cooking"             1
Healthy Eating        1
"studying"            1
"food"                1
Culture               1
"tennis"              1
Technology            1
"animals"             1
Veganism              1
"science"             1
Name: Category, dtype: int64
```

```
#Replace all error values to correct values
df1['Category'] = df1['Category'].replace(['"animals"','animals'],['animals','Animals'])
df1['Category'] = df1['Category'].replace(['"technology"','technology'],['technology','Technology'])
df1['Category'] = df1['Category'].replace(['"culture"','culture'],['culture','Culture'])
df1['Category'] = df1['Category'].replace(['"science"','science'],['science','Science'])
df1['Category'] = df1['Category'].replace(['"food"','food'],['food','Food'])
df1['Category'] = df1['Category'].replace(['"soccer"','soccer'],['soccer','Soccer'])
df1['Category'] = df1['Category'].replace(['"tennis"','tennis'],['tennis','Tennis'])
df1['Category'] = df1['Category'].replace(['"studying"','studying'],['studying','Studying'])
df1['Category'] = df1['Category'].replace(['"veganism"','veganism'],['veganism','Veganism'])
df1['Category'] = df1['Category'].replace(['"public speaking"','public speaking'],['public speaking','Public Speaking'])
df1['Category'] = df1['Category'].replace(['"dogs"','dogs'],['dogs','Dogs'])
df1['Category'] = df1['Category'].replace(['"cooking"','cooking'],['cooking','Cooking'])
df1['Category'] = df1['Category'].replace('fitness','Fitness')
df1['Category'] = df1['Category'].replace('healthy eating','Healthy Eating')
df1['Category'] = df1['Category'].replace('travel','Travel')
df1['Category'] = df1['Category'].replace('education','Education')
```

```
df1.Category.value_counts()
```

```
Technology         73
Animals            72
Travel             69
Science            68
Culture            67
Fitness            66
Food               64
Soccer             64
Healthy Eating     62
Cooking            61
Tennis             59
Education          59
Studying           58
Dogs               58
Public Speaking    50
Veganism           50
Name: Category, dtype: int64
```

```
df1.rename(columns={'Type': 'Content_Type'}, inplace=True)
```

```
df1.head()
```

	Content ID	Content_Type	Category
0	97522e57-d9ab-4bd6-97bf-c24d952602d2	photo	Studying
1	9f737e0a-3cdd-4d29-9d24-753f4e3be810	photo	Healthy Eating
2	230c4e4d-70c3-461d-b42c-ec09396efb3f	photo	Healthy Eating
3	356fff80-da4d-4785-9f43-bc1261031dc6	photo	Technology
4	01ab84dd-6364-4236-abbb-3f237db77180	video	Food

```
df1.shape
```

```
(1000, 3)
```

Reaction Types Dataset:

```
rea_t.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16 entries, 0 to 15
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  16 non-null     int64 
 1   Type        16 non-null     object
 2   Sentiment   16 non-null     object
 3   Score       16 non-null     int64 
dtypes: int64(2), object(2)
memory usage: 640.0+ bytes
```

```
rea_t.describe()
```

	Unnamed: 0	Score
count	16.000000	16.000000
mean	7.500000	39.625000
std	4.760952	26.901983
min	0.000000	0.000000
25%	3.750000	14.250000
50%	7.500000	40.000000
75%	11.250000	66.250000
max	15.000000	75.000000

```
rea_t.shape
```

```
(16, 4)
```

```
rea_t.columns
```

```
Index(['Unnamed: 0', 'Type', 'Sentiment', 'Score'], dtype='object')
```

```
rea_t.isnull().sum()
```

```
Unnamed: 0    0
Type          0
Sentiment     0
Score         0
dtype: int64
```

```
rea_t.duplicated().sum()
```

```
0
```

```
df2 = rea_t.drop('Unnamed: 0',axis = 1)
```

```
df2.shape
```

```
(16, 3)
```

```
df2.rename(columns={'Type': 'Reaction_Type'}, inplace=True)
```

```
df2
```

	Reaction_Type	Sentiment	Score
0	heart	positive	60
1	want	positive	70
2	disgust	negative	0
3	hate	negative	5
4	interested	positive	30
5	indifferent	neutral	20
6	love	positive	65
7	super love	positive	75
8	cherish	positive	70
9	adore	positive	72
10	like	positive	50
11	dislike	negative	10
12	intrigued	positive	45
13	peeking	neutral	35
14	scared	negative	15
15	worried	negative	12

Reactions Dataset:

```
rea.head()
```

	Unnamed: 0	Content ID	User ID	Type	Datetime
0	0	97522e57-d9ab-4bd6-97bf-c24d952602d2	NaN	NaN	2021-04-22 15:17:15
1	1	97522e57-d9ab-4bd6-97bf-c24d952602d2	5d454588-283d-459d-915d-c48a2cb4c27f	disgust	2020-11-07 09:43:50
2	2	97522e57-d9ab-4bd6-97bf-c24d952602d2	92b87fa5-f271-43e0-af66-84fac21052e6	dislike	2021-06-17 12:22:51
3	3	97522e57-d9ab-4bd6-97bf-c24d952602d2	163daa38-8b77-48c9-9af6-37a6c1447ac2	scared	2021-04-18 05:13:58
4	4	97522e57-d9ab-4bd6-97bf-c24d952602d2	34e8add9-0206-47fd-a501-037b994650a2	disgust	2021-01-06 19:13:01

```
rea.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25553 entries, 0 to 25552
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  25553 non-null  int64 
 1   Content ID  25553 non-null  object
 2   User ID     22534 non-null  object
 3   Type        24573 non-null  object
 4   Datetime    25553 non-null  object
dtypes: int64(1), object(4)
memory usage: 998.3+ KB
```

```
rea.shape
```

```
(25553, 5)
```

```
rea.columns
```

```
Index(['Unnamed: 0', 'Content ID', 'User ID', 'Type', 'Datetime'], dtype='object')
```

```
rea.dtypes
```

```
Unnamed: 0     int64
Content ID    object
User ID       object
Type          object
Datetime      object
dtype: object
```

```
df3 = rea.drop('Unnamed: 0',axis = 1)
```

```
df3 = df3.drop('User ID',axis = 1)
```

```
df3.shape
```

```
(25553, 3)
```

```
df3.head()
```

	Content ID	Type	Datetime
0	97522e57-d9ab-4bd6-97bf-c24d952602d2	NaN	2021-04-22 15:17:15
1	97522e57-d9ab-4bd6-97bf-c24d952602d2	disgust	2020-11-07 09:43:50
2	97522e57-d9ab-4bd6-97bf-c24d952602d2	dislike	2021-06-17 12:22:51
3	97522e57-d9ab-4bd6-97bf-c24d952602d2	scared	2021-04-18 05:13:58
4	97522e57-d9ab-4bd6-97bf-c24d952602d2	disgust	2021-01-06 19:13:01

```
df3.isnull().sum()
```

```
Content ID      0
Type          980
Datetime        0
dtype: int64
```

```
df3.dropna(inplace = True)
df3.isnull().sum()
```

```
Content ID    0
Type          0
Datetime      0
dtype: int64
```

```
df3.shape
```

```
(24573, 3)
```

```
df3.rename(columns={'Type': 'Reaction_Type'}, inplace=True)
```

```
df3.head()
```

	Content ID	Reaction_Type	Datetime
1	97522e57-d9ab-4bd6-97bf-c24d952602d2	disgust	2020-11-07 09:43:50
2	97522e57-d9ab-4bd6-97bf-c24d952602d2	dislike	2021-06-17 12:22:51
3	97522e57-d9ab-4bd6-97bf-c24d952602d2	scared	2021-04-18 05:13:58
4	97522e57-d9ab-4bd6-97bf-c24d952602d2	disgust	2021-01-06 19:13:01
5	97522e57-d9ab-4bd6-97bf-c24d952602d2	interested	2020-08-23 12:25:58

Merging Datasets for insight:

```
merge_df = pd.merge(df3, df1, on='Content ID', how='inner')
```

```
merge_df.shape
```

```
(24573, 5)
```

```
merged_df = pd.merge(merge_df, df2, on='Reaction_Type', how='inner')
```

```
merged_df.shape
```

```
(24573, 7)
```

```
merged_df
```

	Content ID	Reaction_Type	Datetime	Content_Type	Category	Sentiment	Score
0	97522e57-d9ab-4bd6-97bf-c24d952602d2	disgust	2020-11-07 09:43:50	photo	Studying	negative	0
1	97522e57-d9ab-4bd6-97bf-c24d952602d2	disgust	2021-01-06 19:13:01	photo	Studying	negative	0
2	97522e57-d9ab-4bd6-97bf-c24d952602d2	disgust	2021-04-09 02:46:20	photo	Studying	negative	0
3	9f737e0a-3cdd-4d29-9d24-753f4e3be810	disgust	2021-03-28 21:15:26	photo	Healthy Eating	negative	0
4	230c4e4d-70c3-461d-b42c-ec09396efb3f	disgust	2020-08-04 05:40:33	photo	Healthy Eating	negative	0
...	...	...	...	...	...	...	...
24568	435007a5-6261-4d8b-b0a4-55fdc189754b	adore	2020-10-04 22:26:33	audio	Veganism	positive	72
24569	435007a5-6261-4d8b-b0a4-55fdc189754b	adore	2020-09-18 10:50:50	audio	Veganism	positive	72

```
merged_df.to_csv('merged_df.csv', index=False)
```

```
merged_df.Score.value_counts()
```

```
70    3040
60    1622
15    1572
35    1559
5     1552
30    1549
10    1548
72    1548
65    1534
0     1526
50    1520
75    1519
20    1512
12    1497
45    1475
Name: Score, dtype: int64
```

```
merged_df.Category.value_counts()
```

```
Animals            1897
Science            1796
Healthy Eating     1717
Food               1699
Technology         1698
Culture            1676
Cooking            1664
Travel             1647
Soccer             1457
Education          1433
Fitness            1395
Studying           1363
Dogs               1338
Tennis             1328
Veganism           1248
Public Speaking    1217
Name: Category, dtype: int64
```

```
merged_df.Reaction_Type.value_counts()
```

```
heart          1622
scared         1572
peeking        1559
hate           1552
interested     1549
dislike        1548
adore          1548
want           1539
love           1534
disgust        1526
like           1520
super love     1519
indifferent    1512
cherish        1501
worried        1497
intrigued      1475
Name: Reaction_Type, dtype: int64
```

```
grouped_df = merged_df.groupby('Category')['Score'].sum().reset_index()
```

```
 grouped_df
```

	Category	Score
0	Animals	74965
1	Cooking	64756
2	Culture	66579
3	Dogs	52511
4	Education	57436
5	Fitness	55323
6	Food	66676
7	Healthy Eating	69339
8	Public Speaking	49264
9	Science	71168
10	Soccer	57783
11	Studying	54269
12	Technology	68738
13	Tennis	50339
14	Travel	64880
15	Veganism	49619

```
sorted_df = grouped_df.sort_values(by='Score', ascending=False)
```

```
sorted_df
```

	Category	Score
0	Animals	74965
9	Science	71168
7	Healthy Eating	69339
12	Technology	68738
6	Food	66676
2	Culture	66579
14	Travel	64880
1	Cooking	64756
10	Soccer	57783
4	Education	57436
5	Fitness	55323
11	Studying	54269
3	Dogs	52511
13	Tennis	50339
15	Veganism	49619
8	Public Speaking	49264

```
top_5_categories = sorted_df.head(5)
```

```
top_5_categories
```

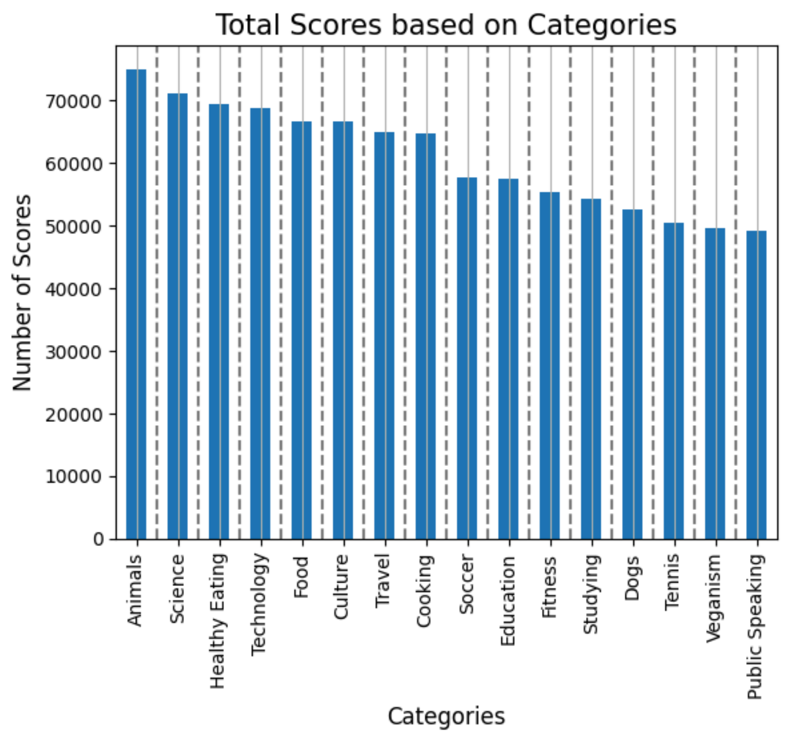
	Category	Score
0	Animals	74965
9	Science	71168
7	Healthy Eating	69339
12	Technology	68738
6	Food	66676

```
top_5_categories.to_csv('top_5_categories.csv', index=False)
```

```
#sns.set_style('whitegrid')

#plt.rcParams['figure.figsize'] = (12,6)
```

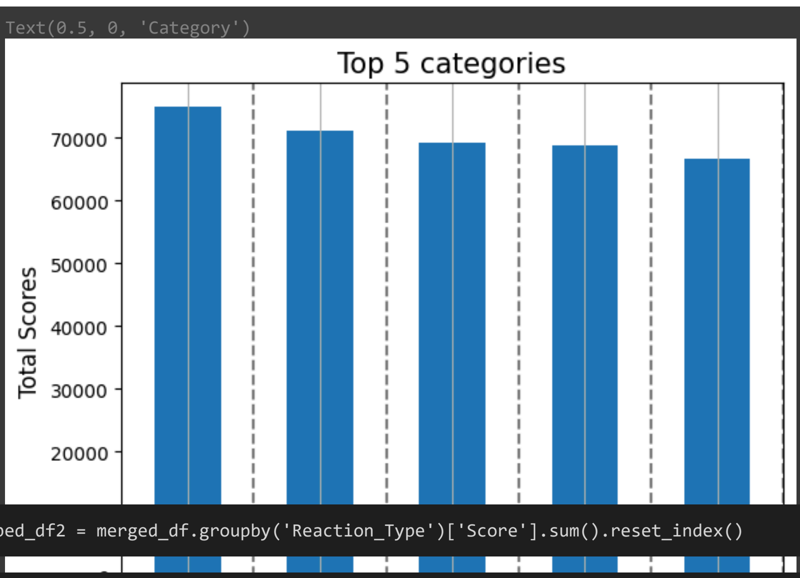
```
#This command shows the graphical view of gender column based on past_3_years_bike_related_purchases column:

graph1 = grouped_df.groupby('Category')['Score'].sum()
separation_lines = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
for line in separation_lines:
    plt.axvline(x=line - 0.5, color='grey', linestyle='--')
graph1 = graph1.sort_values(ascending=False)
graph1.plot(kind='bar')
plt.grid(axis='x')
plt.title('Total Scores based on Categories', fontsize=15)
plt.ylabel('Number of Scores', fontsize=12)
plt.yticks(np.arange(0, 80000, 10000))
plt.xlabel('Categories', fontsize=12)
```

```
Text(0.5, 0, 'Categories')
```



```
graph2 = top_5_categories.groupby('Category')['Score'].sum()
graph2 = graph2.sort_values(ascending=False)
separation_lines = [1, 2, 3, 4, 5]
for line in separation_lines:
    plt.axvline(x=line - 0.5, color='grey', linestyle='--')
graph1 = graph1.sort_values(ascending=False)
graph2.plot(kind='bar')
plt.grid(axis='x')
plt.title('Top 5 categories', fontsize=15)
plt.ylabel('Total Scores', fontsize=12)
plt.xlabel('Category', fontsize=12)
```

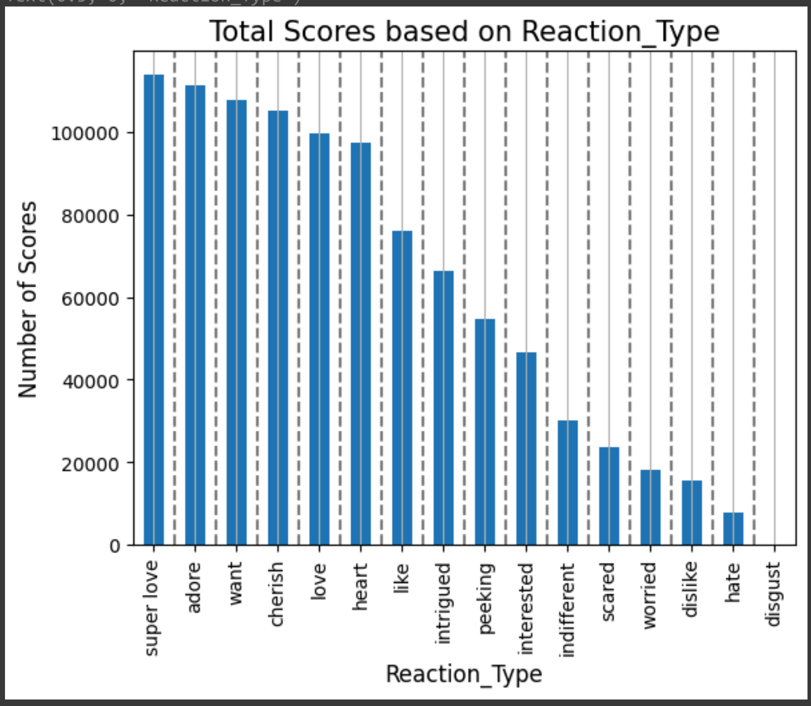
```
Text(0.5, 0, 'Category')
```



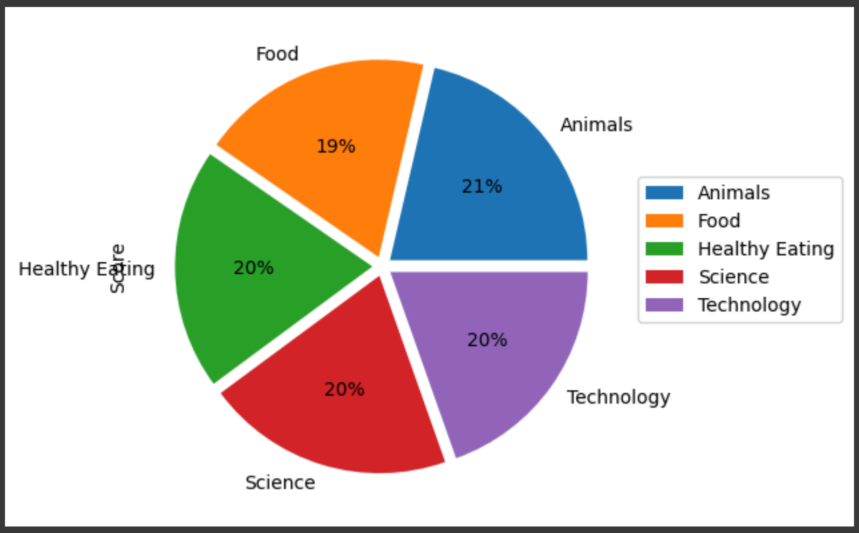
```
grouped_df2 = merged_df.groupby('Reaction_Type')['Score'].sum().reset_index()
```

```
graph3 = grouped_df2.groupby('Reaction_Type')['Score'].sum()
separation_lines = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
for line in separation_lines:
    plt.axvline(x=line - 0.5, color='grey', linestyle='--')
graph3 = graph3.sort_values(ascending=False)
graph3.plot(kind='bar')
plt.grid(axis='x')
plt.title('Total Scores based on Reaction_Type', fontsize=15)
plt.ylabel('Number of Scores', fontsize=12)
plt.xlabel('Reaction_Type', fontsize=12)
```

```
Text(0.5, 0, 'Reaction_Type')
```



```
# Define the ratio of gap of each fragment in a tuple
explode = (0.05, 0.05, 0.05, 0.05, 0.05)

# Plotting the pie chart for above dataframe
pie_chart = top_5_categories.groupby(['Category']).sum().plot(kind='pie', y='Score', autopct='%1.0f%%', explode=explode)

# Adjust legend location to avoid overlapping
plt.legend(loc='upper left', bbox_to_anchor=(1, 0.7))

# Show the plot
plt.show()
```


Food
Animals
Healthy Eating
Score
Technology
Science
Animals
Food
Healthy Eating
Science
Technology