In [1]:
```
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
%matplotlib inline
from matplotlib import style
```

In [21]:
```
mcd = pd.read_csv("C://Users//kolli Mallikarjuna//OneDrive//Desktop//project data//mer
mcd
```

Out[21]:

	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	...
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13.0	20	5.0	25	0.0	...
1	Breakfast	Egg White Delight	4.8 oz (135 g)	250	70	8.0	12	3.0	15	0.0	...
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23.0	35	8.0	42	0.0	...
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	250	28.0	43	10.0	52	0.0	...
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	210	23.0	35	8.0	42	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...
255	Smoothies & Shakes	McFlurry with Oreo Cookies (Small)	10.1 oz (285 g)	510	150	17.0	26	9.0	44	0.5	...
256	Smoothies & Shakes	McFlurry with Oreo Cookies (Medium)	13.4 oz (381 g)	690	200	23.0	35	12.0	58	1.0	...
257	Smoothies & Shakes	McFlurry with Oreo Cookies (Snack)	6.7 oz (190 g)	340	100	11.0	17	6.0	29	0.0	...
258	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Medium)	14.2 oz (403 g)	810	290	32.0	50	15.0	76	1.0	...
259	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Snack)	7.1 oz (202 g)	410	150	16.0	25	8.0	38	0.0	...

260 rows × 24 columns

In [22]:
```
mcd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 260 entries, 0 to 259
Data columns (total 24 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Category                      260 non-null    object
 1   Item                          260 non-null    object
 2   Serving Size                  260 non-null    object
 3   Calories                      260 non-null    int64
 4   Calories from Fat             260 non-null    int64
 5   Total Fat                     260 non-null    float64
 6   Total Fat (% Daily Value)     260 non-null    int64
 7   Saturated Fat                 260 non-null    float64
 8   Saturated Fat (% Daily Value) 260 non-null    int64
 9   Trans Fat                     260 non-null    float64
 10  Cholesterol                   260 non-null    int64
 11  Cholesterol (% Daily Value)   260 non-null    int64
 12  Sodium                        260 non-null    int64
 13  Sodium (% Daily Value)        260 non-null    int64
 14  Carbohydrates                 260 non-null    int64
 15  Carbohydrates (% Daily Value) 260 non-null    int64
 16  Dietary Fiber                 260 non-null    int64
 17  Dietary Fiber (% Daily Value) 260 non-null    int64
 18  Sugars                        260 non-null    int64
 19  Protein                       260 non-null    int64
 20  Vitamin A (% Daily Value)     260 non-null    int64
 21  Vitamin C (% Daily Value)     260 non-null    int64
 22  Calcium (% Daily Value)       260 non-null    int64
 23  Iron (% Daily Value)          260 non-null    int64
dtypes: float64(3), int64(18), object(3)
memory usage: 48.9+ KB
```

In [23]:
```
mcd.describe()
```

Out[23]:

	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	Choleste
count	260.000000	260.000000	260.000000	260.000000	260.000000	260.000000	260.000000	260.000
mean	368.269231	127.096154	14.165385	21.815385	6.007692	29.965385	0.203846	54.942
std	240.269886	127.875914	14.205998	21.885199	5.321873	26.639209	0.429133	87.269
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
25%	210.000000	20.000000	2.375000	3.750000	1.000000	4.750000	0.000000	5.000
50%	340.000000	100.000000	11.000000	17.000000	5.000000	24.000000	0.000000	35.000
75%	500.000000	200.000000	22.250000	35.000000	10.000000	48.000000	0.000000	65.000
max	1880.000000	1060.000000	118.000000	182.000000	20.000000	102.000000	2.500000	575.000

8 rows × 21 columns

In [24]:
```
mcd.columns
```

Out[24]:
```
Index(['Category', 'Item', 'Serving Size', 'Calories', 'Calories from Fat',
       'Total Fat', 'Total Fat (% Daily Value)', 'Saturated Fat',
       'Saturated Fat (% Daily Value)', 'Trans Fat', 'Cholesterol',
       'Cholesterol (% Daily Value)', 'Sodium', 'Sodium (% Daily Value)',
       'Carbohydrates', 'Carbohydrates (% Daily Value)', 'Dietary Fiber',
       'Dietary Fiber (% Daily Value)', 'Sugars', 'Protein',
       'Vitamin A (% Daily Value)', 'Vitamin C (% Daily Value)',
       'Calcium (% Daily Value)', 'Iron (% Daily Value)'],
      dtype='object')
```

In [25]:
```
mcd.isna().sum()
```

Out[25]:
```
Category                          0
Item                              0
Serving Size                      0
Calories                          0
Calories from Fat                 0
Total Fat                         0
Total Fat (% Daily Value)         0
Saturated Fat                     0
Saturated Fat (% Daily Value)     0
Trans Fat                         0
Cholesterol                       0
Cholesterol (% Daily Value)       0
Sodium                            0
Sodium (% Daily Value)            0
Carbohydrates                     0
Carbohydrates (% Daily Value)     0
Dietary Fiber                     0
Dietary Fiber (% Daily Value)     0
Sugars                            0
Protein                           0
Vitamin A (% Daily Value)         0
Vitamin C (% Daily Value)         0
Calcium (% Daily Value)           0
Iron (% Daily Value)              0
dtype: int64
```

In [26]:
```
mcd.shape
```

Out[26]:
```
(260, 24)
```

In [27]:
```
mcd.Category.value_counts()
```

Out[27]:
```
Category
Coffee & Tea          95
Breakfast             42
Smoothies & Shakes    28
Chicken & Fish        27
Beverages             27
Beef & Pork           15
Snacks & Sides        13
Desserts               7
Salads                 6
Name: count, dtype: int64
```

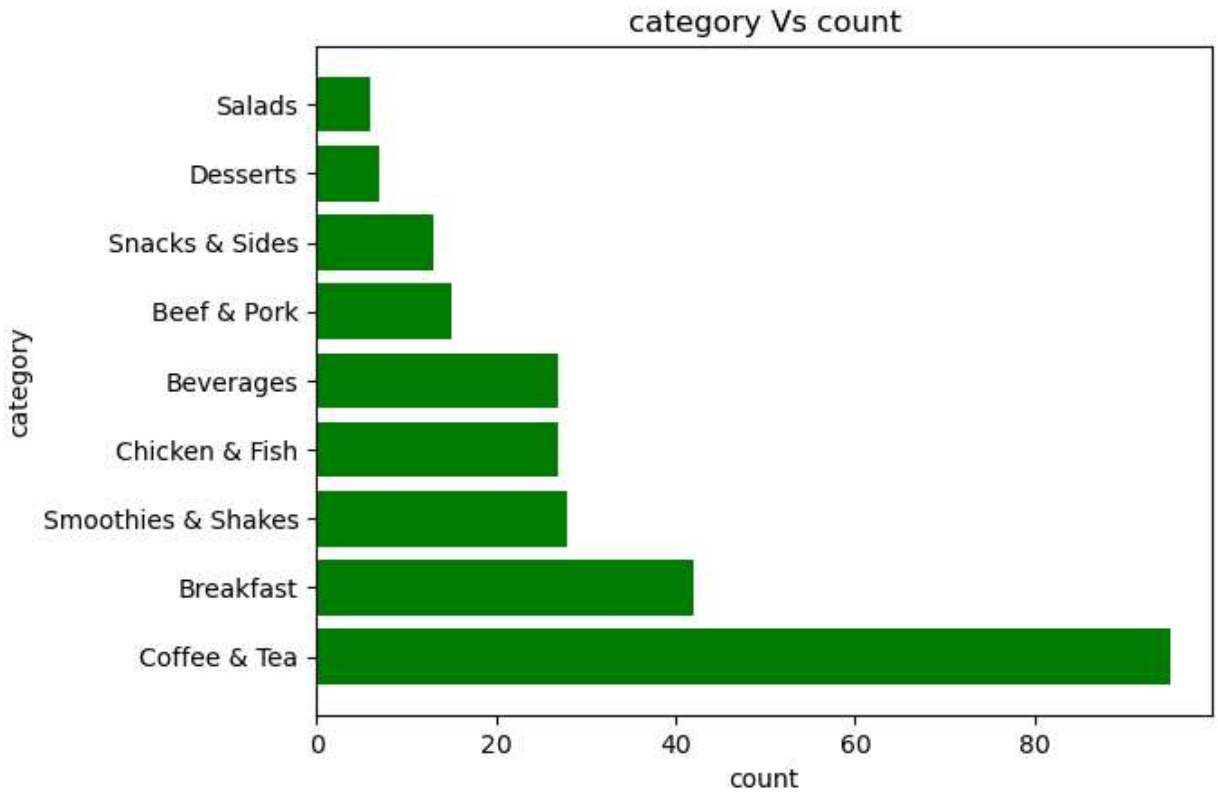
In [28]:
```
cat_val = mcd.Category.value_counts().values
cat_val
```

Out[28]:
```
array([95, 42, 28, 27, 27, 15, 13,  7,  6], dtype=int64)
```

In [30]:
```
cat_name = mcd.Category.value_counts().index
cat_name
```

Out[30]:
```
Index(['Coffee & Tea', 'Breakfast', 'Smoothies & Shakes', 'Chicken & Fish',
       'Beverages', 'Beef & Pork', 'Snacks & Sides', 'Desserts', 'Salads'],
      dtype='object', name='Category')
```

In [31]:
```
plt.barh(cat_name,cat_val,color = 'green')
plt.ylabel('category')
plt.xlabel('count')
plt.title('category Vs count');
```



Highest number of entries are from coffee & tea category and the least entries are from salad category.

In [34]:
```
cat = mcd.groupby(['Category','Item']).size().reset_index().rename(columns= {0:'entrie
cat
```

Out[34]:

	Category	Item	entries
0	Beef & Pork	Bacon Clubhouse Burger	1
1	Beef & Pork	Bacon McDouble	1
2	Beef & Pork	Big Mac	1
3	Beef & Pork	Cheeseburger	1
4	Beef & Pork	Daily Double	1
...	...	...	...
255	Snacks & Sides	Medium French Fries	1
256	Snacks & Sides	Ranch Snack Wrap (Crispy Chicken)	1
257	Snacks & Sides	Ranch Snack Wrap (Grilled Chicken)	1
258	Snacks & Sides	Side Salad	1
259	Snacks & Sides	Small French Fries	1

260 rows × 3 columns

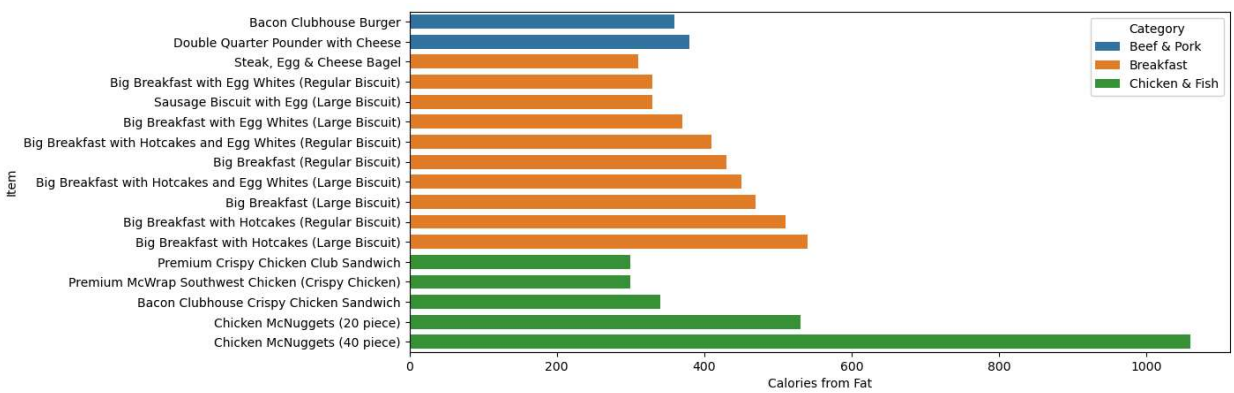
In [53]:
```
cal = mcd[mcd['Calories from Fat']>=300].groupby(['Category','Calories from Fat','Item
cal
```

Out[53]:

	Category	Calories from Fat	Item	entries
0	Beef & Pork	360	Bacon Clubhouse Burger	1
1	Beef & Pork	380	Double Quarter Pounder with Cheese	1
2	Breakfast	310	Steak, Egg & Cheese Bagel	1
3	Breakfast	330	Big Breakfast with Egg Whites (Regular Biscuit)	1
4	Breakfast	330	Sausage Biscuit with Egg (Large Biscuit)	1
5	Breakfast	370	Big Breakfast with Egg Whites (Large Biscuit)	1
6	Breakfast	410	Big Breakfast with Hotcakes and Egg Whites (Re...	1
7	Breakfast	430	Big Breakfast (Regular Biscuit)	1
8	Breakfast	450	Big Breakfast with Hotcakes and Egg Whites (La...	1
9	Breakfast	470	Big Breakfast (Large Biscuit)	1
10	Breakfast	510	Big Breakfast with Hotcakes (Regular Biscuit)	1
11	Breakfast	540	Big Breakfast with Hotcakes (Large Biscuit)	1
12	Chicken & Fish	300	Premium Crispy Chicken Club Sandwich	1
13	Chicken & Fish	300	Premium McWrap Southwest Chicken (Crispy Chicken)	1
14	Chicken & Fish	340	Bacon Clubhouse Crispy Chicken Sandwich	1
15	Chicken & Fish	530	Chicken McNuggets (20 piece)	1
16	Chicken & Fish	1060	Chicken McNuggets (40 piece)	1

In [54]:

```
matplotlib.rcParams['figure.figsize'] = (12,5)
sns.barplot(y='Item',x='Calories from Fat',data = cal, hue ='Category' ,width = 0.7, c
```



Highest calories from fat are from Chiken Mc Nuggets(40 pc), from chicken and fish category; Big breakfast with Hot cakes(Large biscuits) from break fast category; and Double Quarter pounder with cheese from Beef and pork category. All these 3 categories (Beef and pork, Breakfast, Chickedn and fish) are high in calories from fat.

In [67]:

```
zero_calfromfat = mcd[mcd['Calories from Fat']==0]
zero_calfromfat
```

Out[67]:

	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	...
100	Snacks & Sides	Side Salad	3.1 oz (87 g)	20	0	0.0	0	0.0	0	0.0	...
101	Snacks & Sides	Apple Slices	1.2 oz (34 g)	15	0	0.0	0	0.0	0	0.0	...
110	Beverages	Coca-Cola Classic (Small)	16 fl oz cup	140	0	0.0	0	0.0	0	0.0	...
111	Beverages	Coca-Cola Classic (Medium)	21 fl oz cup	200	0	0.0	0	0.0	0	0.0	...
112	Beverages	Coca-Cola Classic (Large)	30 fl oz cup	280	0	0.0	0	0.0	0	0.0	...
113	Beverages	Coca-Cola Classic (Child)	12 fl oz cup	100	0	0.0	0	0.0	0	0.0	...
114	Beverages	Diet Coke (Small)	16 fl oz cup	0	0	0.0	0	0.0	0	0.0	...
115	Beverages	Diet Coke (Medium)	21 fl oz cup	0	0	0.0	0	0.0	0	0.0	...
116	Beverages	Diet Coke (Large)	30 fl oz cup	0	0	0.0	0	0.0	0	0.0	...
117	Beverages	Diet Coke (Child)	12 fl oz cup	0	0	0.0	0	0.0	0	0.0	...
118	Beverages	Dr Pepper (Small)	16 fl oz cup	140	0	0.0	0	0.0	0	0.0	...
119	Beverages	Dr Pepper (Medium)	21 fl oz cup	190	0	0.0	0	0.0	0	0.0	...
120	Beverages	Dr Pepper (Large)	30 fl oz cup	270	0	0.0	0	0.0	0	0.0	...
121	Beverages	Dr Pepper (Child)	12 fl oz cup	100	0	0.0	0	0.0	0	0.0	...
122	Beverages	Diet Dr Pepper (Small)	16 fl oz cup	0	0	0.0	0	0.0	0	0.0	...
123	Beverages	Diet Dr Pepper (Medium)	21 fl oz cup	0	0	0.0	0	0.0	0	0.0	...

	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	...
124	Beverages	Diet Dr Pepper (Large)	30 fl oz cup	0	0	0.0	0	0.0	0	0.0	...
125	Beverages	Diet Dr Pepper (Child)	12 fl oz cup	0	0	0.0	0	0.0	0	0.0	...
126	Beverages	Sprite (Small)	16 fl oz cup	140	0	0.0	0	0.0	0	0.0	...
127	Beverages	Sprite (Medium)	21 fl oz cup	200	0	0.0	0	0.0	0	0.0	...
128	Beverages	Sprite (Large)	30 fl oz cup	280	0	0.0	0	0.0	0	0.0	...
129	Beverages	Sprite (Child)	12 fl oz cup	100	0	0.0	0	0.0	0	0.0	...
131	Beverages	Fat Free Chocolate Milk Jug	1 carton (236 ml)	130	0	0.0	0	0.0	0	0.0	...
132	Beverages	Minute Maid 100% Apple Juice Box	6 fl oz (177 ml)	80	0	0.0	0	0.0	0	0.0	...
133	Beverages	Minute Maid Orange Juice (Small)	12 fl oz cup	150	0	0.0	0	0.0	0	0.0	...
134	Beverages	Minute Maid Orange Juice (Medium)	16 fl oz cup	190	0	0.0	0	0.0	0	0.0	...
135	Beverages	Minute Maid Orange Juice (Large)	22 fl oz cup	280	0	0.0	0	0.0	0	0.0	...
136	Beverages	Dasani Water Bottle	16.9 fl oz	0	0	0.0	0	0.0	0	0.0	...
137	Coffee & Tea	Iced Tea (Small)	16 fl oz cup	0	0	0.0	0	0.0	0	0.0	...
138	Coffee & Tea	Iced Tea (Medium)	21 fl oz cup	0	0	0.0	0	0.0	0	0.0	...

	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	...
139	Coffee & Tea	Iced Tea (Large)	30 fl oz cup	0	0	0.0	0	0.0	0	0.0	...
140	Coffee & Tea	Iced Tea (Child)	12 fl oz cup	0	0	0.0	0	0.0	0	0.0	...
141	Coffee & Tea	Sweet Tea (Small)	16 fl oz cup	150	0	0.0	0	0.0	0	0.0	...
142	Coffee & Tea	Sweet Tea (Medium)	21 fl oz cup	180	0	0.0	0	0.0	0	0.0	...
143	Coffee & Tea	Sweet Tea (Large)	30 fl oz cup	220	0	0.0	0	0.0	0	0.0	...
144	Coffee & Tea	Sweet Tea (Child)	12 fl oz cup	110	0	0.0	0	0.0	0	0.0	...
145	Coffee & Tea	Coffee (Small)	12 fl oz cup	0	0	0.0	0	0.0	0	0.0	...
146	Coffee & Tea	Coffee (Medium)	16 fl oz cup	0	0	0.0	0	0.0	0	0.0	...
147	Coffee & Tea	Coffee (Large)	16 fl oz cup	0	0	0.0	0	0.0	0	0.0	...
163	Coffee & Tea	Nonfat Latte (Small)	12 fl oz cup	100	0	0.0	0	0.0	0	0.0	...
164	Coffee & Tea	Nonfat Latte (Medium)	16 fl oz cup	130	0	0.0	0	0.0	0	0.0	...
165	Coffee & Tea	Nonfat Latte (Large)	20 fl oz cup	170	0	0.5	1	0.0	0	0.0	...
166	Coffee & Tea	Nonfat Caramel Latte (Small)	12 fl oz cup	200	0	0.0	0	0.0	0	0.0	...
167	Coffee & Tea	Nonfat Caramel Latte (Medium)	16 fl oz cup	250	0	0.0	0	0.0	0	0.0	...
168	Coffee & Tea	Nonfat Caramel Latte (Large)	20 fl oz cup	310	0	0.5	1	0.0	0	0.0	...
169	Coffee & Tea	Nonfat Hazelnut Latte (Small)	12 fl oz cup	200	0	0.0	0	0.0	0	0.0	...

	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	...
170	Coffee & Tea	Nonfat Hazelnut Latte (Medium)	16 fl oz cup	250	0	0.0	0	0.0	0	0.0	...
171	Coffee & Tea	Nonfat Hazelnut Latte (Large)	20 fl oz cup	310	0	0.5	1	0.0	0	0.0	...
172	Coffee & Tea	Nonfat French Vanilla Latte (Small)	12 fl oz cup	190	0	0.0	0	0.0	0	0.0	...
173	Coffee & Tea	Nonfat French Vanilla Latte (Medium)	16 fl oz cup	240	0	0.0	0	0.0	0	0.0	...
174	Coffee & Tea	Nonfat French Vanilla Latte (Large)	20 fl oz cup	300	0	0.5	1	0.0	0	0.0	...
175	Coffee & Tea	Nonfat Latte with Sugar Free French Vanilla Sy...	12 fl oz cup	140	0	0.0	0	0.0	0	0.0	...
176	Coffee & Tea	Nonfat Latte with Sugar Free French Vanilla Sy...	16 fl oz cup	170	0	0.0	0	0.0	0	0.0	...
177	Coffee & Tea	Nonfat Latte with Sugar Free French Vanilla Sy...	20 fl oz cup	220	0	0.5	1	0.0	0	0.0	...

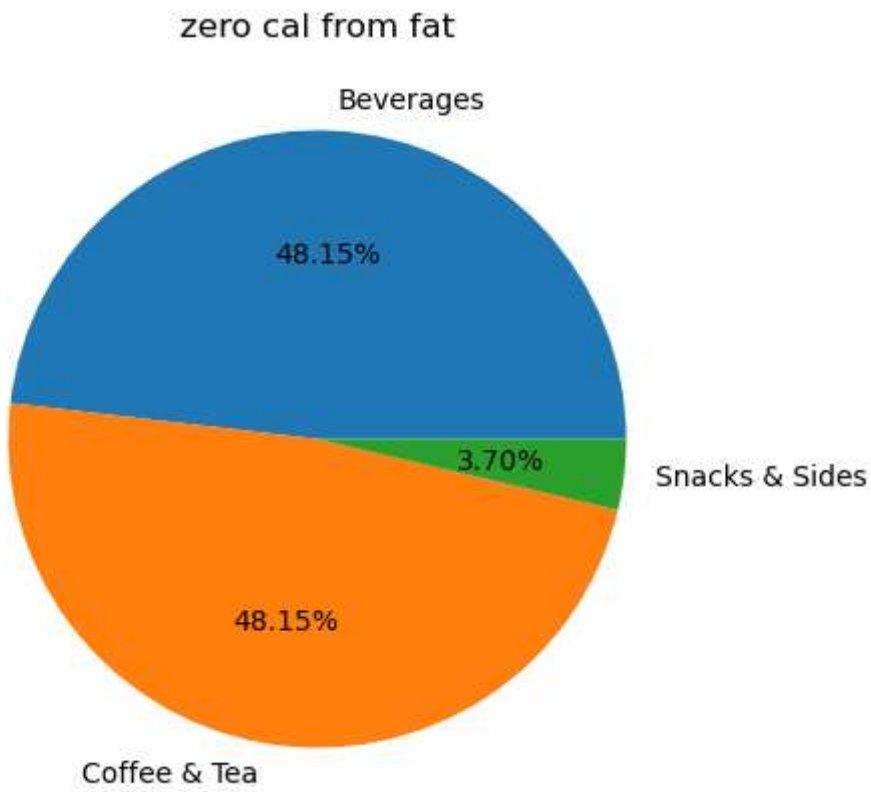
In [70]:
```
Zero_calfromfatcat= zero_calfromfat.Category.value_counts().index
Zero_calfromfatcat
```

Out[70]:
```
Index(['Beverages', 'Coffee & Tea', 'Snacks & Sides'], dtype='object', name='Categor
y')
```

In [71]:
```
Zero_calfromfatcatnum= zero_calfromfat.Category.value_counts().values
Zero_calfromfatcatnum
```

Out[71]:
```
array([26, 26,  2], dtype=int64)
```

In [72]:
```
plt.pie(Zero_calfromfatcatnum,labels = Zero_calfromfatcat, autopct = '%1.2f%%')
plt.title('zero cal from fat');
```



In [78]:
```
chlstrol_fat = mcd.groupby(['Cholesterol','Total Fat', 'Category']).size().reset_index
chlstrol_fat
```

Out[78]:

	Cholesterol	Total Fat	Category	0
0	0	0.0	Beverages	25
1	0	0.0	Coffee & Tea	11
2	0	0.0	Snacks & Sides	2
3	0	5.0	Snacks & Sides	1
4	0	9.0	Breakfast	1
...	...	...	...	...
149	300	23.0	Breakfast	1
150	555	48.0	Breakfast	1
151	555	52.0	Breakfast	1
152	575	56.0	Breakfast	1
153	575	60.0	Breakfast	1

154 rows × 4 columns

In [79]:
```
chlstrol_fat.tail(20)
```

Out[79]:

	Cholesterol	Total Fat	Category	0
134	115	40.0	Beef & Pork	1
135	135	59.0	Chicken & Fish	1
136	160	43.0	Beef & Pork	1
137	250	21.0	Breakfast	1
138	250	26.0	Breakfast	1
139	250	30.0	Breakfast	1
140	250	33.0	Breakfast	1
141	250	37.0	Breakfast	1
142	260	13.0	Breakfast	1
143	265	31.0	Breakfast	1
144	265	118.0	Chicken & Fish	1
145	275	31.0	Breakfast	1
146	280	32.0	Breakfast	1
147	285	28.0	Breakfast	1
148	295	35.0	Breakfast	1
149	300	23.0	Breakfast	1
150	555	48.0	Breakfast	1
151	555	52.0	Breakfast	1
152	575	56.0	Breakfast	1
153	575	60.0	Breakfast	1

In [80]:

```
chlstrol_fat.head(20)
```

Out[80]:

	Cholesterol	Total Fat	Category	0
0	0	0.0	Beverages	25
1	0	0.0	Coffee & Tea	11
2	0	0.0	Snacks & Sides	2
3	0	5.0	Snacks & Sides	1
4	0	9.0	Breakfast	1
5	0	11.0	Snacks & Sides	1
6	0	13.0	Desserts	1
7	0	16.0	Snacks & Sides	1
8	0	24.0	Snacks & Sides	1
9	5	0.0	Beverages	1
10	5	0.0	Coffee & Tea	10
11	5	0.5	Smoothies & Shakes	3
12	5	1.0	Smoothies & Shakes	6
13	5	1.5	Desserts	1
14	5	2.0	Snacks & Sides	1
15	5	4.0	Breakfast	2
16	10	0.5	Coffee & Tea	5
17	10	2.5	Beverages	1
18	10	4.5	Salads	1
19	10	6.0	Desserts	1

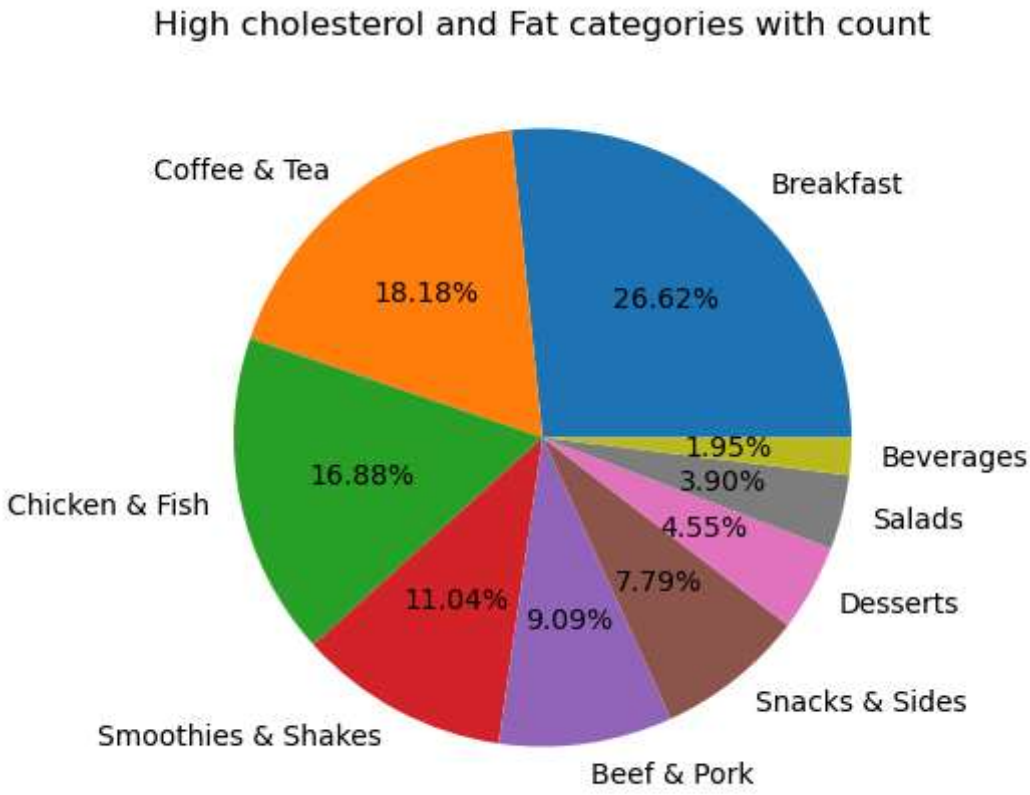
In [83]:
```
CF_cat = chlstrol_fat.Category.value_counts().index
CF_cat
```

Out[83]:
```
Index(['Breakfast', 'Coffee & Tea', 'Chicken & Fish', 'Smoothies & Shakes',
       'Beef & Pork', 'Snacks & Sides', 'Desserts', 'Salads', 'Beverages'],
      dtype='object', name='Category')
```

In [84]:
```
CF_count = chlstrol_fat.Category.value_counts().values
CF_count
```

Out[84]:
```
array([41, 28, 26, 17, 14, 12,  7,  6,  3], dtype=int64)
```

In [85]:
```
plt.pie(CF_count,labels = CF_cat, autopct = '%1.2f%%')
plt.title('High cholesterol and Fat categories with count');
```



In [105…

```
Fibre_Protien = mcd[mcd['Protein']>=30].groupby(['Protein','Dietary Fiber','Category']
Fibre_Protien
```

Out[105]:

	Protein	Dietary Fiber	Category	entries
0	30	3	Beef & Pork	1
1	30	3	Breakfast	2
2	30	3	Chicken & Fish	1
3	31	5	Chicken & Fish	1
4	32	3	Chicken & Fish	2
5	33	3	Breakfast	1
6	35	6	Breakfast	1
7	35	7	Breakfast	1
8	36	3	Chicken & Fish	3
9	36	4	Chicken & Fish	1
10	36	6	Breakfast	1
11	36	7	Breakfast	1
12	37	3	Beef & Pork	2
13	39	4	Beef & Pork	1
14	40	3	Chicken & Fish	1
15	40	4	Chicken & Fish	1
16	44	3	Chicken & Fish	1
17	48	3	Beef & Pork	1
18	87	6	Chicken & Fish	1

In [106…

```
FP_highcat = Fibre_Protien.Category.value_counts().index
FP_highcat
```

Out[106]:

```
Index(['Chicken & Fish', 'Breakfast', 'Beef & Pork'], dtype='object', name='Categor
y')
```

In [107…

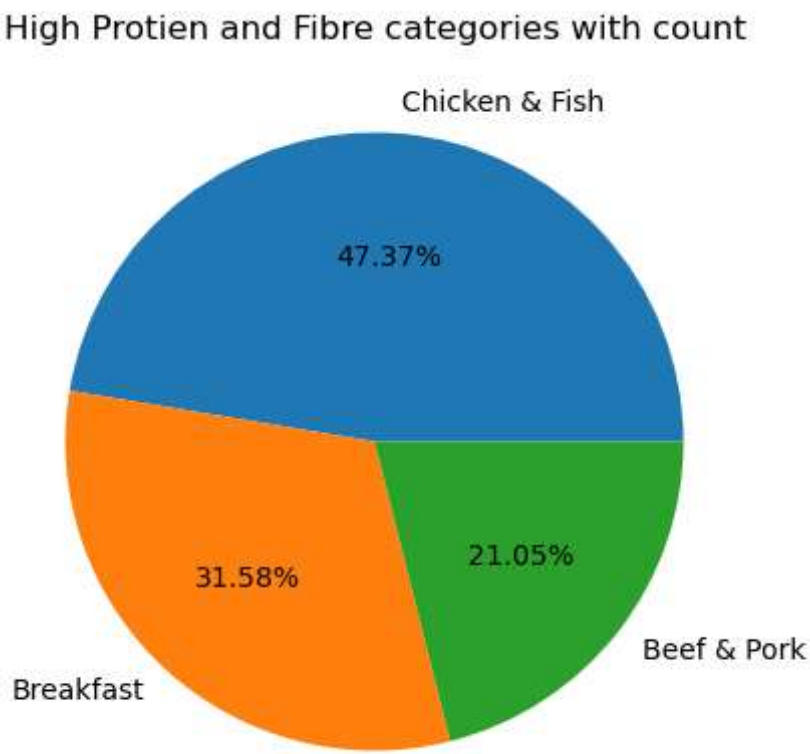
```
FP_highcount = Fibre_Protien.Category.value_counts().values
FP_highcount
```

Out[107]:

```
array([9, 6, 4], dtype=int64)
```

In [108…

```
plt.pie(FP_highcount,labels = FP_highcat, autopct = '%1.2f%%')
plt.title('High Protien and Fibre categories with count');
```



In [140…

```
CoffeeTea= mcd[mcd['Category']=='Coffee & Tea'].groupby(['Calories','Sugars','Calcium
CoffeeTea
```

Out[140]:

	Calories	Sugars	Calcium	Item	entries
0	0	0	0	Coffee (Large)	1
1	0	0	0	Coffee (Medium)	1
2	0	0	0	Coffee (Small)	1
3	0	0	0	Iced Tea (Child)	1
4	0	0	0	Iced Tea (Large)	1
...	...	...	...	...	...
90	550	71	30	Frappé Caramel (Medium)	1
91	630	81	30	Frappé Chocolate Chip (Medium)	1
92	670	88	35	Frappé Caramel (Large)	1
93	670	88	35	Frappé Mocha (Large)	1
94	760	99	35	Frappé Chocolate Chip (Large)	1

95 rows × 5 columns

In [125…

```
healthyCoffeeTea = CoffeeTea[CoffeeTea['Sugars']<20]
healthyCoffeeTea
```

Out[125]:

	Calories	Sugars	Calcium	Item	entries
0	0	0	0	Coffee (Large)	1
1	0	0	0	Coffee (Medium)	1
2	0	0	0	Coffee (Small)	1
3	0	0	0	Iced Tea (Child)	1
4	0	0	0	Iced Tea (Large)	1
5	0	0	0	Iced Tea (Medium)	1
6	0	0	0	Iced Tea (Small)	1
7	80	1	4	Iced Coffee with Sugar Free French Vanilla Syr...	1
8	100	13	30	Nonfat Latte (Small)	1
10	120	2	4	Iced Coffee with Sugar Free French Vanilla Syr...	1
11	120	19	4	French Vanilla Iced Coffee (Small)	1
12	130	16	40	Nonfat Latte (Medium)	1
15	140	13	30	Nonfat Latte with Sugar Free French Vanilla Sy...	1
18	160	2	8	Iced Coffee with Sugar Free French Vanilla Syr...	1
19	170	12	30	Latte (Small)	1
20	170	16	40	Nonfat Latte with Sugar Free French Vanilla Sy...	1
30	210	12	30	Latte with Sugar Free French Vanilla Syrup (Sm...	1
31	210	15	35	Latte (Medium)	1
42	260	15	35	Latte with Sugar Free French Vanilla Syrup (Me...	1

These are the healthy options from the Category of Coffee and Tea

In [135…

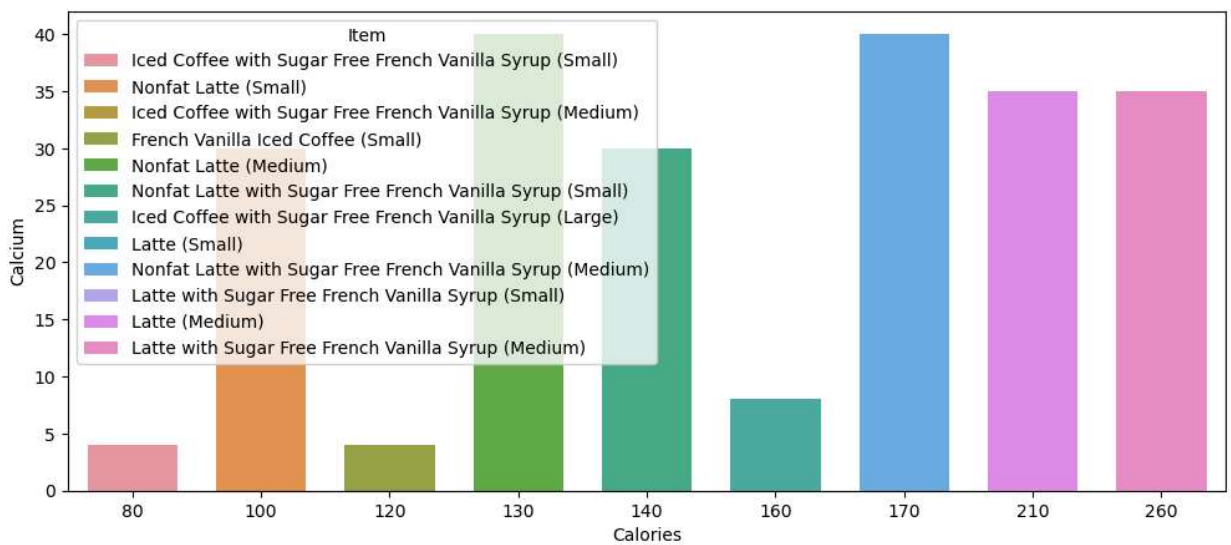
```
hlthyCT = healthyCoffeeTea[healthyCoffeeTea['Calories']>0]
hlthyCT
```

Out[135]:

	Calories	Sugars	Calcium	Item	entries
7	80	1	4	Iced Coffee with Sugar Free French Vanilla Syr...	1
8	100	13	30	Nonfat Latte (Small)	1
10	120	2	4	Iced Coffee with Sugar Free French Vanilla Syr...	1
11	120	19	4	French Vanilla Iced Coffee (Small)	1
12	130	16	40	Nonfat Latte (Medium)	1
15	140	13	30	Nonfat Latte with Sugar Free French Vanilla Sy...	1
18	160	2	8	Iced Coffee with Sugar Free French Vanilla Syr...	1
19	170	12	30	Latte (Small)	1
20	170	16	40	Nonfat Latte with Sugar Free French Vanilla Sy...	1
30	210	12	30	Latte with Sugar Free French Vanilla Syrup (Sm...	1
31	210	15	35	Latte (Medium)	1
42	260	15	35	Latte with Sugar Free French Vanilla Syrup (Me...	1

In [136…

```
matplotlib.rcParams['figure.figsize'] = (12,5)
sns.barplot(x='Calories',y='Calcium',data = hlthyCT, hue = 'Item',width = 0.7, dodge =
```



In [145…

```
breakfast= mcd[mcd['Category']=='Breakfast'].groupby(['Calories','Protein','Dietary Fi
breakfast
```

Out[145]:

	Calories	Protein	Dietary Fiber	Carbohydrates	Item	entries
0	150	1	2	15	Hash Brown	1
1	250	18	4	30	Egg White Delight	1
2	260	5	5	49	Fruit & Maple Oatmeal without Brown Sugar	1
3	290	5	5	58	Fruit & Maple Oatmeal	1
4	300	12	1	26	Sausage Burrito	1
5	300	17	4	31	Egg McMuffin	1
6	350	8	3	60	Hotcakes	1
7	370	14	4	29	Sausage McMuffin	1
8	400	20	2	47	Bacon, Egg & Cheese McGriddles with Egg Whites	1
9	400	21	4	30	Sausage McMuffin with Egg Whites	1
10	410	17	2	41	Southern Style Chicken Biscuit (Regular Biscuit)	1
11	410	20	2	36	Bacon, Egg & Cheese Biscuit with Egg Whites (R...	1
12	420	11	2	44	Sausage McGriddles	1
13	430	11	2	34	Sausage Biscuit (Regular Biscuit)	1
14	430	26	4	31	Steak & Egg McMuffin	1
15	450	21	4	30	Sausage McMuffin with Egg	1
16	460	6	3	66	Cinnamon Melts	1
17	460	18	2	34	Sausage Biscuit with Egg Whites (Regular Biscuit)	1
18	460	19	2	38	Bacon, Egg & Cheese Biscuit (Regular Biscuit)	1
19	460	19	2	48	Bacon, Egg & Cheese McGriddles	1
20	470	17	3	46	Southern Style Chicken Biscuit (Large Biscuit)	1
21	470	20	3	42	Bacon, Egg & Cheese Biscuit with Egg Whites (L...	1
22	480	11	3	39	Sausage Biscuit (Large Biscuit)	1
23	500	21	2	46	Sausage, Egg & Cheese McGriddles with Egg Whites	1
24	510	18	2	36	Sausage Biscuit with Egg (Regular Biscuit)	1
25	520	15	3	61	Hotcakes and Sausage	1
26	520	18	3	40	Sausage Biscuit with Egg Whites (Large Biscuit)	1

	Calories	Protein	Dietary Fiber	Carbohydrates	Item	entries
27	520	19	3	43	Bacon, Egg & Cheese Biscuit (Large Biscuit)	1
28	540	25	2	38	Steak & Egg Biscuit (Regular Biscuit)	1
29	550	20	2	48	Sausage, Egg & Cheese McGriddles	1
30	570	18	3	42	Sausage Biscuit with Egg (Large Biscuit)	1
31	570	30	3	55	Bacon, Egg & Cheese Bagel with Egg Whites	1
32	620	30	3	57	Bacon, Egg & Cheese Bagel	1
33	640	26	3	50	Big Breakfast with Egg Whites (Regular Biscuit)	1
34	670	33	3	56	Steak, Egg & Cheese Bagel	1
35	690	26	4	55	Big Breakfast with Egg Whites (Large Biscuit)	1
36	740	28	3	51	Big Breakfast (Regular Biscuit)	1
37	800	28	4	56	Big Breakfast (Large Biscuit)	1
38	990	35	6	110	Big Breakfast with Hotcakes and Egg Whites (Re...	1
39	1050	35	7	115	Big Breakfast with Hotcakes and Egg Whites (La...	1
40	1090	36	6	111	Big Breakfast with Hotcakes (Regular Biscuit)	1
41	1150	36	7	116	Big Breakfast with Hotcakes (Large Biscuit)	1

In [147…

```
hlthyBF = breakfast[breakfast['Calories']<350]
hlthyBF
```

Out[147]:

	Calories	Protein	Dietary Fiber	Carbohydrates	Item	entries
0	150	1	2	15	Hash Brown	1
1	250	18	4	30	Egg White Delight	1
2	260	5	5	49	Fruit & Maple Oatmeal without Brown Sugar	1
3	290	5	5	58	Fruit & Maple Oatmeal	1
4	300	12	1	26	Sausage Burrito	1
5	300	17	4	31	Egg McMuffin	1

These are the healthy options of breakfast

In [ ]:

In [ ]:

```

```