In [3]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

In [5]:
```python
data = pd.read_csv("C:\\Users\\DHANUSHA\\Desktop\\menu.csv")
data
```

Out[5]:

	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	...	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)
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13.0	20	5.0	25	0.0	...	31	10	4	17	3	17	10	0	25	15
1	Breakfast	Egg White Delight	4.8 oz (135 g)	250	70	8.0	12	3.0	15	0.0	...	30	10	4	17	3	18	6	0	25	8
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23.0	35	8.0	42	0.0	...	29	10	4	17	2	14	8	0	25	10
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	250	28.0	43	10.0	52	0.0	...	30	10	4	17	2	21	15	0	30	15
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	210	23.0	35	8.0	42	0.0	...	30	10	4	17	2	21	6	0	25	10
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
255	Smoothies & Shakes	McFlurry with Oreo Cookies (Small)	10.1 oz (285 g)	510	150	17.0	26	9.0	44	0.5	...	80	27	1	4	64	12	15	0	40	8
256	Smoothies & Shakes	McFlurry with Oreo Cookies (Medium)	13.4 oz (381 g)	690	200	23.0	35	12.0	58	1.0	...	106	35	1	5	85	15	20	0	50	10
257	Smoothies & Shakes	McFlurry with Oreo Cookies (Snack)	6.7 oz (190 g)	340	100	11.0	17	6.0	29	0.0	...	53	18	1	2	43	8	10	0	25	6
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	...	114	38	2	9	103	21	20	0	60	6
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	...	57	19	1	5	51	10	10	0	30	4

260 rows × 24 columns

In [6]:
```python
data.head(5)
```

Out[6]:

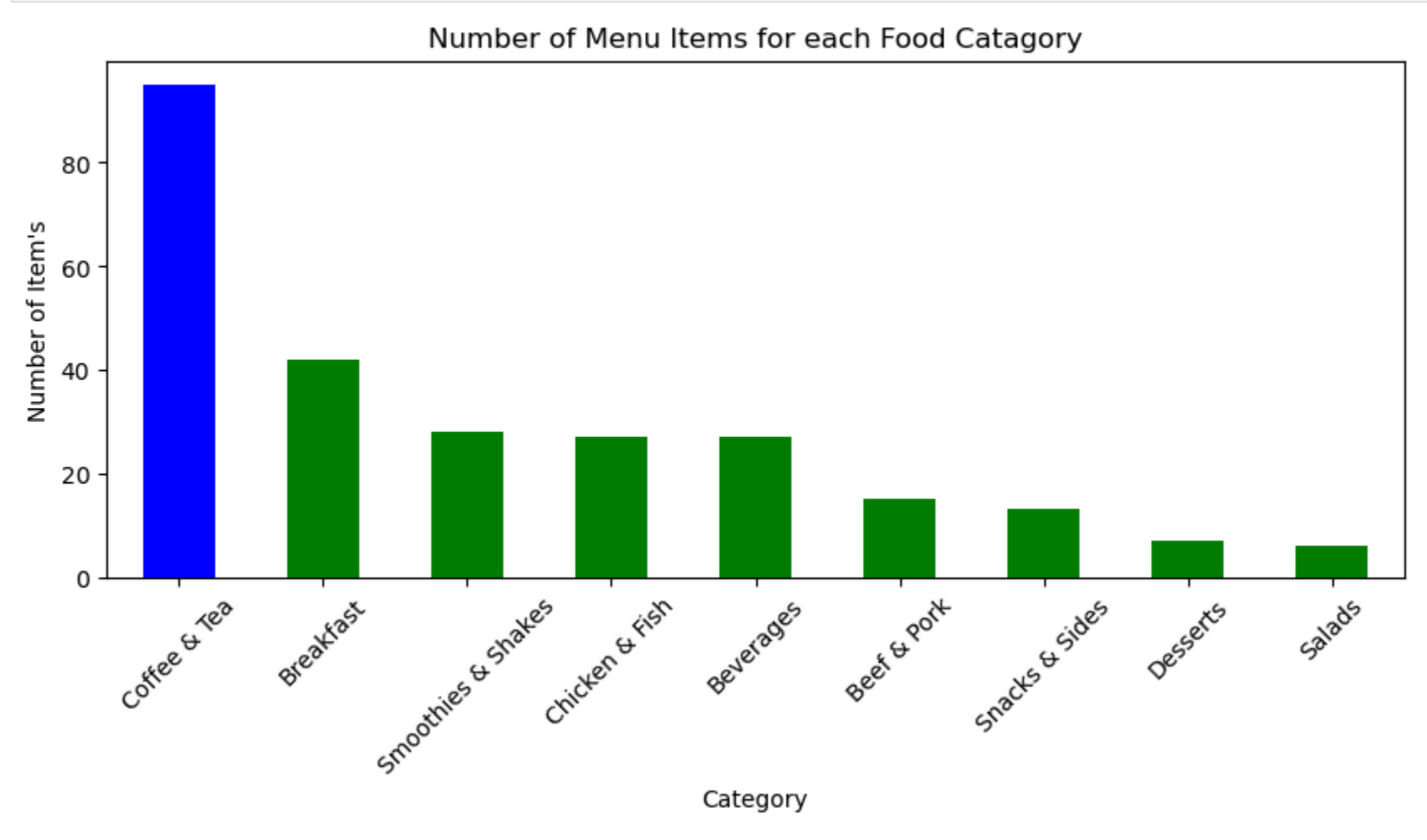
	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	...	Carbohydrates	Carbohydrates (% Daily Value)	Dietary Fiber	Dietary Fiber (% Daily Value)	Sugars	Protein	Vitamin A (% Daily Value)	Vitamin C (% Daily Value)	Calcium (% Daily Value)	Iron (% Daily Value)
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13.0	20	5.0	25	0.0	...	31	10	4	17	3	17	10	0	25	15
1	Breakfast	Egg White Delight	4.8 oz (135 g)	250	70	8.0	12	3.0	15	0.0	...	30	10	4	17	3	18	6	0	25	8
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23.0	35	8.0	42	0.0	...	29	10	4	17	2	14	8	0	25	10
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	250	28.0	43	10.0	52	0.0	...	30	10	4	17	2	21	15	0	30	15
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	210	23.0	35	8.0	42	0.0	...	30	10	4	17	2	21	6	0	25	10

5 rows × 24 columns

In [7]:
```python
data.shape
```

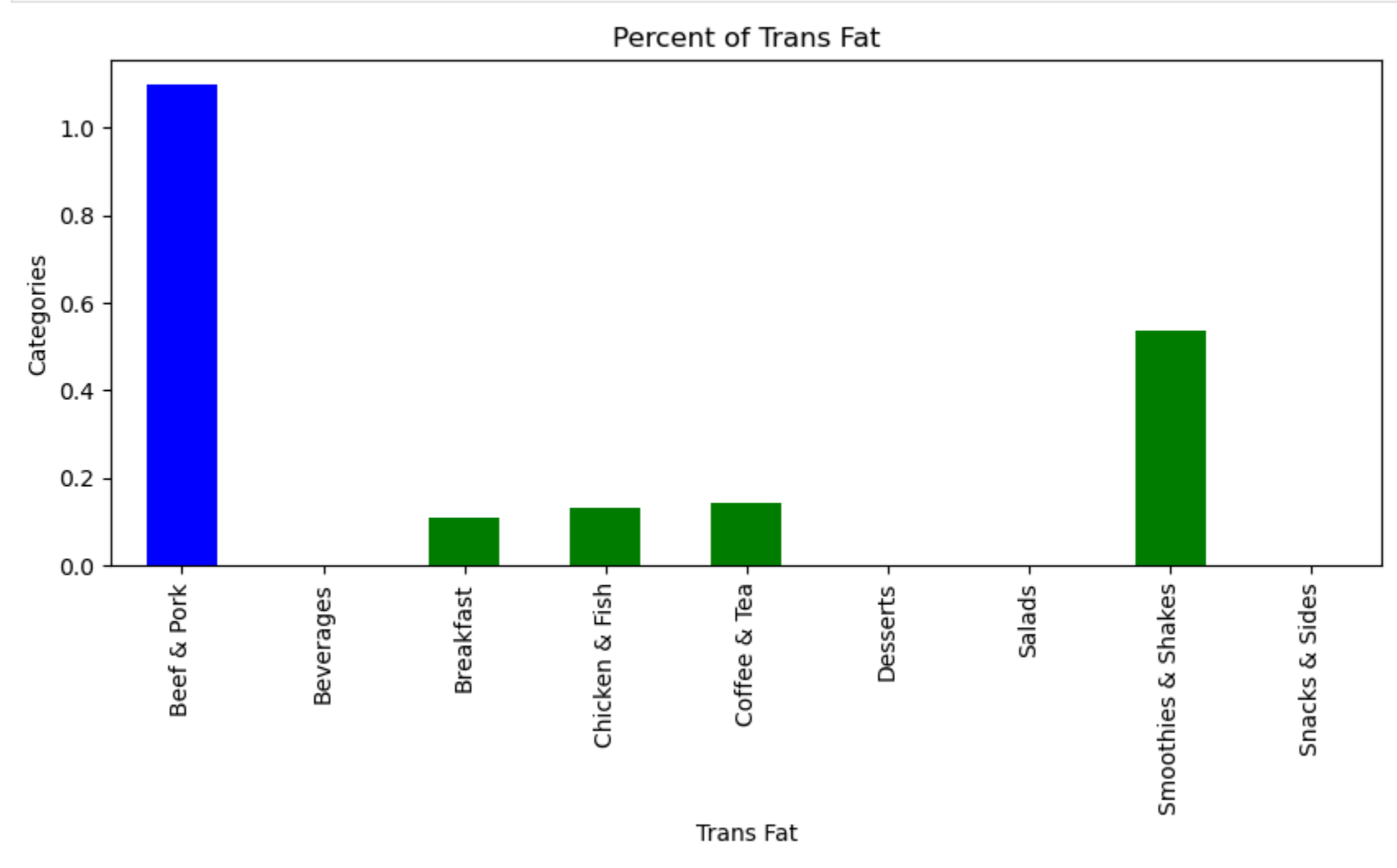
Out[7]:
```
(260, 24)
```

In [8]:
```python
#Graph for Number of Menu Items for each Catagory
plt.figure(figsize=(10, 4), dpi=100)
menu_category = data.Category.value_counts()
menu_category.plot.bar(color = ['blue','green','green','green','green','green','green','green','green']) #graph color
plt.title("Number of Menu Items for each Food Catagory")
plt.ylabel("Number of Item's")
plt.xlabel("Category")
plt.xticks(rotation=45)
plt.show()
```



In [9]:
```python
#Plot graph for relationship between Calories vs Calories from Fat
plt.figure(figsize=(10, 4), dpi=100)
correlation = data['Calories'].corr(data['Calories from Fat'])
plt.scatter(data.Calories, data['Calories from Fat'], color='blue')
plt.text(500,450, 'r = {}'.format(round(correlation,2)))
plt.xlabel("Calories")
plt.ylabel("Calories form Fat")
plt.title("The Relationship Between Calories vs Calories from Fat")
plt.show()
```

In [10]:
```python
#Bar Graph for Percent of Trans Fat
plt.figure (figsize=(10, 4), dpi=100)
menu_category = data.groupby('Category')['Trans Fat'].mean()
menu_category.plot.bar(color = ['blue','green','green','green','green','green','green','green','green'])
plt.title("Percent of Trans Fat")
plt.xlabel("Trans Fat")
plt.ylabel("Categories")
plt.xticks(rotation=90)
plt.show()
```



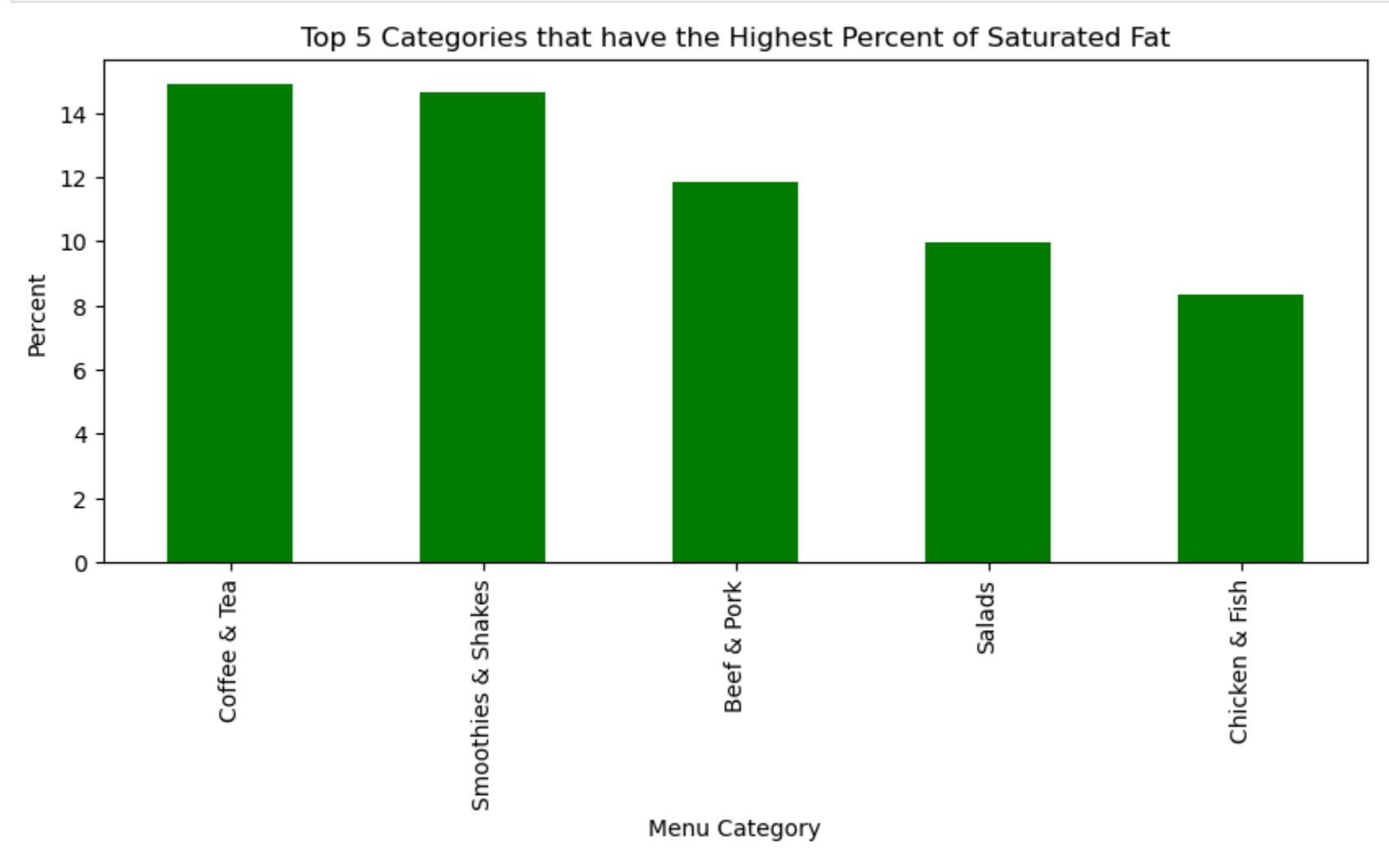
In [11]:
```python
data.groupby('Category')['Trans Fat'].mean()
```

Out[11]:
```
Category
Beef & Pork           1.100000
Beverages             0.000000
Breakfast             0.107143
Chicken & Fish        0.129630
Coffee & Tea          0.142105
Desserts              0.000000
Salads                0.000000
Smoothies & Shakes    0.535714
Snacks & Sides        0.000000
Name: Trans Fat, dtype: float64
```

In [12]:
```python
data['saturated_cholesterol'] = data['Saturated Fat']/data['Cholesterol']*100
```

In [13]:
```python
saturated_cholesterol = data.groupby('Category')['saturated_cholesterol'].mean().dropna().nlargest(5)
```

In [14]:
```python
#Bar Graph for Percent of Saturated Fat
plt.figure(figsize=(10, 4), dpi=100)
saturated_cholesterol.sort_values(ascending=False).plot.bar(color = 'green')
plt.title("Top 5 Categories that have the Highest Percent of Saturated Fat")
plt.ylabel("Percent")
plt.xlabel("Menu Category")
plt.xticks(rotation=90)
plt.show();
```



In [15]:
```python
coffee_tea = data[data.Category == 'Coffee & Tea']
coffee_tea.groupby('Item')['saturated_cholesterol'].mean().sort_values(ascending=False)
```

Out[15]:
```
Item
Frappé Chocolate Chip (Small)     21.538462
Frappé Chocolate Chip (Medium)    21.250000
Frappé Chocolate Chip (Large)     21.052632
Hazelnut Iced Coffee (Small)      20.000000
Hazelnut Latte (Medium)           20.000000
                                    ...
Iced Tea (Small)                        NaN
Sweet Tea (Child)                       NaN
Sweet Tea (Large)                       NaN
Sweet Tea (Medium)                      NaN
Sweet Tea (Small)                       NaN
Name: saturated_cholesterol, Length: 95, dtype: float64
```

In [16]:
```python
shakes = data[data.Category == 'Smoothies & Shakes']
shakes.groupby('Item')['saturated_cholesterol'].mean().sort_values()
```

Out[16]:
```
Item
Blueberry Pomegranate Smoothie (Medium)            0.000000
Blueberry Pomegranate Smoothie (Small)             0.000000
Mango Pineapple Smoothie (Medium)                  0.000000
Mango Pineapple Smoothie (Small)                   0.000000
Strawberry Banana Smoothie (Small)                 0.000000
Strawberry Banana Smoothie (Medium)                0.000000
Blueberry Pomegranate Smoothie (Large)            10.000000
Mango Pineapple Smoothie (Large)                  10.000000
Strawberry Banana Smoothie (Large)                10.000000
Chocolate Shake (Medium)                          16.000000
Vanilla Shake (Medium)                            16.000000
Shamrock Shake (Medium)                           16.000000
Vanilla Shake (Large)                             16.666667
Strawberry Shake (Small)                          16.666667
Strawberry Shake (Large)                          16.666667
Shamrock Shake (Large)                            16.666667
Vanilla Shake (Small)                             16.666667
Chocolate Shake (Small)                           16.666667
Strawberry Shake (Medium)                         17.333333
Chocolate Shake (Large)                           17.647059
McFlurry with Oreo Cookies (Snack)                20.000000
McFlurry with Oreo Cookies (Small)                20.000000
McFlurry with Oreo Cookies (Medium)               21.818182
McFlurry with Reese's Peanut Butter Cups (Medium) 25.000000
McFlurry with M&M’s Candies (Medium)              26.666667
McFlurry with Reese's Peanut Butter Cups (Snack)  26.666667
McFlurry with M&M’s Candies (Small)               28.000000
McFlurry with M&M’s Candies (Snack)               28.571429
Name: saturated_cholesterol, dtype: float64
```

In [17]:
```python
beef_pork = data[data.Category == 'Beef & Pork']
beef_pork.groupby('Item')['saturated_cholesterol'].mean().sort_values()
```

Out[17]:
```
Item
Hamburger                                 10.000000
McDouble                                  10.666667
Bacon McDouble                            11.111111
Cheeseburger                              11.111111
Double Cheeseburger                       11.111111
Daily Double                              11.250000
Jalapeño Double                           11.250000
Big Mac                                   11.764706
Double Quarter Pounder with Cheese        11.875000
Quarter Pounder with Bacon & Cheese       12.380952
Quarter Pounder with Bacon Habanero Ranch 12.380952
Quarter Pounder with Cheese               12.631579
Quarter Pounder Deluxe                    12.941176
Bacon Clubhouse Burger                    13.043478
McRib                                     14.285714
Name: saturated_cholesterol, dtype: float64
```

In [18]:
```python
chicken_fish = data[data.Category == 'Chicken & Fish']
chicken_fish.groupby('Item')['saturated_cholesterol'].mean().sort_values()
```

Out[18]:
```
Item
Premium Grilled Chicken Classic Sandwich             3.076923
Premium McWrap Chicken Sweet Chili (Grilled Chicken)  4.615385
Premium Grilled Chicken Ranch BLT Sandwich           5.000000
Southern Style Crispy Chicken Sandwich               6.666667
Premium Grilled Chicken Club Sandwich                6.666667
Bacon Clubhouse Grilled Chicken Sandwich             7.272727
Premium McWrap Chicken & Bacon (Grilled Chicken)     7.368421
Chicken McNuggets (20 piece)                         7.407407
Premium McWrap Chicken & Ranch (Grilled Chicken)     7.500000
Premium McWrap Southwest Chicken (Grilled Chicken)   7.500000
Chicken McNuggets (6 piece)                          7.500000
Chicken McNuggets (40 piece)                         7.547170
Chicken McNuggets (10 piece)                         7.692308
Premium Crispy Chicken Classic Sandwich              7.777778
Chicken McNuggets (4 piece)                          8.000000
McChicken                                            8.571429
Premium Crispy Chicken Ranch BLT Sandwich            8.571429
Buffalo Ranch McChicken                              8.571429
Premium McWrap Chicken Sweet Chili (Crispy Chicken)  9.000000
Bacon Buffalo Ranch McChicken                       10.000000
Filet-O-Fish                                        10.000000
Premium Crispy Chicken Club Sandwich                10.588235
Bacon Cheddar McChicken                             10.769231
Bacon Clubhouse Crispy Chicken Sandwich             11.111111
Premium McWrap Chicken & Bacon (Crispy Chicken)     11.250000
Premium McWrap Chicken & Ranch (Crispy Chicken)     12.307692
Premium McWrap Southwest Chicken (Crispy Chicken)   13.333333
Name: saturated_cholesterol, dtype: float64
```

In [19]:
```python
data.groupby('Category')['Vitamin A (% Daily Value)'].mean()
```

Out[19]:
```
Category
Beef & Pork             6.933333
Beverages               0.740741
Breakfast               6.928571
Chicken & Fish         20.444444
Coffee & Tea           10.736842
Desserts                5.142857
Salads                146.666667
Smoothies & Shakes     18.750000
Snacks & Sides          4.846154
Name: Vitamin A (% Daily Value), dtype: float64
```

In [20]:
```python
data.groupby('Category')['Vitamin C (% Daily Value)'].mean()
```

Out[20]:
```
Category
Beef & Pork            7.333333
Beverages             23.481481
Breakfast              8.904762
Chicken & Fish        12.629630
Coffee & Tea           0.000000
Desserts               4.142857
Salads                28.333333
Smoothies & Shakes     6.964286
Snacks & Sides        28.153846
Name: Vitamin C (% Daily Value), dtype: float64
```

In [21]:
```python
beverage = data[data.Category == 'Beverages']
beverage.groupby('Item')['Vitamin C (% Daily Value)'].mean().sort_values()
```

Out[21]:
```
Item
Diet Dr Pepper (Small)               0.0
Sprite (Large)                       0.0
Sprite (Child)                       0.0
Fat Free Chocolate Milk Jug          0.0
Dr Pepper (Small)                    0.0
Dr Pepper (Medium)                   0.0
Dr Pepper (Large)                    0.0
Dr Pepper (Child)                    0.0
Sprite (Medium)                      0.0
Diet Dr Pepper (Medium)              0.0
Diet Dr Pepper (Large)               0.0
Sprite (Small)                       0.0
Diet Coke (Small)                    0.0
Diet Coke (Medium)                   0.0
Diet Coke (Large)                    0.0
Diet Coke (Child)                    0.0
Dasani Water Bottle                  0.0
Coca-Cola Classic (Small)            0.0
Coca-Cola Classic (Medium)           0.0
Coca-Cola Classic (Large)            0.0
Coca-Cola Classic (Child)            0.0
Diet Dr Pepper (Child)               0.0
1% Low Fat Milk Jug                  4.0
Minute Maid 100% Apple Juice Box   100.0
Minute Maid Orange Juice (Small)   130.0
Minute Maid Orange Juice (Medium)  160.0
Minute Maid Orange Juice (Large)   240.0
Name: Vitamin C (% Daily Value), dtype: float64
```

In [22]:
```python
iron = data.groupby('Category')['Iron (% Daily Value)'].mean()
calcium = data.groupby('Category')['Calcium (% Daily Value)'].mean()
```

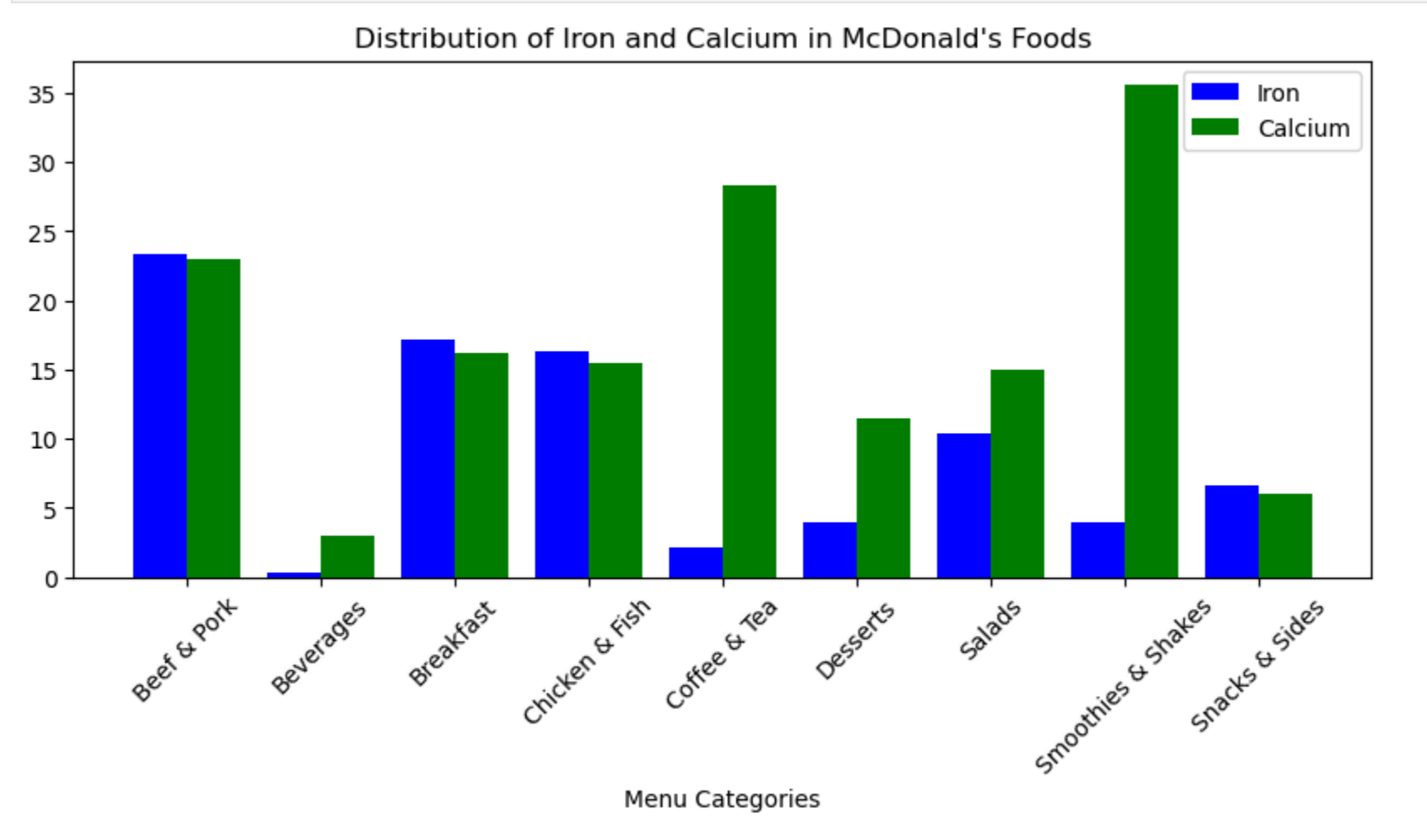
In [23]:
```python
plt.figure(figsize=(10, 4), dpi=100)

category = iron.index
Iron = iron.values
Calcium = calcium.values

X_axis = np.arange(len(category))

plt.bar(X_axis - 0.2, Iron, 0.4, label = 'Iron', color = 'blue')
plt.bar(X_axis + 0.2, Calcium, 0.4, label = 'Calcium', color = 'green')

plt.xticks(X_axis, category)
plt.xlabel("Menu Categories")
plt.title("Distribution of Iron and Calcium in McDonald's Foods")
plt.xticks(rotation=45)
plt.legend()
plt.show()
```



In [26]:
```python
'''This comprehensive analysis delves into the McDonald's Menu, providing detailed nutritional information for each food category. By reviewing this analysis, individuals can make i
```

Out[26]:
```
"This comprehensive analysis delves into the McDonald's Menu, providing detailed nutritional information for each food category. By reviewing this analysis, individuals can make in
formed choices based on their preferences and dietary requirements.Many items on McDonald's menu, particularly salads, are notably high in vitamins like Vitamin A and C, while most
items are low in carbohydrates and contain little to no trans-fat, which is a positive aspect. However, it's crucial to be mindful of the high saturated fat content in many menu it
ems; those seeking to limit their saturated fat intake should opt for items with less than 10% saturated fat. Additionally, the analysis includes insights into essential minerals s
uch as calcium and iron, revealing that menu items containing milk or animal protein are particularly rich in calcium."
```