AICP Internship Task Week 4

McDonald's Menu Nutritional Facts Analysis

This research effort examines McDonald's, a well-known restaurant, in terms of its nutritional content. An American businessman named Roy Kroc, who served as the former CEO of McDonald's, had the vision to open a fast-food joint that consistently served meals of the highest caliber. Every menu item on the US McDonald's menu, including breakfast, beef burgers, chicken and fish sandwiches, fries, salads, drinks, coffee and tea, milkshakes, and desserts, has its nutrition information analyzed in this dataset. We will focus on the correlation between Calories and other independent variables. Find dateset menu.csv in same folder. We will only deal with following attributes

['Calories', 'Total Fat', 'Carbohydrates', 'Dietary Fiber', 'Sugars', 'Protein', 'Vitamin A (% Daily Value)', 'Vitamin C (% Daily Value)', 'Calcium (% Daily Value)', 'Iron (% Daily Value)']

In [3]:
```
!pip install seaborn
```

```
WARNING: pip is being invoked by an old script wrapper. This will
fail in a future version of pip.
Please see https://github.com/pypa/pip/issues/5599 (https://githu
b.com/pypa/pip/issues/5599) for advice on fixing the underlying is
sue.
To avoid this problem you can invoke Python with '-m pip' instead
of running pip directly.
Collecting seaborn
  Downloading seaborn-0.12.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in /home/malik
-m-shahmeer-rashid/snap/jupyter/common/lib/python3.7/site-packages
(from seaborn) (1.21.6)
Requirement already satisfied: pandas>=0.25 in /home/malik-m-shahm
eer-rashid/snap/jupyter/common/lib/python3.7/site-packages (from s
eaborn) (1.3.5)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in /home/ma
lik-m-shahmeer-rashid/snap/jupyter/common/lib/python3.7/site-packa
ges (from seaborn) (3.5.3)
Requirement already satisfied: typing_extensions in /home/malik-m-
shahmeer-rashid/snap/jupyter/common/lib/python3.7/site-packages (f
rom seaborn) (4.7.1)
Requirement already satisfied: cycler>=0.10 in /home/malik-m-shahm
eer-rashid/snap/jupyter/common/lib/python3.7/site-packages (from m
atplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /home/malik-m-
shahmeer-rashid/snap/jupyter/common/lib/python3.7/site-packages (f
rom matplotlib!=3.6.1,>=3.1->seaborn) (4.38.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /home/malik-m-
shahmeer-rashid/snap/jupyter/common/lib/python3.7/site-packages (f
rom matplotlib!=3.6.1,>=3.1->seaborn) (1.4.5)
Requirement already satisfied: packaging>=20.0 in /home/malik-m-sh
ahmeer-rashid/snap/jupyter/common/lib/python3.7/site-packages (fro
m matplotlib!=3.6.1,>=3.1->seaborn) (23.2)
Requirement already satisfied: pillow>=6.2.0 in /home/malik-m-shah
meer-rashid/snap/jupyter/common/lib/python3.7/site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (9.5.0)
Requirement already satisfied: pyparsing>=2.2.1 in /home/malik-m-s
hahmeer-rashid/snap/jupyter/common/lib/python3.7/site-packages (fr
om matplotlib!=3.6.1,>=3.1->seaborn) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /snap/jupyt
er/6/lib/python3.7/site-packages (from matplotlib!=3.6.1,>=3.1->se
aborn) (2.8.0)
Requirement already satisfied: pytz>=2017.3 in /home/malik-m-shahm
eer-rashid/snap/jupyter/common/lib/python3.7/site-packages (from p
andas>=0.25->seaborn) (2024.1)
Requirement already satisfied: six>=1.5 in /snap/jupyter/6/lib/pyt
hon3.7/site-packages (from python-dateutil>=2.7->matplotlib!=3.6.
1,>=3.1->seaborn) (1.12.0)
Downloading seaborn-0.12.2-py3-none-any.whl (293 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 293.3/293.3 kB 900.6 k
B/s eta 0:00:00[36m0:00:01[36m0:00:01:01
Installing collected packages: seaborn
Successfully installed seaborn-0.12.2
```

In [5]:
```
!pip install plotly
```

```
WARNING: pip is being invoked by an old script wrapper. This will
fail in a future version of pip.
Please see https://github.com/pypa/pip/issues/5599 (https://githu
b.com/pypa/pip/issues/5599) for advice on fixing the underlying is
sue.
To avoid this problem you can invoke Python with '-m pip' instead
of running pip directly.
Collecting plotly
  Downloading plotly-5.18.0-py3-none-any.whl.metadata (7.0 kB)
Collecting tenacity>=6.2.0 (from plotly)
  Downloading tenacity-8.2.3-py3-none-any.whl.metadata (1.0 kB)
Requirement already satisfied: packaging in /home/malik-m-shahmeer
-rashid/snap/jupyter/common/lib/python3.7/site-packages (from plot
ly) (23.2)
Downloading plotly-5.18.0-py3-none-any.whl (15.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15.6/15.6 MB 9.8 MB/s
eta 0:00:000m eta 0:00:01[36m0:00:01
Downloading tenacity-8.2.3-py3-none-any.whl (24 kB)
Installing collected packages: tenacity, plotly
Successfully installed plotly-5.18.0 tenacity-8.2.3
```

Q.1:Import libraries (Numpy, pandas, matplotlib, plotly and seaborn) and then read csv file.

In [6]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

menu_df = pd.read_csv('menu.csv')

menu_df.head()
```

Out[6]:

	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trar F
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13.0	20	5.0	25	0
1	Breakfast	Egg White Delight	4.8 oz (135 g)	250	70	8.0	12	3.0	15	0
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23.0	35	8.0	42	0
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	250	28.0	43	10.0	52	0
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	210	23.0	35	8.0	42	0

5 rows × 24 columns

Q.2: Check statistical facts by checking all columns. Then calculate the maximum value of the following attributes:

['Calories', 'Total Fat', 'Carbohydrates', 'Dietary Fiber', 'Sugars', 'Protein', 'Vitamin A (% Daily Value)', 'Vitamin C (% Daily Value)', 'Calcium (% Daily Value)', 'Iron (% Daily Value)']

In [7]:

```
stats = menu_df.describe()
print("Statistical Facts for all columns:")
print(stats)

max_values = menu_df[['Calories', 'Total Fat', 'Carbohydrates', 'Di
                      'Protein', 'Vitamin A (% Daily Value)', 'Vita
                      'Calcium (% Daily Value)', 'Iron (% Daily Val
print("\nMaximum values of specified attributes:")
print(max_values)
```

Statistical Facts for all columns:

	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)
count	260.000000	260.000000	260.000000	260.000000
mean	368.269231	127.096154	14.165385	21.815385
std	240.269886	127.875914	14.205998	21.885199
min	0.000000	0.000000	0.000000	0.000000
25%	210.000000	20.000000	2.375000	3.750000
50%	340.000000	100.000000	11.000000	17.000000
75%	500.000000	200.000000	22.250000	35.000000
max	1880.000000	1060.000000	118.000000	182.000000

	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	Cholesterol
count	260.000000	260.000000	260.000000	260.000000
mean	6.007692	29.965385	0.203846	54.942308
std	5.321873	26.639209	0.429133	87.269257
min	0.000000	0.000000	0.000000	0.000000
25%	1.000000	4.750000	0.000000	5.000000
50%	5.000000	24.000000	0.000000	35.000000
75%	10.000000	48.000000	0.000000	65.000000
max	20.000000	102.000000	2.500000	575.000000

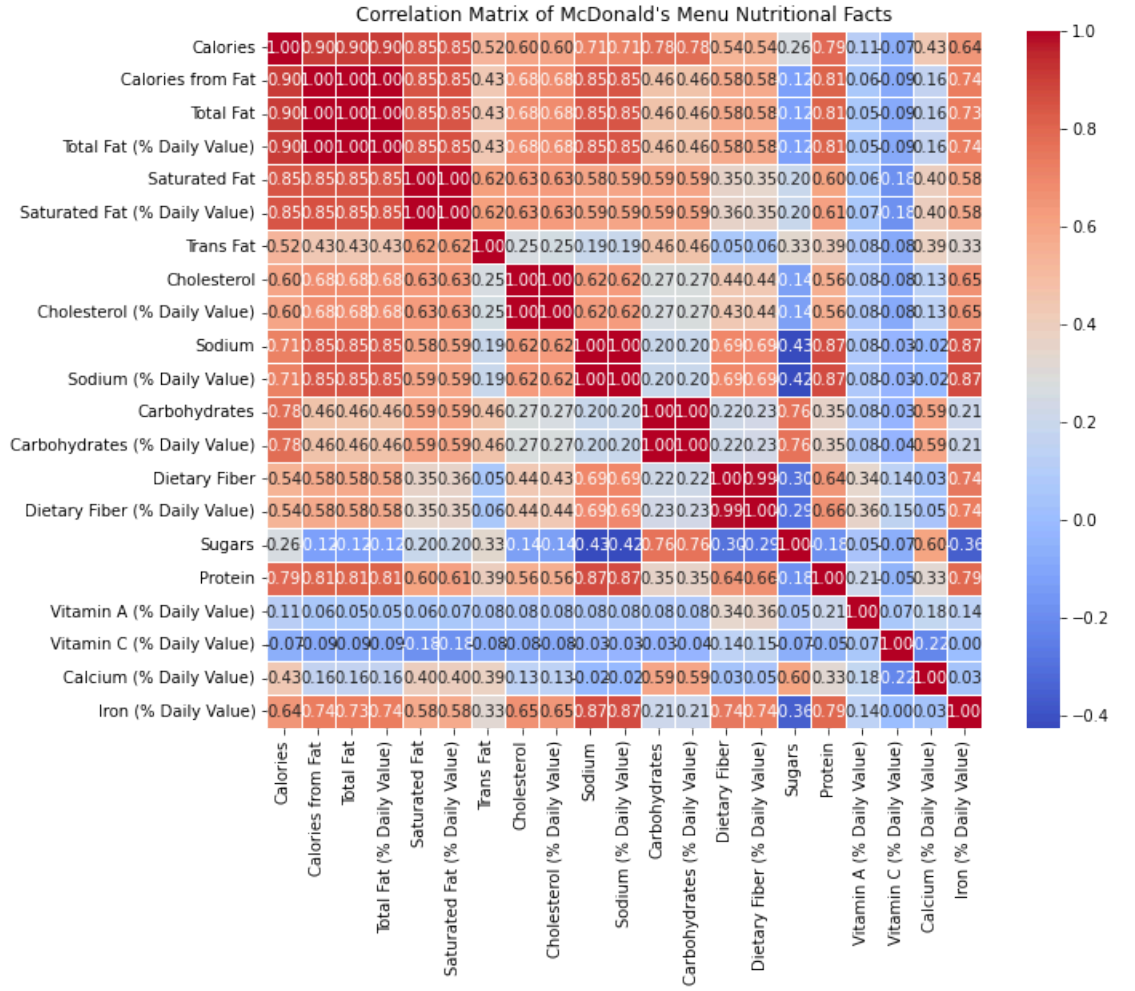
	Cholesterol (% Daily Value)	Sodium	...	Carbohydrates
count	260.000000	260.000000	...	260.000000
mean	18.392308	495.750000	...	47.346154
std	29.091653	577.026323	...	28.252232
min	0.000000	0.000000	...	0.000000
25%	2.000000	107.500000	...	30.000000
50%	11.000000	190.000000	...	44.000000
75%	21.250000	865.000000	...	60.000000
max	192.000000	3600.000000	...	141.000000

	Carbohydrates (% Daily Value)	Dietary Fiber
count	260.000000	260.000000
mean	15.780769	1.630769

```
std                          9.419544       1.567717
min                          0.000000       0.000000
25%                         10.000000       0.000000
50%                         15.000000       1.000000
75%                         20.000000       3.000000
max                         47.000000       7.000000

       Dietary Fiber (% Daily Value)      Sugars     Protein  \
count                     260.000000  260.000000  260.000000
mean                        6.530769   29.423077   13.338462
std                         6.307057   28.679797   11.426146
min                         0.000000    0.000000    0.000000
25%                         0.000000    5.750000    4.000000
50%                         5.000000   17.500000   12.000000
75%                        10.000000   48.000000   19.000000
max                        28.000000  128.000000   87.000000

       Vitamin A (% Daily Value)  Vitamin C (% Daily Value)  \
count                 260.000000                 260.000000
mean                   13.426923                   8.534615
std                    24.366381                  26.345542
min                     0.000000                   0.000000
25%                     2.000000                   0.000000
50%                     8.000000                   0.000000
75%                    15.000000                   4.000000
max                   170.000000                 240.000000

       Calcium (% Daily Value)  Iron (% Daily Value)
count               260.000000            260.000000
mean                 20.973077              7.734615
std                  17.019953              8.723263
min                   0.000000              0.000000
25%                   6.000000              0.000000
50%                  20.000000              4.000000
75%                  30.000000             15.000000
max                  70.000000             40.000000

[8 rows x 21 columns]

Maximum values of specified attributes:
Calories                     1880.0
Total Fat                     118.0
Carbohydrates                 141.0
Dietary Fiber                   7.0
Sugars                        128.0
Protein                        87.0
Vitamin A (% Daily Value)     170.0
Vitamin C (% Daily Value)     240.0
Calcium (% Daily Value)        70.0
Iron (% Daily Value)           40.0
dtype: float64
```

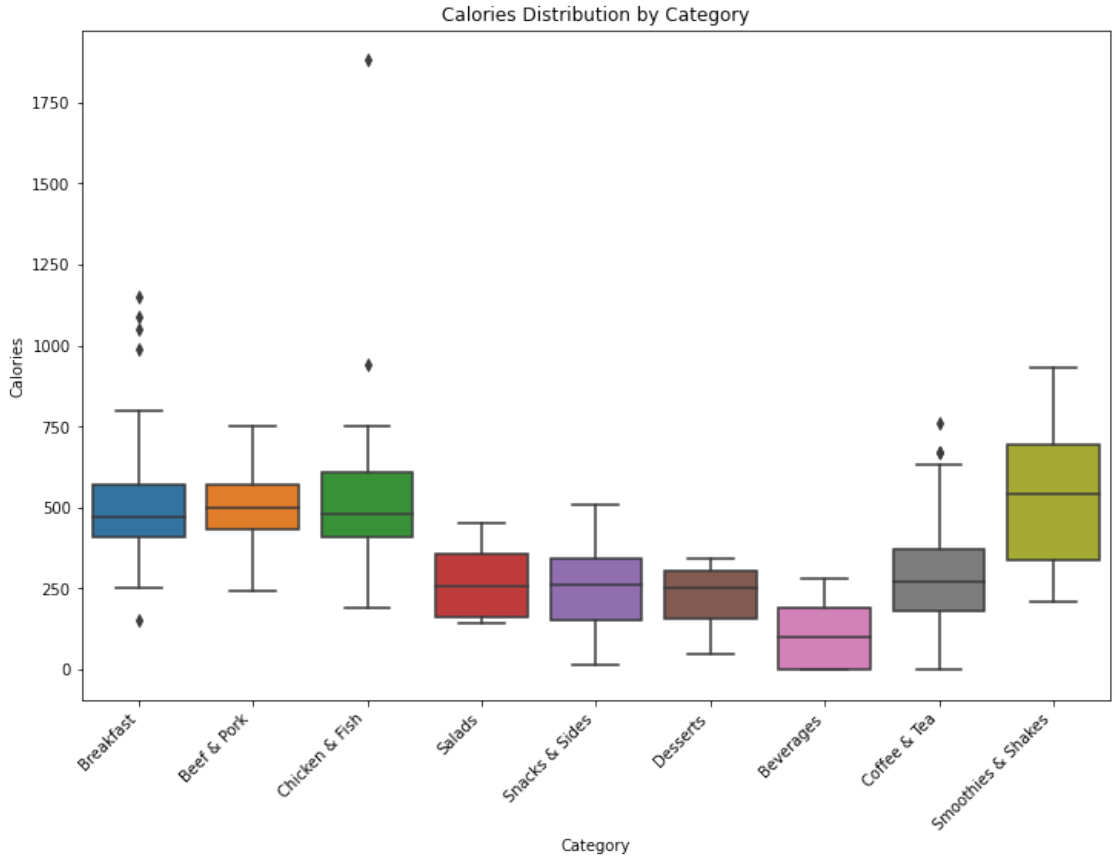
Q.3: Check to see if infact there is any correlation between Calories and other independent variables by plotting a correlation matrix next.

In [8]:
```
correlation_matrix = menu_df.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".
plt.title('Correlation Matrix of McDonald\'s Menu Nutritional Facts
plt.show()
```



Q.4: Draw boxplot for Calories vs Category to spot outliers and max calories category.

In [9]:
```
plt.figure(figsize=(12, 8))

sns.boxplot(x='Category', y='Calories', data=menu_df)
plt.title('Calories Distribution by Category')
plt.xticks(rotation=45, ha='right')

plt.show()
```



Q.5: Figure out which exact item contains a high quantity for ['Calories', 'Total Fat', 'Carbohydrates', 'Dietary Fiber', 'Sugars', 'Protein', 'Vitamin A (% Daily Value)', 'Vitamin C (% Daily Value)', 'Calcium (% Daily Value)', 'Iron (% Daily Value)'.

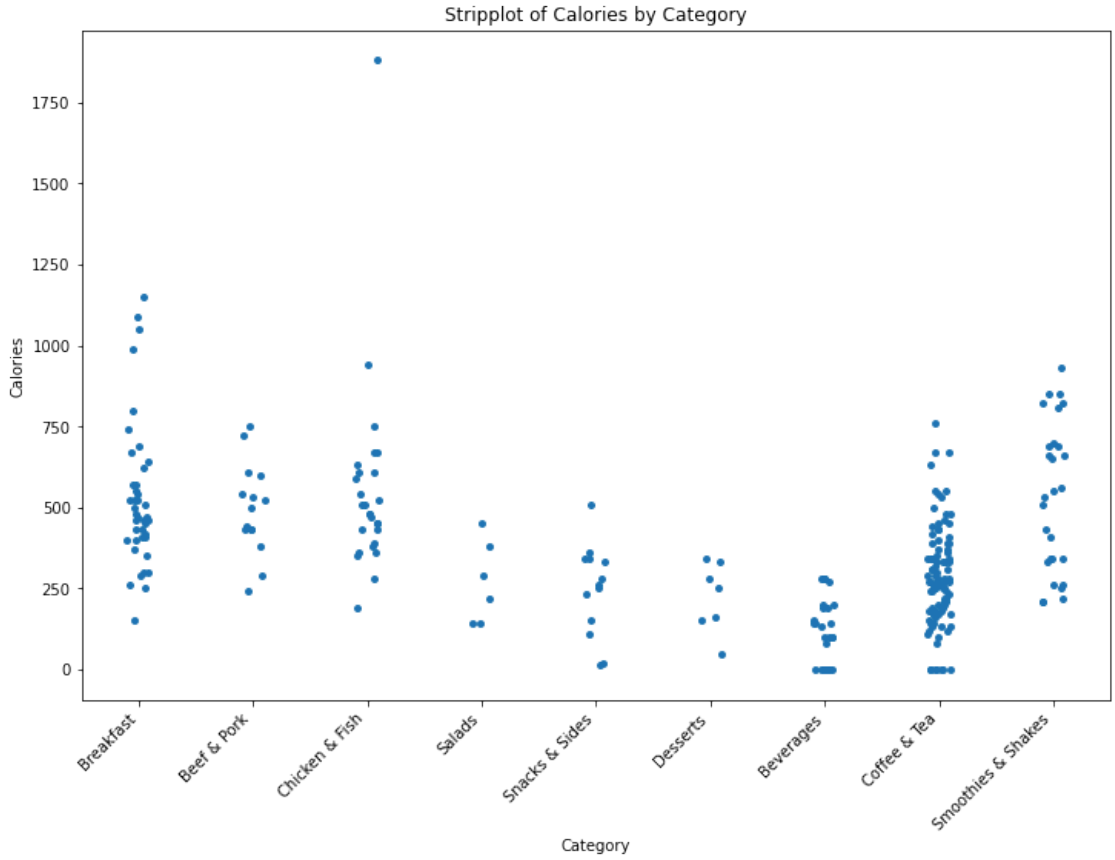
In [10]:
```
max_calories_item = menu_df.loc[menu_df['Calories'].idxmax()]
max_total_fat_item = menu_df.loc[menu_df['Total Fat'].idxmax()]
max_carbohydrates_item = menu_df.loc[menu_df['Carbohydrates'].idxma
max_dietary_fiber_item = menu_df.loc[menu_df['Dietary Fiber'].idxma
max_sugars_item = menu_df.loc[menu_df['Sugars'].idxmax()]
max_protein_item = menu_df.loc[menu_df['Protein'].idxmax()]
max_vitamin_a_item = menu_df.loc[menu_df['Vitamin A (% Daily Value)
max_vitamin_c_item = menu_df.loc[menu_df['Vitamin C (% Daily Value)
max_calcium_item = menu_df.loc[menu_df['Calcium (% Daily Value)'].i
max_iron_item = menu_df.loc[menu_df['Iron (% Daily Value)'].idxmax(

print("Item with the highest quantity for each attribute:")
print("Calories:", max_calories_item['Item'])
print("Total Fat:", max_total_fat_item['Item'])
print("Carbohydrates:", max_carbohydrates_item['Item'])
print("Dietary Fiber:", max_dietary_fiber_item['Item'])
print("Sugars:", max_sugars_item['Item'])
print("Protein:", max_protein_item['Item'])
print("Vitamin A (% Daily Value):", max_vitamin_a_item['Item'])
print("Vitamin C (% Daily Value):", max_vitamin_c_item['Item'])
print("Calcium (% Daily Value):", max_calcium_item['Item'])
print("Iron (% Daily Value):", max_iron_item['Item'])
```

```
Item with the highest quantity for each attribute:
Calories: Chicken McNuggets (40 piece)
Total Fat: Chicken McNuggets (40 piece)
Carbohydrates: Chocolate Shake (Large)
Dietary Fiber: Big Breakfast with Hotcakes (Large Biscuit)
Sugars: McFlurry with M&M’s Candies (Medium)
Protein: Chicken McNuggets (40 piece)
Vitamin A (% Daily Value): Premium Bacon Ranch Salad (without Chic
ken)
Vitamin C (% Daily Value): Minute Maid Orange Juice (Large)
Calcium (% Daily Value): Strawberry Shake (Large)
Iron (% Daily Value): Big Breakfast with Hotcakes (Regular Biscui
t)
```

Q.6: Draw Stripplot for each category against the following attributes

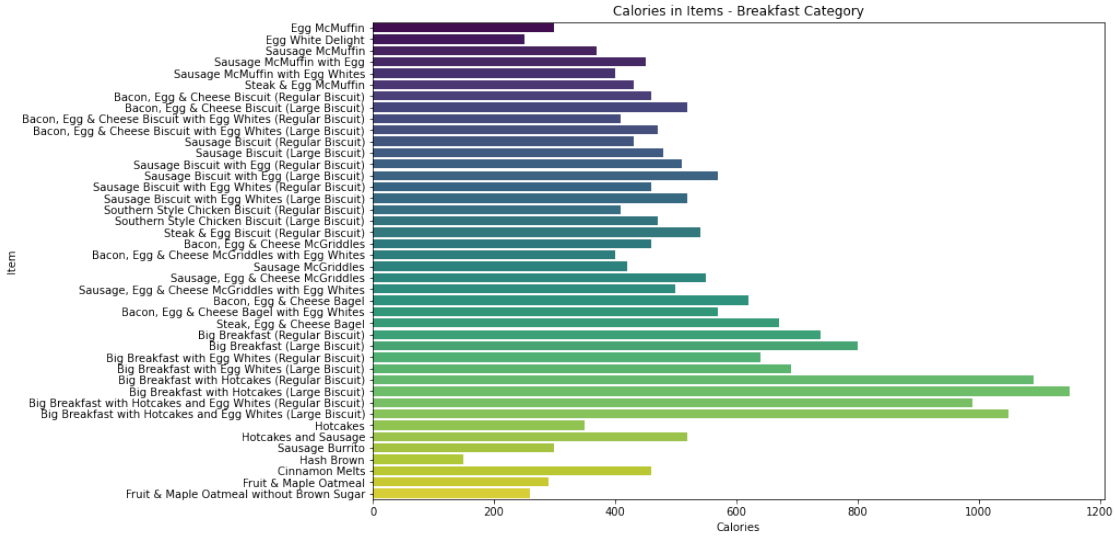
['Calories', 'Total Fat', 'Carbohydrates', 'Dietary Fiber', 'Sugars', 'Protein', 'Vitamin A (% Daily Value)', 'Vitamin C (% Daily Value)', 'Calcium (% Daily Value)', 'Iron(% Daily Value '. Here is one sample, you have to do it for all above mentioned attributes in list.

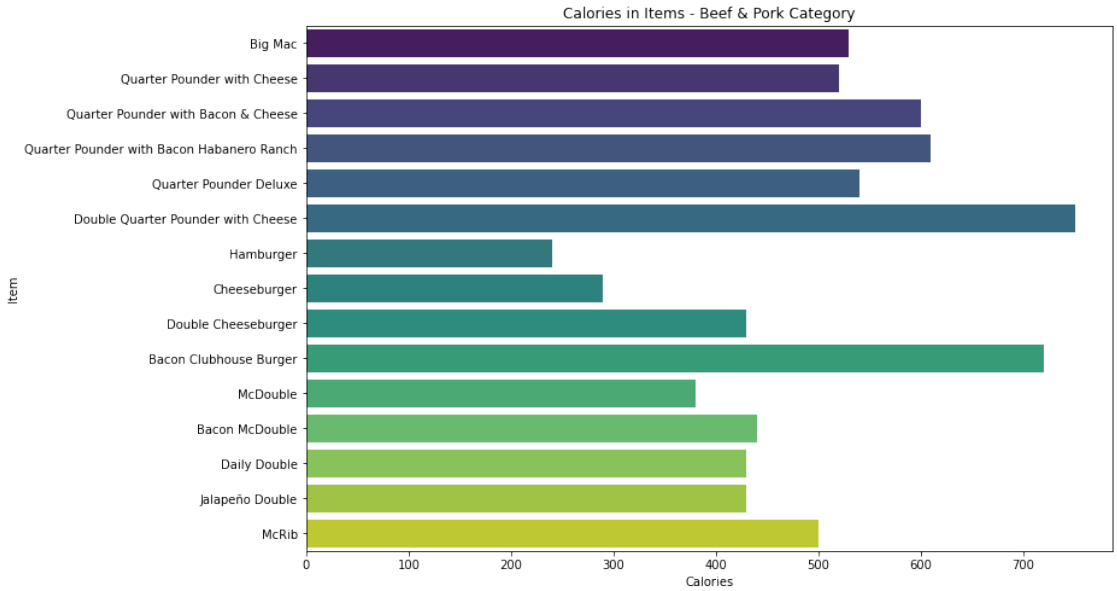
In [11]:
```
plt.figure(figsize=(12, 8))

sns.stripplot(x='Category', y='Calories', data=menu_df, jitter=True
plt.title('Stripplot of Calories by Category')
plt.xticks(rotation=45, ha='right')

plt.show()
```

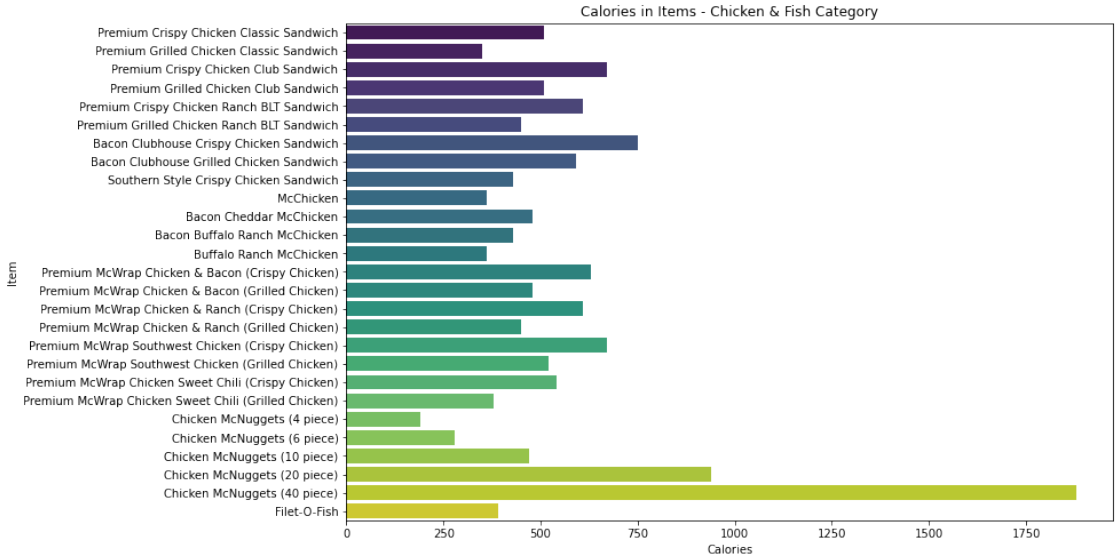


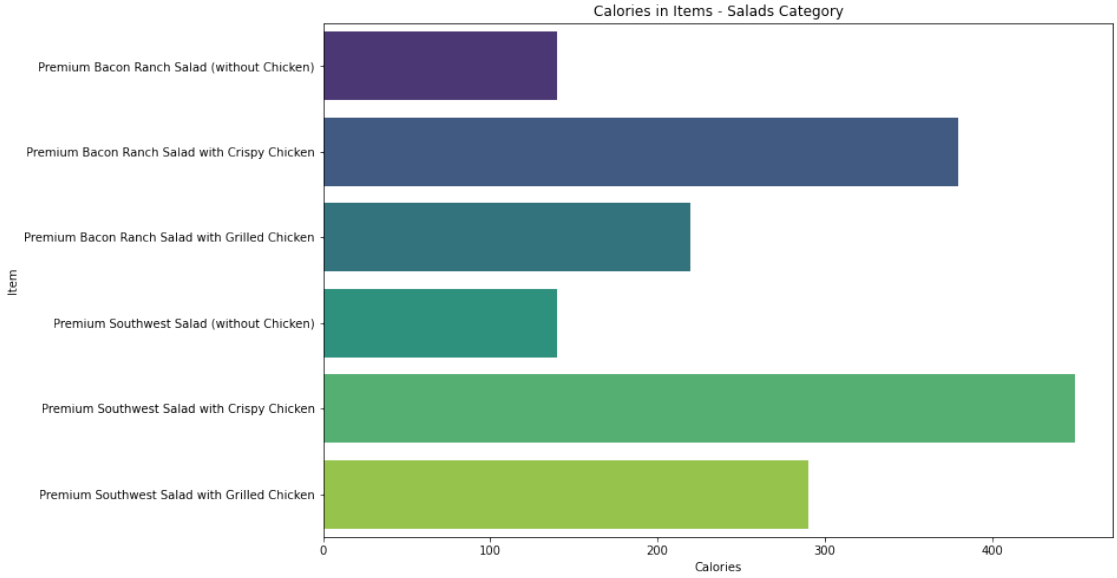
Q.7: Draw a horizontal bar graph for items in each category against the calories. Here is one sample for all items in Beef & Pork, you have to do it for items in each category. Also, write your observation.

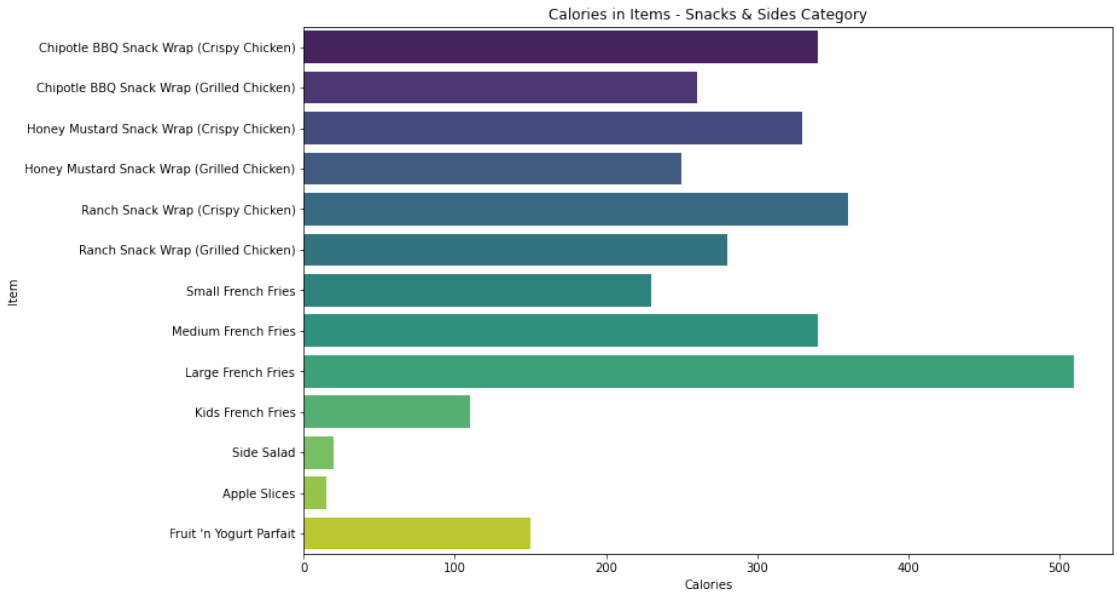
In [13]:
```
categories = menu_df['Category'].unique()

for category in categories:

    category_df = menu_df[menu_df['Category'] == category]

    plt.figure(figsize=(12, 8))

    sns.barplot(x='Calories', y='Item', data=category_df, palette='
    plt.title(f'Calories in Items - {category} Category')
    plt.xlabel('Calories')
    plt.ylabel('Item')

    plt.show()
```

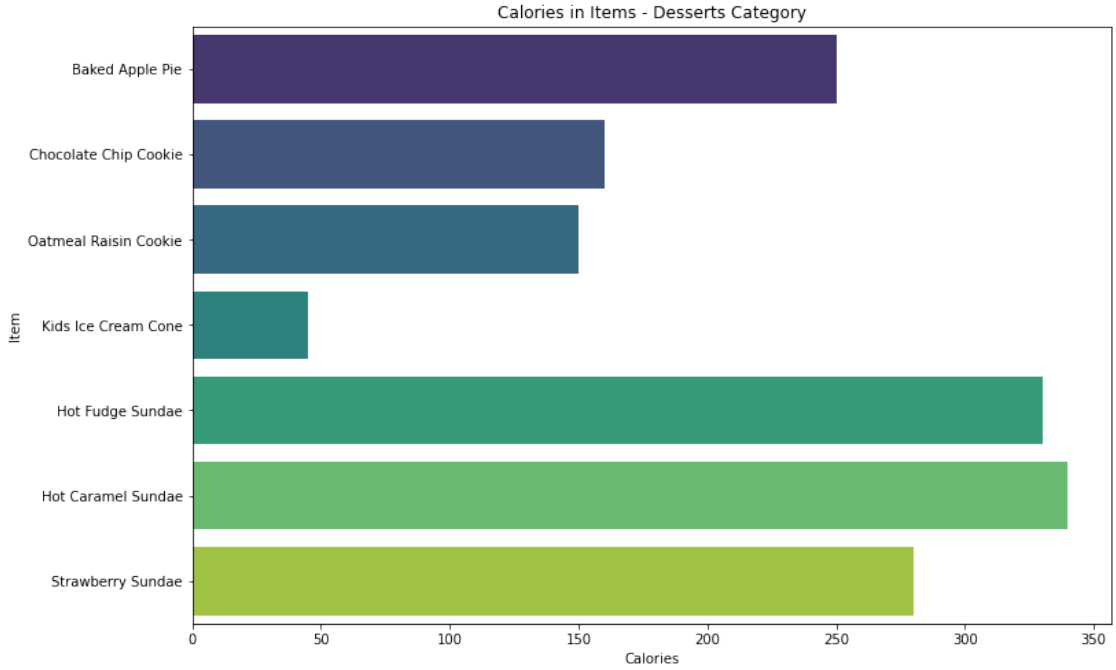


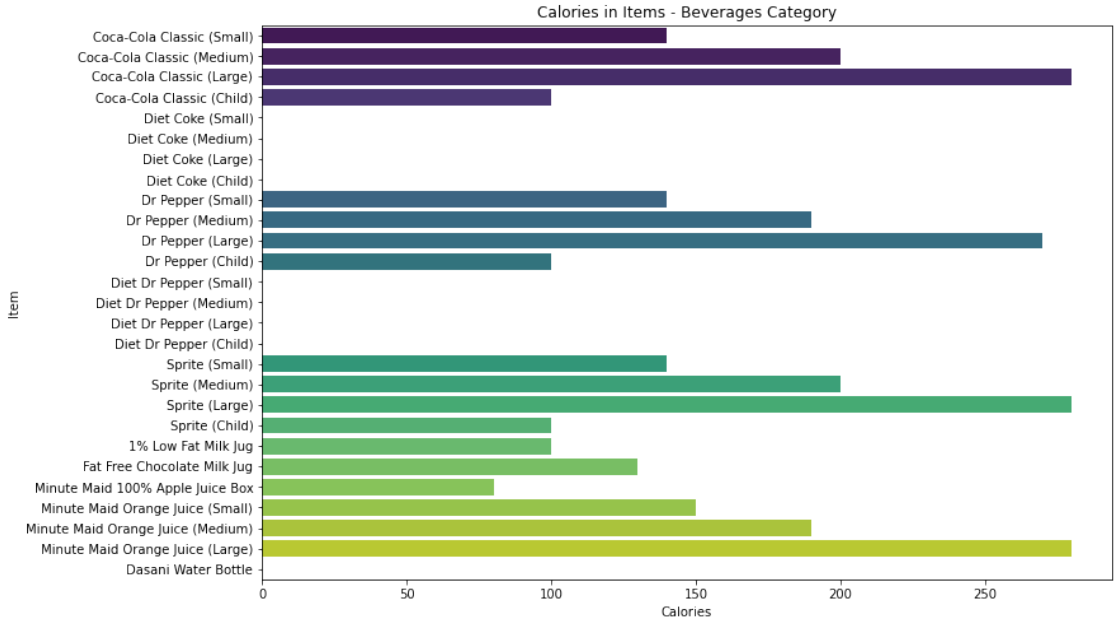


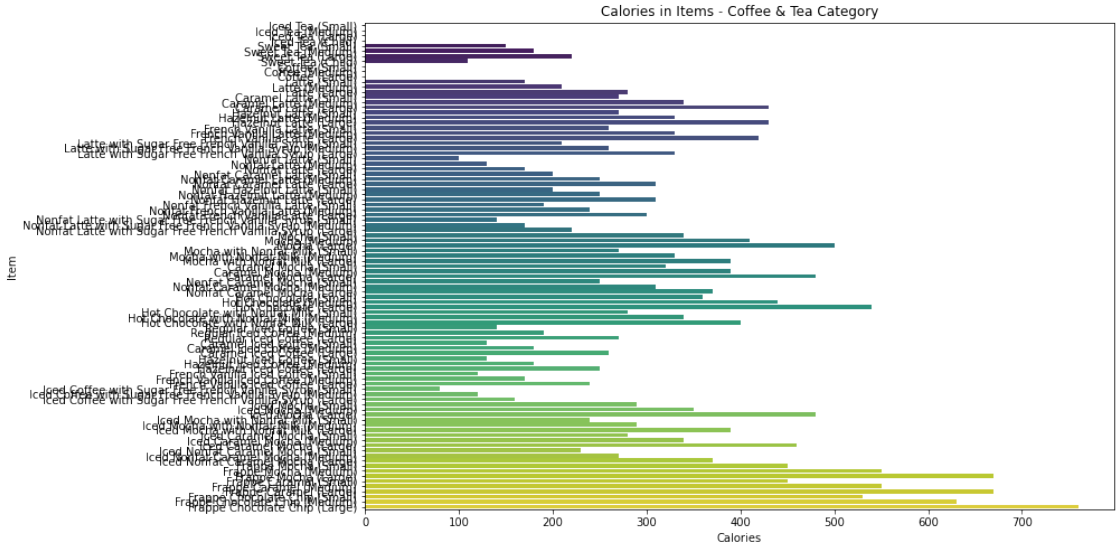

Calories in Items - Chicken & Fish Category
Premium Crispy Chicken Classic Sandwich
Premium Grilled Chicken Classic Sandwich
Premium Crispy Chicken Club Sandwich
Premium Grilled Chicken Club Sandwich
Premium Crispy Chicken Ranch BLT Sandwich
Premium Grilled Chicken Ranch BLT Sandwich
Bacon Clubhouse Crispy Chicken Sandwich
Bacon Clubhouse Grilled Chicken Sandwich
Southern Style Crispy Chicken Sandwich
McChicken
Bacon Cheddar McChicken
Bacon Buffalo Ranch McChicken
Buffalo Ranch McChicken
Premium McWrap Chicken & Bacon (Crispy Chicken)
Premium McWrap Chicken & Bacon (Grilled Chicken)
Premium McWrap Chicken & Ranch (Crispy Chicken)
Premium McWrap Chicken & Ranch (Grilled Chicken)
Premium McWrap Southwest Chicken (Crispy Chicken)
Premium McWrap Southwest Chicken (Grilled Chicken)
Premium McWrap Chicken Sweet Chili (Crispy Chicken)
Premium McWrap Chicken Sweet Chili (Grilled Chicken)
Chicken McNuggets (4 piece)
Chicken McNuggets (6 piece)
Chicken McNuggets (10 piece)
Chicken McNuggets (20 piece)
Chicken McNuggets (40 piece)
Filet-O-Fish
Item
0
250
500
750
1000
1250
1500
1750
Calories

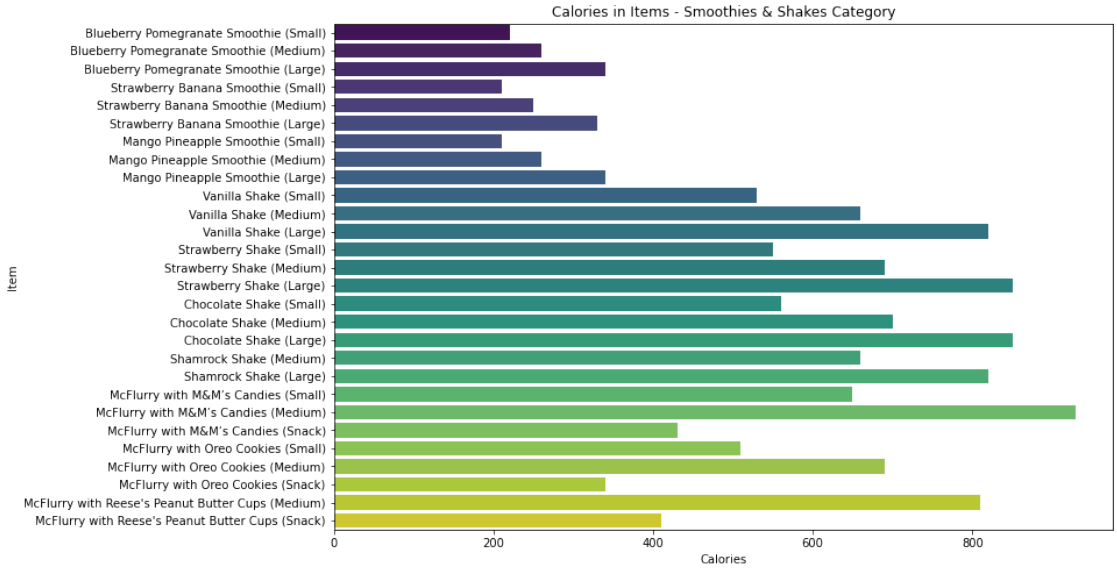

Calories in Items - Salads Category
Premium Bacon Ranch Salad (without Chicken)
Premium Bacon Ranch Salad with Crispy Chicken
Premium Bacon Ranch Salad with Grilled Chicken
Premium Southwest Salad (without Chicken)
Premium Southwest Salad with Crispy Chicken
Premium Southwest Salad with Grilled Chicken
Item
0
100
200
300
400
Calories


Calories in Items - Snacks & Sides Category
Chipotle BBQ Snack Wrap (Crispy Chicken)
Chipotle BBQ Snack Wrap (Grilled Chicken)
Honey Mustard Snack Wrap (Crispy Chicken)
Honey Mustard Snack Wrap (Grilled Chicken)
Ranch Snack Wrap (Crispy Chicken)
Ranch Snack Wrap (Grilled Chicken)
Small French Fries
Medium French Fries
Large French Fries
Kids French Fries
Side Salad
Apple Slices
Fruit 'n Yogurt Parfait
Item
0
100
200
300
400
500
Calories


Calories in Items - Desserts Category
Baked Apple Pie
Chocolate Chip Cookie
Oatmeal Raisin Cookie
Kids Ice Cream Cone
Hot Fudge Sundae
Hot Caramel Sundae
Strawberry Sundae
Item
0
50
100
150
200
250
300
350
Calories


Calories in Items - Beverages Category
Coca-Cola Classic (Small)
Coca-Cola Classic (Medium)
Coca-Cola Classic (Large)
Coca-Cola Classic (Child)
Diet Coke (Small)
Diet Coke (Medium)
Diet Coke (Large)
Diet Coke (Child)
Dr Pepper (Small)
Dr Pepper (Medium)
Dr Pepper (Large)
Dr Pepper (Child)
Diet Dr Pepper (Small)
Diet Dr Pepper (Medium)
Diet Dr Pepper (Large)
Diet Dr Pepper (Child)
Sprite (Small)
Sprite (Medium)
Sprite (Large)
Sprite (Child)
1% Low Fat Milk Jug
Fat Free Chocolate Milk Jug
Minute Maid 100% Apple Juice Box
Minute Maid Orange Juice (Small)
Minute Maid Orange Juice (Medium)
Minute Maid Orange Juice (Large)
Dasani Water Bottle
Item
0
50
100
150
200
250
Calories


Calories in Items - Coffee & Tea Category
Item
0
100
200
300
400
500
600
700
Calories



by: Malik M Shahmeer Rashid