MT-XYZ – Probability & Statistics Assignment

Statistical Analysis of McDonald’s Menu

Group Members:

- Zain Ammad – CS-19079
- Taimoor Jawaid – CS-19083
- Rohan – CS-19091
- Sohaib Ahmed Abbasi – CS-19096

Initial data exploration / understanding

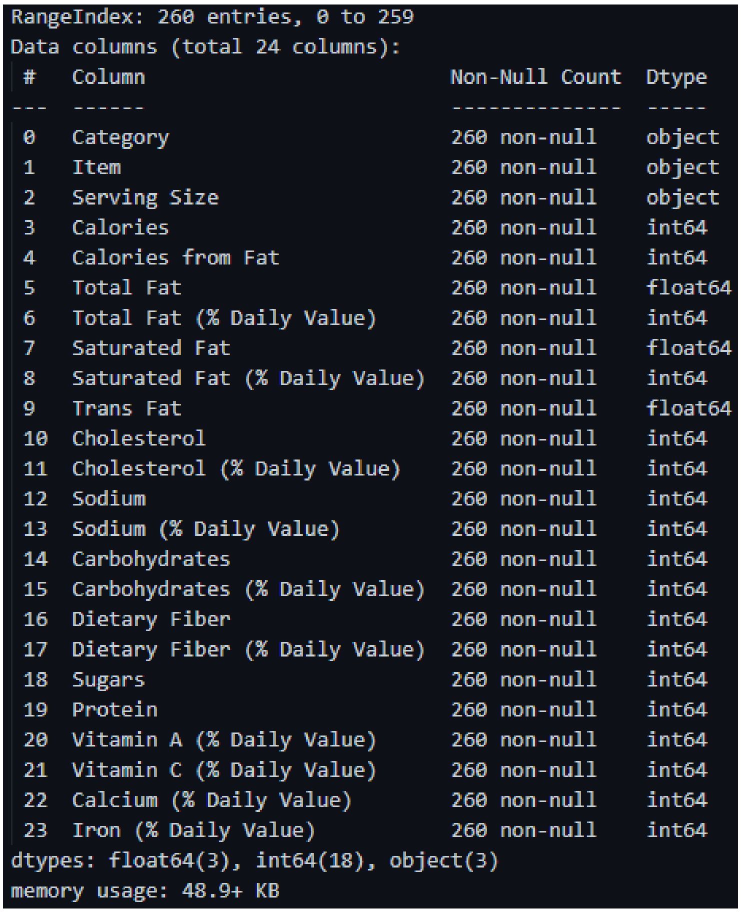
Getting required libraries for data analysis

```
import pandas as pd
import matplotlib.pyplot as plt
```

Reading the data set and show initial information related to it

```
dataset = pd.read_csv('McDonalds Menu.csv')  # read the data set
dataset.info()  # show the information of the data set
```

output of last line:



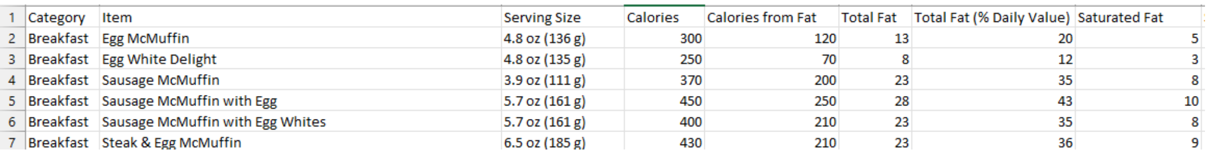
```
RangeIndex: 260 entries, 0 to 259
Data columns (total 24 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Category                        260 non-null    object
 1   Item                            260 non-null    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
```

This shows all the columns of the data set, how many values are non-null (not empty) in each column (Non-Null Count), and data type of each column (Dtype) (Dtype object means text/string, int64 means integer number and float64 real number.)

From this we get the following information:

- There are 260 rows and 24 columns in the dataset.
- No column has null value (i.e no cell is empty).
- For each menu item, the dataset has its category, name of the item, size of one serving, followed by a bunch of nutritional information like calories, total fat, protein, etc.

Here is a sample of the dataset (first few rows and some columns shown):



1	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat
2	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13	20	5
3	Breakfast	Egg White Delight	4.8 oz (135 g)	250	70	8	12	3
4	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23	35	8
5	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	250	28	43	10
6	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	210	23	35	8
7	Breakfast	Steak & Egg McMuffin	6.5 oz (185 g)	430	210	23	36	9

Now let’s answer some questions related to this dataset and find out some interesting insights.

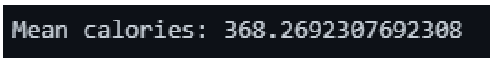
Q1: What is the average number of calories in each category of the menu (and how do those averages compare with overall average calories of the entire menu)?

Calorie of a food item is a measure of amount of energy one gains by having that food. Let’s find out the mean number of calories in each category of McDonalds and how do those individuals averages compare to overall mean calories of the entire menu.

First lets find out the mean calories overall.

```
# find mean calories overall
menu_mean = dataset['Calories'].mean()
print(f"Mean calories: {menu_mean}")
```

Output:

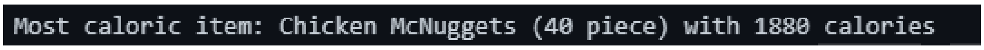


```
Mean calories: 368.2692307692308
```

So the average amount of calories per serving in a McDonalds meal is around 368.27 calories. That isn’t a lot. How about the most number of calories in an item? And what item is it? Let’s find that out.

```
most_calorie_item = dataset.nlargest(1, 'Calories')
print(f"Most caloric item: {most_calorie_item['Item'].values[0]} with 
{most_calorie_item['Calories'].values[0]} calories")
```

Output:



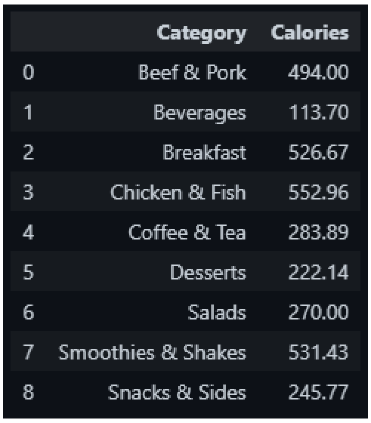
```
Most caloric item: Chicken McNuggets (40 piece) with 1880 calories
```

Wow there’s an item with 1880 calories per serving! That’s a lot for one person to eat in a meal! But it makes sense because that item is 40 pieces of Chicken nuggets, which is surely meant for multiple people to eat, not one.

Now let’s find out average Calories of each category

```
# make a new data frame with categories column and average calories for each category
categories = dataset.groupby('Category')['Calories'].mean().round(2)
categories.reset_index(inplace=False)
```

Output:



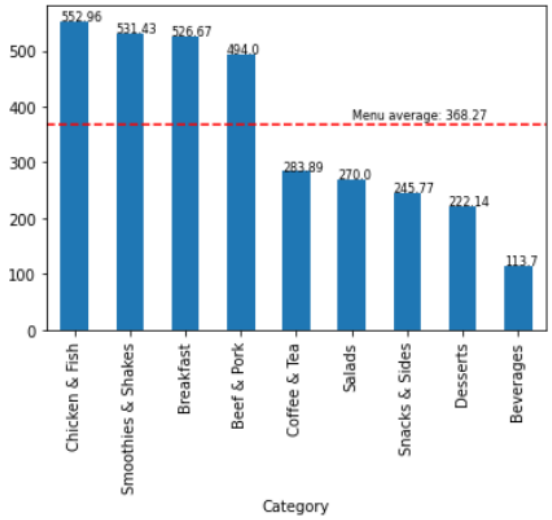
	Category	Calories
0	Beef & Pork	494.00
1	Beverages	113.70
2	Breakfast	526.67
3	Chicken & Fish	552.96
4	Coffee & Tea	283.89
5	Desserts	222.14
6	Salads	270.00
7	Smoothies & Shakes	531.43
8	Snacks & Sides	245.77

Let’s sort these values and plot them for better understanding

```
# sort categories by calories
categories = categories.sort_values(ascending=False)

# plot categories as bar chart
categories.plot(kind='bar')
for index, data in enumerate(categories):
    plt.text(x=index-0.25 , y =data+2 , s=f"{round(data, 2)}" , fontdict=dict(fontsize=8))
# plot menu mean line on top of categories and add label to it
plt.axhline(y=menu_mean, color='r', linestyle='--')
plt.text(x=5, y=menu_mean+10, s=f"Menu average: {round(menu_mean, 2)}", 
fontdict=dict(fontsize=8))
```

Output:



This shows that Chicken & Fish category on average has the most calories per item and beverages is the category with the least average per item calories. Also a surprising fact is discovered that Smoothies & Shakes categories is 2nd in the list of most average calories.

Q2: What are the least and most calorie dense (eatable) food items in each category.

Calorie density refers to the amount of calories per unit food. This is a very important fact to know specially if one is looking to loose fat as fat loss occurs due to a calorie deficit (more calories lost/used up compared to consumed). A low calorie dense food can be had in higher amount for equal or less calories than a high calorie dense food which can lead to the person eating such a food staying full and satisfied for longer and help them stay in a calorie deficit.

So let's find out the top 3 most calorie dense food items in each category which one should try and avoid and top 3 least calorie dense food items in each category that one should look to have if having a meal at McDonald's.

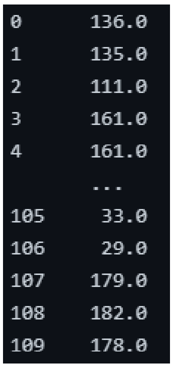
First we will separate out only eatable food items from the menu (i.e items whose categories aren't 'Beverages', 'Coffee & Tea', or 'Smoothies & Shakes'.

```
eatables = dataset[~dataset['Category'].isin(['Beverages', 'Coffee & Tea', 'Smoothies &
Shakes'])]
```

Next, we need to do some data cleaning. As it can be seen from the sample of the dataset shown before, the Serving Size column has values in ounces and grams both. Let's clean it up by only having values of grams in the serving size column.

```
eatables['Serving Size'] = eatables['Serving Size'].str.split('(',
expand=True)[1].str.split('g', expand=True)[0].astype(float)
eatables['Serving Size']
```

Output: (this is how the Serving Size column is now)

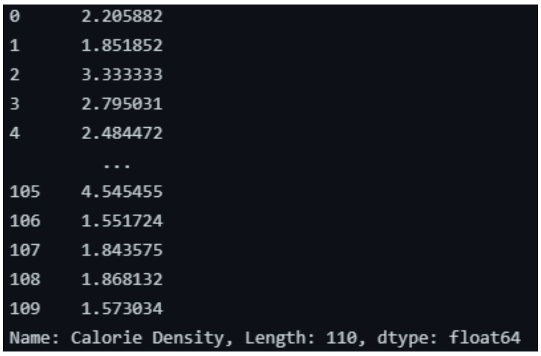


```
0       136.0
1       135.0
2       111.0
3       161.0
4       161.0
        ...
105      33.0
106      29.0
107     179.0
108     182.0
109     178.0
```

Now let’s find out calorie density of each item (we will add a new column for it). Calorie density is the amount of caloires per unit food as discussed above. So we can obtain calorie density for each food by dividing amount of calories per serving by the size of one serving, as done in the following code:

```
# add a new column for calorie density of each non-drink item
eatables['Calorie Density'] = eatables['Calories'] / eatables['Serving Size']
# show calorie density column
eatables['Calorie Density']
```

Output:

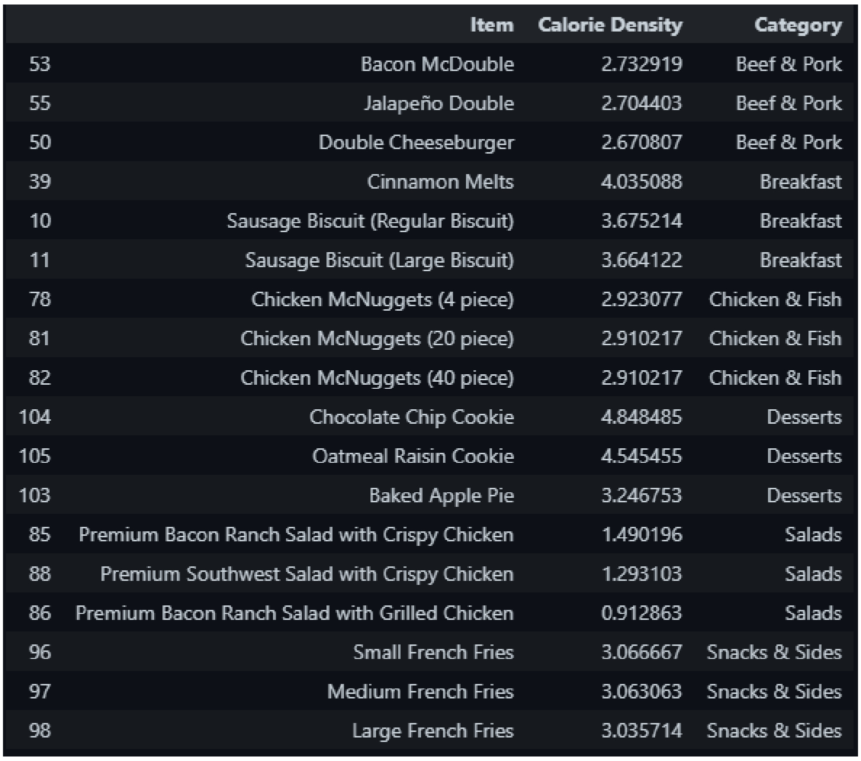


```
0      2.205882
1      1.851852
2      3.333333
3      2.795031
4      2.484472
         ...
105    4.545455
106    1.551724
107    1.843575
108    1.868132
109    1.573034
Name: Calorie Density, Length: 110, dtype: float64
```

Now let’s find out top 3 most calorie dense food in each category.

```
# show Item column for top 3 items by calorie density for each Category
most_calorie_dense = eatables.sort_values(by=["Category", 'Calorie Density'],
ascending=[True, False]).groupby('Category')[["Item", "Calorie Density",
"Category"]].head(3)
most_calorie_dense
```

Output:



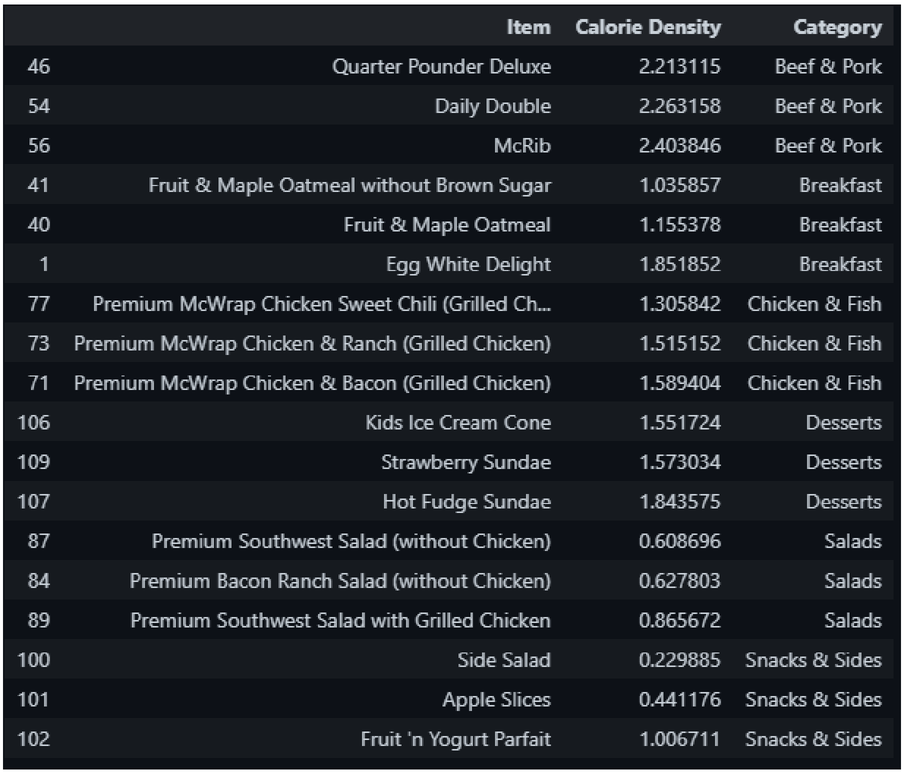
	Item	Calorie Density	Category
53	Bacon McDouble	2.732919	Beef & Pork
55	Jalapeño Double	2.704403	Beef & Pork
50	Double Cheeseburger	2.670807	Beef & Pork
39	Cinnamon Melts	4.035088	Breakfast
10	Sausage Biscuit (Regular Biscuit)	3.675214	Breakfast
11	Sausage Biscuit (Large Biscuit)	3.664122	Breakfast
78	Chicken McNuggets (4 piece)	2.923077	Chicken & Fish
81	Chicken McNuggets (20 piece)	2.910217	Chicken & Fish
82	Chicken McNuggets (40 piece)	2.910217	Chicken & Fish
104	Chocolate Chip Cookie	4.848485	Desserts
105	Oatmeal Raisin Cookie	4.545455	Desserts
103	Baked Apple Pie	3.246753	Desserts
85	Premium Bacon Ranch Salad with Crispy Chicken	1.490196	Salads
88	Premium Southwest Salad with Crispy Chicken	1.293103	Salads
86	Premium Bacon Ranch Salad with Grilled Chicken	0.912863	Salads
96	Small French Fries	3.066667	Snacks & Sides
97	Medium French Fries	3.063063	Snacks & Sides
98	Large French Fries	3.035714	Snacks & Sides

These show 3 items in each category that one should try and avoid if they want to consume low calorie dense foods

Next let’s find out top 3 least calorie dense food in each category

```
# show Item column for bottom 3 items by calorie density for each Category
least_calorie_dense = eatables.sort_values(by=["Category", 'Calorie Density'],
ascending=[True, True]).groupby('Category')[["Item", "Calorie Density",
"Category"]].head(3)
least_calorie_dense
```

Output:



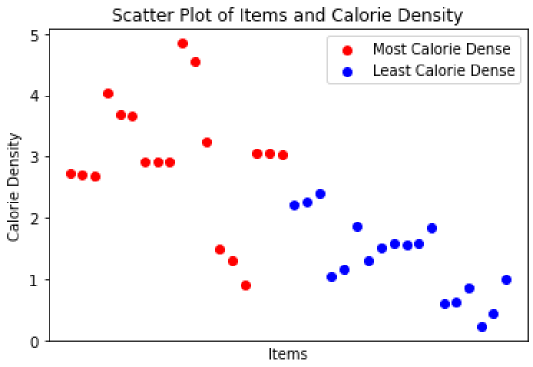
	Item	Calorie Density	Category
46	Quarter Pounder Deluxe	2.213115	Beef & Pork
54	Daily Double	2.263158	Beef & Pork
56	McRib	2.403846	Beef & Pork
41	Fruit & Maple Oatmeal without Brown Sugar	1.035857	Breakfast
40	Fruit & Maple Oatmeal	1.155378	Breakfast
1	Egg White Delight	1.851852	Breakfast
77	Premium McWrap Chicken Sweet Chili (Grilled Ch...	1.305842	Chicken & Fish
73	Premium McWrap Chicken & Ranch (Grilled Chicken)	1.515152	Chicken & Fish
71	Premium McWrap Chicken & Bacon (Grilled Chicken)	1.589404	Chicken & Fish
106	Kids Ice Cream Cone	1.551724	Desserts
109	Strawberry Sundae	1.573034	Desserts
107	Hot Fudge Sundae	1.843575	Desserts
87	Premium Southwest Salad (without Chicken)	0.608696	Salads
84	Premium Bacon Ranch Salad (without Chicken)	0.627803	Salads
89	Premium Southwest Salad with Grilled Chicken	0.865672	Salads
100	Side Salad	0.229885	Snacks & Sides
101	Apple Slices	0.441176	Snacks & Sides
102	Fruit 'n Yogurt Parfait	1.006711	Snacks & Sides

These are the foods that one should be looking to have at McDonald's if they want to consume low calorie dense foods.

Here is a comparison of the calorie density of the items shown in the two tables above shown using a scatter plot.

```
plt.scatter(most_calorie_dense['Item'], most_calorie_dense['Calorie Density'],
color='red')
plt.scatter(least_calorie_dense['Item'], least_calorie_dense['Calorie Density'],
color='blue')
plt.xticks([])
plt.xlabel('Items')
plt.ylabel('Calorie Density')
plt.title('Scatter Plot of Items and Calorie Density')
plt.legend(['Most Calorie Dense', 'Least Calorie Dense'])
plt.show()
```

Output:



Q3: Are grilled chicken options better than crispy chicken ones?

One may think that grilled chicken options would be a healthier choice than crispy chicken ones, but is that actually true? Let’s find out by comparing various nutritional facts for grilled and crispy chicken menu options.

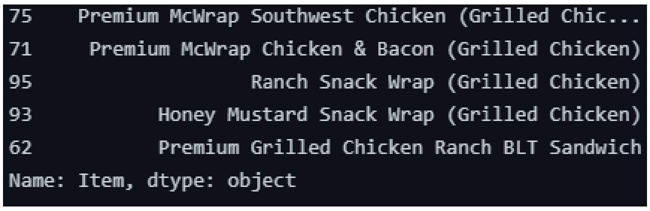
First let’s separate out grilled and crispy chicken options

```
# seperate out grilled chicken and crispy chicken items
grilled_chicken_items = dataset[dataset['Item'].str.contains('Grilled')]
crispy_chicken_items = dataset[dataset['Item'].str.contains('Crispy')]
```

Let’s see a sample of both of these sub-datasets

```
grilled_chicken_items.sample(5)["Item"]
```

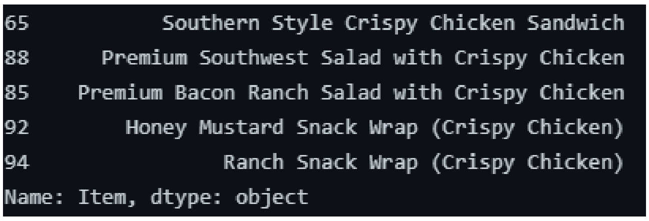
Output:



```
75    Premium McWrap Southwest Chicken (Grilled Chic...
71     Premium McWrap Chicken & Bacon (Grilled Chicken)
95                   Ranch Snack Wrap (Grilled Chicken)
93            Honey Mustard Snack Wrap (Grilled Chicken)
62            Premium Grilled Chicken Ranch BLT Sandwich
Name: Item, dtype: object
```

```
crispy_chicken_items.sample(5)["Item"]
```

Output:



```
65           Southern Style Crispy Chicken Sandwich
88       Premium Southwest Salad with Crispy Chicken
85     Premium Bacon Ranch Salad with Crispy Chicken
92          Honey Mustard Snack Wrap (Crispy Chicken)
94                 Ranch Snack Wrap (Crispy Chicken)
Name: Item, dtype: object
```

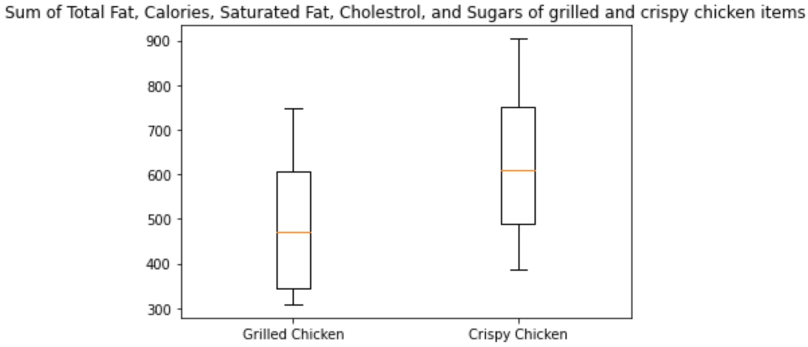
Let us focus our comparison of the two types of chicken items on the following 2 categories of nutritional values:

1) Good nutritional values (that we want more of): Dietary Fiber and Protein
2) Bad nutritional values (that we want less of): Total Fat, Calories, Saturated Fat, Cholesterol, and Sugars

First lets see which chicken type has less of the bad nutritional values.

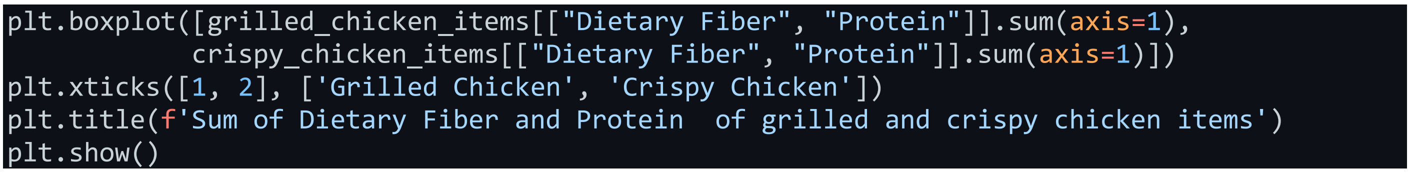
```
plt.boxplot([grilled_chicken_items[["Total Fat", "Calories", "Saturated Fat",
"Cholesterol", "Sugars"]].sum(axis=1),
            crispy_chicken_items[["Total Fat", "Calories", "Saturated Fat", "Cholesterol",
"Sugars"]].sum(axis=1)])
plt.xticks([1, 2], ['Grilled Chicken', 'Crispy Chicken'])
plt.title(f'Sum of Total Fat, Calories, Saturated Fat, Cholestrol, and Sugars of grilled
and crispy chicken items')
plt.show()
```

Output:



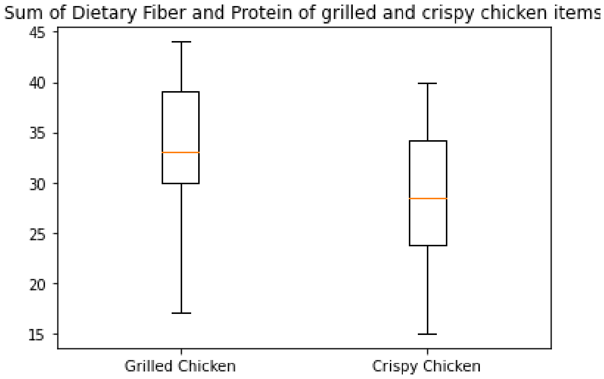
This shows that most grilled chicken items have less of the bad nutrition values so most of them are better, although some of them have more of the bad nutritional values than some crispy chicken items (as shown by the maximum value of grilled chicken being higher than minimum of crispy chicken). On average though grilled chicken items are better.

Next let’s see how the two types compare with the amount of good nutritional values



```
plt.boxplot([grilled_chicken_items[["Dietary Fiber", "Protein"]].sum(axis=1),
            crispy_chicken_items[["Dietary Fiber", "Protein"]].sum(axis=1)])
plt.xticks([1, 2], ['Grilled Chicken', 'Crispy Chicken'])
plt.title(f'Sum of Dietary Fiber and Protein  of grilled and crispy chicken items')
plt.show()
```

Output:



Grilled chicken wins here again as most items in that category have more of the good nutritional values (although not all) and average is also better.

This analysis shows that indeed, for most cases grilled chicken items are better than crispy chicken ones but one must still choose carefully as some grilled chicken items are worse than some crispy chicken items.