In [1]:
```
#Importing all important libraries and packages:

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

In [2]:
```
#Checking the current working directory.

os.getcwd()
```

Out[2]:
```
'C:\\Users\\user'
```

In [3]:
```
#Changing the working directory where the 'McDonald.csv' file is stored.

os.chdir('D:\\SHUBHANK !\\GL\\3. TOPIC 2 - Statistical Methods for Decision Ma
king\\SMDM- Week 2')
```

In [4]:
```
#Again checking the directory after changing it.

os.getcwd()
```

Out[4]:
```
'D:\\SHUBHANK !\\GL\\3. TOPIC 2 - Statistical Methods for Decision Making\\SM
DM- Week 2'
```

Now opening and reading the data from the McDonald.csv file:

In [5]:
```
McData = pd.read_csv('Mcdonald.csv')
```

Now fetching and displaying the top 10 records from the file:

In [7]:
```
McData.head(10)
```

Out[7]:

	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	..
5	Breakfast	Steak & Egg McMuffin	6.5 oz (185 g)	430	210	23.0	36	9.0	46	1.0	..
6	Breakfast	Bacon, Egg & Cheese Biscuit (Regular Biscuit)	5.3 oz (150 g)	460	230	26.0	40	13.0	65	0.0	..
7	Breakfast	Bacon, Egg & Cheese Biscuit (Large Biscuit)	5.8 oz (164 g)	520	270	30.0	47	14.0	68	0.0	..
8	Breakfast	Bacon, Egg & Cheese Biscuit with Egg Whites (R...	5.4 oz (153 g)	410	180	20.0	32	11.0	56	0.0	..
9	Breakfast	Bacon, Egg & Cheese Biscuit with Egg Whites (L...	5.9 oz (167 g)	470	220	25.0	38	12.0	59	0.0	..

10 rows × 24 columns

In [9]:
```
#to describe the five number summary and some other important numerical inform
ation about the dataset:

McData.describe()
```

Out[9]:

	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	C
count	260.000000	260.000000	260.000000	260.000000	260.000000	260.000000	260.000000	2
mean	368.269231	127.096154	14.165385	21.815385	6.007692	29.965385	0.203846	
std	240.269886	127.875914	14.205998	21.885199	5.321873	26.639209	0.429133	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	210.000000	20.000000	2.375000	3.750000	1.000000	4.750000	0.000000	
50%	340.000000	100.000000	11.000000	17.000000	5.000000	24.000000	0.000000	
75%	500.000000	200.000000	22.250000	35.000000	10.000000	48.000000	0.000000	
max	1880.000000	1060.000000	118.000000	182.000000	20.000000	102.000000	2.500000	5

8 rows × 21 columns

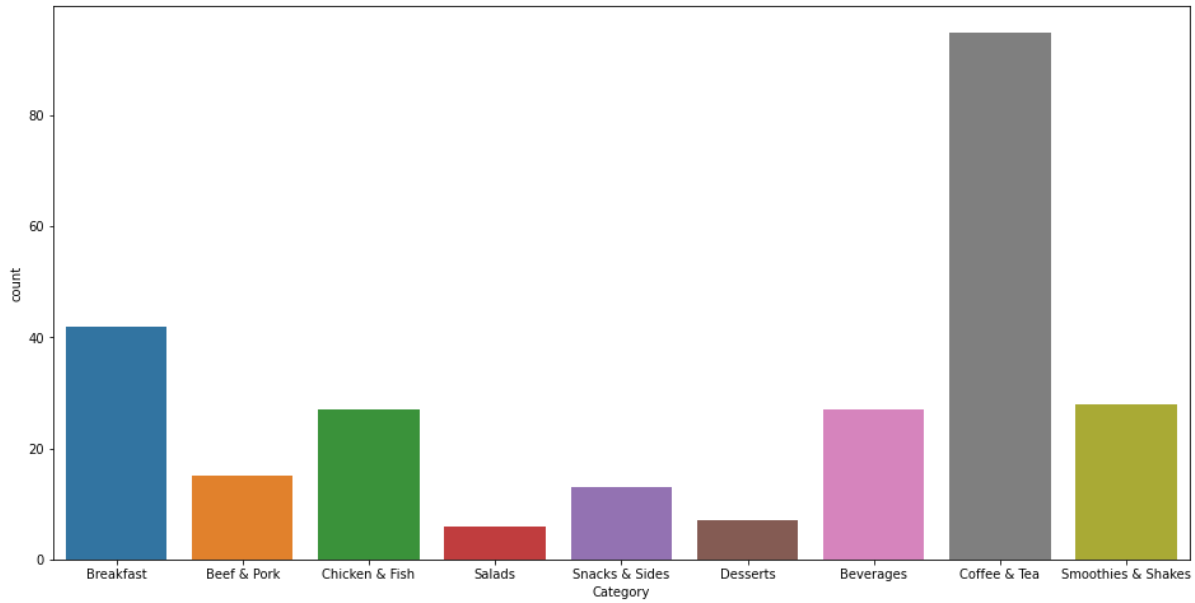
In [10]:
```
#getting the basic info of the dataset like about datatype, about missing valu
es, total entries, etc..

McData.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
```

Q1. Plot graphically which food categories have the highest and lowest varieties.

Solution: The below countplot will display the food categories with hignest and lowest varieties:

In [11]:
```
plt.figure(figsize=(16,8))              #defining the figure size of the plot
sns.countplot(McData['Category']);
```

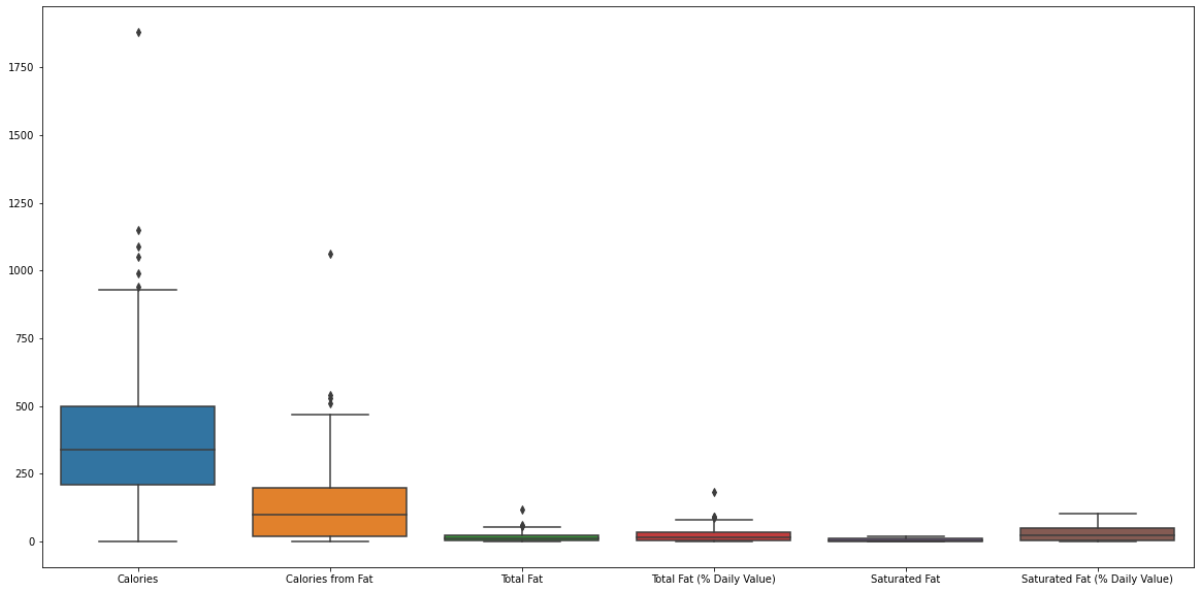


So, from the above plot we can see that 'Coffee & Tea' have the most varities and 'Salads' have the least varities.

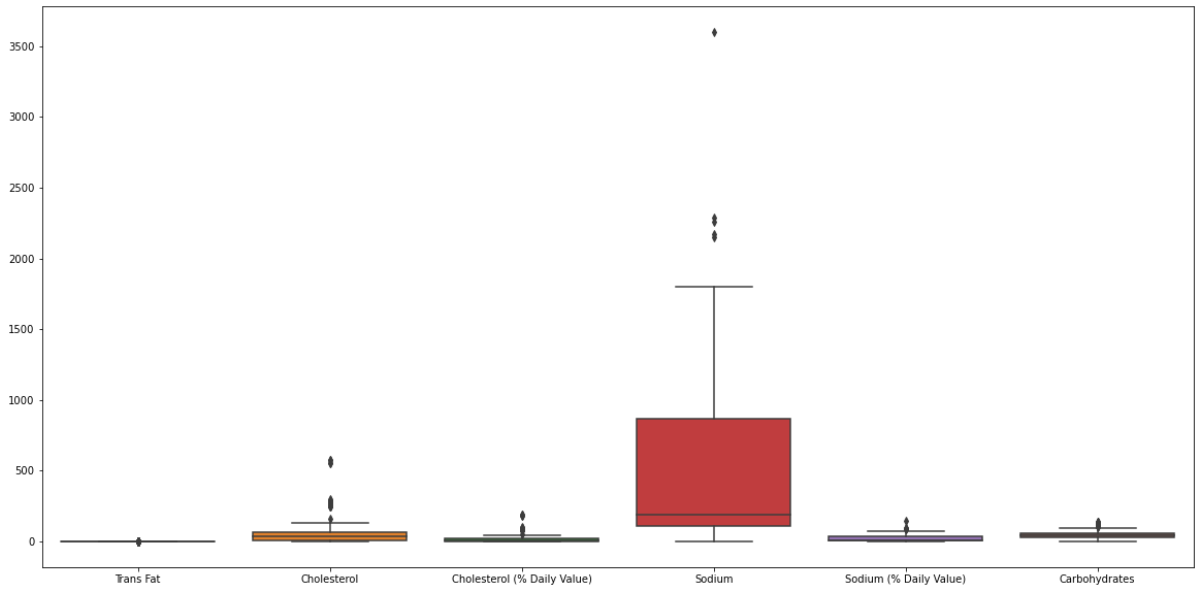
Q2. Which all variables have an outlier ?

Solution: We can check the outliers only for the columns with Numerical data.

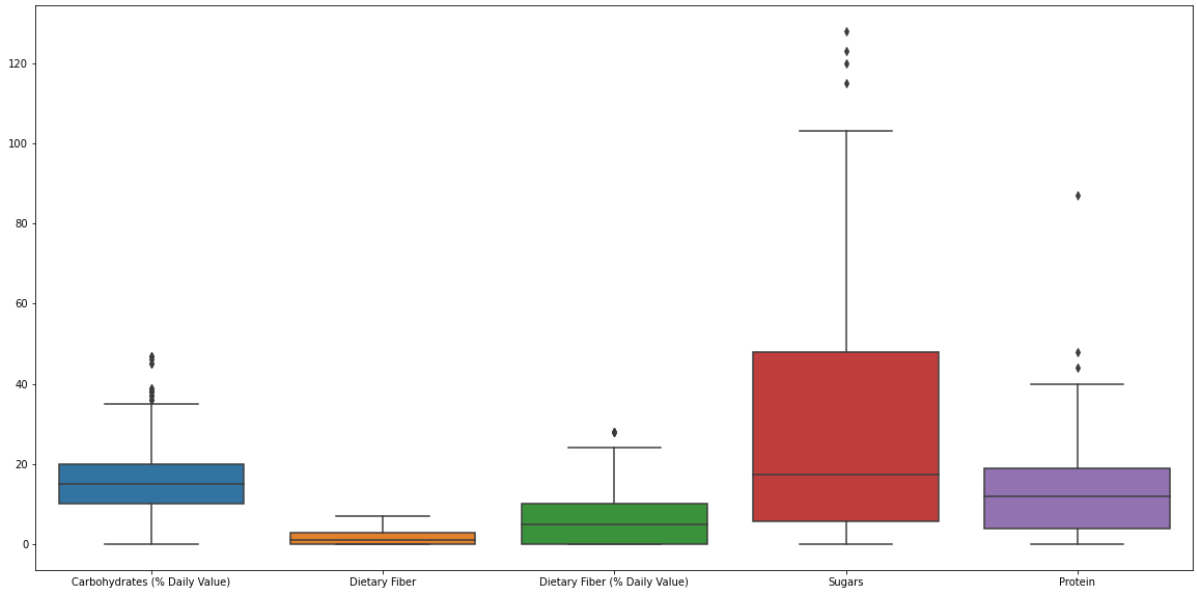
In [15]:
```
#Showing the boxplot and checking the outliers for the below variables:
#'Calories','Calories from Fat','Total Fat','Total Fat (% Daily Value)','Saturated Fat','Saturated Fat (% Daily Value).


plt.figure(figsize=(20,10))

sns.boxplot(data = McData[['Calories','Calories from Fat','Total Fat','Total Fat (% Daily Value)','Saturated Fat'
                           ,'Saturated Fat (% Daily Value)']]);
```



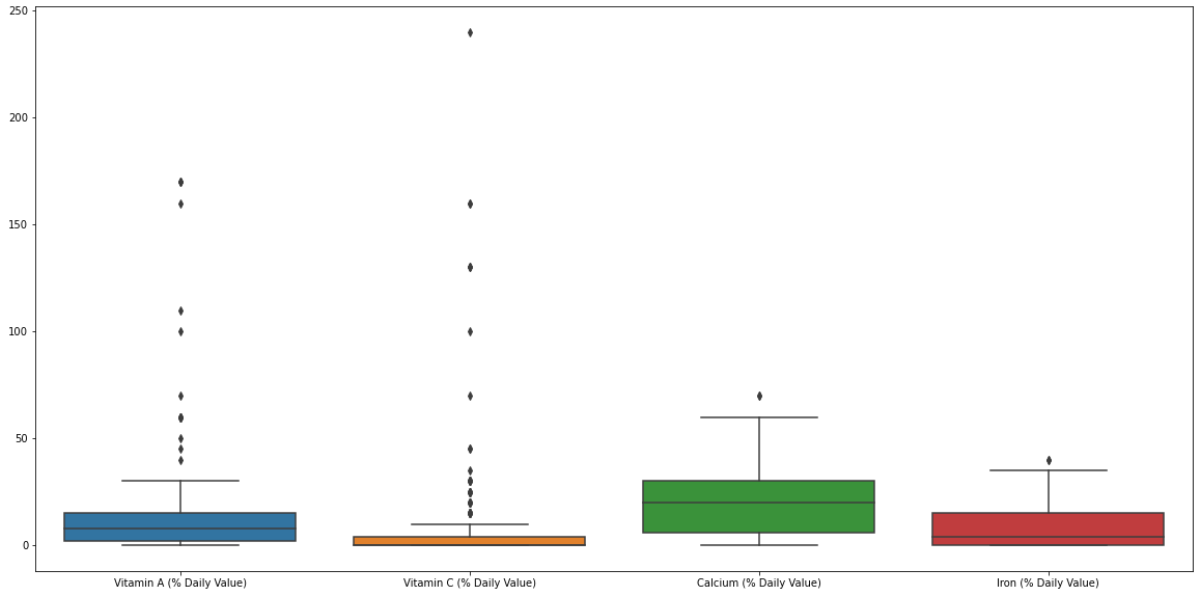
In [17]:
```
#Showing the boxplot and checking the outliers for the below variables:
#'Trans Fat','Cholesterol','Cholesterol (% Daily Value)','Sodium','Sodium (% D
aily Value)','Carbohydrates.


plt.figure(figsize=(20,10))

sns.boxplot(data = McData[['Trans Fat','Cholesterol','Cholesterol (% Daily Val
ue)','Sodium','Sodium (% Daily Value)'
                           ,'Carbohydrates']]);
```



In [19]:
```
#Showing the boxplot and checking the outliers for the below variables:
#'Carbohydrates (% Daily Value)','Dietary Fiber','Dietary Fiber (% Daily Valu
e)','Sugars','Protein.

plt.figure(figsize=(20,10))

sns.boxplot(data = McData[['Carbohydrates (% Daily Value)','Dietary Fiber','Di
etary Fiber (% Daily Value)','Sugars'
                           ,'Protein']]);
```



In [20]:
```
#Showing the boxplot and checking the outliers for the below variables:
#Vitamin A (% Daily Value),Vitamin C (% Daily Value),Calcium (% Daily Value),I
ron (% Daily Value)

plt.figure(figsize=(20,10))

sns.boxplot(data = McData[['Vitamin A (% Daily Value)','Vitamin C (% Daily Val
ue)','Calcium (% Daily Value)',
                           'Iron (% Daily Value)']]);
```



So, from the above boxplots we can conclude that:

The variables with outliers are :-

***Calories**

***Calories from Fat**

***Total fat**

***Saturated Fat (% Daily Value)**

***Cholestrol**

***Cholestrol (% Daily Value)**

***Calories from Fat**

***Cholestrol**

***Cholestrol (% Daily Value)**

***Total Fat**

***Total Fat (% Daily Value)**

***Sodium**

***Sodium (% Daily Value)**

The variables which don't have outliers are :-

***Trans Fat**

***Saturated Fat**

***Dietary Fiber**

Q3. Which variables have the highest correlation? Plot them and find out the value?

Solution: First we will create the correlation matrices between the variables:

In [14]:
```
corr = McData.corr()

corr
```

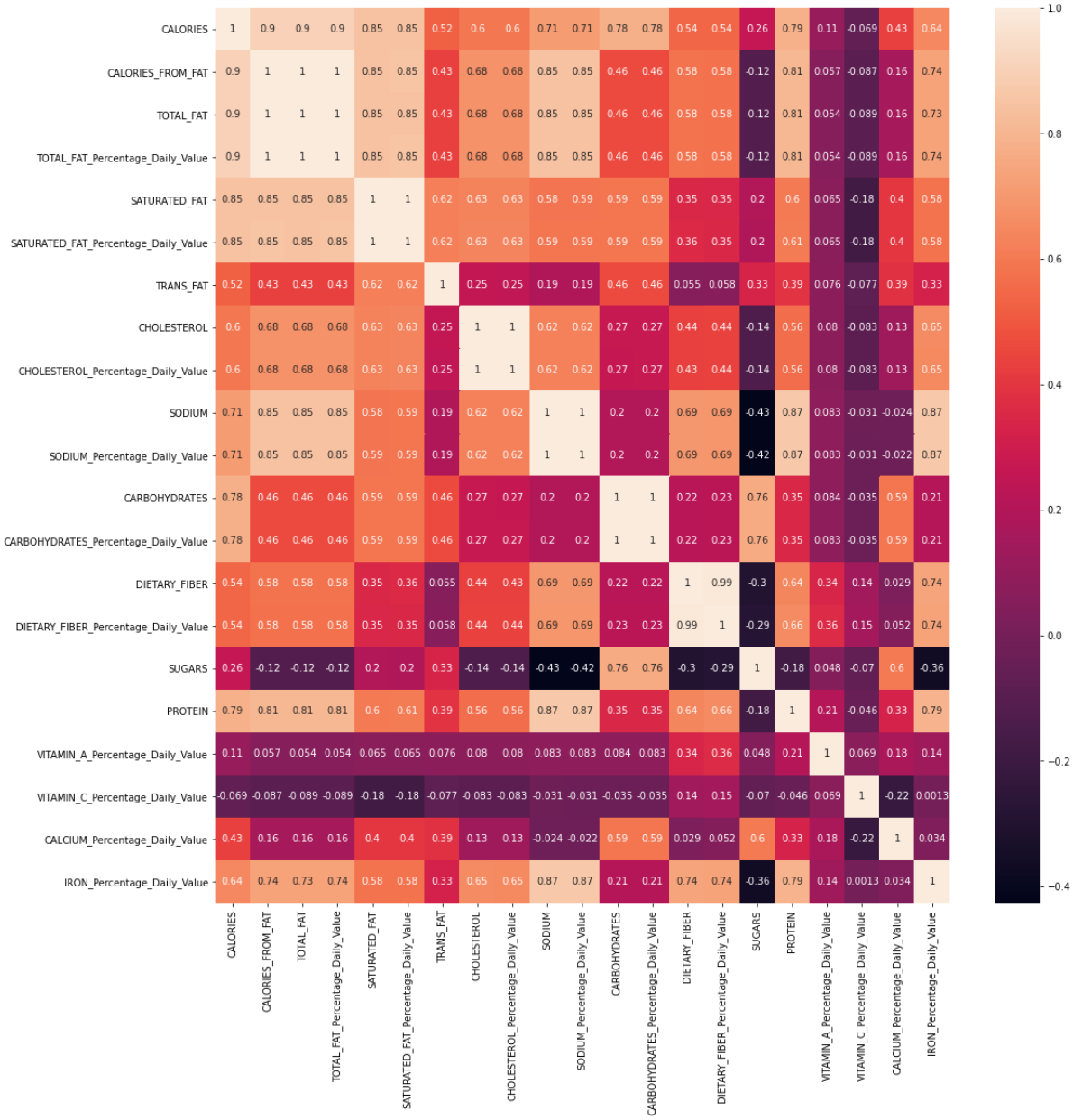
Out[14]:

	CALORIES	CALORIES_FROM_FAT	TOTAL_FAT	TOTA
CALORIES	1.000000	0.904588	0.904409	
CALORIES_FROM_FAT	0.904588	1.000000	0.999663	
TOTAL_FAT	0.904409	0.999663	1.000000	
TOTAL_FAT_Percentage_Daily_Value	0.904123	0.999725	0.999765	
SATURATED_FAT	0.845564	0.847008	0.846707	
SATURATED_FAT_Percentage_Daily_Value	0.847631	0.849592	0.849293	
TRANS_FAT	0.522441	0.433686	0.431453	
CHOLESTEROL	0.596399	0.682161	0.680547	
CHOLESTEROL_Percentage_Daily_Value	0.595208	0.681607	0.680000	
SODIUM	0.712309	0.846624	0.846158	
SODIUM_Percentage_Daily_Value	0.713415	0.847276	0.846780	
CARBOHYDRATES	0.781539	0.461672	0.461213	
CARBOHYDRATES_Percentage_Daily_Value	0.781242	0.461463	0.461005	
DIETARY_FIBER	0.538894	0.581274	0.580837	
DIETARY_FIBER_Percentage_Daily_Value	0.540014	0.575621	0.575206	
SUGARS	0.259598	-0.115285	-0.115446	
PROTEIN	0.787847	0.807913	0.807773	
VITAMIN_A_Percentage_Daily_Value	0.108844	0.056731	0.054434	
VITAMIN_C_Percentage_Daily_Value	-0.068747	-0.087331	-0.089354	
CALCIUM_Percentage_Daily_Value	0.428426	0.161034	0.162860	
IRON_Percentage_Daily_Value	0.643552	0.735894	0.734685	

21 rows × 21 columns

In [15]:
```
#Now we will plot the heatmap to display the correlation between the variable
s:

fig, ax = plt.subplots(figsize=(17,17))
sns.heatmap(corr,annot=True)
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1ad29a674c0>



So here in the above heatmap we can see that the light colours shows the highest correlation.

Highest correlation variables are :

(1)'Dietary_fiber' and 'Dietary_fiber_percentage_daily_value' (0.99 approx to 1)

(2)'Protein' and 'Sodium' (0.87)

(3) 'Sodium' and 'Iron' (0.85)

(4)'Protein' and 'Iron_percentage_daily_value' (0.79)

.

Q4. Which category contributes to the maximum % of Cholesterol in a diet (% daily value)?

Solution: First we will count the total Unique Categories in the dataset and then will solve the given question.

In [31]:
```
McData['Category'].nunique()
```

Out[31]: 9

In [40]:
```
Required_result_1 = pd.pivot_table(McData, 'Cholesterol (% Daily Value)', index=['Category'])

print('\n \n The list of different categories with their respective % Cholesterol is:-\n\n', Required_result_1)
```

```
 The list of different categories with their respective % Cholesterol is:-

                    Cholesterol (% Daily Value)
Category
Beef & Pork                           28.933333
Beverages                              0.185185
Breakfast                             50.952381
Chicken & Fish                        25.222222
Coffee & Tea                           9.378947
Desserts                               4.857143
Salads                                17.333333
Smoothies & Shakes                    14.714286
Snacks & Sides                         6.230769
```

So, from the above information we can see that the 'Breakfast' category contributes to the maximum % of Cholesterol in a diet.

Q5. Which item contributes maximum to the Sodium intake ?

Solution: First we will check the total Items present in the dataset:

In [43]:
```
McData['Item'].nunique()
```

Out[43]: 260

Now we will create the pivot table for the columns 'Sodium' and 'Item' :

In [48]:
```
P_table = pd.pivot_table(McData, 'Sodium', index=['Item'])

P_table
```

Out[48]:

Item	Sodium
1% Low Fat Milk Jug	125
Apple Slices	0
Bacon Buffalo Ranch McChicken	1260
Bacon Cheddar McChicken	1260
Bacon Clubhouse Burger	1470
...	...
Sweet Tea (Medium)	10
Sweet Tea (Small)	10
Vanilla Shake (Large)	260
Vanilla Shake (Medium)	200
Vanilla Shake (Small)	160

260 rows × 1 columns

Now we need to sort the data in the descending order of the sodium values so that we can find out the item that contributes maximum to the sodium intake:

In [50]:
```
Required_result_2 = P_table.sort_values(('Sodium'), ascending=False)

Required_result_2
```

Out[50]:

Item	Sodium
Chicken McNuggets (40 piece)	3600
Big Breakfast with Hotcakes and Egg Whites (Large Biscuit)	2290
Big Breakfast with Hotcakes (Large Biscuit)	2260
Big Breakfast with Hotcakes and Egg Whites (Regular Biscuit)	2170
Big Breakfast with Hotcakes (Regular Biscuit)	2150
...	...
Coca-Cola Classic (Child)	0
Dasani Water Bottle	0
Minute Maid Orange Juice (Medium)	0
Apple Slices	0
Minute Maid Orange Juice (Small)	0

260 rows × 1 columns

So from the above result we can see the item 'Chicken McNuggets' contributes maximum to the Sodium intake.

Q6. Which 4 food items contains the most amount of Saturated Fat ?

Solution: We will create the pivot table between the columns 'Saturated fat' and 'Items' and then we will sort the table to get the top 4 food items contains the most amount of Saturated Fat :

In [53]:
```
P_table_1 = pd.pivot_table(McData, 'Saturated Fat', index=['Item'])

Required_result_3 = P_table_1.sort_values(('Saturated Fat'), ascending=False)

Required_result_3
```

Out[53]:

Item	Saturated Fat
McFlurry with M&M's Candies (Medium)	20.0
Big Breakfast with Hotcakes (Large Biscuit)	20.0
Chicken McNuggets (40 piece)	20.0
Frappé Chocolate Chip (Large)	20.0
Double Quarter Pounder with Cheese	19.0
...	...
Diet Dr Pepper (Small)	0.0
Diet Dr Pepper (Medium)	0.0
Diet Dr Pepper (Large)	0.0
Diet Dr Pepper (Child)	0.0
Minute Maid Orange Juice (Large)	0.0

260 rows × 1 columns

So from the above data we can see that the top 4 food items are as follows:

* McFlurry with M&M's candies (Medium)

* Big Breakfast with Hotcakes (Large Biscuit)

* Chicken McNuggets (40 piece)

* Frappe Chocolate Chip (Large)