In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
plt.style.use ('ggplot')
import warnings
warnings.filterwarnings ('ignore')
```

In [3]:

```
df = pd.read_csv (r'F:\Certified Nutritionist\nutrients_csvfile_new.csv')
df.head(8)
```

Out[3]:

	Food	Measure	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs	Category
0	Cows' milk	1 qt.	976	660	32	40	36	0	48	Dairy products
1	Milk skim	1 qt.	984	360	36	t	t	0	52	Dairy products
2	Buttermilk	1 cup	246	127	9	5	4	0	13	Dairy products
3	Evaporated, undiluted	1 cup	252	345	16	20	18	0	24	Dairy products
4	Fortified milk	6 cups	1,419	1,373	89	42	23	1.4	119	Dairy products
5	Powdered milk	1 cup	103	515	27	28	24	0	39	Dairy products
6	skim, instant	1 1/3 cups	85	290	30	t	t	0	42	Dairy products
7	skim, non-instant	2/3 cup	85	290	30	t	t	1	42	Dairy products

Data Cleaning.

In [4]:

```
df = df.replace ("t", 0)
df = df.replace ("t'",0)
df.head(8) # Replacing "t" with 0.
```

Out[4]:

	Food	Measure	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs	Category
0	Cows' milk	1 qt.	976	660	32	40	36	0	48	Dairy products
1	Milk skim	1 qt.	984	360	36	0	0	0	52	Dairy products
2	Buttermilk	1 cup	246	127	9	5	4	0	13	Dairy products
3	Evaporated, undiluted	1 cup	252	345	16	20	18	0	24	Dairy products
4	Fortified milk	6 cups	1,419	1,373	89	42	23	1.4	119	Dairy products
5	Powdered milk	1 cup	103	515	27	28	24	0	39	Dairy products
6	skim, instant	1 1/3 cups	85	290	30	0	0	0	42	Dairy products
7	skim, non-instant	2/3 cup	85	290	30	0	0	1	42	Dairy products

In [5]:

```
df=df.replace(",","", regex=True)
df['Fiber']=df['Fiber'].replace("a","", regex=True)
df['Calories'][91]=(8+44)/2
# removed all commas in words and in numbers.
```

In [7]:

```
df.dtypes
```

Out[7]:

```
Food        object
Measure     object
Grams       object
Calories    object
Protein     object
Fat         object
Sat.Fat     object
Fiber       object
Carbs       object
Category    object
dtype: object
```

All are object but we know that there are continous variables as in real world nutrients cant be object as they will be continous.

Grams, Calories, Protein, Fat, Sat.Fat, Fiber, Carbs

In [8]:

```
df['Grams'] = pd.to_numeric (df['Grams'])
df['Calories'] = pd.to_numeric (df['Calories'])
df['Protein'] = pd.to_numeric (df['Protein'])
df['Fat'] = pd.to_numeric (df['Fat'])
df['Sat.Fat'] = pd.to_numeric (df['Sat.Fat'])
df['Fiber'] = pd.to_numeric (df['Fiber'])
df['Carbs'] = pd.to_numeric (df['Carbs'])
```

In [9]:

```
df.dtypes
```

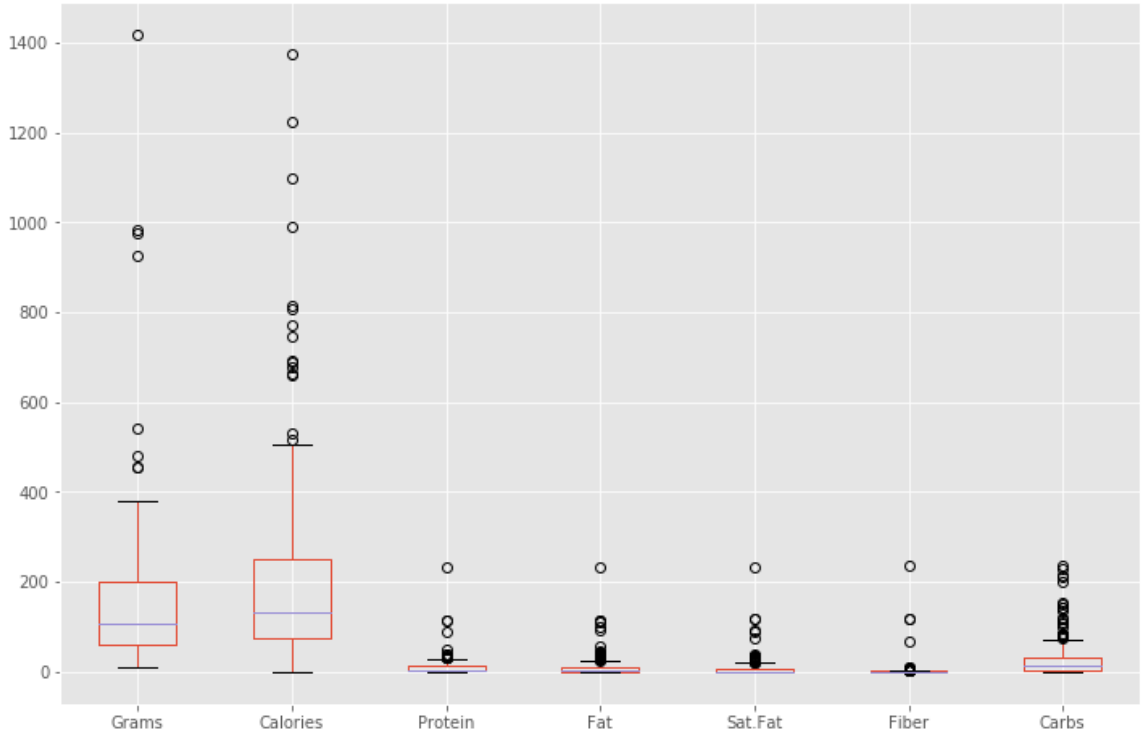
Out[9]:

```
Food         object
Measure      object
Grams         int64
Calories    float64
Protein       int64
Fat           int64
Sat.Fat     float64
Fiber       float64
Carbs       float64
Category     object
dtype: object
```

EDA

In [11]:

```
# checking outliers.
plt.figure (figsize=(12,8))
df.boxplot (vert='0')
plt.show()
```



Every column is having outliers, we know that each food has different micro and macro nutrients,hence treating them are not valid point.hence we will be replacing missing values with their median.

In [19]:

```
df.shape
```

Out[19]:

```
(335, 10)
```

In [10]:

```
df.isnull().sum()
```

Out[10]:

```
Food        0
Measure     0
Grams       0
Calories    1
Protein     0
Fat         0
Sat.Fat     2
Fiber       1
Carbs       0
Category    0
dtype: int64
```

In [16]:

```
print (df.Calories.median())

print (df.Fiber.median())
```

```
131.0
0.2
```

In [17]:

```
df ['Calories'].fillna (df['Calories'].median(),inplace=True)
```

In [18]:

```
df['Fiber'].fillna (df['Fiber'].median(),inplace=True)
```

In [21]:

```
df = df.dropna() # dropping Sat.Fat row
df.shape
```

Out[21]:

```
(333, 10)
```

In [22]:

```
df.isnull().sum()
```

Out[22]:

```
Food        0
Measure     0
Grams       0
Calories    0
Protein     0
Fat         0
Sat.Fat     0
Fiber       0
Carbs       0
Category    0
dtype: int64
```

In [23]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 333 entries, 0 to 334
Data columns (total 10 columns):
Food        333 non-null object
Measure     333 non-null object
Grams       333 non-null int64
Calories    333 non-null float64
Protein     333 non-null int64
Fat         333 non-null int64
Sat.Fat     333 non-null float64
Fiber       333 non-null float64
Carbs       333 non-null float64
Category    333 non-null object
dtypes: float64(4), int64(3), object(3)
memory usage: 38.6+ KB
```

In [25]:

```
df.describe()
```

Out[25]:

	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs
count	333.000000	333.000000	333.000000	333.000000	333.000000	333.000000	333.000000
mean	143.396396	188.348348	8.579580	8.426426	6.438438	2.383814	25.130030
std	139.005414	183.822872	17.783431	19.687119	18.517656	16.101853	35.889878
min	11.000000	0.000000	-1.000000	0.000000	0.000000	0.000000	0.000000
25%	60.000000	75.000000	1.000000	0.000000	0.000000	0.000000	3.000000
50%	108.000000	131.000000	3.000000	1.000000	0.000000	0.200000	14.000000
75%	200.000000	250.000000	12.000000	10.000000	8.000000	1.000000	31.000000
max	1419.000000	1373.000000	232.000000	233.000000	234.000000	235.000000	236.000000

Data cleaning and preprocessing has been done.

In [ ]:

```

```

Univariate, Bi-variate and Multivariate Analysis.

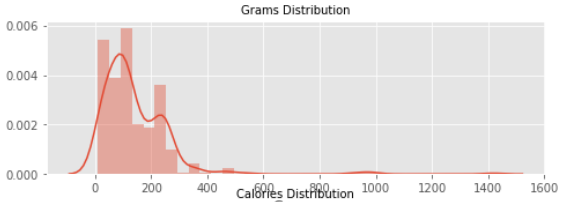
For continous variables. Grams, Calories, Protein, Fat, Sat.Fat, Fiber, Carbs

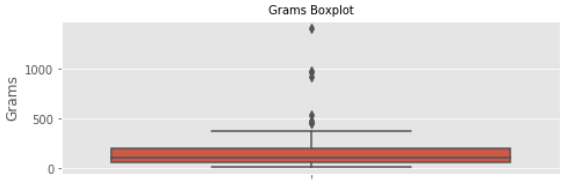
In [31]:

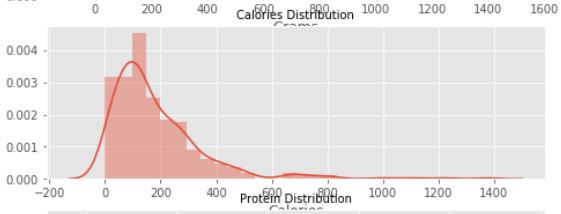
```
fig, axes = plt.subplots (nrows=7,ncols=2)
fig.set_size_inches (17,20)

a = sns.distplot (df['Grams'], ax=axes [0][0]);
a.set_title ('Grams Distribution',fontsize=10)
a = sns.boxplot (df['Grams'],orient='v',ax=axes [0][1]);
a.set_title ('Grams Boxplot',fontsize=10)

a = sns.distplot (df['Calories'],ax=axes [1][0]);
a.set_title ('Calories Distribution',fontsize=10)
a = sns.boxplot (df['Calories'],orient='v',ax=axes [1][1]);
a.set_title ('Calories Boxplot',fontsize=10)

a = sns.distplot (df['Protein'],ax=axes [2][0]);
a.set_title ('Protein Distribution',fontsize=10)
a = sns.boxplot (df['Protein'],orient='v',ax=axes [2][1]);
a.set_title ('Protein Boxplot',fontsize=10)

a = sns.distplot (df['Fat'],ax=axes [3][0]);
a.set_title ('Fat Distribution',fontsize=10)
a = sns.boxplot (df['Fat'],orient='v',ax=axes [3][1]);
a.set_title ('Fat Boxplot',fontsize=10)

a = sns.distplot (df['Sat.Fat'],ax=axes [4][0]);
a.set_title ('Sat.Fat Distribution',fontsize=10)
a = sns.boxplot (df['Sat.Fat'],orient='v',ax=axes [4][1]);
a.set_title ('Sat.Fat Boxplot',fontsize=10)

a = sns.distplot (df['Fiber'],ax=axes[5][0]);
a.set_title ('Fiber Distribution',fontsize=10)
a = sns.boxplot (df['Fiber'],orient='v',ax=axes [5][1]);
a.set_title ('Fiber Boxplot',fontsize=10)

a = sns.distplot (df['Carbs'],ax=axes [6][0]);
a.set_title ('Carbs Distribution',fontsize=10)
a = sns.boxplot (df['Carbs'],orient='v',ax=axes [6][1]);
a.set_title ('Carbs Boxplot',fontsize=10)
```

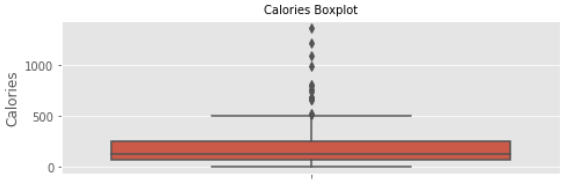
Out[31]:

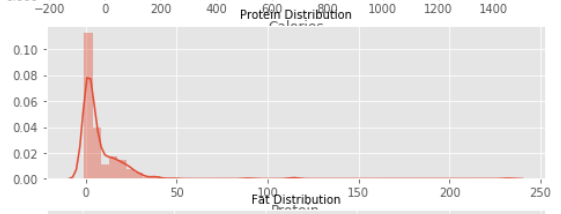
```
Text(0.5, 1.0, 'Carbs Boxplot')
```

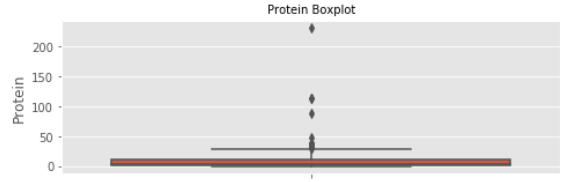

Grams Distribution

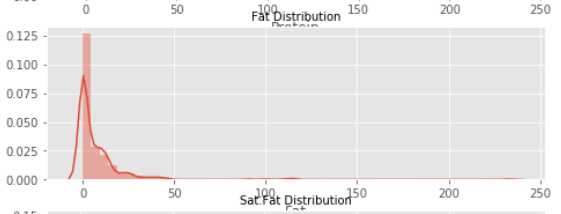

Grams Boxplot
Grams

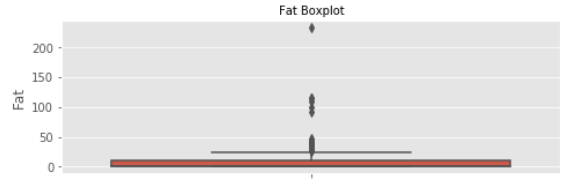

Calories Distribution

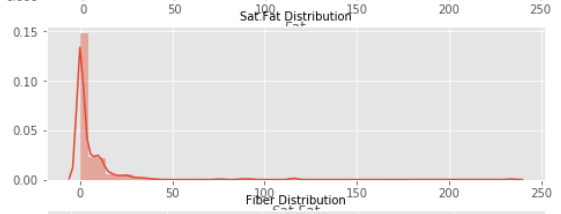

Calories Boxplot
Calories

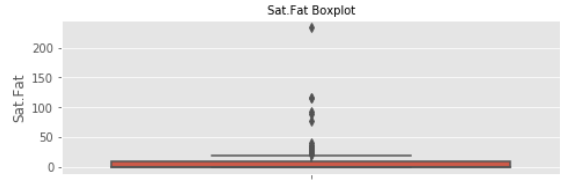

Protein Distribution

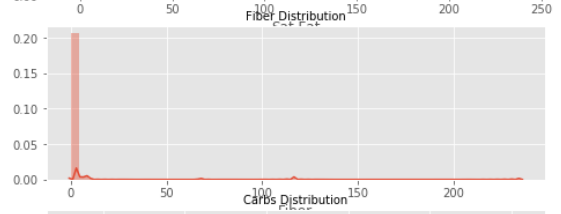

Protein Boxplot
Protein

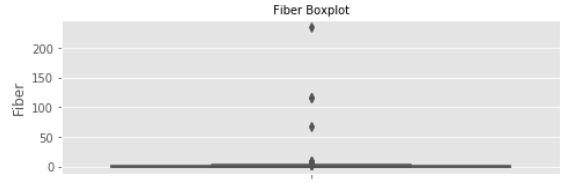

Fat Distribution

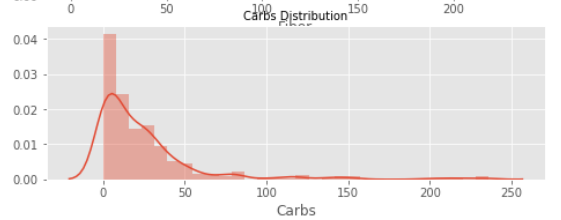

Fat Boxplot
Fat

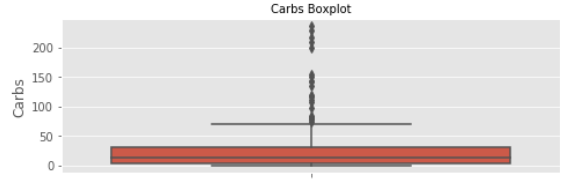

Sat.Fat Distribution


Sat.Fat Boxplot
Sat.Fat


Fiber Distribution


Fiber Boxplot
Fiber


Carbs Distribution
Carbs


Carbs Boxplot
Carbs

As we can see that all the continous variables have right skewed as the flat line to the end and its totally related with the population data also they all have outliers which can be seen in Boxplot.

In [32]:

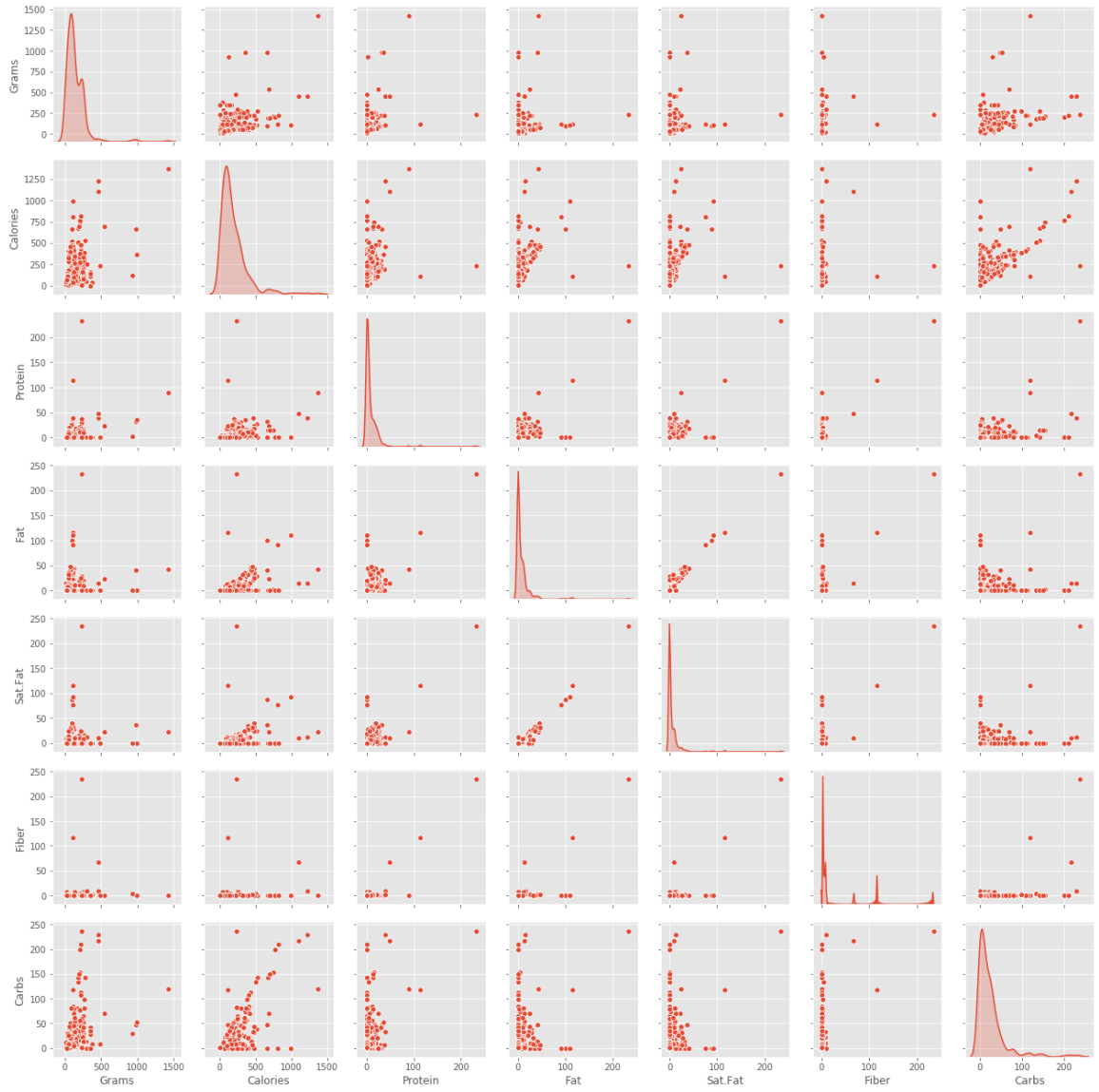
```
# correlation.

df.corr(method='pearson')
```

Out[32]:

	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs
Grams	1.000000	0.405786	0.218453	0.031836	0.039623	0.059406	0.363071
Calories	0.405786	1.000000	0.297692	0.388889	0.320538	0.061260	0.616406
Protein	0.218453	0.297692	1.000000	0.742977	0.755505	0.834724	0.390383
Fat	0.031836	0.388889	0.742977	1.000000	0.977957	0.733635	0.227446
Sat.Fat	0.039623	0.320538	0.755505	0.977957	1.000000	0.794014	0.264885
Fiber	0.059406	0.061260	0.834724	0.733635	0.794014	1.000000	0.450309
Carbs	0.363071	0.616406	0.390383	0.227446	0.264885	0.450309	1.000000

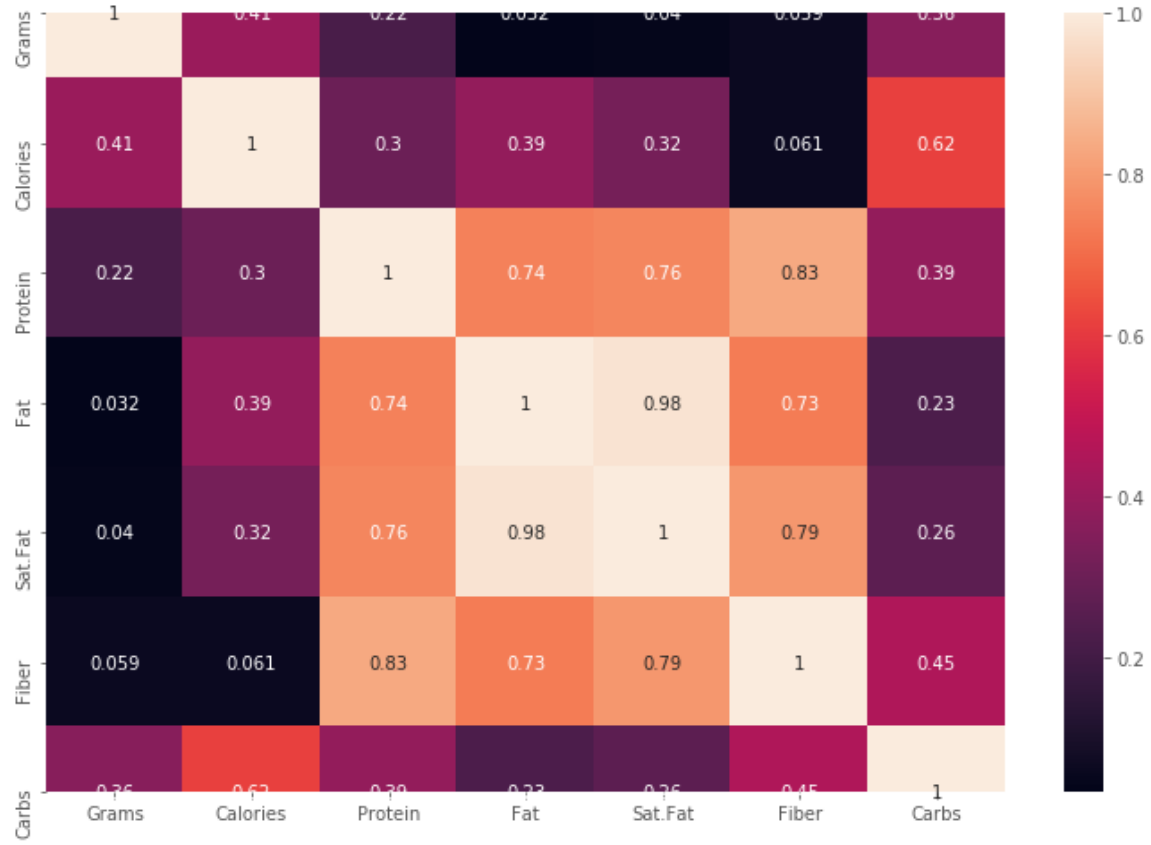
In [34]:

```
# pairplot.
sns.pairplot (df, diag_kind='kde');
```


Grams
Calories
Protein
Fat
Sat.Fat
Fiber
Carbs

In [44]:

```
plt.figure(figsize=(12,8))
sns.heatmap(df.corr(),annot=True);
plt.show()
```



Above analysis shows that every continous variables have a good correlation with each other as all are in positive.

In [ ]:

```

```

In [ ]:

```

```

Lets explore the Insights now.

In [45]:

```
### Categories.
df.Category.value_counts()
```

Out[45]:

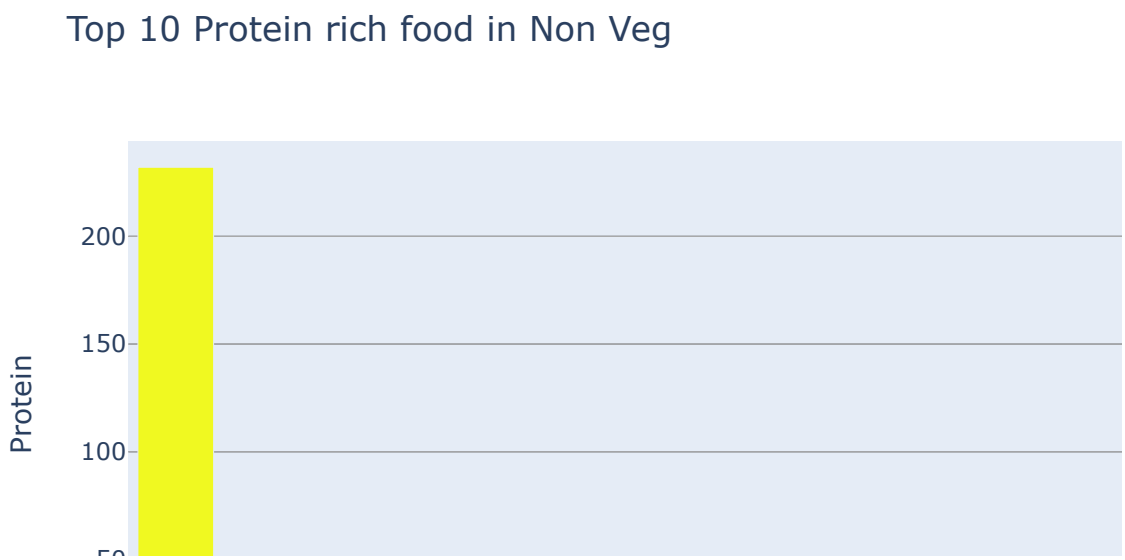
```
Breads cereals fastfoodgrains    45
Meat Poultry                     29
Desserts sweets                  29
Fruits G-P                       28
Dairy products                   28
Vegetables R-Z                   28
Vegetables A-E                   27
Fruits A-F                       22
Fish Seafood                     19
Fats Oils Shortenings            14
Vegetables F-P                   14
DrinksAlcohol Beverages          12
Seeds and Nuts                   12
Soups                            10
Fruits R-Z                        8
Jams Jellies                      8
Name: Category, dtype: int64
```

1: Proteint Rich Food in All the Categories, we will start from:

Meat Poultry (all non vegeterian).

In [52]:

```
import plotly.express as px

nv = ['Meat Poultry','Fish Seafood']

protnv = df [df ['Category'].isin (nv)]

protein_rich_nv = protnv.sort_values (by='Protein',ascending=False)

top_20 = protein_rich_nv.head(17)

fig = px.bar (top_20, x='Food', y='Protein', color='Protein', title= ' Top 10 Protein r
ich food in Non Veg')
fig.show()
```

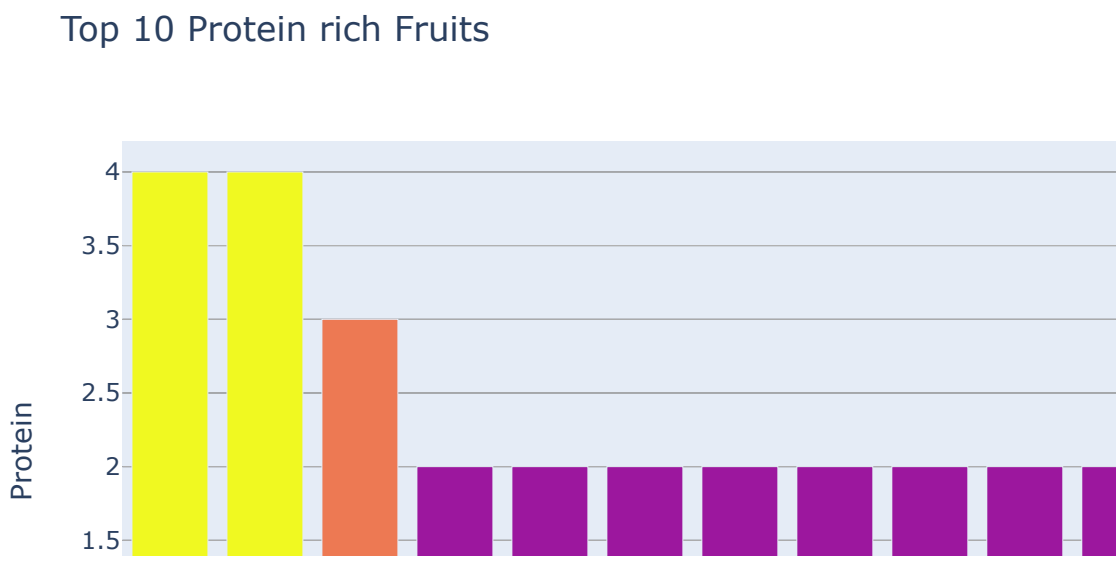


Top 10 Non-Veg Protein Foods in India			
Sr.No	Name	Serving Size in Grams	Protein Value
1	Fried Chicken breast	85	25
2	Leg	85	25
3	Thigh Chicken	85	25
4	Roasted chicken	100	25
5	Tuna	85	25
6	Chicken	85	23
7	Chicken livers fried	100	22
8	Fish sticks fried	112	19
9	Salmon	85	17
10	Clams	85	12

Top Protein Foods in Fruits.

In [60]:

```
fruits = ['Fruits G-P','Fruits A-F', 'Fruits R-Z']

protfruits = df [df ['Category'].isin (fruits)]

protein_rich_fruits = protfruits.sort_values (by='Protein',ascending=False)

top_20 = protein_rich_fruits.head(17)

fig = px.bar (top_20, x='Food', y='Protein', color='Protein', title= ' Top 10 Protein r
ich Fruits')
fig.show()
```

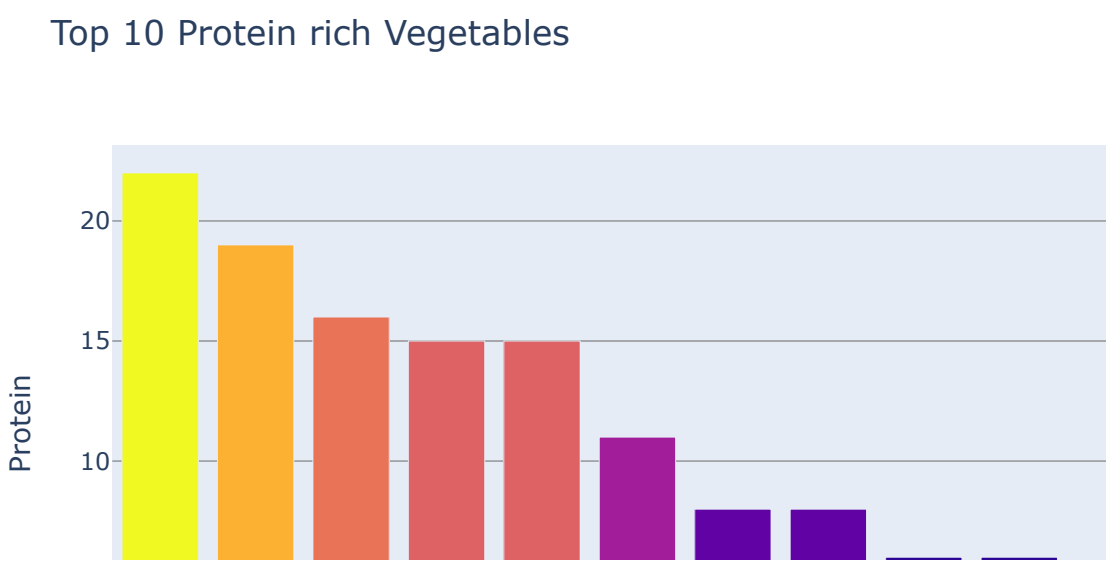


Top 10 Protein Fruits in India			
Sr.No	Name	Serving Size in Grams	Protein Value
1	Dates	178 or 1 cup	4
2	Blackberries	144 or 1 cup	2
3	Watermelon	925	2
4	Orange juice	1 or 1.5	2
5	Papaya	200 or 1/2	1
6	Peaches	257 or 1 cup	1
7	Pears	255 or 1 cup	1
8	Grapes	153 or 1 cup	1
9	Pineapple	122 or 1 Large size	0
10	Lemon juice	125 or 1/2	0

Top Protein Foods in Vegetables.

In [63]:

```
vegetables = ['Vegetables R-Z','Vegetables A-E', 'Vegetables F-P']

protveg = df [df ['Category'].isin (vegetables)]

protein_rich_veg = protveg.sort_values (by='Protein',ascending=False)

top_20 = protein_rich_veg.head(17)

fig = px.bar (top_20, x='Food', y='Protein', color='Protein', title= ' Top 10 Protein r
ich Vegetables')
fig.show()
```

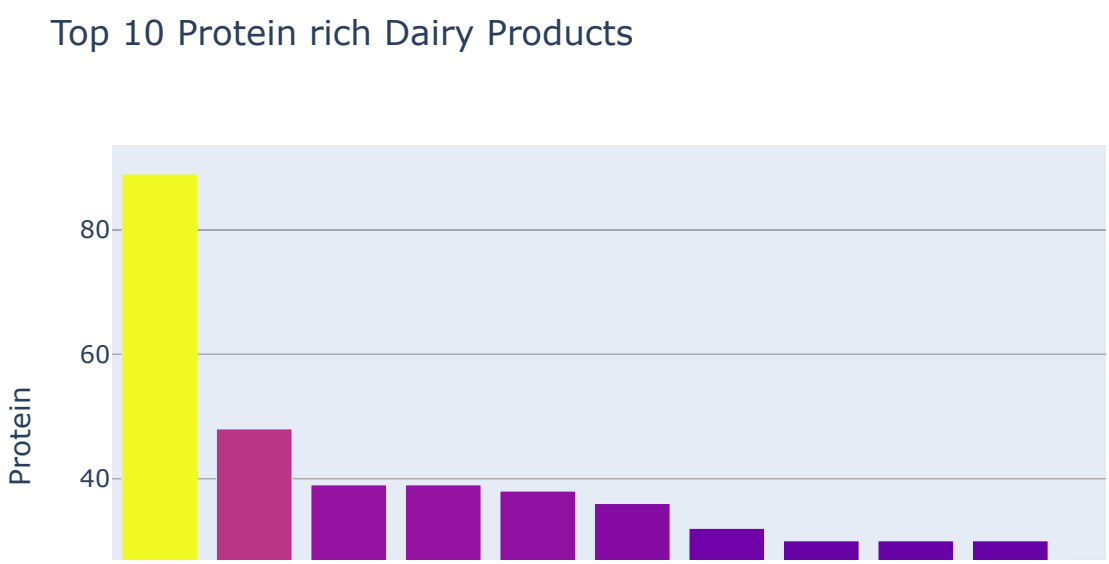


Top 10 Vegetables Protein in India			
Sr.No	Name	Serving Size in Grams	Protein Value
1	Soybeans	200 or 1 cup	22
2	Red kidney	260 or 1 cup	15
3	Lentils	200 or 1 cup	15
4	Cooked peas	100 or 4 cups	8
5	Brussels sprouts	130 or 1 cup	6
6	Broccoli	150 or 1 cup	5
7	Scalloped with cheese potat	100 or 3/4 cup	6
8	Frozen peas	100 or 1 cup	5
9	Potatoes Mashed with milk a	200 or 1 cup	4
10	Corn	100 or 1	3

Top Protein Foods in Dairy products.

In [65]:

```
dairy = ['Breads cereals fastfoodgrains','Dairy products']

protdairy = df [df ['Category'].isin (dairy)]

protein_rich_dairy = protdairy.sort_values (by='Protein',ascending=False)

top_20 = protein_rich_dairy.head(17)

fig = px.bar (top_20, x='Food', y='Protein', color='Protein', title= ' Top 10 Protein r
ich Dairy Products')
fig.show()
```

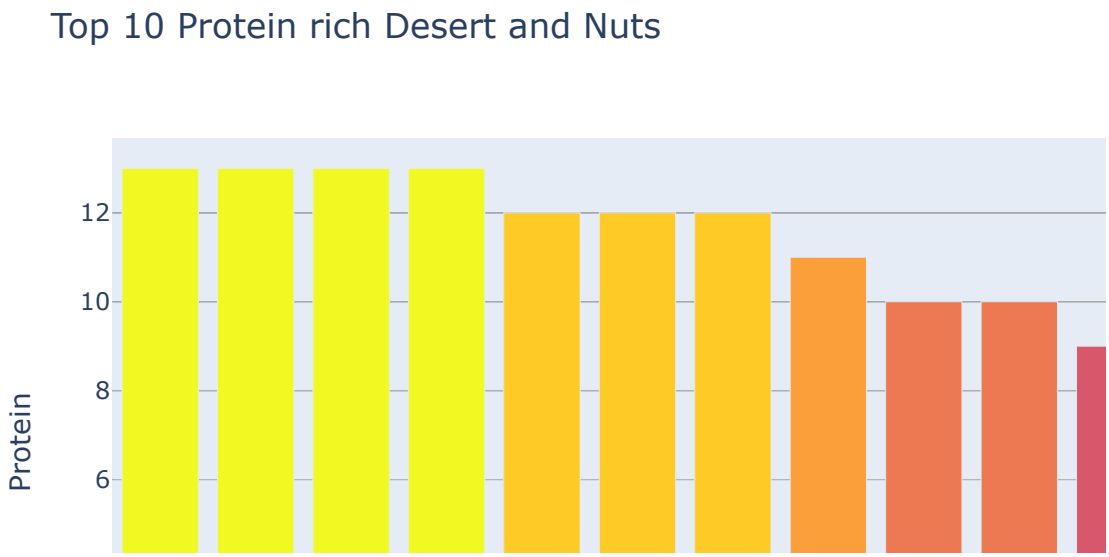


Top 10 Protein in Dairy products in India			
Sr.No	Name	Serving Size in Grams	Protein Value
1	Fortified milk	236.5 or 1 cup	14.8
2	Cows Milk	1 cup	5
3	Cheese	1 cup	30
4	Powdered milk	1 cup	27
5	Whole-wheat	200	24
6	Cornflakes	1 cup	2
7	Popcorn salted	28 or 2 cup	3
8	Biscuits	38	3
9	Cream cheese	28 or 1 hand	2
10	wheat biscuit	28 or 1 hand	3

Top Protein in Sweets and Nuts.

In [69]:

```
sn = ['Desserts sweets','Seeds and Nuts']

protsn = df [df ['Category'].isin (sn)]

protein_rich_sn = protsn.sort_values (by='Protein',ascending=False)

top_20 = protein_rich_sn.head(17)

fig = px.bar (top_20, x='Food', y='Protein', color='Protein', title= ' Top 10 Protein r
ich Desert and Nuts')
fig.show()
```

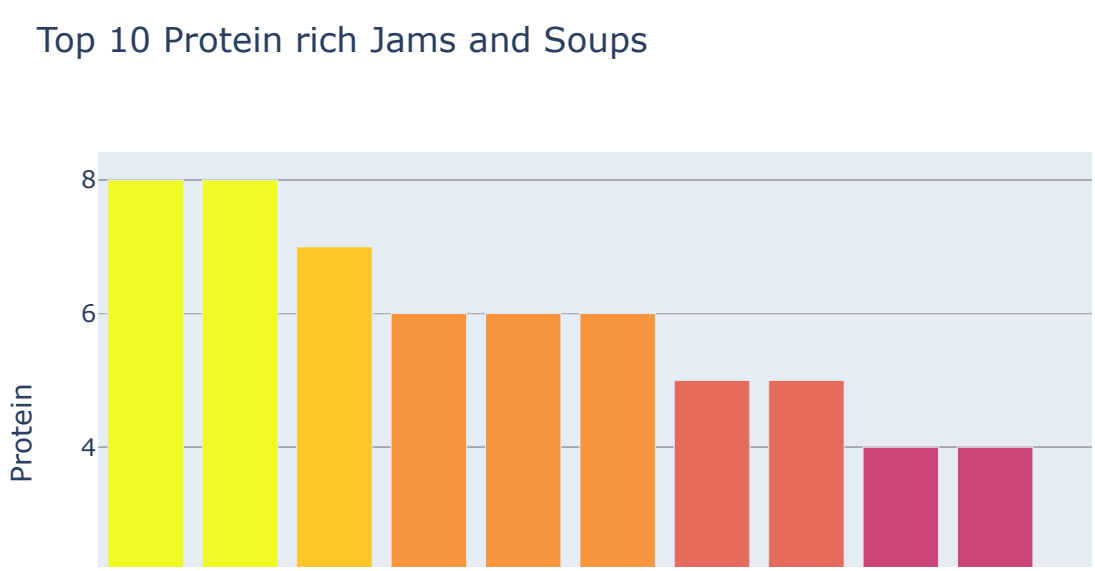


Top 10 Protein foods in Sweet and Nuts in India			
Sr.No	Name	Serving Size in Grams	Protein Value
1	Peanut butter natural	50 or 1/3	13
2	Almonds	70 or 1/2 cup	13
3	Peanuts	50 or 1/3 cup	13
4	Cashews	70 or 1/2cup	12
5	Walnuts	50 or 1/2 cup	7
6	Custard	130 or 1 slice	7
7	Cakes	40 or 1 slice	3
8	Doughnuts	33 or 1	2
9	Ice cream	250 or 2 cups	0
10	3 teaspoons sugar	12 or 1 T	0

Top Protein in Jams and Soups.

In [72]:

```
js = ['Jams Jellies','Soups']

protjs = df [df ['Category'].isin (js)]

protein_rich_js = protjs.sort_values (by='Protein',ascending=False)

top_20 = protein_rich_js.head(17)

fig = px.bar (top_20, x='Food', y='Protein', color='Protein', title= ' Top 10 Protein r
ich Jams and Soups')
fig.show()
```



Top 10 Protein foods in Jams and Soups in India			
Sr.No	Name	Serving Size in Grams	Protein Value
1	Bean soups	250 or 1 cup	8
2	Cream soups	255 or 1 cup	7
3	Tomato soup	245 or 1 cup	6
4	Noodle	250 or 1 cup	6
5	Vegetable	250 or 1 cup	4
6	chicken soup	250 or 1 cup	4
7	Honey	42 or 2 T	0
8	Jellies	20 or 1 T	0
9	preserves	20 or 1 T	0
10	Brown firm-packed dark sugar	220 or 1 cup	0

Top 10 Protein in Beverages.

In [75]:

```
bv = ['DrinksAlcohol Beverages','Fats Oils Shortenings']

protbv = df [df ['Category'].isin (bv)]

protein_rich_bv = protbv.sort_values (by='Protein',ascending=False)

top_20 = protein_rich_bv.head(17)

fig = px.bar (top_20, x='Food', y='Protein', color='Protein', title= ' Top 10 Protein i
n Beverages')
fig.show()
```

Top 10 Protein foods in Beverages in India			
Sr.No	Name	Serving Size in Grams	Protein Value
1	Butter	112 or 1/2 cup	114
2	Yolks	34 or 2	6
3	Coffee	1 cup	0
4	Fruit-flavored soda	1 cup	0
5	Cola drinks	1 cup	0
6	Club soda	1 cup	0
7	Safflower seed oil	1 cup	0
8	Olive oil	1 cup	0
9	Corn oil	1 cup	0
10	Tea	1 cup0	0

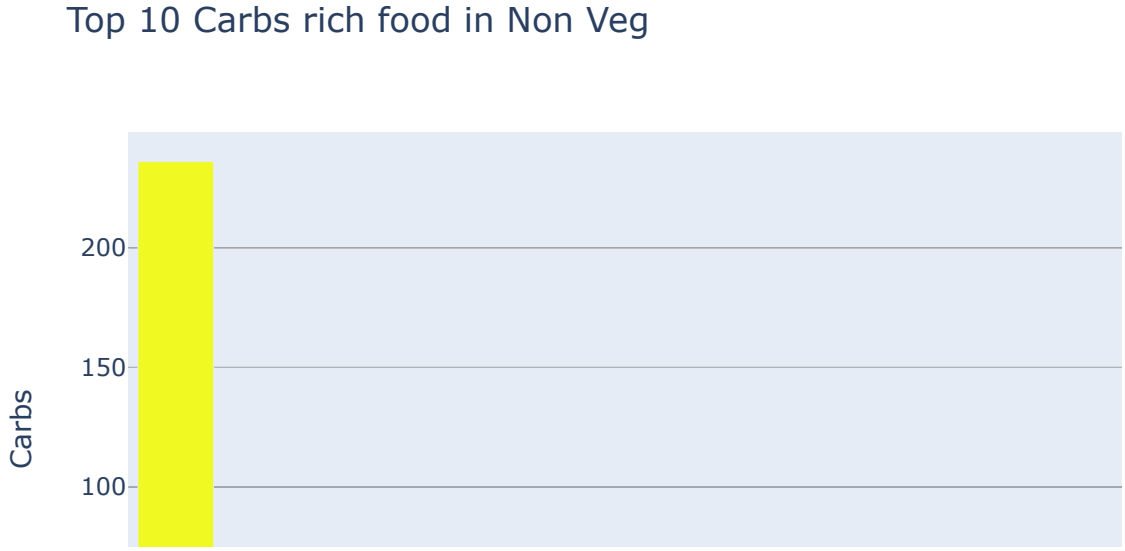
In [ ]:

2: Carbs Rich Food in All the Categories, we will start from:

Meat Poultry (all non vegeterian).

In [77]:

```
nv = ['Meat Poultry','Fish Seafood']

carbnv = df [df ['Category'].isin (nv)]

carb_rich_nv = carbnv.sort_values (by='Carbs',ascending=False)

top_20 = carb_rich_nv.head(17)

fig = px.bar (top_20, x='Food', y='Carbs', color='Carbs', title= ' Top 10 Carbs rich fo
od in Non Veg')
fig.show()
```

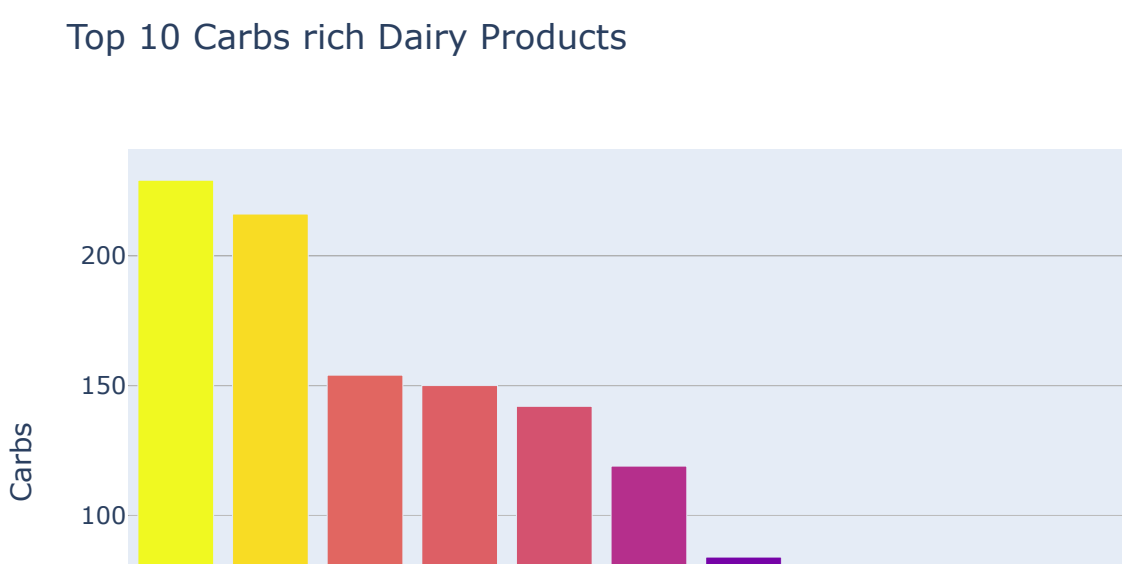


Top 10 Carbs foods in Non Veg in India			
Sr.No	Name	Serving Size in Grams	Carbs Value
1	Oysters	230	236
2	Pot-pie	227	32
3	Fish sticks fried	112	8
4	Haddock	85	6
5	Chicken livers fried	100	2.3

Top Carbs Foods in Dairy products.

In [79]:

```
dairy = ['Breads cereals fastfoodgrains','Dairy products']

carbdairy = df [df ['Category'].isin (dairy)]

carb_rich_dairy = carbdairy.sort_values (by='Carbs',ascending=False)

top_20 = carb_rich_dairy.head(17)

fig = px.bar (top_20, x='Food', y='Carbs', color='Carbs', title= ' Top 10 Carbs rich Da
iry Products')
fig.show()
```



Top 10 Carbs foods in Dairy Product in India			
Sr.No	Name	Serving Size in Grams	Carbs Value
1	White bread 20 slices	1 packet	229
2	Whole-wheat	100	50
3	White Rice	1 cup	150
4	Wheat (whole)	1 cup	79
5	Corn meal	1 cup	74
6	Cows' milk	100	5
7	Noodles	1 cup	37
8	Eggs Scrambled or fried	2 piece	1
9	Cheese	1 cup	6
10	Goats' milk	1 cup	11

Top Carbs in Sweets and Nuts.

In [84]:

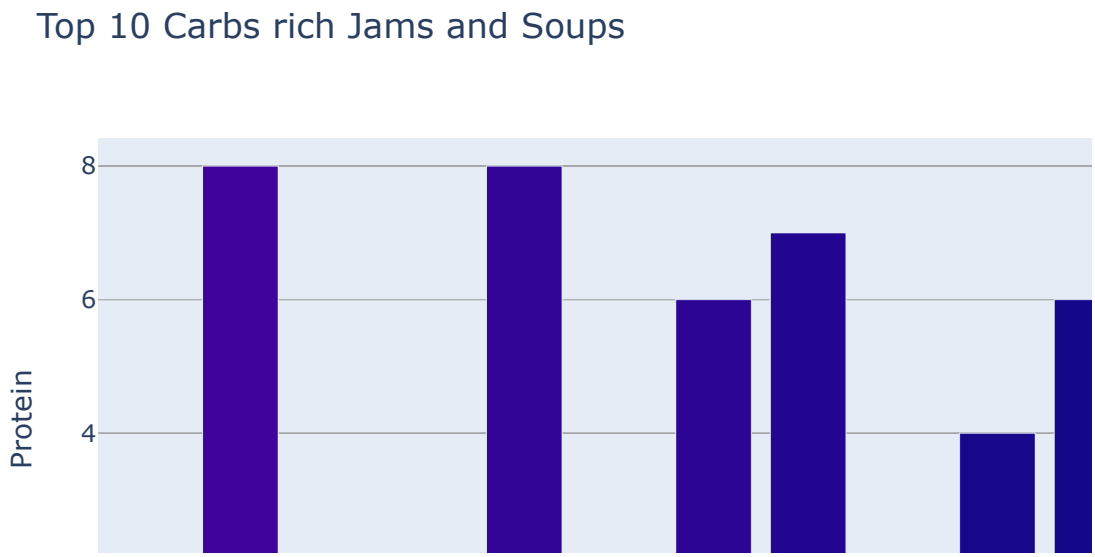
```
dairy = ['Desserts sweets','Seeds and Nuts']

carbsn = df [df ['Category'].isin (dairy)]

carb_rich_sn = carbsn.sort_values (by='Carbs',ascending=False)

top_20 = carb_rich_sn.head(17)

fig = px.bar (top_20, x='Food', y='Carbs', color='Carbs', title= ' Top 10 Carbs rich Sweet and Nuts')
fig.show()
```

Top 10 Carbs foods in Sweet and Nuts in India			
Sr.No	Name	Serving Size in Grams	Carbs Value
1	Puddings Sugar	1 cup	199
2	Bread pudding	1 cup	15
3	Milk chocolate	1 full	44
4	Pumpkin Pie	1 slice	34
5	Custard	1 slice	34
6	Cupcake	1 slice	31
7	Cashews	1/2 cup	20
8	Almonds	1/2 cup	13
9	Peanuts	1/3 cup	9
10	Walnuts	1/2 cup	8

Top Carbs in Jams and Soups.

In [88]:

```
js = ['Jams Jellies','Soups']

carbjs = df [df ['Category'].isin (js)]

carbs_rich_js = carbjs.sort_values (by='Carbs',ascending=False)

top_20 = carbs_rich_js.head(17)

fig = px.bar (top_20, x='Food', y='Protein', color='Carbs', title= ' Top 10 Carbs rich
 Jams and Soups')
fig.show()
```

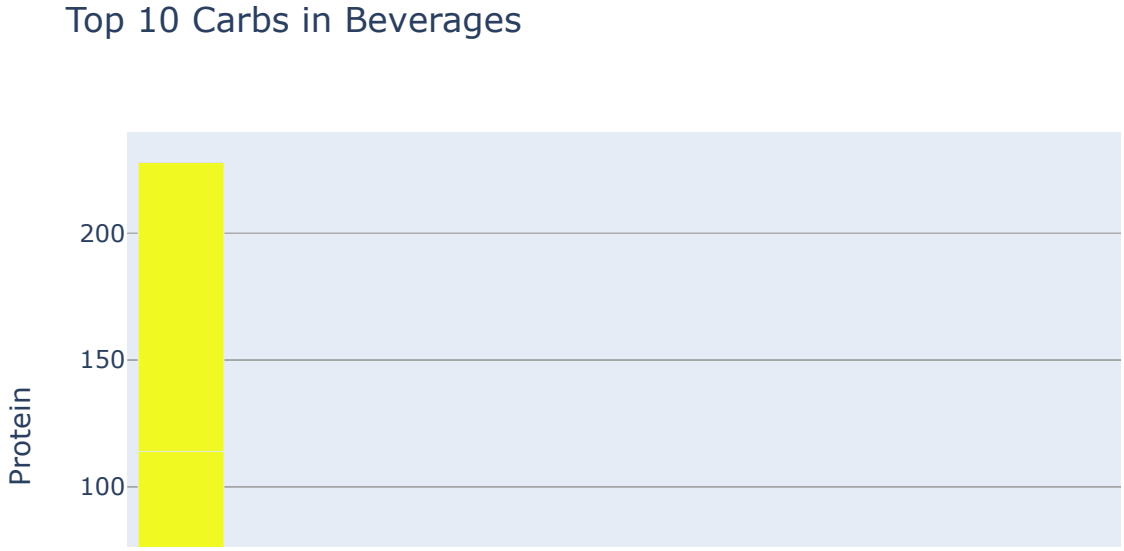


Top 10 Carbs foods in Jams and Soups in India			
Sr.No	Name	Serving Size in Grams	Carbs Value
1	Honey	2 Teaspoon	30
2	Tomato soup	1 cup	22
3	Vegetable	1 cup	14
4	Chicken soup	1 cup	10
5	Bean soups	1 cup	30
6	Syrup	2 Teaspoon	25
7	Vegetable	1 cup	14
8	Jellies	1 Teaspoon	13
9	table blends sugar	2 Teaspoon	29
10	Cane Syrup	1 Teaspoon	13

Top 10 Carbs in Beverages.

In [91]:

```
bv = ['DrinksAlcohol Beverages','Fats Oils Shortenings']

carbbv = df [df ['Category'].isin (bv)]

carb_rich_bv = carbbv.sort_values (by='Carbs',ascending=False)

top_20 = carb_rich_bv.head(17)

fig = px.bar (top_20, x='Food', y='Protein', color='Carbs', title= ' Top 10 Carbs in Be
verages')
fig.show()
```



Top 10 Carbs foods in Beverages in India			
Sr.No	Name	Serving Size in Grams	Carbs Value
1	Butter	1/2 cup	118
2	Fruit-flavored soda	300 Ml or 1 Bottle	42
3	Cola drinks	300 Ml or 1 Bottle	38
4	Root beer	300 Ml or 1 Bottle	35
5	Ginger ale	300 Ml or 1 Bottle	28
6	Wines	1/2 cup	9
7	Coffee	1 cup	1
8	Tea	1 cup	1
9	Safflower seed oil	1 Teaspoon	0
10	Olive oil	1 Teaspoon	0

Saturated Fat are not consider to be Good, but they are require on daily basis in small quantity, lets see what are top most Saturated Fat foods in all category.

In [100]:

```
Saturated= df.sort_values(by='Sat.Fat', ascending= False)
top_20_Sat_fat=Saturated.head(20)
fig = px.bar(top_20_Sat_fat, x='Food', y='Calories', color='Calories', title=' SatFat C
ontent and Calories')
fig.show()
```

Analysing Categories, Grouping the Data into categories so that we can get total counts on all the Nutrients.

In [101]:

```
cat_dis = df.groupby (['Category']).sum()
cat_dis
```

Out[101]:

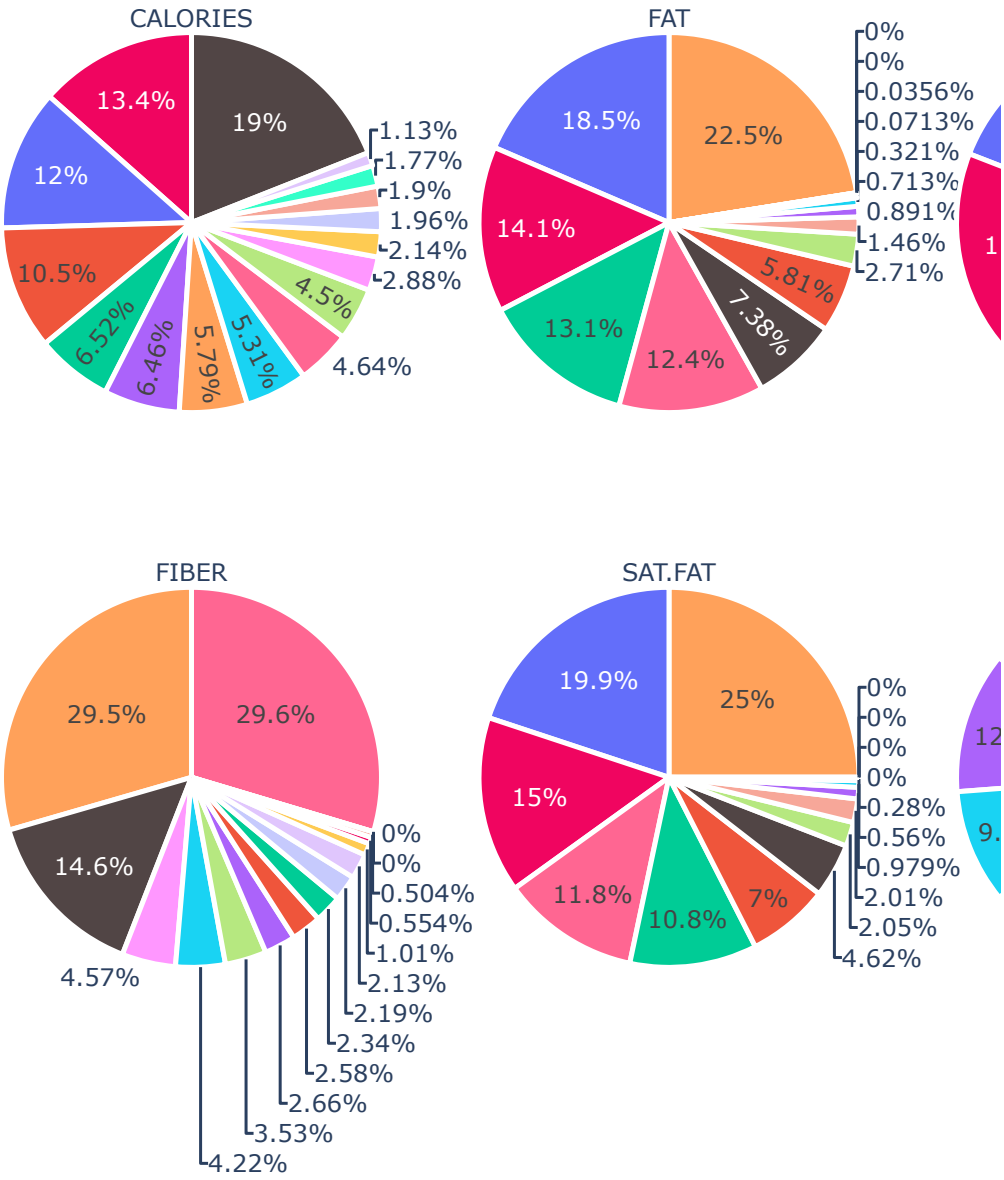
Category	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs
Breads cereals fastfoodgrains	5253	11921.0	403	207	99.0	115.91	2059.0
Dairy products	7412	8434.0	503	396	322.0	4.40	651.0
Desserts sweets	2958	6608.0	78	163	150.0	20.50	1184.0
DrinksAlcohol Beverages	3284	1112.0	0	0	0.0	0.00	167.0
Fats Oils Shortenings	695	3629.0	234	631	536.0	234.00	239.0
Fish Seafood	1892	2912.0	606	347	252.0	235.20	263.0
Fruits A-F	3844	3328.0	29	20	12.0	33.50	812.0
Fruits G-P	5412	4054.0	28	25	21.0	21.10	1009.0
Fruits R-Z	1973	1228.0	7	1	0.0	17.40	330.0
Jams Jellies	422	1345.0	0	0	0.0	8.00	345.0
Meat Poultry	2724	7529.0	546	520	427.0	0.00	57.3
Seeds and Nuts	682	4089.0	120	368	232.0	18.60	140.0
Soups	2495	1191.0	59	41	43.0	4.00	155.0
Vegetables A-E	3520	1804.0	101	9	6.0	36.30	356.0
Vegetables F-P	1725	711.0	40	2	0.0	16.90	142.0
Vegetables R-Z	3460	2825.0	103	76	44.0	28.00	459.0

We will visualise this information.

In [102]:

```
cat_dis = df.groupby (['Category']).sum()
from plotly.subplots import make_subplots
import plotly.graph_objects as go

fig = make_subplots(
    rows=2, cols=3,
    specs=[[{"type": "domain"},{"type": "domain"},{"type": "domain"}],[{"type": "domai
n"},{"type": "domain"},{"type": "domain"}]])

fig.add_trace(go.Pie(values=cat_dis['Calories'].values, title='CALORIES', labels=cat_di
s.index,marker=dict(colors=['#100b','#f00560'], line=dict(color='#FFFFFF', width=2.5
))),
              row=1, col=1)

fig.add_trace(go.Pie(values=cat_dis['Fat'].values,title='FAT', labels=cat_dis.index,mar
ker=dict(colors=['#100b','#f00560'], line=dict(color='#FFFFFF', width=2.5))),
              row=1, col=2)

fig.add_trace(go.Pie(values=cat_dis['Protein'].values,title='PROTEIN', labels=cat_dis.i
ndex,marker=dict(colors=['#100b','#f00560'], line=dict(color='#FFFFFF', width=2.5))),
              row=1, col=3)

fig.add_trace(go.Pie(values=cat_dis['Fiber'].values,title='FIBER', labels=cat_dis.index
,marker=dict(colors=['#100b','#f00560'], line=dict(color='#FFFFFF', width=2.5))),
              row=2, col=1)

fig.add_trace(go.Pie(values=cat_dis['Sat.Fat'].values,title='SAT.FAT', labels=cat_dis.i
ndex,marker=dict(colors=['#100b','#f00560'], line=dict(color='#FFFFFF', width=2.5))),
              row=2, col=2)

fig.add_trace(go.Pie(values=cat_dis['Carbs'].values,title='CARBS', labels=cat_dis.index
,marker=dict(colors=['#100b','#f00560'], line=dict(color='#FFFFFF', width=2.5))),
              row=2, col=3)
fig.update_layout(title_text="Category wise distribution of all metrics",height=700, wi
dth=1000)

fig.show()
```

Category wise distribution of all metrics



Inferences from the Above plots.

1: Highest Protein is in the Sea Food as expected followed by Meat and Poultry and Dairy Products (milk,whole wheat etc).

2: Highest Carbs (Simple) are in Breads Cereals Fastfoodgrains followed by Desert sweets (all are simple carbs) and then Fruits.

3: Highest Fats found in Fats oils shortneings (example:Fish oil) followed by Milk and Poultry and Dairy Products.

4: Highest SatFats found in Fats oils shortneings (example:Fish oil) followed by Milk and Poultry and Dairy Products.

5: Highest Fiber rich food are Seafood and Fats oil.

6: Highest Calories are in Breads Cereals Fastfoodgrains followed by Dairy products and Meat and Poultry.

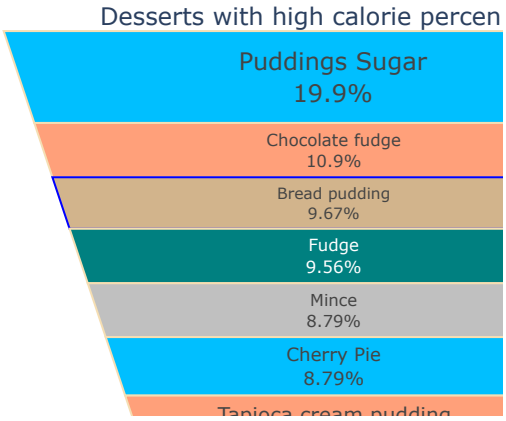
It has been observed that Dairy products are almost in every category which makes them good input for taking all the Micro and Macro nutrients but every one should consider this based on which dairy product they consume.

In [ ]:

Analysing Drinks,Beverages,Alcohol and Deserts.

In [103]:

```
drinks= df[df['Category'].isin(['Fish Seafood','Desserts sweets'])]
drinks_top=drinks.sort_values(by='Calories', ascending= False)
drinks_top=drinks_top.head(10)

fig = go.Figure(go.Funnelarea(values=drinks_top['Calories'].values, text=drinks_top['Fo
od'],
                              title = { "text": "Desserts with high calorie percentage
s"},
                marker = {"colors": ["deepskyblue", "lightsalmon", "tan", "teal", "silve
r","deepskyblue", "lightsalmon", "tan", "teal", "silver"],
                 "line": {"color": ["wheat", "wheat", "blue", "wheat", "wheat","wheat",
"wheat", "blue", "wheat", "wheat"]}}))



fig.show()
```

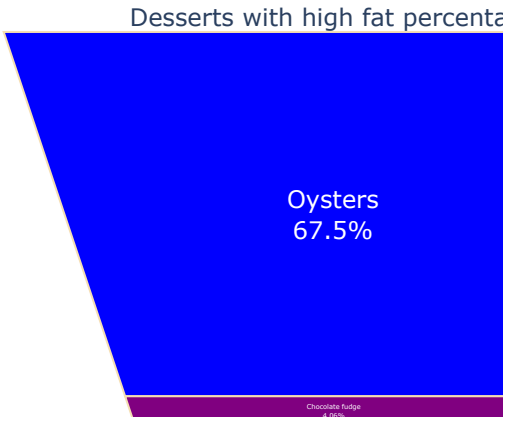


Imferences:

Puddings Sugar has more calories followed by Chocolate and Bread.

In [104]:

```
drinks_fatty=drinks.sort_values(by='Fat', ascending= False)
drinks_fatty=drinks_fatty.head(10)

fig = go.Figure(go.Funnelarea(values=drinks_fatty['Fat'].values, text=drinks_fatty['Food'],
                              title = { "text": "Desserts with high fat percentage"},
                marker = {"colors": ["blue", "purple", "pink", "teal", "silver","yellow", "lightsalmon", "tan", "teal", "silver"],
                 "line": {"color": ["wheat", "wheat", "blue", "wheat", "wheat","wheat", "wheat", "blue", "wheat", "wheat"]}}))
fig.show()
```



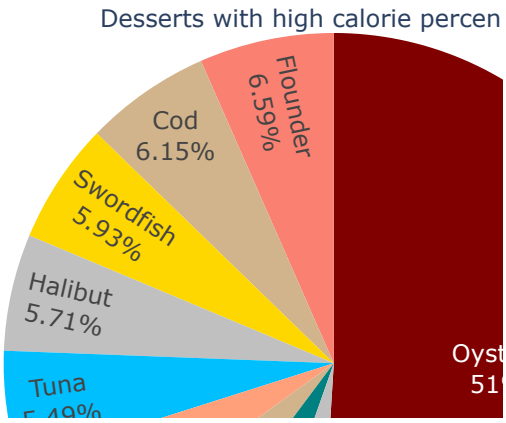
Inferences:

Pies and fudges have the highest percentage for fat.

Analysing Meat,Poulty and SeaFood.

In [106]:

```
meat= df[df['Category'].isin(['Fish Seafood','Meat Poultry'])]
meats_top=drinks.sort_values(by='Protein', ascending= False)
meats_top=meats_top.head(10)

fig = go.Figure(go.Pie(values=meats_top['Protein'].values, text=meats_top['Food'],
                               title = { "text": "Desserts with high calorie percentage
s"},
                marker = {"colors": ["maroon", "salmon", "tan", "gold", "silver","deepsk
yblue", "lightsalmon", "tan", "teal", "silver"],
                 "line": {"color": ["wheat", "wheat", "blue", "wheat", "wheat","wheat",
"wheat", "blue", "wheat", "wheat"]}}))
fig.show()
```



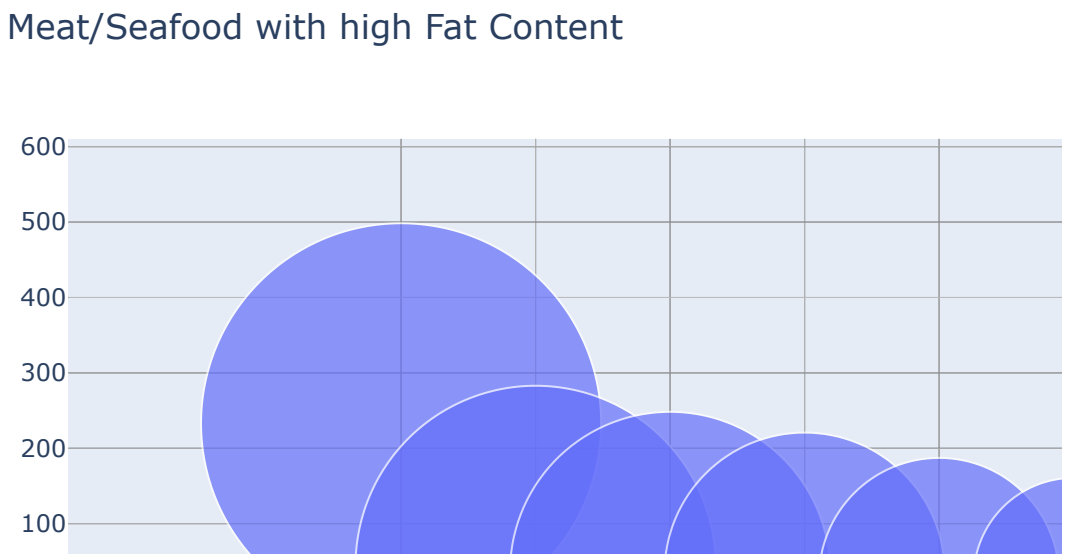
Inferences:

Oysters have large amount of protein followed by flounder.

Which Fish has the high amount of Fat (Good Fat).

In [107]:

```
top_10_fattest= meat.sort_values(by='Fat', ascending=False)
top_10_fattest=top_10_fattest.head(10)
fig = go.Figure(data=[go.Scatter(
    x=top_10_fattest['Food'], y=top_10_fattest['Fat'],
    mode='markers',
    marker_size=[200,180,160,140,120, 100 ,80 , 60 ,40,20])
])
fig.update_layout(title='Meat/Seafood with high Fat Content')
fig.show()
```



Inferences:

Oysters have the most number of good fat followed by pork and roast beef.

Meat with Most Fiber.

In [108]:

```
top_10_fibrous= meat.sort_values(by='Fiber', ascending=False)
top_10_fibrous=top_10_fibrous.head(10)
top_10_fibrous
```

Out[108]:

	Food	Measure	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs	Category
82	Oysters	6-8 med.	230	231.0	232	233	234.0	235.0	236.0	Fish Seafood
81	Mackerel	3 oz.	85	155.0	18	9	0.0	0.2	0.0	Fish Seafood
43	Bacon	2 slices	16	95.0	4	8	7.0	0.0	1.0	Meat Poultry
78	Halibut	3 1/2 oz.	100	182.0	26	8	0.0	0.0	0.0	Fish Seafood
69	Turkey	3 1/2 oz.	100	265.0	27	15	0.0	0.0	0.0	Meat Poultry
70	Veal	3 oz.	85	185.0	23	9	8.0	0.0	0.0	Meat Poultry
71	Roast	3 oz.	85	305.0	13	14	13.0	0.0	0.0	Meat Poultry
72	Clams	3 oz.	85	87.0	12	1	0.0	0.0	2.0	Fish Seafood
73	Cod	3 1/2 oz.	100	170.0	28	5	0.0	0.0	0.0	Fish Seafood
74	Crab meat	3 oz.	85	90.0	14	2	0.0	0.0	1.0	Fish Seafood

Inferences:

Oysters have high amount of Fiber too followed by Mackerel.

In India its difficult to find this types of Food hence we will be taking the most common foods which are easily available with good amount of Nutrients and Minerals.

In [ ]:

```

```