**##########################################################**

**# CST8333 2018 Final Project #**

**# Created by Jay Italia #**

**# November 26 ,2018 #**

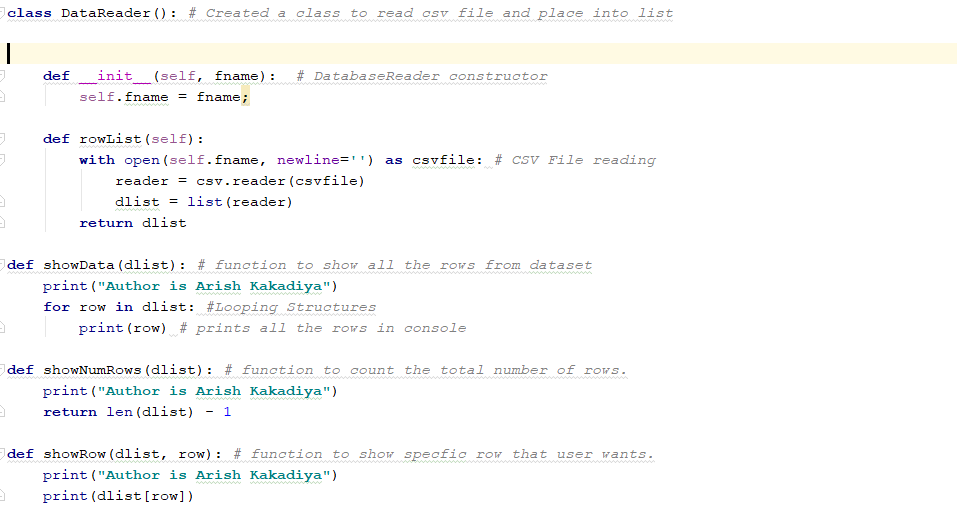
**# #**

**##########################################################**

I chose Python as the language I wanted to research, and my project is centered around an 32100054.csv food database. Which is having Food data for Canada Geo location and have Food categories, Commodity UOM, UOM\_ID etc. as column data.

I have implemented Feature on the given data base are listed below:

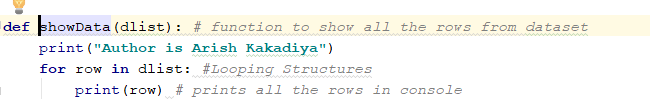
*Created a class to read csv file and place into list*



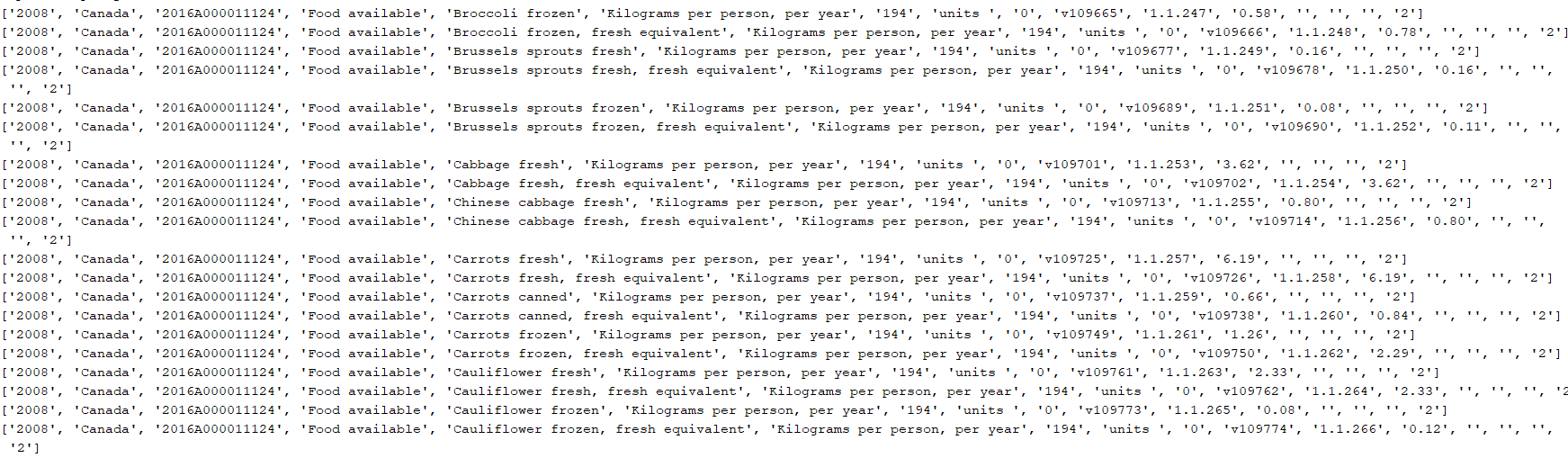
**Feacture & Functions are listed below:**

showData(dList):

To show the all the rows and column data I have used For Loop on RowList and Panda DataFrame to display output data in terminal as shown below:



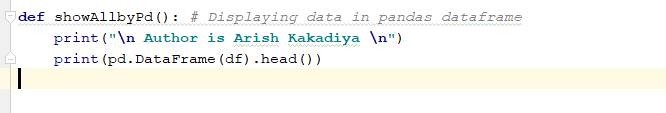
And output for this code is :



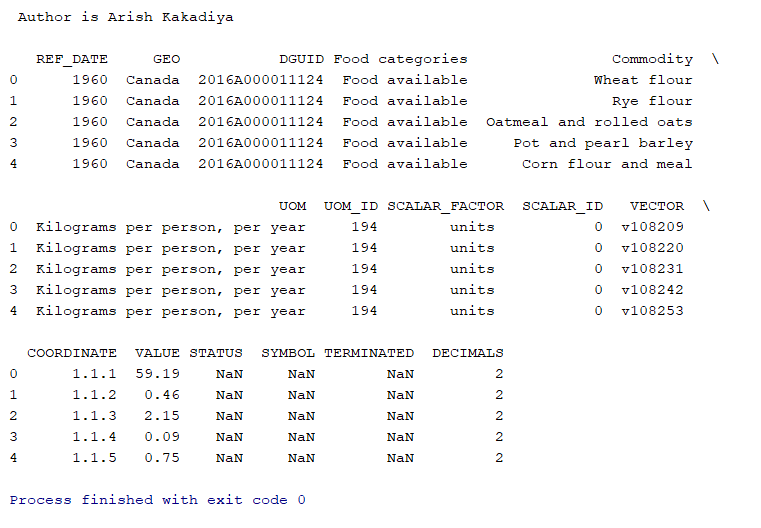
Here Data is displayed using pandas DataFrame:

Head will print 5 rows by default

(Ref. <https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.head.html>)

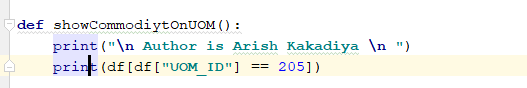


And output for this code this:

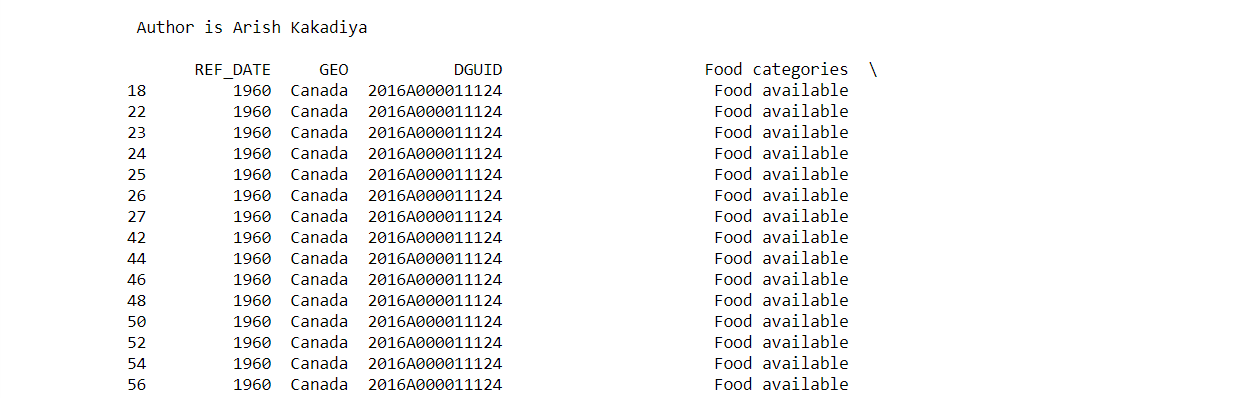


**showCommodiytOnUOM()** # Function for Showing all rows Commodity based on UOM

Here I have displayed Commodity data on given UOM\_ID which is done using Pandas Data Frame.



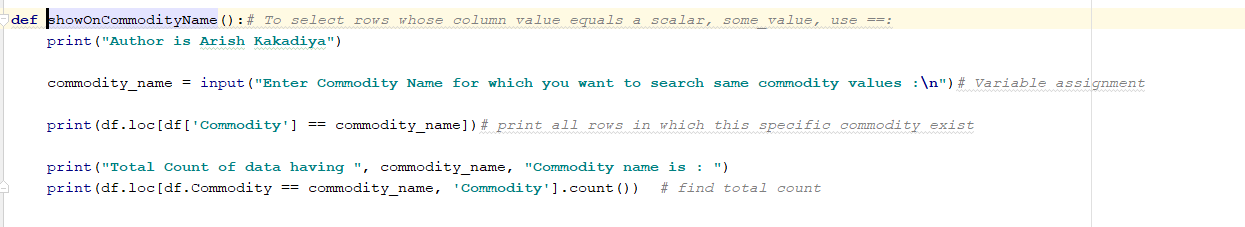
Output:



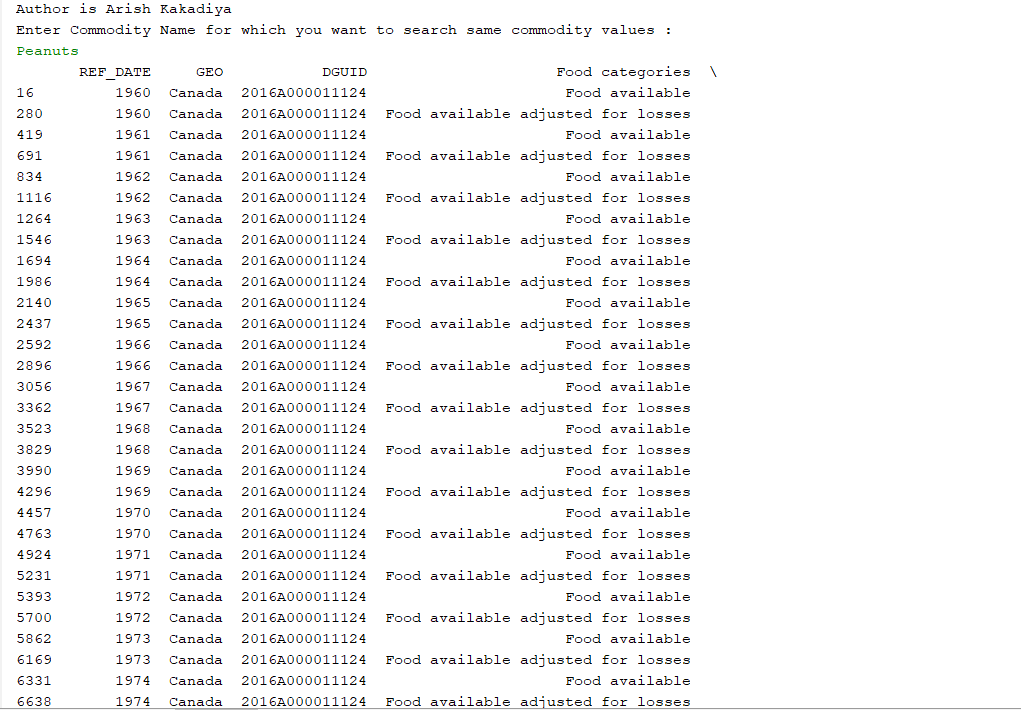
**showOnCommodityName()***#Function for Showing all rows having specific commodity name*

To show data Rows for given specific column values in input which was taken as input using Input function and Also have shown Total count of data rows which is having Specific commodity name which was done using the loc indexer for Pandas Dataframe is used for integer-location based indexing / selection by position.

Ref.( <https://pandas.pydata.org/pandas-docs/version/0.23/generated/pandas.DataFrame.loc.html> )



Output:



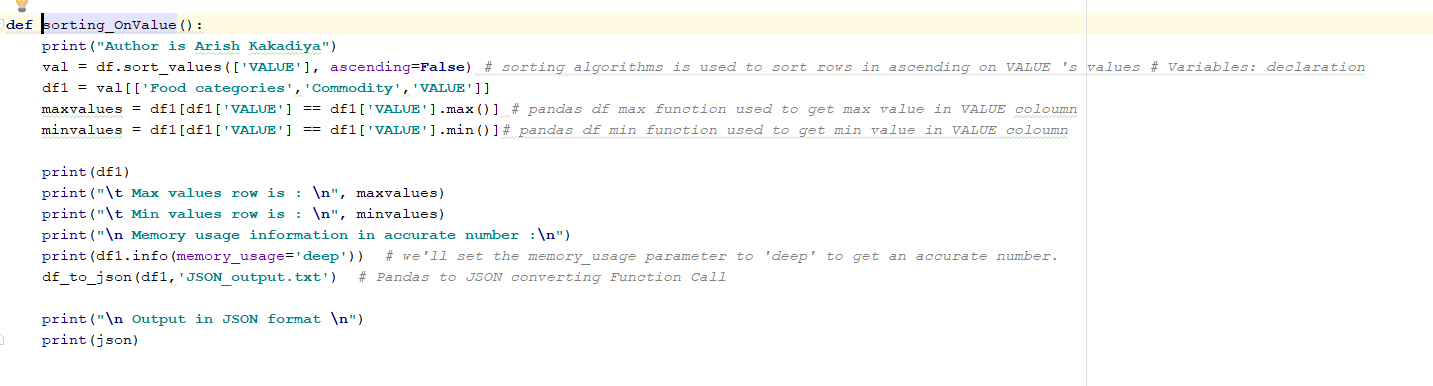


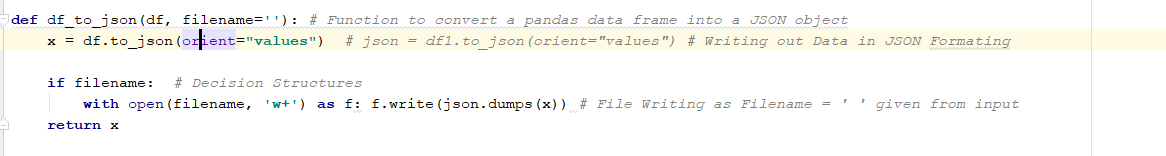
**sorting\_OnValue()** *# function for sorting Values in ascending or descending order*

This function is used to sort the given dataset based on VALUE’s data value to sort I used sort function and calculated Max and Min data Values of VALUE column using Max and Min function.

And to show Memory Usage during execution, I used **df1.info(memory\_usage='deep')** which gives summary of a Data-Frame and returns None. And shown Sorted data in JSON format by converting csv to JSON.

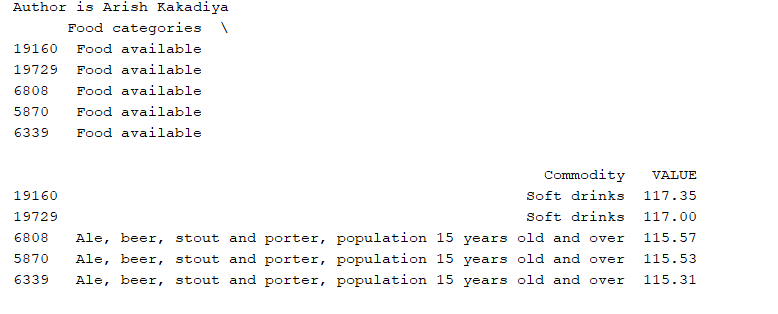
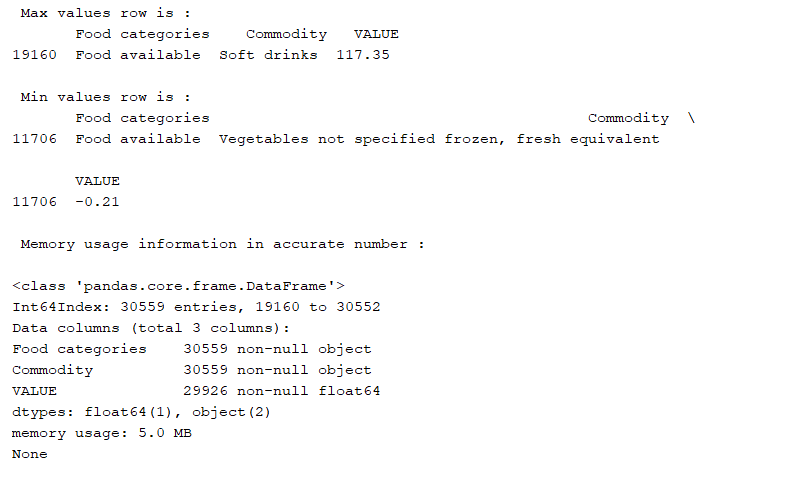
Ref.( <https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.info.html> )

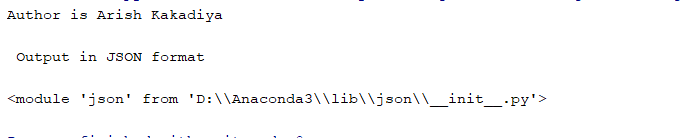




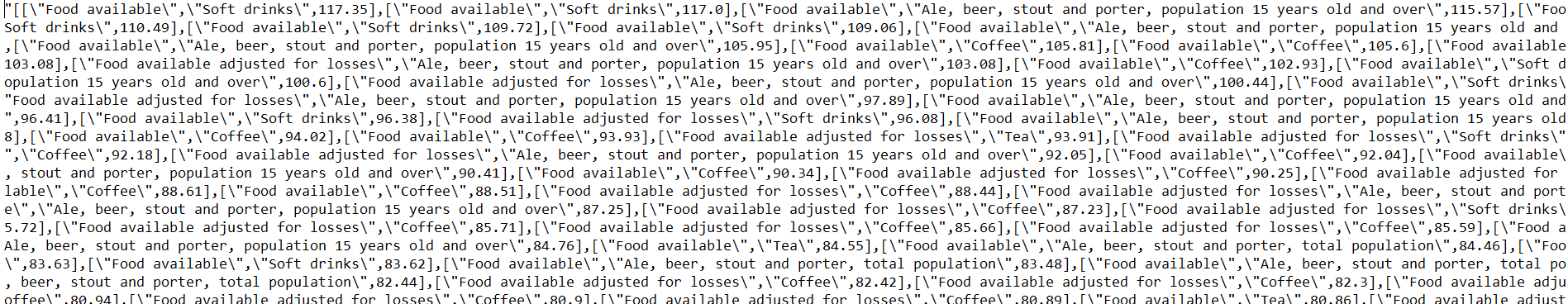
Output:

Output in sorted order (Descending order )

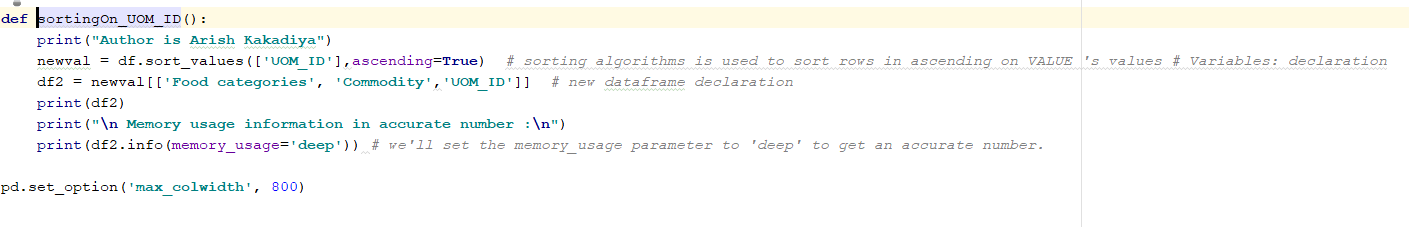


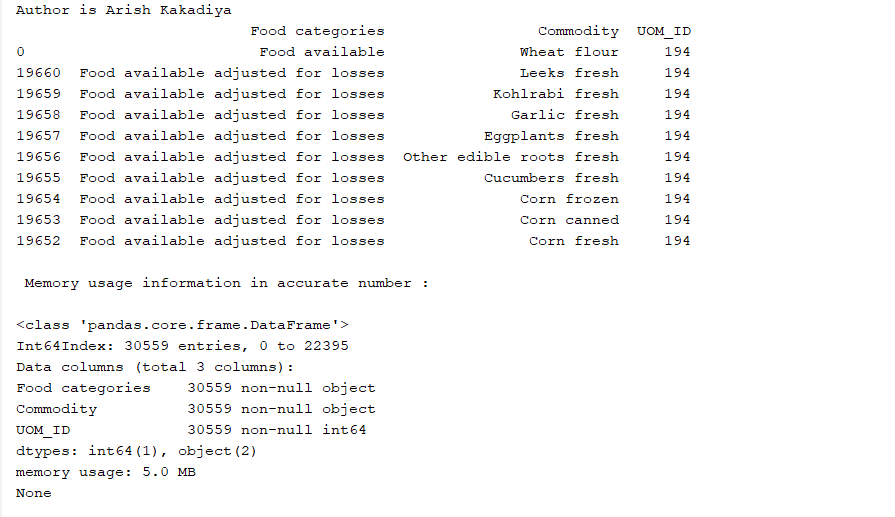
Output JSON file is created in file directory and Data In JSON format will be like :



**sortingOn\_UOM\_ID()*#*** *function for sorting Values in ascending or descending order*

This function is used to sort the given dataset based on UOM\_ID data value to sort I used sort function And to show Memory Usage during execution, I used **df2.info(memory\_usage='deep')** which gives summary of a Data-Frame and returns None. And shown Sorted data in JSON format by converting csv to JSON.



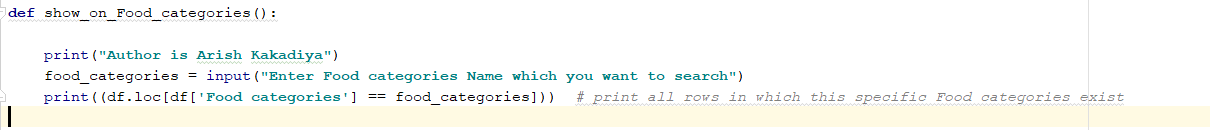


**show\_on\_Food\_categories()**

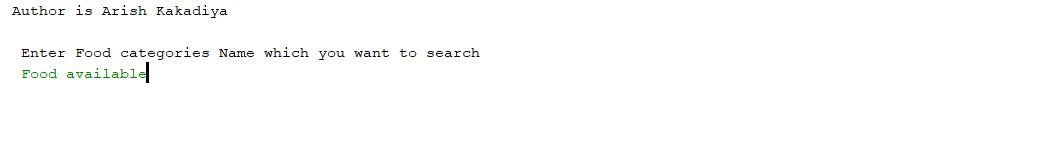
#function for showing all rows having specific food category

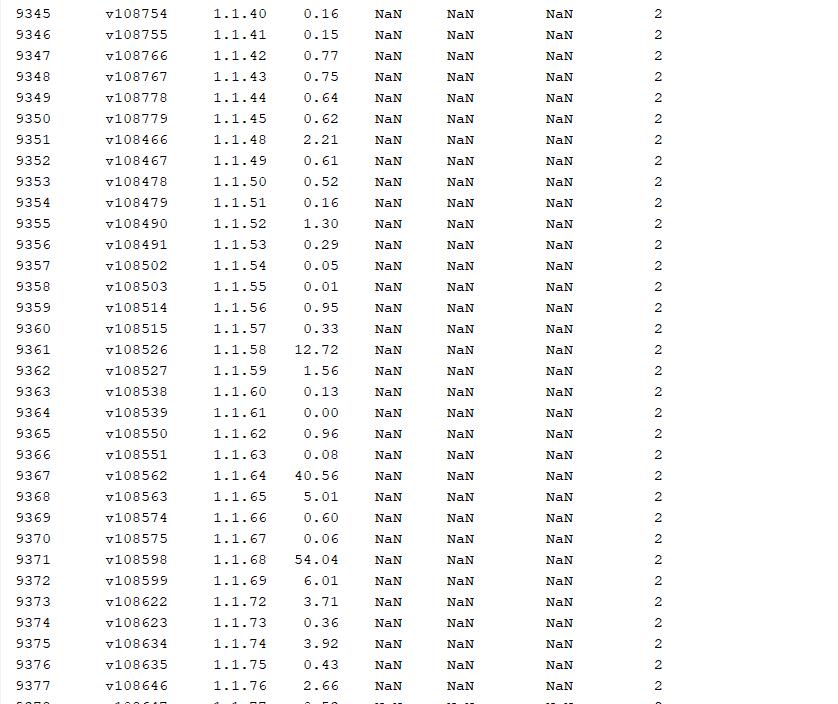
To display all the row having Specific searched Food Category , I used loc to select rows and column in Pandas Dataframe.

Ref.( <https://pandas.pydata.org/pandas-docs/version/0.23/generated/pandas.DataFrame.loc.html> )



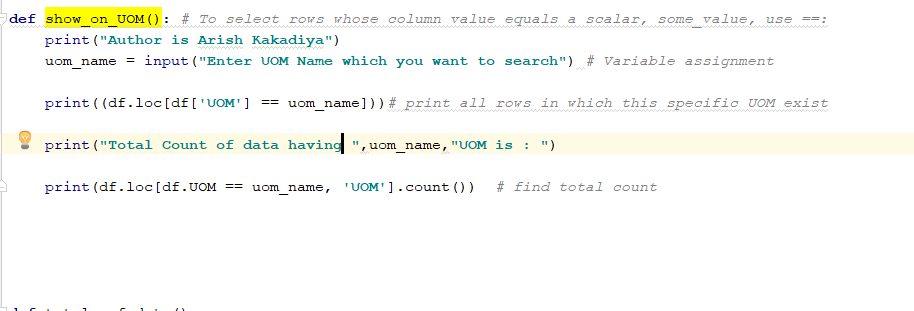
Output:





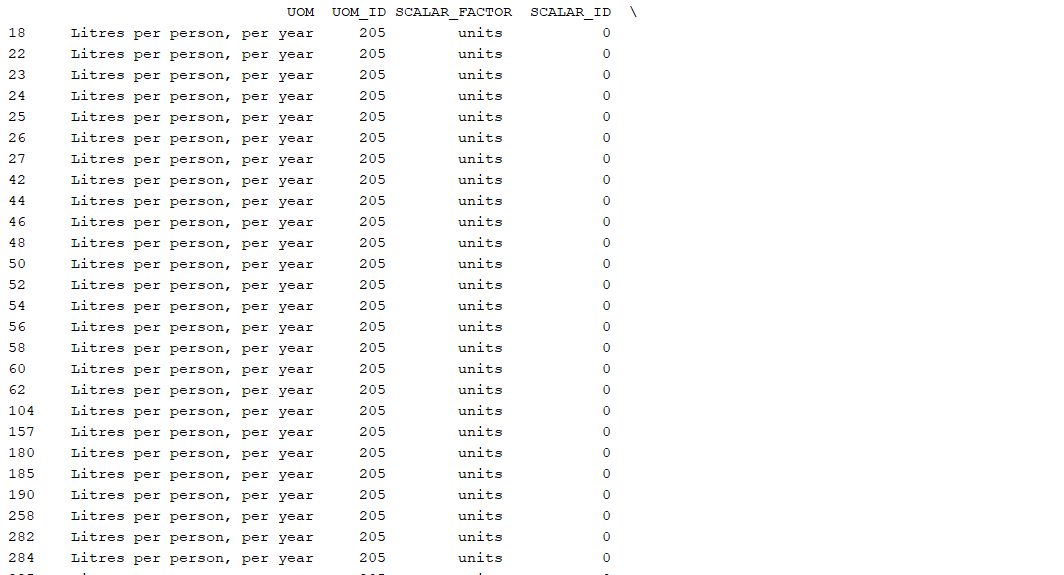
show\_on\_UOM()

To display data on searched UOM name and selected rows , column using .loc and counted total such rows using count function .



Output:





***# Multithreading to execute two given process***

*Multithreading is a way of achieving multitasking. In multithreading, the concept of threads is used.*

*To create a new thread, I create an object of Thread class. It takes following arguments:*

*target: the function to be executed by thread*

*args: the arguments to be passed to the target function*

*I created 2 threads with different target functions*

t1 = threading.Thread(target=sorting\_OnValue)  
t2 = threading.Thread(target=sortingOn\_UOM\_ID)

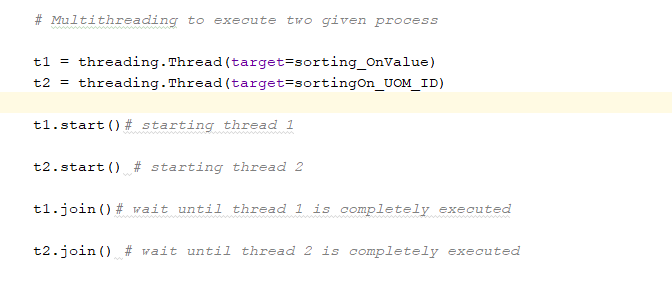
To start a thread, I will use start method of Thread class.

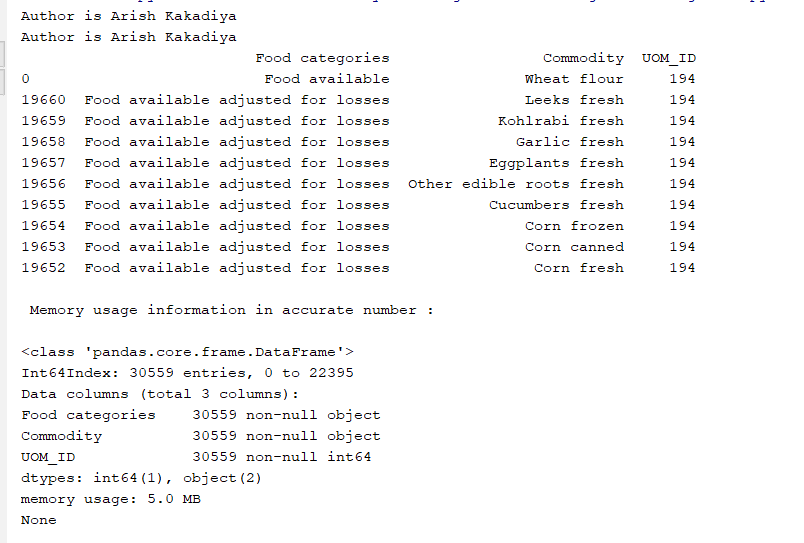
t1.start()*# starting thread 1*t2.start() *# starting thread 2*

*Once the threads start, the current program (you can think of it like a main thread) also keeps on executing. In order to stop execution of current program until a thread is complete, I use join method.*

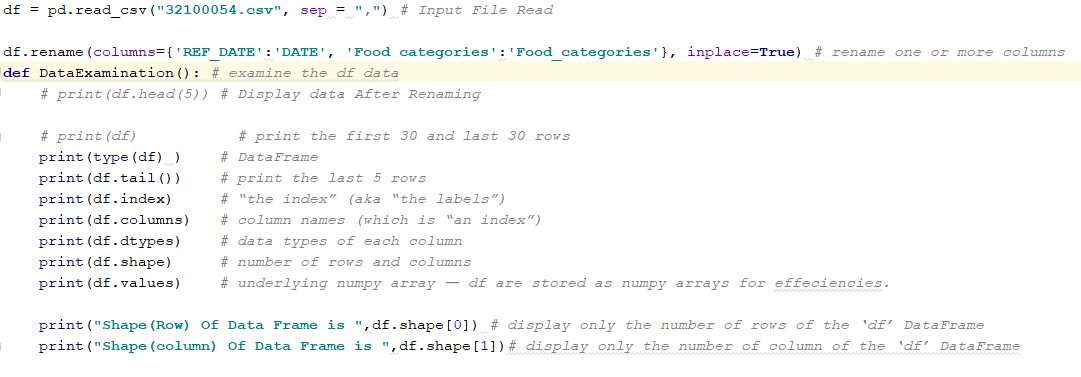
*t1.join()*

*t2.join()*





DataExamination() # Method to examine the DataFrame Dataset



End

**#####################################################################################**