In [32]:
```
import pandas as pd
import numpy as np

from numpy.random import randn
```

In [33]:
```
# Because, i want to Check wheather the 'plotly' has been 'imported' or 'not' :

from plotly import __version__
```

In [34]:
```
# So, i want to Verify wheather the 'Cufflinks' also 'exist' or 'not' :

import cufflinks as cf
```

In [35]:
```
init_notebook_mode(connected = True)
```

In [39]:
```
cf.go_offline()
```

1. Now particular Data Frame :

page 02 BOOK 4

In [65]:
```
df = pd.DataFrame(np.random.randn(200,4), columns = 'A B C D'.split())
df.head()
```

Out[65]:

	A	B	C	D
0	-0.201626	3.258483	0.655303	1.109467
1	1.330618	-0.867787	0.328051	1.260478
2	1.085319	0.493355	0.117959	2.229773
3	-0.838138	-0.638371	0.183775	1.943851
4	-0.213938	0.587816	1.366488	0.993348

In [71]:
```
df
```

Out[71]:

	A	B	C	D
0	-0.201626	3.258483	0.655303	1.109467
1	1.330618	-0.867787	0.328051	1.260478
2	1.085319	0.493355	0.117959	2.229773
3	-0.838138	-0.638371	0.183775	1.943851
4	-0.213938	0.587816	1.366488	0.993348
...	...	...	...	...
195	0.820415	-1.329971	0.629896	0.107500
196	-0.913967	0.141321	-0.697182	0.261392
197	-1.020456	-0.552893	0.509566	0.320884
198	-0.306798	-0.646119	-1.253757	0.084196
199	1.642455	-0.649336	0.046080	0.323306

200 rows × 4 columns

2. Now Second particular DataFrame :

page 03 book 4

In [52]:

```
# category and the values where [32,43,50]

df2 = pd.DataFrame({'category':['A','B','C'], 'values': [32,43,50]})
df2
```

Out[52]:

	category	values
0	A	32
1	B	43
2	C	50

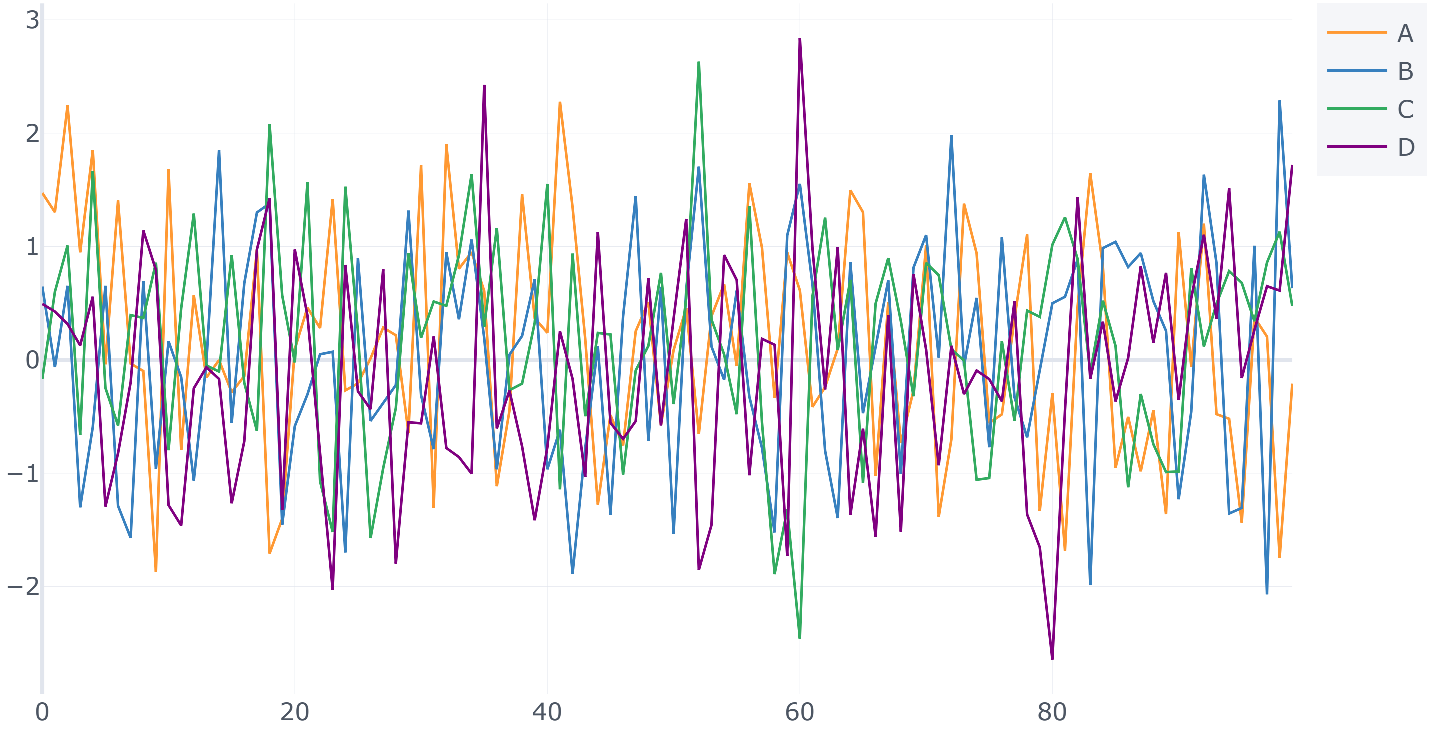
3. iplot( ) : Advanced Library iplotting in plotting methods( ) : vvimp

page 03 book 4

In [69]:

```
# Here, i didn't mentioned any color combination for the Variables, data and everything it has been.
# and didn't Declared parameters also : it's an 'Very Highend Plotting Point'
# Whenever We give the 'Cursor' - We are getting 'Data Point' exactly with an a 'Color Combination'.
```

In [54]:
```
df.iplot()
```

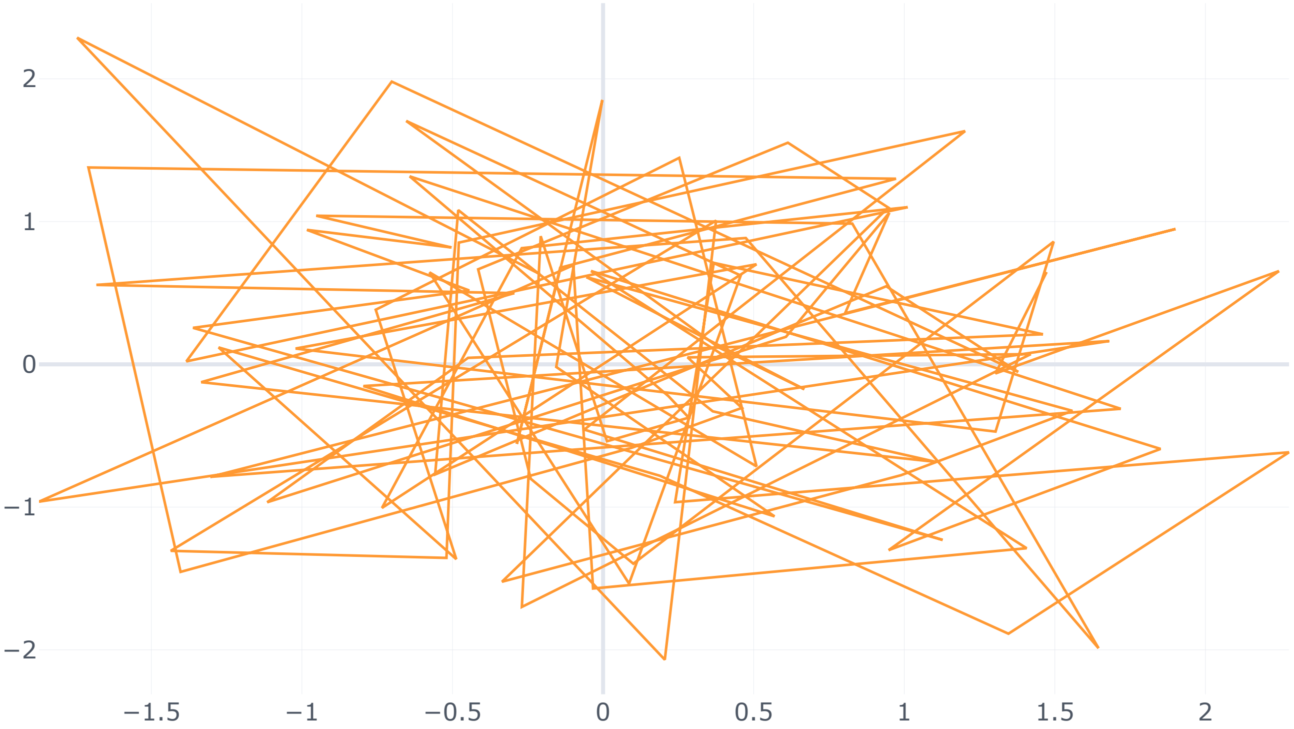


Export to plot.ly »

4. SCATTER WORKING ENVIRONMENT :

PAGE 04 BOOK 04

```
In [57]: df.iplot(kind = 'scatter', x = 'A', y = 'B')
```



Export to plot.ly »

5. Another DataFrame :

In [60]:

```
df3 = pd.DataFrame({'x':[1,2,3,4,5], 'y':[10,20,30,40,50], 'z':[5,4,3,2,1]})
df3
```

Out[60]:

	x	y	z
0	1	10	5
1	2	20	4
2	3	30	3
3	4	40	2
4	5	50	1

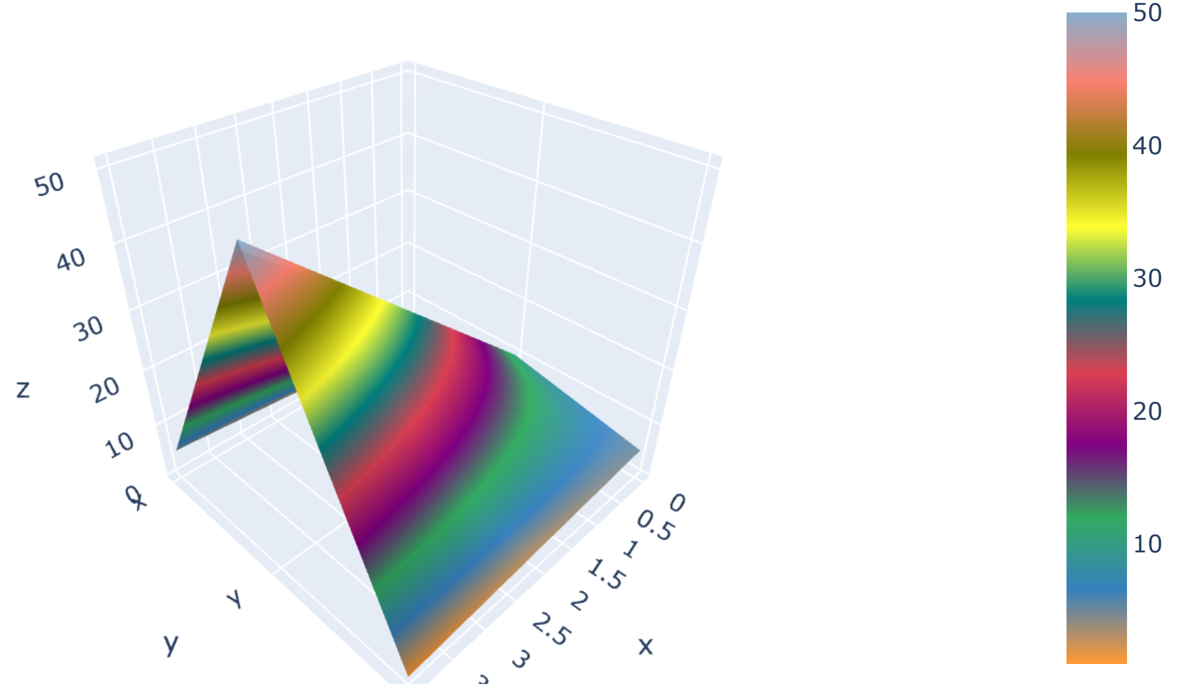
6. '3D' REPRESENTATION USING 'iplot' : Above sum values :

page 05 book 4

In [70]:

```
# This is an a '3D' REPRESENTATION OF DATA :
# Here, just i have taken '3' Dimension Data, But Everythin has been Done very perfectly by
# 'iplot' network itself.
```

In [75]:
```
df3.iplot(kind = 'surface')
```



Export to plot.ly »

7. Another Example on '3D' REPRESENTATION : Using previously created 'df' Data :

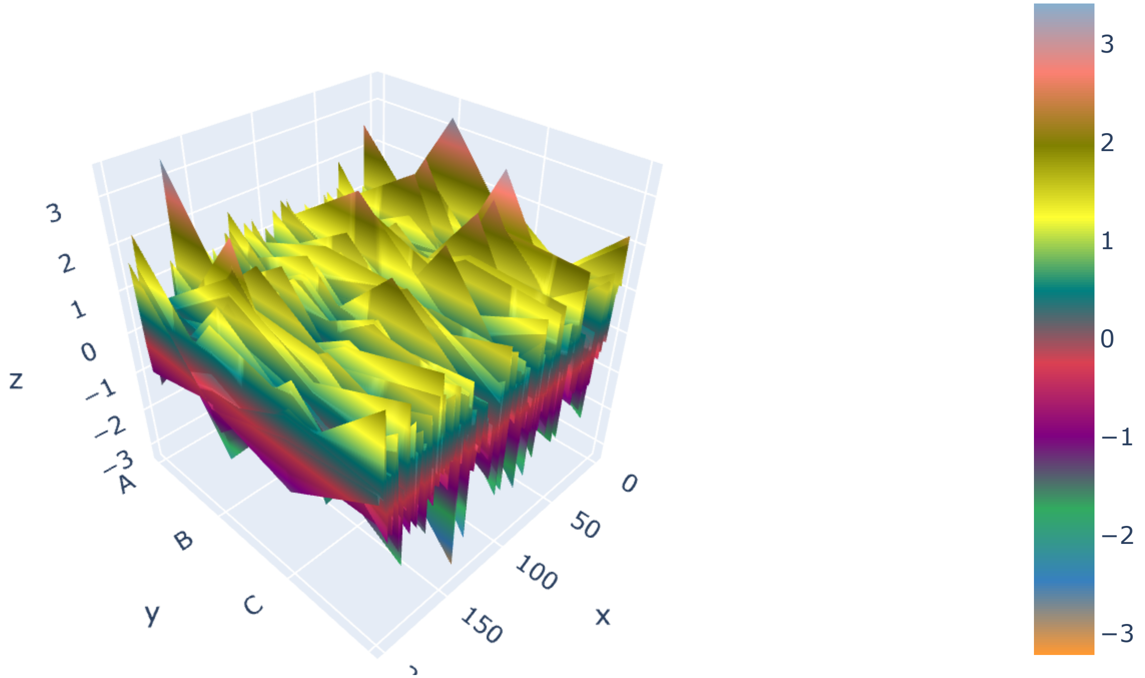
In [73]:
```
df
```

Out[73]:

	A	B	C	D
0	-0.201626	3.258483	0.655303	1.109467
1	1.330618	-0.867787	0.328051	1.260478
2	1.085319	0.493355	0.117959	2.229773
3	-0.838138	-0.638371	0.183775	1.943851
4	-0.213938	0.587816	1.366488	0.993348
...	...	...	...	...
195	0.820415	-1.329971	0.629896	0.107500
196	-0.913967	0.141321	-0.697182	0.261392
197	-1.020456	-0.552893	0.509566	0.320884
198	-0.306798	-0.646119	-1.253757	0.084196
199	1.642455	-0.649336	0.046080	0.323306

200 rows × 4 columns

```
In [72]: df.iplot(kind = 'surface')
```



Export to plot.ly »

8. TEST CASES :

PAGE 05 BOOK 04

SOME PARTICULAR QUESTIONS AND WORKING ENVIRONMENT :

In [76]:

```
# STEP 1 : 'chipotle.tsv' upload - tsv -> 'The Tab Seperated Value'
```

In [78]:

```
# Now, let's call the required libraries to our working environment :
# i have done with an all the Libraries, which i required for the 'Machine Learning' :
```

In [79]:

```
# STEP 2 : import required liberaries :

import pandas as pd
import numpy as np
```

STEP 3 : ASSIGN it to a VARIABLE Called Chipo :

page 06

In [81]:
```
chipo = pd.read_csv("19.10.chipotle.tsv", sep = '\t')
chipo
```

Out[81]:

	order_id	quantity	item_name	choice_description	item_price
0	1	1	Chips and Fresh Tomato Salsa	NaN	$2.39
1	1	1	Izze	[Clementine]	$3.39
2	1	1	Nantucket Nectar	[Apple]	$3.39
3	1	1	Chips and Tomatillo-Green Chili Salsa	NaN	$2.39
4	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	$16.98
...	...	...	...	...	...
4617	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Sour ...	$11.75
4618	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Sour Cream, Cheese...	$11.75
4619	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	$11.25
4620	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Lettu...	$8.75
4621	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	$8.75

4622 rows × 5 columns

STEP 4 : SEE THE FIRST '10' ENTRIES :

PAGE 07 BOOK 4

In [83]:

```python
chipo.head(10)
```

Out[83]:

	order_id	quantity	item_name	choice_description	item_price
0	1	1	Chips and Fresh Tomato Salsa	NaN	$2.39
1	1	1	Izze	[Clementine]	$3.39
2	1	1	Nantucket Nectar	[Apple]	$3.39
3	1	1	Chips and Tomatillo-Green Chili Salsa	NaN	$2.39
4	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	$16.98
5	3	1	Chicken Bowl	[Fresh Tomato Salsa (Mild), [Rice, Cheese, Sou...	$10.98
6	3	1	Side of Chips	NaN	$1.69
7	4	1	Steak Burrito	[Tomatillo Red Chili Salsa, [Fajita Vegetables...	$11.75
8	4	1	Steak Soft Tacos	[Tomatillo Green Chili Salsa, [Pinto Beans, Ch...	$9.25
9	5	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Pinto...	$9.25

STEP 5 : WHAT IS THE NUMBER OF OBSERVATIONS IN THE DATA SET ?

PAGE 07 BOOK 04

(A) SOLUTION -1

In [129]:

```python
chipo.shape[0]
```

Out[129]:

```
4622
```

(B) SOLUTION -2 : with names

In [130]:
```
chipo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4622 entries, 0 to 4621
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   order_id            4622 non-null   int64  
 1   quantity            4622 non-null   int64  
 2   item_name           4622 non-null   object 
 3   choice_description  3376 non-null   object 
 4   item_price          4622 non-null   float64
dtypes: float64(1), int64(2), object(2)
memory usage: 180.7+ KB
```

STEP 6 : WHAT IS THE NUMBER OF COLUMNS IN THE DATASET :

PAGE 07 BOOK 04

In [131]:
```
chipo.columns
```

Out[131]:
```
Index(['order_id', 'quantity', 'item_name', 'choice_description',
       'item_price'],
      dtype='object')
```

In [88]:
```
# STEP 7 : PRINT THE NAMES OF THE COLUMNS : CHECK STEP 5 SOLUTION 2
```

STEP 8 : HOW IS THE DATASET INDEXED :

PAGE 08 BOOK 4

In [132]:

```
chipo.index
```

Out[132]:

```
RangeIndex(start=0, stop=4622, step=1)
```

STEP 9 : WHICH WAS THE MOST-ORDERED ITEM : USING (GROUP BY) :

In [133]:

```
c = chipo.groupby('item_name')
c = c.sum()
c = c.sort_values(['quantity'], ascending = False)
c.head()
```

Out[133]:

	order_id	quantity	item_price
item_name			
Chicken Bowl	713926	761	7342.73
Chicken Burrito	497303	591	5575.82
Chips and Guacamole	449959	506	2201.04
Steak Burrito	328437	386	3851.43
Canned Soft Drink	304753	351	438.75

In [134]:

```
c.head(1)
```

Out[134]:

	order_id	quantity	item_price
item_name			
Chicken Bowl	713926	761	7342.73

STEP 10 : FOR THE MOST ORDERED ITEM, HOW MANY ITEMS WHERE ORDERED :

PAGE 08 BOOK 04

In [135]:
```
# Same step 9 question, diffently asked :

c = chipo.groupby('item_name')
c = c.sum()
c = c.sort_values(['quantity'], ascending = False)
c.head()
```

Out[135]:

	order_id	quantity	item_price
item_name			
Chicken Bowl	713926	761	7342.73
Chicken Burrito	497303	591	5575.82
Chips and Guacamole	449959	506	2201.04
Steak Burrito	328437	386	3851.43
Canned Soft Drink	304753	351	438.75

STEP 11 : WHAT WAS THE MOST ORDERED ITEM IN THE CHOICE_DESCRIPTION COLUMN :

PAGE 09 BOOK 4

In [136]:

```python
c = chipo.groupby('choice_description').sum()
c.sort_values(['quantity'], ascending = False)
```

Out[136]:

	order_id	quantity	item_price
choice_description			
[Diet Coke]	123455	159	326.71
[Coke]	122752	143	288.79
[Sprite]	80426	89	133.93
[Fresh Tomato Salsa, [Rice, Black Beans, Cheese, Sour Cream, Lettuce]]	43088	49	432.25
[Fresh Tomato Salsa, [Rice, Black Beans, Cheese, Sour Cream]]	36041	42	372.64
...	...	...	...
[Roasted Chili Corn Salsa, [Fajita Vegetables, Rice, Pinto Beans, Guacamole, Lettuce]]	577	1	11.25
[Roasted Chili Corn Salsa, [Fajita Vegetables, Rice, Sour Cream, Lettuce]]	585	1	8.75
[Roasted Chili Corn Salsa, [Fajita Vegetables, Sour Cream, Lettuce, Guacamole]]	235	1	11.75
[Roasted Chili Corn Salsa, [Guacamole, Sour Cream, Rice, Fajita Vegetables, Lettuce]]	987	1	11.25
[[Tomatillo-Red Chili Salsa (Hot), Tomatillo-Green Chili Salsa (Medium)], [Rice, Pinto Beans, Fajita Veggies, Lettuce]]	1299	1	8.99

1043 rows × 3 columns

STEP 12 : HOW MANY ITEMS WHERE ORDERED IN TOTAL : TOTAL NO.OF ITEMS :

PAGE 09 BOOK 04

In [137]:

```python
total_items_ordered = chipo.quantity.sum()
total_items_ordered
```

Out[137]:

```
4972
```

STEP 13 : TURN THE ITEM PRICE INTO A 'FLOAT' :

(A) CHECK THE ITEM PRICE TYPE :

```
In [141]: chipo.item_price.dtype
```

```
Out[141]: dtype('float64')
```

(B) CREATE A 'LAMBDA' FUNCTION AND CHANGE THE TYPE OF ITEM PRICE :

In [142]:
```
dollarizer = lambda x:float(x[1:-1])
chipo.item_price = chipo.item_price.apply(dollarizer)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_8964\1103613255.py in <module>
      1 dollarizer = lambda x:float(x[1:-1])
----> 2 chipo.item_price = chipo.item_price.apply(dollarizer)

~\anaconda3\lib\site-packages\pandas\core\series.py in apply(self, func, convert_dtype, args, **kwargs)
   4431         dtype: float64
   4432         """
-> 4433         return SeriesApply(self, func, convert_dtype, args, kwargs).apply()
   4434 
   4435     def _reduce(

~\anaconda3\lib\site-packages\pandas\core\apply.py in apply(self)
   1086             return self.apply_str()
   1087 
-> 1088         return self.apply_standard()
   1089 
   1090     def agg(self):

~\anaconda3\lib\site-packages\pandas\core\apply.py in apply_standard(self)
   1141                 # List[Union[Callable[..., Any], str]]]]]"; expected
   1142                 # "Callable[[Any], Any]"
-> 1143                 mapped = lib.map_infer(
   1144                     values,
   1145                     f,  # type: ignore[arg-type]

~\anaconda3\lib\site-packages\pandas\_libs\lib.pyx in pandas._libs.lib.map_infer()

~\AppData\Local\Temp\ipykernel_8964\1103613255.py in <lambda>(x)
----> 1 dollarizer = lambda x:float(x[1:-1])
      2 chipo.item_price = chipo.item_price.apply(dollarizer)

TypeError: 'float' object is not subscriptable
```

(C) CHECK THE ITEM PRICE TYPE :

In [144]:

```
# It is changed to 'float' :

chipo.item_price.dtype
```

Out[144]:

```
dtype('float64')
```

STEP 14 : HOW MUCH WAS THE 'REVENUE' FOR THE PERIOD IN THE DATASET :

PAGE 10 BOOK 04

In [150]:

```
revenue = (chipo['quantity'] * chipo['item_price']).sum()
print("Revenue was : $", str(np.round(revenue,2)))
```

```
Revenue was : $ 39237.02
```

STEP 15 : HOW MANY ORDERS WERE MADE IN THE PERIOD :

PAGE 10 BOOK 04

In [152]:

```
# This is the particular Average revenue for a period :

chipo['revenue'] = chipo['quantity'] + chipo['item_price']
order_grouped = chipo.groupby(by = ['order_id']).sum()
order_grouped.mean()['revenue']
```

Out[152]:

```
21.522442748091617
```

STEP 16 : WHAT IS THE AVERAGE REVENUE AMOUNT FOR THE ORDER :

In [ ]:

```

```

STEP 17 : HOW MANY DIFFERENT ITEMS ARE SOLD :

PAGE 11 BOOK 04

In [153]:

```
chipo.item_name.value_counts().count()
```

Out[153]:

```
50
```

In [155]:

```
# FOR MORE TEST CASES : CHECK THE DRIVE AND PRACTICE :
```

In [ ]:

```

```