from pprint import pprint

import interface

def new\_function(row\_dict\_list, column\_type\_dict, columns, number\_of\_rows, \*\*kwargs):

"""

Given a list of row dictionaries, column names, and number of rows,

prints counts and percentages of categorical variable of columns

selected by the user.

"""

columns\_to\_analyze = []

for column in columns:

if interface.input\_is\_yes(f"Analyze {column}?", default='n'):

columns\_to\_analyze.append(column)

if columns\_to\_analyze == []:

print("No columns selected")

return

for selected\_column in columns\_to\_analyze:

print(f"{selected\_column}:")

count\_dict = {}

width = 25

for row\_dict in row\_dict\_list:

value = row\_dict[selected\_column]

width = max(width, len(str(value)))

if value in count\_dict:

count\_dict[value] += 1

else:

count\_dict[value] = 1

# print(number\_of\_rows)

# pprint(count\_dict)

for category, count in count\_dict.items():

print(f"{category:>{width}} {count:>5} {count/number\_of\_rows:>5.1%}")

print()

input("Press enter to return to main menu. ")

def get\_numerical\_columns\_list(column\_type\_dict, \*\*kwargs):

"""

Returns list of column names having numerical data types.

Will return empty list if no columns are numerical.

"""

numerical\_columns = []

numerical\_types = [int, float]

for column, data\_type in column\_type\_dict.items():

if data\_type in numerical\_types:

numerical\_columns.append(column)

return numerical\_columns

def your\_new\_function(row\_dict\_list, \*\*kwargs):

"""

TODO: Implement your new function.

Change this docstring and the code inside it.

"""

print("This is new functionality")

parameters = locals()

print("Parameters available here:")

pprint(parameters)