

```
#!/usr/bin/env python
# coding: utf-8
```

```
# In[ ]:
```

```
def enter_filename():
    filename = input("Enter name of file to open: ")
    return filename
```

```
def load_data(filename):
    import csv
    datasets = []
    try:
        with open(filename, 'r') as file_in:
            read_contents = csv.reader(file_in)
            imported_data = list(read_contents)
            file_in.close()
            print("")
            print(filename, "has been loaded.")
            return imported_data
    except FileNotFoundError:
        print("File not found.")
    except ValueError:
        print("Bad format in file")
```

```
#####
##### DISPLAY DATA #####
#####
```

```
def display_data(imported_data):
    for rows in imported_data:
        display_data= {row[0]: row[1:] for row in imported_data}
    print('')
    print('')
    i = 0
    for display_data, data in display_data.items():
        print(" ")
        print(display_data)
        length = len(display_data)
        i = 0
        while i < length:
            print('-', end=" ")
            i = i + 1
        print('')
        print(*data, sep=', ')
    return imported_data
```

```
#####
##### MENU INPUT #####
#####
```

```
def display_main_menu():
    print("")
    print("")
    print("")
    print("Please choose from the following options:")
    print("")
    1 - Load data from a file
    2 - Display the data to the screen
    3 - Rename a set
    4 - Sort a set
    5 - Analyse a set
    6 - Quit
```

```

"""
def set_menu(imported_data):
    data = []
    menu = []
    row_count = sum(1 for row in imported_data)
    try:
        for row in imported_data:
            data.append(row)
            v = row[0]
            if v not in menu:
                menu.append(v)

        i = 1
        for item in menu:
            print('          ', "%d - %s" % (i, item))
            i += 1

        #read user input
        print('')
        uservalue = get_uservalue()
        index = int(uservalue) - 1
        selected = menu[index]
        for row in data:
            if row[0] == selected:
                isolated_row = row
        return isolated_row, uservalue
    except IndexError:
        print("Invalid selection.")
    except ValueError:
        print("Invalid selection.")
    except TypeError:
        print("Invalid selection.")

def get_uservalue():
    try:
        uservalue = int(input(">>>"))
        while uservalue < 1:
            uservalue = input("Invalid selection. Please make a selection: ")
        return uservalue
    except TypeError:
        print("TypeError from get_uservalue")
    except ValueError:
        print("Invalid selection.")

#####
##### RENAME A SET #####
#####

def get_newname():
    try:
        print("")
        print("Please enter a new name: ")
        print("")
        new_name = input(">>>")
        while len(new_name) < 1:
            new_name = input("Name cannot be blank. Please enter a new name: ")
        # while new_name == old_name
        # new_name = input("Name must be unique. Please enter a new name: ")
        return new_name
    except ValueError:
        print("Invalid input. ")

def rename_set(imported_data, uservalue):
    try:

```

```

        uservalue = uservalue[1]
        old_name = imported_data[uservalue - 1][0]
        new_name = get_newname()
        imported_data[uservalue - 1][0] = new_name
        print("")
        print(old_name, "renamed to", new_name)
        return imported_data
    except IndexError:
        print("Invalid selection. ")
    except TypeError:
        print("Invalid selection. ")

#####
##### SORT A SET #####
#####

def sort_set(imported_data, uservalue):
    try:
        uservalue = uservalue[1]
        old_list = imported_data[uservalue - 1][1:]
        new_list = list(map(int, imported_data[uservalue - 1][1:]))
        new_list = sorted(new_list)
        imported_data[uservalue - 1][1:] = new_list
        return imported_data
    except IndexError:
        print("Index error in sort_set")

#####
##### ANALYSE A SET #####
#####

def analyse_set(isolated_row):
    analysis_display = isolated_row[0]
    numbers = analysis_display[1:]
    numbers = [int(i) for i in numbers]
    name = analysis_display[0]
    print("")
    print(name)
    length = len(name)
    print('-'*length)
    number_of_values(numbers)
    calculate_minimum(numbers)
    calculate_maximum(numbers)
    calculate_mean(numbers)
    calculate_median(numbers)
    calculate_mode(numbers)
    calculate_sd(numbers)
    return numbers

def number_of_values(numbers):
    number_of_values = len(numbers)
    print("number of values (n): ", "{:01.0f}".format(number_of_values))

def calculate_minimum(numbers):
    x = ' '
    len1 = len("number of values (n)")
    len2 = len("minimum:")
    smallest = numbers[0]
    for number in numbers:
        if number < smallest:
            smallest = number
    print((len1-len2)*x, "minimum: ", "{:01.0f}".format(smallest))

def calculate_maximum(numbers):
    x = ' '
    len1 = len("number of values (n)")

```

```

len3 = len("maximum:")
largest = numbers[0]
for number in numbers:
    if number > largest:
        largest = number
print((len1-len3)*x, "maximum: ", "{:01.0f}".format(largest))

def calculate_mean(numbers):
    x = ' '
    len1 = len("number of values (n)")
    len4 = len("mean:")
    mean = sum(numbers)/len(numbers)
    print((len1-len4)*x, "mean: ", "{:01.2f}".format(mean))

def calculate_median(numbers):
    x = ' '
    len1 = len("number of values (n)")
    len5 = len("median:")
    n = len(numbers)
    numbers = sorted(numbers)
    if n % 2 == 0:
        median1 = numbers[n//2]
        median2 = numbers[n//2 - 1]
        median = (median1 + median2)/2
    else:
        median = numbers[n//2]
    print((len1-len5)*x, "median: ", "{:01.2f}".format(median))

def calculate_mode(numbers):
    x = ' '
    len1 = len("number of values (n)")
    len6 = len("mode:")

    count_dict = {}
    for e in numbers:
        count = numbers.count(e)
        if e not in count_dict.keys():
            count_dict[e] = count
    max_count = 0
    for key in count_dict:
        if count_dict[key] >= max_count:
            max_count = count_dict[key]
    corr_keys = []
    for corr_key, count_value in count_dict.items():
        if count_dict[corr_key] == max_count:
            corr_keys.append(corr_key)
    if max_count == 1 and len(count_dict) != 1:
        print((len1-len6)*x, "mode: ", 'none' )
    else:
        mode = sorted(corr_keys)
        print((len1-len6)*x, "mode: ", str(mode)[1:-1] )
        return mode

def calculate_variance(numbers):
    n = len(numbers)
    mean = sum(numbers) / n
    deviations = [(x - mean) ** 2 for x in numbers]
    variance = sum(deviations) / n
    return variance

def calculate_sd(numbers):
    x = ' '
    len1 = len("number of values (n)")
    len7 = len("standard deviation:")
    number = calculate_variance(numbers)
    sqrt = number ** 0.5

```

```
print((len1-len7)*x, "standard deviation: ", "{:01.2f}".format(sqrt))
```

```
#####  
##### MAIN #####  
#####
```

```
def main():  
    print("Welcome to The Statulator, \nprogrammed by Laura Vodden")  
    display_main_menu()  
    ans = input(">>>")  
    try:  
        while ans != "7":  
            if ans == "1":  
                print("""You have selected option 1 - Load data from a file.  
                """)  
                filename = enter_filename()  
                imported_data = load_data(filename)  
            elif ans == "2":  
                print("""You have selected option 2 - Display the data to the screen.  
                """)  
                display_data(imported_data)  
            elif ans == "3":  
                print("")  
                print("""You have selected option 3 - Rename a set.  
                """)  
                print("""Which set would you like to rename?  
                """)  
                uservalue = set_menu(imported_data)  
                rename_set(imported_data, uservalue)  
            elif ans == "4":  
                print("")  
                print("""You have selected option 4 - Sort a set.  
                """)  
                print("""Which set would you like to sort?  
                """)  
                uservalue = set_menu(imported_data)  
                sort_set(imported_data, uservalue)  
            elif ans == "5":  
                print("")  
                print("""You have selected option 5 - Analyse a set.  
                """)  
                print('Which set would you like to analyse?')  
                print("")  
                isolated_row = set_menu(imported_data)  
                numbers = analyse_set(isolated_row)  
            else:  
                print("")  
                print("Invalid selection")  
                display_main_menu()  
                ans = input(">>>")  
        print("")  
        print("Thanks for Statulating!")  
    except UnboundLocalError:  
        print("No data have been loaded. Please re-run program.")
```

```
main()
```

```
# In[ ]:
```

```
# In[ ]:
```

