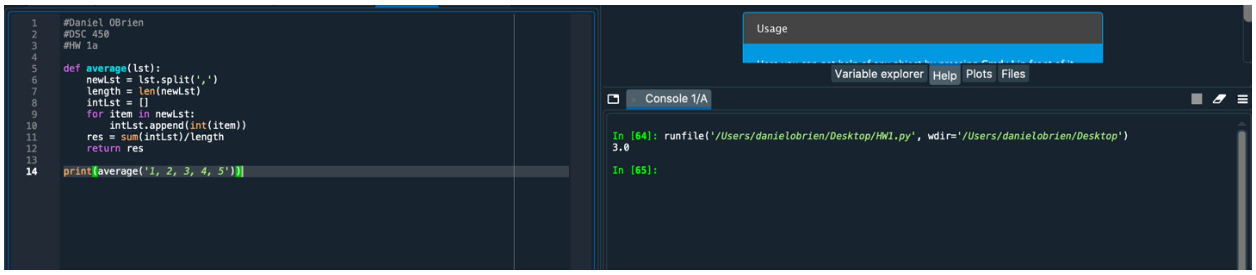
Assignment 1
DSC 450
Daniel O’Brien
01/10/21
Part 1

A.
For part A, I created a function that received a string of numbers. I then removed the commas from the string using the split function. I found the length of the string to use when solving for the average. I then added each item in the string to a list as an integer. Finally, I found the sum of the list and divided it by the length of the list to find the average. I returned the average as the variable, ‘res’.

Python Code:

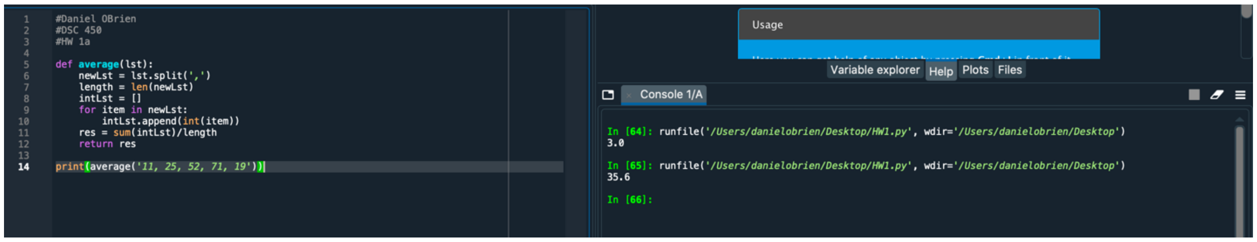
```
def average(lst):
   newLst = lst.split(',')
   length = len(newLst)
   intLst = []
   for item in newLst:
      intLst.append(int(item))
   res = sum(intLst)/length
   return res

print(average('1, 2, 3, 4, 5'))


def average(lst):
   newLst = lst.split(',')
   length = len(newLst)
   intLst = []
   for item in newLst:
      intLst.append(int(item))
   res = sum(intLst)/length
   return res

print(average('11, 25, 52, 71, 19'))
```

Output:



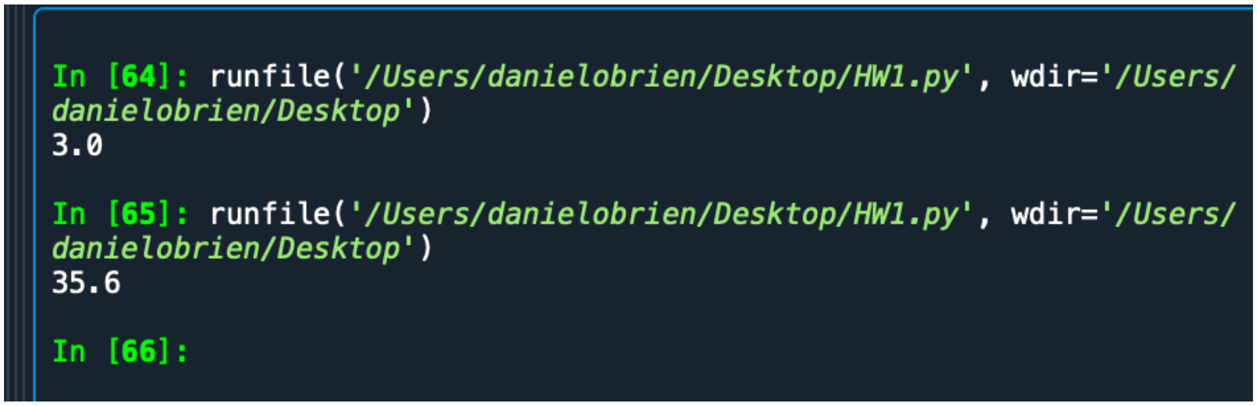
```
#Daniel OBrien
#DSC 450
#HW 1a

def average(lst):
    newLst = lst.split(',')
    length = len(newLst)
    intLst = []
    for item in newLst:
        intLst.append(int(item))
    res = sum(intLst)/length
    return res

print(average('1, 2, 3, 4, 5'))
```

```
In [64]: runfile('/Users/danielobrien/Desktop/HW1.py', wdir='/Users/danielobrien/Desktop')
3.0

In [65]:
```



```
#Daniel OBrien
#DSC 450
#HW 1a

def average(lst):
    newLst = lst.split(',')
    length = len(newLst)
    intLst = []
    for item in newLst:
        intLst.append(int(item))
    res = sum(intLst)/length
    return res

print(average('11, 25, 52, 71, 19'))
```

```
In [64]: runfile('/Users/danielobrien/Desktop/HW1.py', wdir='/Users/danielobrien/Desktop')
3.0

In [65]: runfile('/Users/danielobrien/Desktop/HW1.py', wdir='/Users/danielobrien/Desktop')
35.6

In [66]:
```



```
In [64]: runfile('/Users/danielobrien/Desktop/HW1.py', wdir='/Users/
danielobrien/Desktop')
3.0

In [65]: runfile('/Users/danielobrien/Desktop/HW1.py', wdir='/Users/
danielobrien/Desktop')
35.6

In [66]:
```

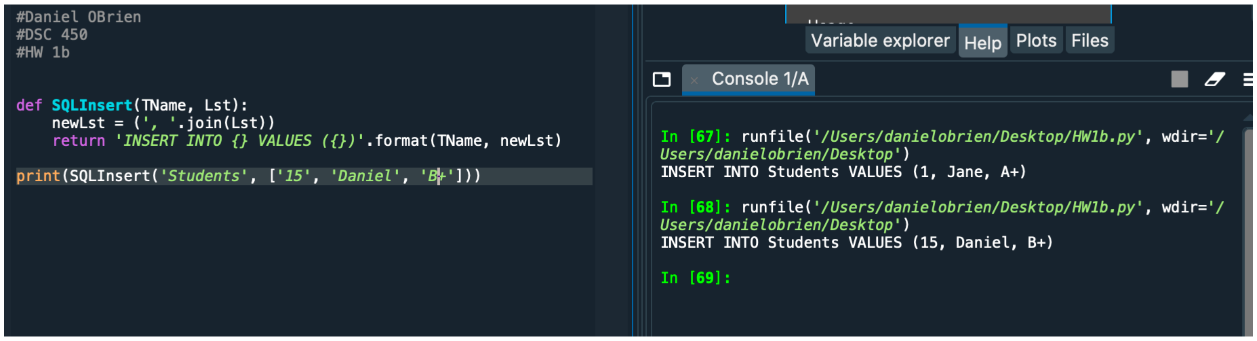
B.
For this code, I created a function that received two different variables, TName and Lst. The first variable indicates what the other items of the list will be inserted into. I then removed commas from the list to better format the return statement. I used brackets and the .format option to create a return statement as requested.

Python Code:

```
def SQLInsert(TName, Lst):
   newLst = (', '.join(Lst))
   return 'INSERT INTO {} VALUES ({})'.format(TName, newLst)

print(SQLInsert('Students', ['1', ‘Jane', 'A+']))


def SQLInsert(TName, Lst):
   newLst = (', '.join(Lst))
   return 'INSERT INTO {} VALUES ({})'.format(TName, newLst)
```

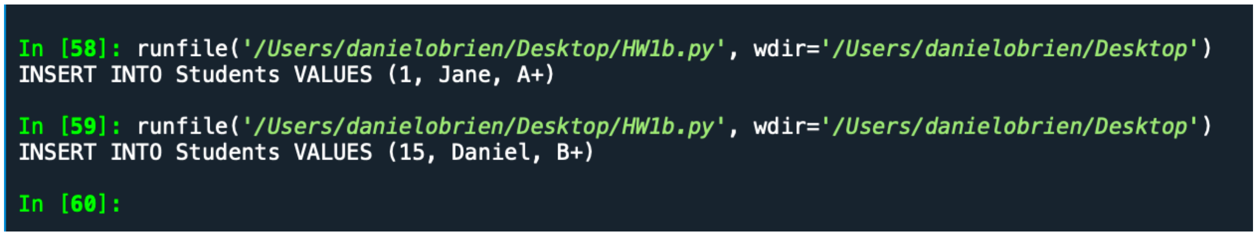
print(SQLInsert('Students', ['15', 'Daniel', 'B+']))

Output:



```
#Daniel OBrien
#DSC 450
#HW 1b


def SQLInsert(TName, Lst):
    newLst = (', '.join(Lst))
    return 'INSERT INTO {} VALUES ({})'.format(TName, newLst)

print(SQLInsert('Students', ['15', 'Daniel', 'B+']))
```

```
In [67]: runfile('/Users/danielobrien/Desktop/HW1b.py', wdir='/
Users/danielobrien/Desktop')
INSERT INTO Students VALUES (1, Jane, A+)

In [68]: runfile('/Users/danielobrien/Desktop/HW1b.py', wdir='/
Users/danielobrien/Desktop')
INSERT INTO Students VALUES (15, Daniel, B+)

In [69]:
```



```
In [58]: runfile('/Users/danielobrien/Desktop/HW1b.py', wdir='/Users/danielobrien/Desktop')
INSERT INTO Students VALUES (1, Jane, A+)

In [59]: runfile('/Users/danielobrien/Desktop/HW1b.py', wdir='/Users/danielobrien/Desktop')
INSERT INTO Students VALUES (15, Daniel, B+)

In [60]:
```