```
                        Analyzing Customer Feedback From Surveys
                                PYTHON AUTOMATION PROJECT
```

```python
## Given a list of customer ratings from 1 to 10, how many are negative, neutral, and positive?

def score_counter (score_list):
    negative_scores = 0
    neutral_scores = 0
    positive_scores = 0

    for score in score_list:
        if score >= 9:
            positive_scores += 1
        elif score >= 6:
            neutral_scores += 1
        else:
            negative_scores += 1

    print('Negative:', negative_scores)
    print('Neutral:', neutral_scores)
    print('Positive:', positive_scores)

## Testing the function
import random
random.seed(42)

possible_scores = list(range(1,11))
score_list1 = random.choices(possible_scores, weights=[8,8,8,8,8,3,3,4,20,30], k=10)
score_list2 = random.choices(possible_scores, weights=[1,2,3,4,5,10,15,15,7,9], k=450)
score_list3 = random.choices(possible_scores, weights=[1,2,3,4,4,5,5,10,15,25], k=10000)

print('Test 1:')
score_counter(score_list1)
print('\nTest 2:')
score_counter(score_list2)
print('\nTest 3:')
score_counter(score_list3)
```

```
Test 1:
Negative: 5
Neutral: 1
Positive: 4

Test 2:
Negative: 85
Neutral: 253
Positive: 112

Test 3:
Negative: 1935
Neutral: 2652
Positive: 5413
```

```
## How many of the feedback IDs are not from verified customers, and what percentage of the total feedback does that represent?

def id_validator(verified_ids, feedback_ids):
    unverified_feedback = 0

    for id in feedback_ids:
        if id not in verified_ids:
            unverified_feedback += 1
    percent_unverified = unverified_feedback / len(feedback_ids) * 100
    print(unverified_feedback, 'of', len(feedback_ids), 'IDs unverified.')
    print(str(round(percent_unverified, 2)) + '% unverified.')
verified_ids = [str(i) for i in range(1,501)]
feedback_ids = [str(i) for i in range(1,701)]

print("\manual Test 1:")
id_validator(verified_ids, feedback_ids)
```

```
\manual Test 1:
200 of 700 IDs unverified.
28.57% unverified.
```

```
## Given a list of what each customer bought, figure out how many customers spent $100 or more.

def purchases_100(sales):
    count = 0                           # Set a counter of totals > 100
    for customer in sales:              # Loop over each inner list
        customer_total = 0              # Set 0 value of purchases for the inner list
        for purchase in customer:       # For price in inner list:
            customer_total += purchase  # Add the price to value of purchases for inner list
        if customer_total >= 100:       # After looping over all prices in inner list, if
                                        # total >= 100
            count+=1                    # Increment the counter
    return count

# Test function

sales1 = [[100,50,20],[70,40,21],[80,50,30],[49,39,20],[40,21,40]]
purchases_100(sales1)
```

```
5
```