Simulation Practice

- Based on sales, frozen yogurt flavors have the following relative frequencies:
- 38% chocolate
- 42% vanilla
- 20% strawberry

Suppose a group of 5 customers comes into the store to purchase yogurt. What is the probability that at least two of them choose chocolate?

Simulating a single customer

- Generate a random integer between 1 and 100 to simulate their ice cream choice
- #'s 1-38 represent purchasing chocolate
    - 38% chance
- #'s 39-80 represent purchasing vanilla
    - 42% chance
- #'s 81-100 represent purchasing Strawberry
    - 20% chance

In [54]:

```
# import the numpy package
import numpy

# Generate random number between 1-100
single_customer = numpy.random.randint(1,101)

# Test the percentage ranges
if single_customer <= 38:
    print("Chocolate")
elif single_customer <= 80:
    print("Vanilla")
else:
    print("Strawberry")
```

```
Chocolate
```

Copy and paste code into a function

- Return a String of the ice cream chosen
- Abstraction!

In [55]:

```
# Create function to simulate a single customer
def single_customer_trial():
    # Generate random number between 1-100
    single_customer = numpy.random.randint(1,101)

    # Test the percentage ranges
    if single_customer <= 38:
        return "Chocolate"
    elif single_customer <= 80:
        return "Vanilla"
    else:
        return "Strawberry"
```

In [56]:

```
# Call the function and print its return value to verify it works!
print(single_customer_trial())
```

```
Chocolate
```

Simulate a single trail of 5 customers

- Create three variables for Chocolate, Vanilla, and Strawberry counts
- Repeat 5 times
  - Execute a single trial
  - Check which ice cream was chosen
  - Update variable appropriately
- Return "true" if chocolate's count is 2 or higher, false otherwise

In [57]:

```
# Create variables
chocolate = 0
vanilla = 0
strawberry= 0

# For loop
for i in range(5):
    customer_choice = single_customer_trial()
    if customer_choice == "Chocolate":
        chocolate += 1
    elif customer_choice == "Vanilla":
        vanilla += 1
    else:
        strawberry += 1

# Print the end result of ice cream choices
# print(chocolate, vanilla, strawberry)

# Print whether chocolate was chosen at least twice
if chocolate >=2:
    print(True)
else:
    print(False)
```

```
True
```

Copy and paste code into a function

- Return true/false whether chocolate was chosen at least twice
- Abstraction!

In [58]:

```
# Create function to simulate 5 customers
def five_customer_trial():
    # Create variables
    chocolate = 0
    vanilla = 0
    strawberry= 0

    # For loop
    for i in range(5):
        customer_choice = single_customer_trial()
        if customer_choice == "Chocolate":
            chocolate += 1
        elif customer_choice == "Vanilla":
            vanilla += 1
        else:
            strawberry += 1

    # Print the end result of ice cream choices
    # print(chocolate, vanilla, strawberry)

    # Print whether chocolate was chosen at least twice
    if chocolate >=2:
        return True
    else:
        return False
```

In [59]:

```
# Call the function and print its return value to verify it works!
print(five_customer_trial())
```

```
False
```

Finding the probabilty that if 5 customers come in, at least two of them purchase chocolate

- Create an array to hold your results
- Create a loop to run 10000 times
  - Add result of each 5-customer trial to the array

In [60]:

```
# Create variables
ice_cream_results = []

# Create a loop that repeats a large number of trials (5 customers), in this ca
for i in range(10000):
    result = five_customer_trial()
    ice_cream_results.append(result)

# Print the results
print(ice_cream_results)
```

```
alse, True, True, False, True, False, False, True, True, True, False, Fals
e, True, True, False, True, True, False, True, True, True, False, False, Tr
ue, False, False, False, False, True, True, False, True, False, False, Tru
e, False, True, False, True, True, False, False, True, True, False, False,
True, False, True, True, False, True, False, False, False, True, False, Fal
se, False, True, True, True, True, True, True, True, True, True, True, Tru
e, False, True, True, False, True, True, True, True, False, True, True, Tru
e, True, False, True, True, True, True, True, False, True, False, False, Tr
ue, False, False, False, True, True, True, True, True, True, True, True, Tr
ue, True, False, False, True, True, False, True, True, True, False, False,
True, True, True, False, True, True, True, True, False, True, True, True, T
rue, True, True, True, True, True, False, True, True, True, False, True, Tr
ue, True, True, False, False, True, False, False, False, True, False, Fals
e, False, False, False, True, True, False, False, False, False, True, True,
True, False, True, True, True, True, False, False, False, True, False, Fals
e, False, True, False, False, True, True, True, True, True, True, True, Tru
e, False, True, False, False, False, True, True, True, True, False, True, T
rue, True, False, True, True, True, False, True, True, False, True, True, T
rue, False, False, True, True, False, True, True, True, False, True, True,
False, True, True, False, True, True, True, True, True, True, False, False,
```

Analyze the results

Find the total number of successes (true)

In [61]:

```
# Create variables
num_true = 0
num_false = 0

# Loop through each entry, updating the number of successful trials (true)
for trial in ice_cream_results:
    if trial == True:
        num_true += 1
    else:
        num_false += 1


# Print the results
print(num_true, num_false)
```

```
6260 3740
```

Calculate the probability

- Probability = (successes/total # trials)

In [62]:

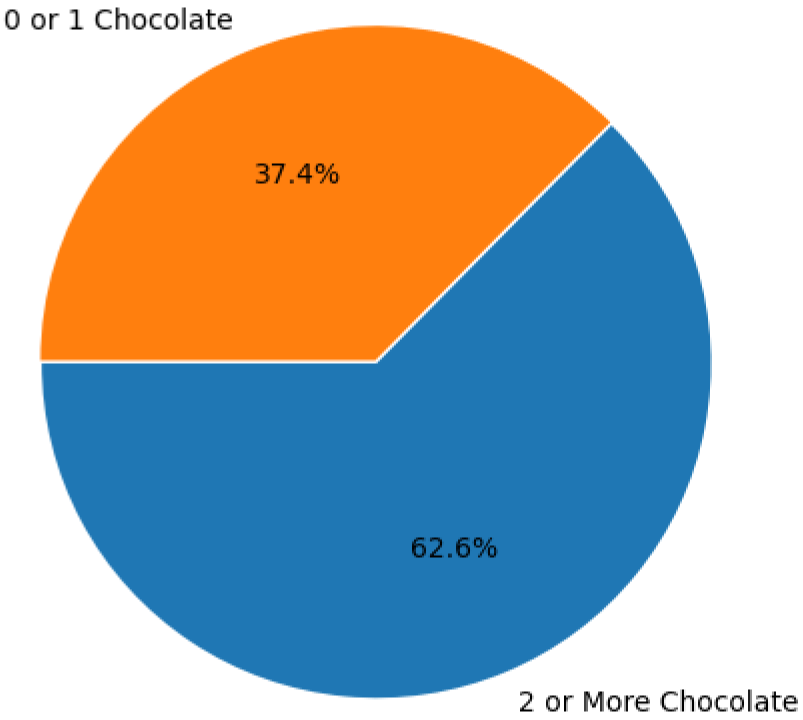
```
# Calculate probability
prob = num_true / 10000

# Print results
print(prob)
```

```
0.626
```

Graph the results

In [65]:

```
# Import the library
import matplotlib.pyplot as plot
%matplotlib inline
# Magic to allow the graph to display directly in this notebook

# Create an array of labels
labels = ["2 or More Chocolate", "0 or 1 Chocolate"]

# Create an array of your results
results = [num_true, num_false]

# Explode option
    # 'Slices' appear distanced from the center
        # Larger numbers = further explosion
    # Explode array should be same size as labels
explode = (0.01, 0)

# Use matplotlib module subplots() to get data for various charts
    # Returns a tuple in the form (figure,axes)
fig1,ax1 = plot.subplots()

# Use axes to create a pie chart
    # ax1.pie(data array, explode array, labels array, starting angle)
ax1.pie(results, explode, labels, autopct='%1.1f%%', startangle=180)

# Equal aspect ratio ensures that pie is drawn as a circle.
ax1.axis('equal')

plot.show()
```



Don't forget to answer the original question!

- Answer should be in the form of a complete sentence

Based on my simulation, I predict that if 5 people walk into my ice cream store that there is a 62.6% chance that 2 or more customers will purchase chocolate

Type *Markdown* and LaTeX: α^2