Simulation Practice - Wait Time

- Every morning, there is a 15% chance that your friend picks you up late on the way to school.

How many school days in a row will typically pass until your friend is late?

Simulating a single day

- Generate a random integer between 1 and 100
- #'s 1-15 represent your friend showing up late
    - 15% chance
- #'s 16-100 represent your friend showing up on time
    - 85% chance

In [905]:

```
# import the numpy package
import numpy

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

# Test the percentage ranges
if single_day <= 15:
    print("Late")
else:
    print("On Time")
```

```
Late
```

Copy and paste code into a function

- Return a String of "Late" or "On Time"
- Abstraction!

In [906]:

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

    # Test the percentage ranges
    if single_day <= 15:
        return("Late")
    else:
        return("On Time")
```

In [907]:

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

```
Late
```

Simulate a single trial until your friend shows up late

- Create a variable that tracks the number of consecutive days on time
  - Start at 0!
- Utilize a "while" loop
  - While your single day trial returns "On Time", add one to your consecutive days variable
- Print the variable tracking the number of consecutive days before your friend shows up late

In [908]:

```
#Import the numpy package
import numpy as np

# Create variables
consecutive_days = 0

# While loop
while single_day_trial() == "On Time":
    consecutive_days += 1

# Print results
print(consecutive_days)
```

```
3
```

Copy and paste code into a function

- Return the number of consecutive days your friend showed up on time
- Abstraction!

In [909]:

```
# Create a function that simulates a complete single trial
def consecutive_days_trial():
    # Create variables
    consecutive_days = 0

    # While Loop
    while single_day_trial() == "On Time":
        consecutive_days += 1

    # Print results
    return(consecutive_days)
```

In [910]:

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

```
8
```

Finding the average number of consecutive days until your friend shows up late

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

In [911]:
```
# Create variables
consecutive_days_results = []

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

# Print the results
print(consecutive_days_results)
```

```
[16, 2, 0, 7, 3, 10, 16, 6, 5, 7, 13, 2, 4, 5, 1, 5, 0, 15, 1, 3, 4, 13, 3,
14, 8, 1, 3, 0, 5, 2, 8, 7, 11, 10, 9, 1, 4, 5, 5, 0, 2, 14, 4, 5, 3, 14,
8, 0, 4, 0, 2, 1, 13, 3, 5, 12, 1, 5, 7, 7, 0, 4, 3, 0, 3, 0, 2, 3, 4, 27,
5, 1, 2, 5, 3, 4, 3, 8, 18, 0, 0, 0, 11, 4, 12, 6, 20, 5, 1, 2, 6, 2, 5, 8,
4, 11, 0, 3, 11, 0, 1, 12, 1, 3, 4, 4, 4, 5, 0, 10, 14, 6, 8, 5, 1, 2, 19,
3, 1, 1, 5, 2, 1, 2, 7, 16, 1, 24, 2, 15, 5, 11, 0, 12, 0, 7, 4, 13, 6, 3,
5, 3, 1, 5, 1, 4, 3, 15, 2, 3, 7, 20, 0, 7, 0, 15, 0, 5, 10, 4, 1, 6, 12,
8, 18, 1, 4, 1, 1, 3, 1, 6, 6, 3, 2, 3, 8, 0, 12, 1, 1, 1, 5, 2, 6, 12, 0,
32, 0, 2, 6, 0, 0, 6, 11, 1, 1, 11, 3, 5, 4, 0, 1, 1, 0, 8, 18, 2, 6, 2, 7,
8, 0, 3, 4, 1, 3, 9, 9, 2, 1, 7, 4, 5, 3, 0, 7, 2, 1, 6, 7, 1, 1, 8, 7, 4,
1, 1, 10, 4, 3, 5, 2, 3, 1, 8, 8, 38, 1, 3, 2, 7, 24, 2, 0, 14, 7, 11, 0,
2, 4, 14, 10, 14, 4, 0, 2, 1, 5, 7, 7, 0, 5, 0, 0, 1, 4, 17, 12, 5, 0, 14,
3, 2, 0, 2, 1, 11, 6, 9, 42, 0, 2, 6, 7, 2, 1, 3, 8, 7, 22, 3, 21, 12, 0,
1, 2, 0, 1, 1, 1, 0, 9, 0, 0, 3, 3, 12, 2, 5, 13, 14, 0, 4, 14, 11, 1, 3,
0, 0, 1, 0, 2, 5, 39, 4, 10, 2, 0, 1, 1, 16, 12, 12, 4, 3, 1, 11, 0, 0, 9,
1, 0, 1, 11, 10, 2, 0, 4, 0, 11, 4, 5, 0, 2, 2, 2, 3, 2, 0, 3, 2, 6, 24, 1
0, 3, 2, 1, 8, 0, 1, 2, 2, 1, 1, 1, 3, 0, 6, 8, 0, 0, 6, 11, 9, 0, 13, 3,
1, 29, 1, 5, 8, 0, 10, 1, 14, 6, 4, 0, 5, 0, 2, 2, 1, 12, 2, 5, 8, 7, 5, 6,
17, 10, 0, 3, 1, 6, 2, 0, 6, 14, 4, 2, 3, 4, 1, 6, 6, 4, 0, 8, 2, 12, 2, 4,
```

Analyze the results

Calculate the sum of all trials in the list

In [912]:
```
# Create variables
sum_of_trials = 0

# Loop through each entry, adding it to the sum
for trial in consecutive_days_results:
    sum_of_trials += trial

# Print the results
print(sum_of_trials)
```

```
56325
```

Calculate the average

- Average = sum of trials / total trials

In [913]:

```
# Calculate the average
avg = sum_of_trials / 10000

# Print the results
print(avg)
```

```
5.6325
```

Graph the results

In [914]:

```
# Count the results using bincount()
count_consecutive_days = numpy.bincount(consecutive_days_results)
print(count_consecutive_days)
print(max(consecutive_days_results))
```

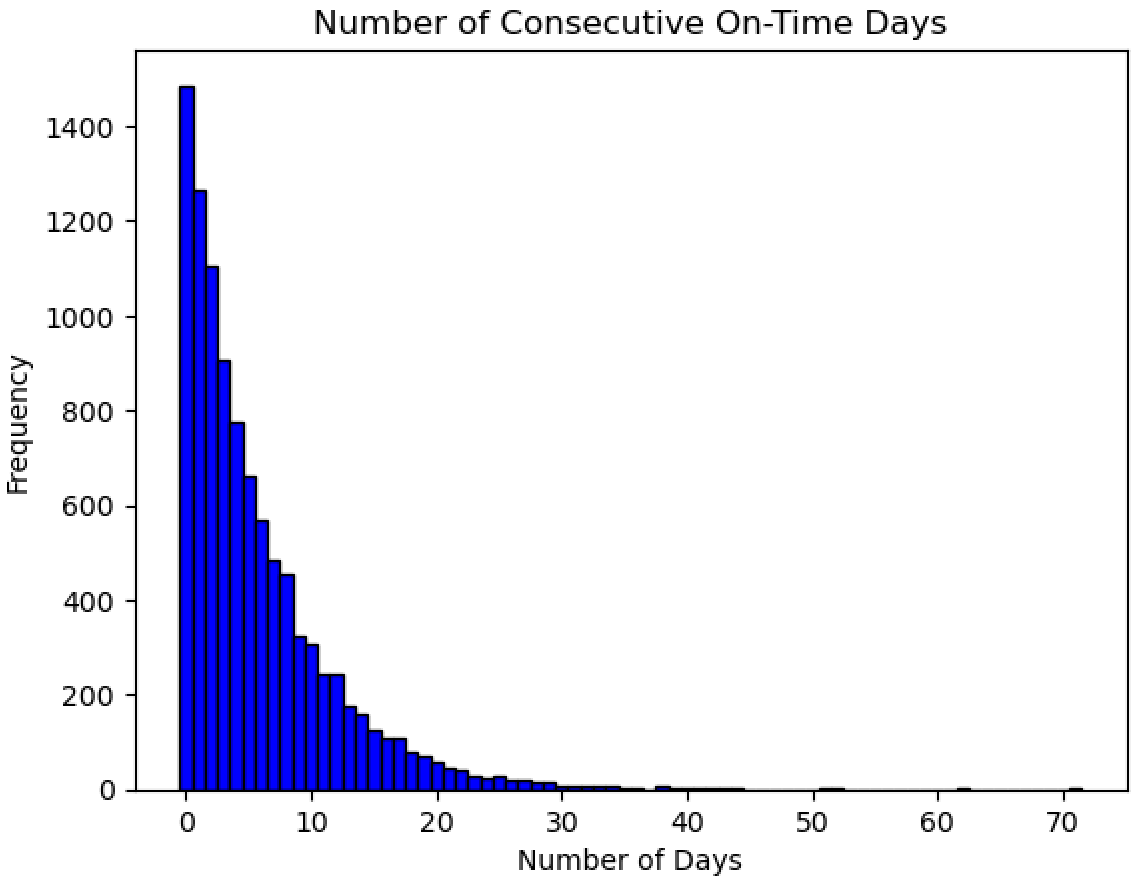
```
[1486 1264 1104  906  777  662  567  483  453  326  307  244  242  178
  161  125  108  108   78   70   56   44   39   27   25   27   20   20
   15   15    9    7    9    6    7    2    3    0    6    3    2    1
    1    2    1    0    0    0    0    0    0    1    1    0    0    0
    0    0    0    0    0    0    1    0    0    0    0    0    0    0
    0    0    1]
72
```

Histogram

In [915]:

```
import matplotlib.pyplot as plt #importing a module for plotting
%matplotlib inline
# this allows us to show graphs in a notebook

# Create the bins array
length = len(count_consecutive_days)
bins = []
for i in range(length):
    bins.append(i)
print(bins)

# plt.hist(array, dividers/bins, alignment, graph color, border color)
plt.hist(consecutive_days_results,bins,align='left',color="blue",edgecolor='bla

# Labels
plt.title("Number of Consecutive On-Time Days")
plt.xlabel("Number of Days")
plt.ylabel("Frequency")

# Show
plt.show()
```

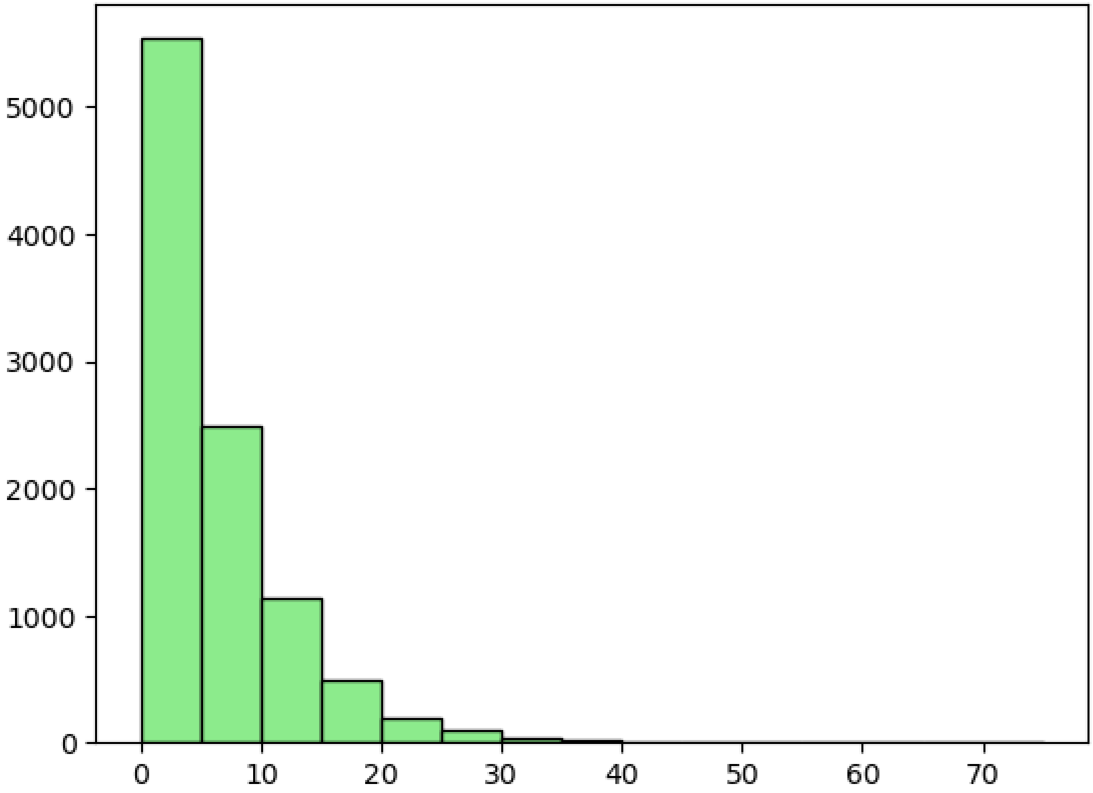
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 2
1, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 4
0, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 5
9, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72]
```


Number of Consecutive On-Time Days
Frequency
1400
1200
1000
800
600
400
200
0
0
10
20
30
40
50
60
70
Number of Days

"Grouped" Histogram

In [916]:

```
# Import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline

# Create bins array
bins = [0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75]

# Create title and lables


# plt.hist(array, dividers/bins, alignment, graph color, border color)
plt.hist(consecutive_days_results, bins, align="mid", color="lightgreen", edge

# Show
plt.show()
```



Don't forget to answer the original question!

- Answer should be in the form of a complete sentence

Based on our simulation, we would estimate that my friend on average would show up 5.6 days in a row before they are late

Type *Markdown* and LaTeX: α^2