

In [1]:

```
# Q1. Create a List, iterate over the List and sort your results, generate random numbers,
# add to the list, and then print your results.
```

```
import random

# Step 1: Create an empty List
numbers = []

# Step 2: Generate 10 random numbers between 1 and 100 and add them to the List
for _ in range(10):
    num = random.randint(1, 100)
    numbers.append(num)

# Step 3: Iterate over the List and print the unsorted numbers
print("Unsorted numbers:")
for num in numbers:
    print(num)

# Step 4: Sort the List
numbers.sort()

# Step 5: Print the sorted results
print("Sorted numbers:")
for num in numbers:
    print(num)
```

Unsorted numbers:

```
6
92
20
76
77
18
7
6
22
77
```

Sorted numbers:

```
6
6
7
18
20
22
76
77
77
92
```

In [2]:

```
# Q2. Create a Line chart with Matplotlib.
```

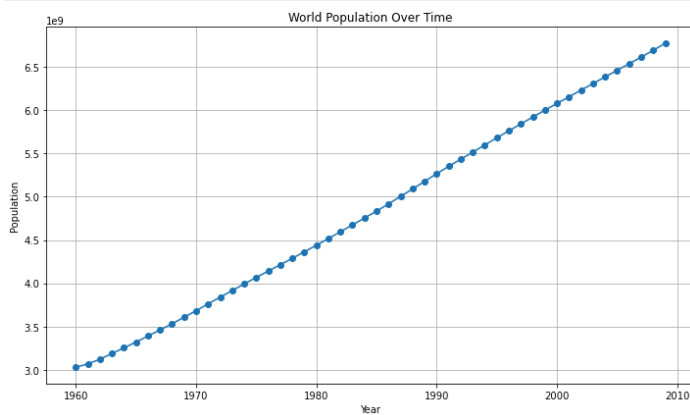
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the Excel file
file_path = (r"C:\Users\Omie\Desktop\MSDS\540_Data Preparation\Week 1-2\world-population.xlsx")
excel_file = pd.ExcelFile(file_path)

# Load the data
df = excel_file.parse('world-population')

# Create a Line chart
plt.figure(figsize=(10, 6))
plt.plot(df['Year'], df['Population'], marker='o', linestyle='-')
plt.title('World Population Over Time')
plt.xlabel('Year')
plt.ylabel('Population')
plt.grid(True)
plt.tight_layout()

# Show the plot
plt.show()
```



In [3]:

```
# Activity 1.01: Handling Lists
```

```
# Generate a List of 100 random integers between 0 and 100
random_number_list = [random.randint(0, 100) for x in range(0, 100)]

# Display the list of random numbers
random_number_list
```

Out[3]:

```
[33,
87,
16,
64,
75,
27,
75,
48,
51,
27,
21,
```

4,
45,
1,
89,
44,
6,
15,
28,
90,
21,
23,
40,
59,
38,
1,
55,
61,
13,
89,
45,
76,
85,
59,
46,
0,
50,
80,
50,
71,
68,
5,
68,
72,
73,
14,
15,
20,
8,
17,
60,
92,
7,
75,
2,
58,
10,
26,
0,
71,
55,
5,
23,
69,
98,
56,
60,
86,
74,
5,
34,
99,
44,
77,
68,
76,
66,
77,
48,
58,
25,
79,
25,
21,
50,
94,
60,
96,
67,
43,
63,
77,
17,
4,
36,
43,
75,
82,
5,
2]

```
In [4]: # Create a new List with numbers divisible by 3 from the original List
list_with_divisible_by_3 = [a for a in random_number_list if a % 3 == 0]

# Display the filtered List
list_with_divisible_by_3
```

```
Out[4]: [33,
87,
75,
27,
75,
48,
51,
27,
21,
45,
6,
15,
90,
21,
45,
0,
72,
15,
60,
75,
0,
69,
60,
99,
66,
48,
```

```
21,
60,
96,
63,
36,
75]
```

```
In [5]: # Get the Length of the original List
length_of_random_list = len(random_number_list)

# Get the Length of the List with numbers divisible by 3
length_of_3_divisible_list = len(list_with_divisible_by_3)

# Calculate how many numbers are not divisible by 3
difference = length_of_random_list - length_of_3_divisible_list

# Display the difference
difference
```

```
Out[5]: 68
```

```
In [6]: # Set the number of times we want to repeat the experiment
NUMBER_OF_EXPERIMENTS = 10

# Create an empty List to store the difference results from each experiment
difference_list = []

# Run the experiment 10 times
for i in range(0, NUMBER_OF_EXPERIMENTS):
    # Generate a new List of 100 random numbers for each experiment
    random_number_list = [random.randint(0, 100) for x in range(0, 100)]
    # Filter out numbers divisible by 3
    list_with_divisible_by_3 = [a for a in random_number_list if a % 3 == 0]

    # Calculate the lengths and their difference
    length_of_random_list = len(random_number_list)
    length_of_3_divisible_list = len(list_with_divisible_by_3)
    difference = length_of_random_list - length_of_3_divisible_list

    # Append the result to the list of differences
    difference_list.append(difference)

# Show the list of differences from all experiments
difference_list
```

```
Out[6]: [56, 75, 68, 60, 68, 66, 70, 62, 65, 63]
```

```
In [7]: # Calculate the average difference over all experiments
avg_diff = sum(difference_list) / float(len(difference_list))

# Display the average difference
avg_diff
```

```
Out[7]: 65.3
```

```
In [8]: # Activity 1.02: Analyzing a Multiline String and Generating the Unique Word Count

# Define a multiline string
multiline_text= """It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife.

However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered

"My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield Park is let at last?"

Mr. Bennet replied that he had not.

"But it is," returned she; "for Mrs. Long has just been here, and she told me all about it."

Mr. Bennet made no answer.

"Do you not want to know who has taken it?" cried his wife impatiently.

"You want to tell me, and I have no objection to hearing it."

This was invitation enough.

"Why, my dear, you must know, Mrs. Long says that Netherfield is taken by a young man of large fortune from the north of England; that he came down on Monday in a chaise and four to see the place

"What is his name?""
```

```
In [9]: # Check the type of the text
type(multiline_text)
```

```
Out[9]: str
```

```
In [10]: # Check how many characters are in the text
len(multiline_text)
```

```
Out[10]: 1228
```

```
In [11]: # Remove newline characters from the text
multiline_text = multiline_text.replace('\n', '')

# Display the cleaned text
multiline_text
```

```
Out[11]: 'It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife.However little known the feelings or views of such a man may be on his first e
ntering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered the rightful property of some one or other of their daughters."My dear Mr. Benn
et," said his lady to him one day, "have you heard that Netherfield Park is let at last?"Mr. Bennet replied that he had not."But it is," returned she; "for Mrs. Long has just been here, and she to
ld me all about it."Mr. Bennet made no answer."Do you not want to know who has taken it?" cried his wife impatiently."You want to tell me, and I have no objection to hearing it."This was invitatio
n enough."Why, my dear, you must know, Mrs. Long says that Netherfield is taken by a young man of large fortune from the north of England; that he came down on Monday in a chaise and four to see t
he place, and was so much delighted with it, that he agreed with Mr. Morris immediately; that he is to take possession before Michaelmas, and some of his servants are to be in the house by the end
of next week.""What is his name?"
```

```
In [13]: # Initialize an empty string to store cleaned characters
cleaned_multiline_text = ""

# Iterate through each character in the text
for char in multiline_text:
    if char == " ":
        # Keep spaces
        cleaned_multiline_text += char
    elif char.isalnum():
```

```
# Keep Letters and numbers
cleaned_multiline_text += char
else:
    # Replace punctuation with a space
    cleaned_multiline_text += " "
```

```
In [14]: # Display the cleaned version of the text with no punctuation
cleaned_multiline_text
```

```
Out[14]: 'It is a truth universally acknowledged that a single man in possession of a good fortune must be in want of a wife However little known the feelings or views of such a man may be on his first entering a neighbourhood this truth is so well fixed in the minds of the surrounding families that he is considered the rightful property of some one or other of their daughters My dear Mr Bennet said his lady to him one day have you heard that Netherfield Park is let at last Mr Bennet replied that he had not But it is returned she for Mrs Long has just been here and she told me all about it Mr Bennet made no answer Do you not want to know who has taken it cried his wife impatiently You want to tell me and I have no objection to hearing it This was invitation enough Why my dear you must know Mrs Long says that Netherfield is taken by a young man of large fortune from the north of England that he came down on Monday in a chaise and four to see the place and was so much delighted with it that he agreed with Mr Morris immediately that he is to take possession before Michaelmas and some of his servants are to be in the house by the end of next week What is his name '
```

```
In [15]: # Split the cleaned text into a list of words
list_of_words = cleaned_multiline_text.split()

# Show the list of individual words
list_of_words
```

```
Out[15]: ['It',
'is',
'a',
'truth',
'universally',
'acknowledged',
'that',
'a',
'single',
'man',
'in',
'possession',
'of',
'a',
'good',
'fortune',
'must',
'be',
'in',
'want',
'of',
'a',
'wife',
'However',
'little',
'known',
'the',
'feelings',
'or',
'views',
'of',
'such',
'a',
'man',
'may',
'be',
'on',
'his',
'first',
'entering',
'a',
'neighbourhood',
'this',
'truth',
'is',
'so',
'well',
'fixed',
'in',
'the',
'minds',
'of',
'the',
'surrounding',
'families',
'that',
'he',
'is',
'considered',
'the',
'rightful',
'property',
'of',
'some',
'one',
'or',
'other',
'of',
'their',
'daughters',
'My',
'dear',
'Mr',
'Bennet',
'said',
'his',
'lady',
'to',
'him',
'one',
'day',
'have',
'you',
'heard',
'that',
'Netherfield',
'Park',
'is',
'let',
'at',
'last',
'Mr',
'Bennet',
'replied',
'that',
'he',
'had',
'not',
```

'But',
'it',
'is',
'returned',
'she',
'for',
'Mrs',
'Long',
'has',
'just',
'been',
'here',
'and',
'she',
'told',
'me',
'all',
'about',
'it',
'Mr',
'Bennet',
'made',
'no',
'answer',
'Do',
'you',
'not',
'want',
'to',
'know',
'who',
'has',
'taken',
'it',
'cried',
'his',
'wife',
'impatiently',
'You',
'want',
'to',
'tell',
'me',
'and',
'I',
'have',
'no',
'objection',
'to',
'hearing',
'it',
'This',
'was',
'invitation',
'enough',
'Why',
'my',
'dear',
'you',
'must',
'know',
'Mrs',
'Long',
'says',
'that',
'Netherfield',
'is',
'taken',
'by',
'a',
'young',
'man',
'of',
'large',
'fortune',
'from',
'the',
'north',
'of',
'England',
'that',
'he',
'came',
'down',
'on',
'Monday',
'in',
'a',
'chaise',
'and',
'four',
'to',
'see',
'the',
'place',
'and',
'was',
'so',
'much',
'delighted',
'with',
'it',
'that',
'he',
'agreed',
'with',
'Mr',
'Morris',
'immediately',
'that',
'he',
'is',
'to',
'take',
'possession',
'before',
'Michaelmas',
'and',
'some',
'of',
'his',
'servants',
'are',

```
'to',
'be',
'in',
'the',
'house',
'by',
'the',
'end',
'of',
'next',
'week',
'What',
'is',
'his',
'name']
```

```
In [16]: # Count how many words are in the list
len(list_of_words)
```

```
Out[16]: 236
```

```
In [17]: # Create a dictionary with all unique words as keys
unique_words_as_dict = dict.fromkeys(list_of_words)

# Count how many unique words are in the text
len(list(unique_words_as_dict.keys()))
```

```
Out[17]: 135
```

```
In [18]: # Count the number of times the unique word has appeared
for word in list_of_words:
    if unique_words_as_dict[word] is None:
        # If we haven't counted the word yet, initialize to 1
        unique_words_as_dict[word] = 1
    else:
        # If it already exists, increment the count
        unique_words_as_dict[word] += 1

# Show the dictionary of words and their frequencies
unique_words_as_dict
```

```
Out[18]: {'It': 1,
'is': 8,
'a': 8,
'truth': 2,
'universally': 1,
'acknowledged': 1,
'that': 8,
'single': 1,
'man': 3,
'in': 5,
'possession': 2,
'of': 10,
'good': 1,
'fortune': 2,
'must': 2,
'be': 3,
'want': 3,
'wife': 2,
'However': 1,
'little': 1,
'known': 1,
'the': 8,
'feelings': 1,
'or': 2,
'views': 1,
'such': 1,
'may': 1,
'on': 2,
'his': 5,
'first': 1,
'entering': 1,
'neighbourhood': 1,
'this': 1,
'so': 2,
'well': 1,
'fixed': 1,
'minds': 1,
'surrounding': 1,
'families': 1,
'he': 5,
'considered': 1,
'rightful': 1,
'property': 1,
'some': 2,
'one': 2,
'other': 1,
'their': 1,
'daughters': 1,
'My': 1,
'dear': 2,
'Mr': 4,
'Bennet': 3,
'said': 1,
'lady': 1,
'to': 7,
'him': 1,
'day': 1,
'have': 2,
'you': 3,
'heard': 1,
'Netherfield': 2,
'Park': 1,
'let': 1,
'at': 1,
'last': 1,
'replied': 1,
'had': 1,
'not': 2,
'But': 1,
'it': 5,
'returned': 1,
'she': 2,
'for': 1,
'Mrs': 2,
'Long': 2,
'has': 2,
'just': 1,
'been': 1,
```

```
'here': 1,
'and': 5,
'told': 1,
'me': 2,
'all': 1,
'about': 1,
'made': 1,
'no': 2,
'answer': 1,
'Do': 1,
'know': 2,
'who': 1,
'taken': 2,
'cried': 1,
'impatiently': 1,
'You': 1,
'tell': 1,
'I': 1,
'objection': 1,
'hearing': 1,
'This': 1,
'was': 2,
'invitation': 1,
'enough': 1,
'Why': 1,
'my': 1,
'says': 1,
'by': 2,
'young': 1,
'large': 1,
'from': 1,
'north': 1,
'England': 1,
'came': 1,
'down': 1,
'Monday': 1,
'chaise': 1,
'four': 1,
'see': 1,
'place': 1,
'much': 1,
'delighted': 1,
'with': 2,
'agreed': 1,
'Morris': 1,
'immediately': 1,
'take': 1,
'before': 1,
'Michaelmas': 1,
'servants': 1,
'are': 1,
'house': 1,
'end': 1,
'next': 1,
'week': 1,
'What': 1,
'name': 1}
```

```
In [19]: # Sort the words by their frequency in descending order
top_words = sorted(unique_words_as_dict.items(), key=lambda key_val_tuple: key_val_tuple[1], reverse=True)

# Display the top 25 most frequent words
top_words[:25]
```

```
Out[19]: [('of', 10),
('is', 8),
('a', 8),
('that', 8),
('the', 8),
('to', 7),
('in', 5),
('his', 5),
('he', 5),
('it', 5),
('and', 5),
('Mr', 4),
('man', 3),
('be', 3),
('want', 3),
('Bennet', 3),
('you', 3),
('truth', 2),
('possession', 2),
('fortune', 2),
('must', 2),
('wife', 2),
('or', 2),
('on', 2),
('so', 2)]
```

```
In [20]: # Activity 2.01: Permutation, Iterator, Lambda, and List

from itertools import permutations, dropwhile
permutations?
dropwhile?
```

```
In [21]: # Generate all permutations of the numbers [0, 1, 2]
permutations(range(3))
```

```
Out[21]: <itertools.permutations at 0x199af083860>
```

```
In [22]: # Loop through each permutation
for number_tuple in permutations(range(3)):
    print(number_tuple)
    # Assert ensures each output is a tuple
    assert isinstance(number_tuple, tuple)
```

```
(0, 1, 2)
(0, 2, 1)
(1, 0, 2)
(1, 2, 0)
(2, 0, 1)
(2, 1, 0)
```

```
In [23]: # For each permutation, drop all elements from the beginning that are <= 0
for number_tuple in permutations(range(3)):
```

```
# Convert the result to a list and print it
print(list(dropwhile(lambda x: x <= 0, number_tuple)))
```

```
[1, 2]
[2, 1]
[1, 0, 2]
[1, 2, 0]
[2, 0, 1]
[2, 1, 0]
```

```
In [25]: import math

# Function to convert a list of digits into a full number
def convert_to_number(number_stack):
    final_number = 0
    for i in range(0, len(number_stack)):
        # Pop the last element, multiply by 10^i, and add it to the total
        final_number += (number_stack.pop() * (math.pow(10, i)))
    return final_number

# For each permutation of [0, 1, 2]:
for number_tuple in permutations(range(3)):
    # Remove leading zeros by dropping elements <= 0 from the start
    number_stack = list(dropwhile(lambda x: x <= 0, number_tuple))
    # Convert the remaining digits into a number and print
    print(convert_to_number(number_stack))
```

```
12.0
21.0
102.0
120.0
201.0
210.0
```

```
In [26]: # Activity 2.02: Designing Your Own CSV Parser

from itertools import zip_longest
```

```
In [27]: # Function to convert a list of CSV values into a dictionary using the header
def return_dict_from_csv_line(header, line):
    # Pair each header field with the corresponding value from the line
    # If line has fewer values than header, fill missing ones with None
    zipped_line = zip_longest(header, line, fillvalue=None)

    # Use dictionary comprehension to turn the pairs into a dictionary
    ret_dict = {kv[0]: kv[1] for kv in zipped_line}
    return ret_dict
```

```
In [31]: # Open the CSV file in read mode
with open("C:/Users/Omie/Desktop/MSDS/540_Data Preparation/Week 1-2/sales_record.csv", "r") as fd:
    # Read the first line to extract the header
    first_line = fd.readline()

    # Remove the newline character and split by comma to get a list of column names
    header = first_line.replace("\n", "").split(",")

    # Loop through each remaining line in the file
    for i, line in enumerate(fd):
        # Stop after processing the first 10 lines to avoid too much output
        if i > 10:
            break

        # Clean the line and split it into individual values
        line = line.replace("\n", "").split(",")

        # Convert the line into a dictionary using the header
        d = return_dict_from_csv_line(header, line)

        # Print the resulting dictionary
        print(d)
```

```
{'Region': 'Central America and the Caribbean', 'Country': 'Antigua and Barbuda', 'Item Type': 'Baby Food', 'Sales Channel': 'Online', 'Order Priority': 'M', 'Order Date': '12/20/2013', 'Order ID': '957081544', 'Ship Date': '1/11/2014', 'Units Sold': '552', 'Unit Price': '255.28', 'Unit Cost': '159.42', 'Total Revenue': '140914.56', 'Total Cost': '87999.84', 'Total Profit': '52914.72'}
{'Region': 'Central America and the Caribbean', 'Country': 'Panama', 'Item Type': 'Snacks', 'Sales Channel': 'Offline', 'Order Priority': 'C', 'Order Date': '7/5/2010', 'Order ID': '301644504', 'Ship Date': '7/26/2010', 'Units Sold': '2167', 'Unit Price': '152.58', 'Unit Cost': '97.44', 'Total Revenue': '330640.86', 'Total Cost': '211152.48', 'Total Profit': '119488.38'}
{'Region': 'Europe', 'Country': 'Czech Republic', 'Item Type': 'Beverages', 'Sales Channel': 'Offline', 'Order Priority': 'C', 'Order Date': '9/12/2011', 'Order ID': '478051030', 'Ship Date': '9/29/2011', 'Units Sold': '4778', 'Unit Price': '47.45', 'Unit Cost': '31.79', 'Total Revenue': '226716.1', 'Total Cost': '151892.62', 'Total Profit': '74823.48'}
{'Region': 'Asia', 'Country': 'North Korea', 'Item Type': 'Cereal', 'Sales Channel': 'Offline', 'Order Priority': 'L', 'Order Date': '5/13/2010', 'Order ID': '892599952', 'Ship Date': '6/15/2010', 'Units Sold': '9016', 'Unit Price': '205.7', 'Unit Cost': '117.11', 'Total Revenue': '1854591.2', 'Total Cost': '1055863.76', 'Total Profit': '798727.44'}
{'Region': 'Asia', 'Country': 'Sri Lanka', 'Item Type': 'Snacks', 'Sales Channel': 'Offline', 'Order Priority': 'C', 'Order Date': '7/20/2015', 'Order ID': '571902596', 'Ship Date': '7/27/2015', 'Units Sold': '7542', 'Unit Price': '152.58', 'Unit Cost': '97.44', 'Total Revenue': '1150758.36', 'Total Cost': '734892.48', 'Total Profit': '415865.88'}
{'Region': 'Middle East and North Africa', 'Country': 'Morocco', 'Item Type': 'Personal Care', 'Sales Channel': 'Offline', 'Order Priority': 'L', 'Order Date': '11/8/2010', 'Order ID': '412882792', 'Ship Date': '11/22/2010', 'Units Sold': '48', 'Unit Price': '81.73', 'Unit Cost': '56.67', 'Total Revenue': '3923.04', 'Total Cost': '2720.16', 'Total Profit': '1202.88'}
{'Region': 'Australia and Oceania', 'Country': 'Federated States of Micronesia', 'Item Type': 'Clothes', 'Sales Channel': 'Offline', 'Order Priority': 'H', 'Order Date': '3/28/2011', 'Order ID': '932776868', 'Ship Date': '5/10/2011', 'Units Sold': '8258', 'Unit Price': '109.28', 'Unit Cost': '35.84', 'Total Revenue': '902434.24', 'Total Cost': '295966.72', 'Total Profit': '606467.52'}
{'Region': 'Europe', 'Country': 'Bosnia and Herzegovina', 'Item Type': 'Clothes', 'Sales Channel': 'Online', 'Order Priority': 'M', 'Order Date': '10/14/2013', 'Order ID': '919133651', 'Ship Date': '11/4/2013', 'Units Sold': '927', 'Unit Price': '109.28', 'Unit Cost': '35.84', 'Total Revenue': '101302.56', 'Total Cost': '33223.68', 'Total Profit': '68078.88'}
{'Region': 'Middle East and North Africa', 'Country': 'Afghanistan', 'Item Type': 'Clothes', 'Sales Channel': 'Offline', 'Order Priority': 'M', 'Order Date': '8/27/2016', 'Order ID': '579814469', 'Ship Date': '10/5/2016', 'Units Sold': '8841', 'Unit Price': '109.28', 'Unit Cost': '35.84', 'Total Revenue': '966144.48', 'Total Cost': '316861.44', 'Total Profit': '649283.04'}
{'Region': 'Sub-Saharan Africa', 'Country': 'Ethiopia', 'Item Type': 'Baby Food', 'Sales Channel': 'Online', 'Order Priority': 'M', 'Order Date': '4/13/2015', 'Order ID': '192993152', 'Ship Date': '5/7/2015', 'Units Sold': '9817', 'Unit Price': '255.28', 'Unit Cost': '159.42', 'Total Revenue': '2506083.76', 'Total Cost': '1565026.14', 'Total Profit': '941057.62'}
{'Region': 'Middle East and North Africa', 'Country': 'Turkey', 'Item Type': 'Office Supplies', 'Sales Channel': 'Offline', 'Order Priority': 'C', 'Order Date': '9/25/2013', 'Order ID': '557156026', 'Ship Date': '10/15/2013', 'Units Sold': '3704', 'Unit Price': '651.21', 'Unit Cost': '524.96', 'Total Revenue': '2412081.84', 'Total Cost': '1944451.84', 'Total Profit': '467630'}
```

```
In [ ] :
```