Student: Balakrishna Mupparaju

Assignment Week: 1 & 2

1

Create a Jupyter notebook where you create a list, iterate over the list and sort your results, generate random numbers, add to the list, and then print your results.

In [66]:
```
import random

# Step 1: Create a list
numbers = [12, 45, 2, 8, 19]

# Step 2: Iterate over the list and print each element
print("Original List:")
for num in numbers:
    print(num)

# Step 3: Sort the list
numbers.sort()
print("\nSorted List:", numbers)

# Step 4: Generate and add random numbers
random_numbers = [random.randint(1, 100) for _ in range(5)]
numbers.extend(random_numbers)
print("\nList after adding random numbers:", numbers)

# Step 5: Sort again after adding numbers
numbers.sort()
print("\nFinal Sorted List:", numbers)
```

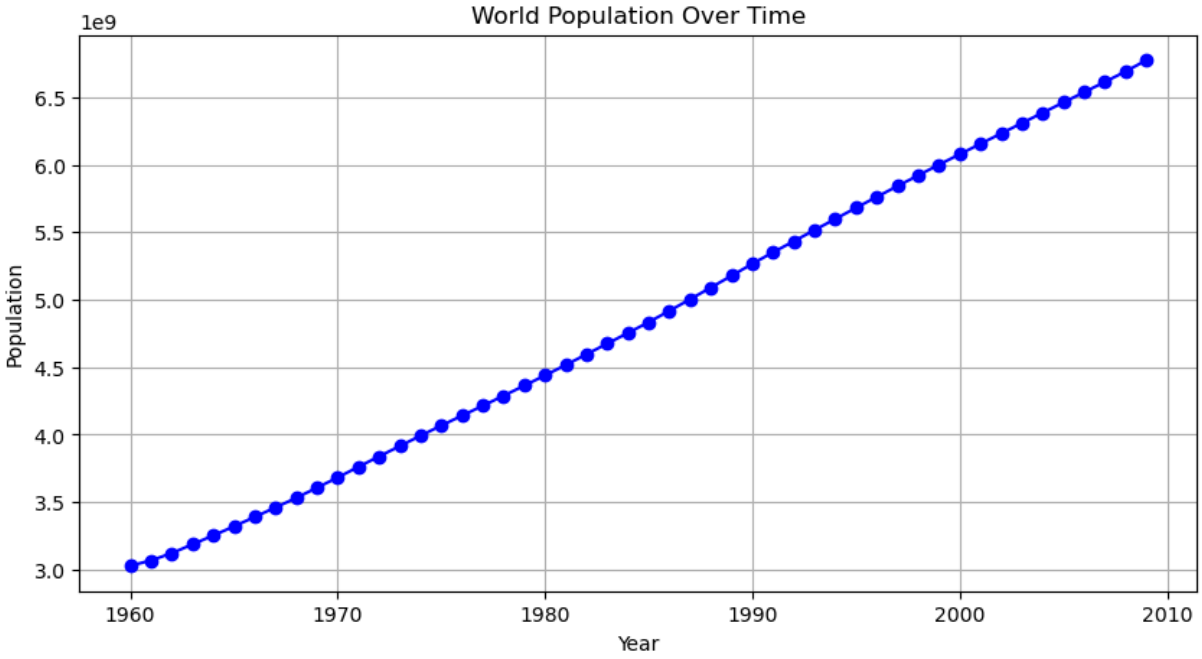
```
Original List:
12
45
2
8
19

Sorted List: [2, 8, 12, 19, 45]

List after adding random numbers: [2, 8, 12, 19, 45, 45, 90, 53, 60, 12]

Final Sorted List: [2, 8, 12, 12, 19, 45, 45, 53, 60, 90]
```

2

Create a line chart with Matplotlib and the following data file.

In [74]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the world population data
file_path = "/Users/balakrishnamupparaju/Downloads/world-population.xlsm"
df = pd.read_excel(file_path)

# Display first few rows to understand the structure
print(df.head())

# The columns are named 'Year' and 'Population.'
plt.figure(figsize=(10,5))
plt.plot(df['Year'], df['Population'], marker='o', linestyle='-', color='b')

# Formatting the chart
plt.title('World Population Over Time')
plt.xlabel('Year')
plt.ylabel('Population')
plt.grid(True)

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

```
   Year  Population
0  1960  3028654024
1  1961  3068356747
2  1962  3121963107
3  1963  3187471383
4  1964  3253112403
```



Activity 1.01

In [77]:
```
#The random module is imported to generate random numbers.
import random
#A list comprehension is used to create random_number_list containing 100 ra
random_number_list = [random.randint(0, 100) for x in range(0, 100)]
print(random_number_list)

#A new list list_with_divisible_by_3 is created, filtering numbers divisible
random_number_list
list_with_divisible_by_3 = [a for a in random_number_list if a % 3 == 0]
print(list_with_divisible_by_3)

#The length of both lists is calculated.
length_of_random_list = len(random_number_list)
length_of_3_divisible_list = len(list_with_divisible_by_3)

#The difference between the lengths is stored in difference.
difference = length_of_random_list - length_of_3_divisible_list
print(difference)

#Defines how many times the experiment will be repeated.
NUMBER_OF_EXPERIMENTS = 10
#Initializes an empty list to store the differences.
difference_list = []

#Runs the same process 10 times:
for i in range(0, NUMBER_OF_EXPERIMENTS):
    #Generates a new list of 100 random numbers.
    random_number_list = [random.randint(0, 100) for x in range(0, 100)]
    #Filters numbers divisible by 3.
    list_with_divisible_by_3 = [a for a in random_number_list if a % 3 == 0]
    #Calculates the difference in lengths.
    length_of_random_list = len(random_number_list)
    #Stores the difference in difference_list
    length_of_3_divisible_list = len(list_with_divisible_by_3)
    difference = length_of_random_list - length_of_3_divisible_list
    difference_list.append(difference)
print(difference_list)
#Computes the arithmetic mean of all stored differences.
#Converts len(difference_list) to float to ensure accurate division
avg_diff = sum(difference_list) / float(len(difference_list))
print(avg_diff)
```

```
[17, 29, 31, 71, 34, 57, 44, 42, 65, 90, 60, 34, 94, 1, 81, 24, 33, 66, 77,
67, 89, 82, 71, 51, 71, 26, 10, 17, 70, 29, 8, 26, 65, 76, 55, 72, 29, 89, 5
5, 65, 44, 54, 62, 9, 24, 29, 3, 5, 87, 98, 22, 7, 72, 68, 66, 92, 57, 67, 6
0, 51, 42, 54, 59, 61, 98, 62, 85, 79, 60, 77, 15, 99, 46, 23, 81, 95, 84, 6
7, 42, 6, 42, 71, 7, 60, 41, 34, 67, 70, 46, 15, 83, 55, 50, 32, 23, 39, 47,
28, 84, 62]
[57, 42, 90, 60, 81, 24, 33, 66, 51, 72, 54, 9, 24, 3, 87, 72, 66, 57, 60, 5
1, 42, 54, 60, 15, 99, 81, 84, 42, 6, 42, 60, 15, 39, 84]
66
[66, 73, 65, 60, 69, 59, 61, 71, 64, 61]
64.9
```

Activity 1.02

he Data Wrangling Workshop: Activity 1.02

In [80]:
```
multiline_text= """It is a truth universally acknowledged, that a single mar

However little known the feelings or views of such a man may be on his first

"My dear Mr. Bennet," said his lady to him one day, "have you heard that Net

Mr. Bennet replied that he had not.

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

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

"What is his name?"""
```

In [82]:
```
#Confirms that multiline_text is a string (str).
type(multiline_text)
#Counts the total number of characters (including spaces, punctuation, and n
len(multiline_text)
```

Out[82]: 1228

In [84]:
```
#The .replace('\n', "") method removes all newline characters, ensuring the
multiline_text = multiline_text.replace('\n', "")
multiline_text
```

Out[84]:
```
'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 t
ruth 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."M
y dear Mr. Bennet," said his lady to him one day, "have you heard that Neth
erfield Park is let at last?"Mr. Bennet replied that he had not."But it i
s," returned she; "for Mrs. Long has just been here, and she told me all ab
out it."Mr. Bennet made no answer."Do you not want to know who has taken i
t?" cried his wife impatiently."You want to tell me, and I have no objectio
n to hearing it."This was invitation enough."Why, my dear, you must know, M
rs. Long says that Netherfield is taken by a young man of large fortune fro
m 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. M
orris immediately; that he is to take possession before Michaelmas, and som
e of his servants are to be in the house by the end of next week.""What is
his name?'
```

In [86]:
```python
#Remove Special Characters and Punctuation
#Keeps spaces (" ") unchanged to preserve word separation.
#Keeps letters and numbers (using isalnum()).
#Replaces punctuation and special characters with spaces to avoid merged wor

cleaned_multiline_text = ""
for char in multiline_text:
    if char == " ":
        cleaned_multiline_text += char
    elif char.isalnum():  # isalnum() checks if the character is a letter or
        cleaned_multiline_text += char
    else:
        cleaned_multiline_text += " "

#Tokenize the Text (Convert to List of Words)
#Splits the cleaned text into individual words using .split(), which default

list_of_words = cleaned_multiline_text.split()

#Returns the total number of words in the text.
#This is different from len(multiline_text), which counted characters.

len(list_of_words)

#Create a Dictionary of Unique Words
#Creates a dictionary where each word is a key, with None as its initial val
#Counts the unique words (removing duplicates).

unique_words_as_dict = dict.fromkeys(list_of_words)
len(list(unique_words_as_dict.keys()))

list_of_words = ["My", "dear", "Mr", "Bennet", "dear", "lady"]
unique_words_as_dict = {"My": None, "dear": None, "Mr": None, "Bennet": None
len(unique_words_as_dict.keys())  # Returns 5 (since "dear" is counted only


#Count the Frequency of Each Word
for word in list_of_words:
```

```
    if unique_words_as_dict[word] is None:
        unique_words_as_dict[word] = 1
    else:
        unique_words_as_dict[word] += 1
```

In [88]:
```
#Sort Words by Frequency (Most Common First)
#Converts the dictionary into a list of tuples ((word, count)).
#Sorts by word frequency (value in the dictionary).
#Reverse=True → Sorts in descending order (most frequent words first).
#Displays the top 25 most common words in the text.
top_words = sorted(unique_words_as_dict.items(), key=lambda key_val_tuple: k
top_words[:25]
```

Out[88]:
```
[('dear', 2), ('My', 1), ('Mr', 1), ('Bennet', 1), ('lady', 1)]
```

Activity 2.01

The Data Wrangling Workshop: Activity 2.01

In [91]:
```
#itertools.permutations generates all possible ordered arrangements of eleme
#itertools.dropwhile removes elements from the start of an iterable as long
#math.pow is used for exponentiation (raising numbers to a power).
from itertools import permutations, dropwhile
import math
```

In [93]:
```
#Understanding permutations()
#permutations(range(3)) generates all ordered arrangements of (0, 1, 2).
permutations(range(3))
#The function returns an iterator, which we can loop through to get the actu
for number_tuple in permutations(range(3)):
    print(number_tuple)
    assert isinstance(number_tuple, tuple)  # Ensures that each result is a
#Each permutation is a tuple of numbers from {0, 1, 2} in different orders.
```

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

In [95]:
```
#Using dropwhile()
#skips elements until it finds an element greater than 0.
#It returns the remaining elements as an iterator, which we convert to a lis

for number_tuple in permutations(range(3)):
    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 [97]:
```python
#Converting List of Digits to a Number

#Converts a list of digits into an actual number.
#Uses pop() to extract digits from the right (last element) and places them
#Uses math.pow(10, i) to position digits correctly.
import math
def convert_to_number(number_stack):
    final_number = 0
    for i in range(0, len(number_stack)):
        final_number += (number_stack.pop() * (math.pow(10, i)))
    return final_number


#Applying dropwhile() and convert_to_number() Together
#Remove leading zeros using dropwhile()
#Convert remaining numbers into an integer using convert_to_number()
for number_tuple in permutations(range(3)):
    number_stack = list(dropwhile(lambda x: x <= 0, number_tuple))
    print(convert_to_number(number_stack))
```

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

Activity 2.02

The Data Wrangling Workshop: Activity 2.02

In [100…
```python
#Import Necessary Module
#zip_longest() is a function from itertools that pairs elements from multipl

from itertools import zip_longest
```

In [102…
```python
#2. Function: return_dict_from_csv_line()
"""
Input:
Header: A list of column names (first line of the CSV).
Line: A list of values from a single CSV row.
Processing:
Uses zip_longest() to pair each column name with its corresponding value.
It uses dictionary comprehension to create a dictionary where keys are colum
Output: A dictionary representing a single CSV row.
 Why zip_longest()?
```

```
If a row has missing values, zip_longest() fills them with None, preventing
"""
def return_dict_from_csv_line(header, line):
    # Zip them
    zipped_line = zip_longest(header, line, fillvalue=None)
    # Use dict comprehension to generate the final dict
    ret_dict = {kv[0]: kv[1] for kv in zipped_line}
    return ret_dict
```

In [104…

```
#Opens the file sales_record.csv in read mode ("r").
#fd is the file object used to read the contents.
with open("/Users/balakrishnamupparaju/Downloads/sales_record.csv", "r") as

    """Reads the first line of the file (which contains column names).
    Removes the newline character (\n) to prevent formatting issues.
    Splits the line into a list using split(","), assuming CSV values are co

    first_line = fd.readline()
    header = first_line.replace("\n", "").split(",")
    # Loops through each line in the file (starting from the second line).
    #enumerate(fd) keeps track of the line index (i).


    for i, line in enumerate(fd):
        # Here we loop over the first 10 lines in order to not to make the c
        # Removes newline characters (\n).
        #Splits the line into a list of values using split(",").
        line = line.replace("\n", "").split(",")

        # Passes the header and current row (line) to return_dict_from_csv_l
        #Stores the resulting dictionary (d) and prints it.
        d = return_dict_from_csv_line(header, line)
        print(d)
        #Stops the loop after processing 10 rows, preventing excessive outpu


        if i > 10:
            break
```

{'Region': 'Central America and the Caribbean', 'Country': 'Antigua and Barb
uda ', 'Item Type': 'Baby Food', 'Sales Channel': 'Online', 'Order Priorit
y': '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 T
ype': 'Snacks', 'Sales Channel': 'Offline', 'Order Priority': 'C', 'Order Da
te': '7/5/2010', 'Order ID': '301644504', 'Ship Date': '7/26/2010', 'Units S
old': '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': '9/12/201
1', 'Order ID': '478051030', 'Ship Date': '9/29/2011', 'Units Sold': '4778',
'Unit Price': '47.45', 'Unit Cost': '31.79', 'Total Revenue': '226716.10',
'Total Cost': '151892.62', 'Total Profit': '74823.48'}
{'Region': 'Asia', 'Country': 'North Korea', 'Item Type': 'Cereal', 'Sales C
hannel': 'Offline', 'Order Priority': 'L', 'Order Date': '5/13/2010', 'Order
ID': '892599952', 'Ship Date': '6/15/2010', 'Units Sold': '9016', 'Unit Pric
e': '205.70', 'Unit Cost': '117.11', 'Total Revenue': '1854591.20', 'Total C
ost': '1055863.76', 'Total Profit': '798727.44'}
{'Region': 'Asia', 'Country': 'Sri Lanka', 'Item Type': 'Snacks', 'Sales Cha
nnel': 'Offline', 'Order Priority': 'C', 'Order Date': '7/20/2015', 'Order I
D': '571902596', 'Ship Date': '7/27/2015', 'Units Sold': '7542', 'Unit Pric
e': '152.58', 'Unit Cost': '97.44', 'Total Revenue': '1150758.36', 'Total Co
st': '734892.48', 'Total Profit': '415865.88'}
{'Region': 'Middle East and North Africa', 'Country': 'Morocco', 'Item Typ
e': 'Personal Care', 'Sales Channel': 'Offline', 'Order Priority': 'L', 'Ord
er Date': '11/8/2010', 'Order ID': '412882792', 'Ship Date': '11/22/2010',
'Units Sold': '48', 'Unit Price': '81.73', 'Unit Cost': '56.67', 'Total Reve
nue': '3923.04', 'Total Cost': '2720.16', 'Total Profit': '1202.88'}
{'Region': 'Australia and Oceania', 'Country': 'Federated States of Micrones
ia', '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': '60
6467.52'}
{'Region': 'Europe', 'Country': 'Bosnia and Herzegovina', 'Item Type': 'Clot
hes', 'Sales Channel': 'Online', 'Order Priority': 'M', 'Order Date': '10/1
4/2013', 'Order ID': '919133651', 'Ship Date': '11/4/2013', 'Units Sold': '9
27', '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 T
ype': 'Clothes', 'Sales Channel': 'Offline', 'Order Priority': 'M', 'Order D
ate': '8/27/2016', 'Order ID': '579814469', 'Ship Date': '10/5/2016', 'Units
Sold': '8841', 'Unit Price': '109.28', 'Unit Cost': '35.84', 'Total Revenu
e': '966144.48', 'Total Cost': '316861.44', 'Total Profit': '649283.04'}
{'Region': 'Sub-Saharan Africa', 'Country': 'Ethiopia', 'Item Type': 'Baby F
ood', 'Sales Channel': 'Online', 'Order Priority': 'M', 'Order Date': '4/13/
2015', 'Order ID': '192993152', 'Ship Date': '5/7/2015', 'Units Sold': '981
7', 'Unit Price': '255.28', 'Unit Cost': '159.42', 'Total Revenue': '250608
3.76', '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', 'Uni
ts Sold': '3704', 'Unit Price': '651.21', 'Unit Cost': '524.96', 'Total Reve

nue': '2412081.84', 'Total Cost': '1944451.84', 'Total Profit': '467630.00'}
{'Region': 'Middle East and North Africa', 'Country': 'Oman', 'Item Type': 'Cosmetics', 'Sales Channel': 'Online', 'Order Priority': 'M', 'Order Date': '5/12/2013', 'Order ID': '741101920', 'Ship Date': '5/17/2013', 'Units Sold': '7382', 'Unit Price': '437.20', 'Unit Cost': '263.33', 'Total Revenue': '3227410.40', 'Total Cost': '1943902.06', 'Total Profit': '1283508.34'}