Bilenkin540Week_1_&_2_Exercises

March 12, 2025

2. 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.

[331]:
```python
# Importiomg necessary library for the random module
import random

# Creating a list with random numbers
my_random_numbers_list = [5, 12, 3, 8, 1]

# Iterating over the list and printing each item
print("The original list of numbers:")
for item in my_random_numbers_list:
    print(item)

# Sorting the numbers in the list
my_random_numbers_list.sort()
print("\nSorted list:", my_random_numbers_list)

# Generating 3 random numbers and adding them to the list
random_numbers = [random.randint(1, 20) for _ in range(3)]
my_random_numbers_list.extend(random_numbers)

# Printing the updated list
print("\nUpdated list with added random numbers:", my_random_numbers_list)

# Sorting again and printing the final list with added random numbers
my_random_numbers_list.sort()
print("\nFinal sorted list:", my_random_numbers_list)
```

```
The original list of numbers:
5
12
3
8
1

Sorted list: [1, 3, 5, 8, 12]

Updated list with added random numbers: [1, 3, 5, 8, 12, 18, 2, 1]
```

```
Final sorted list: [1, 1, 2, 3, 5, 8, 12, 18]
```

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

[332]:
```python
import pandas as pd

# Defining the correct file path and storing into the variable for future use
downloaded_file_path = r"C:\Users\maxim\OneDrive\Desktop\BU\DSC␣
↪540\world-population.xlsm"

# Loading the Excel file
df = pd.read_excel(downloaded_file_path, engine="openpyxl")

# Displaying the first five rows to see the insight of the dataset
df.head()
```

[332]:
```
   Year  Population
0  1960  3028654024
1  1961  3068356747
2  1962  3121963107
3  1963  3187471383
4  1964  3253112403
```

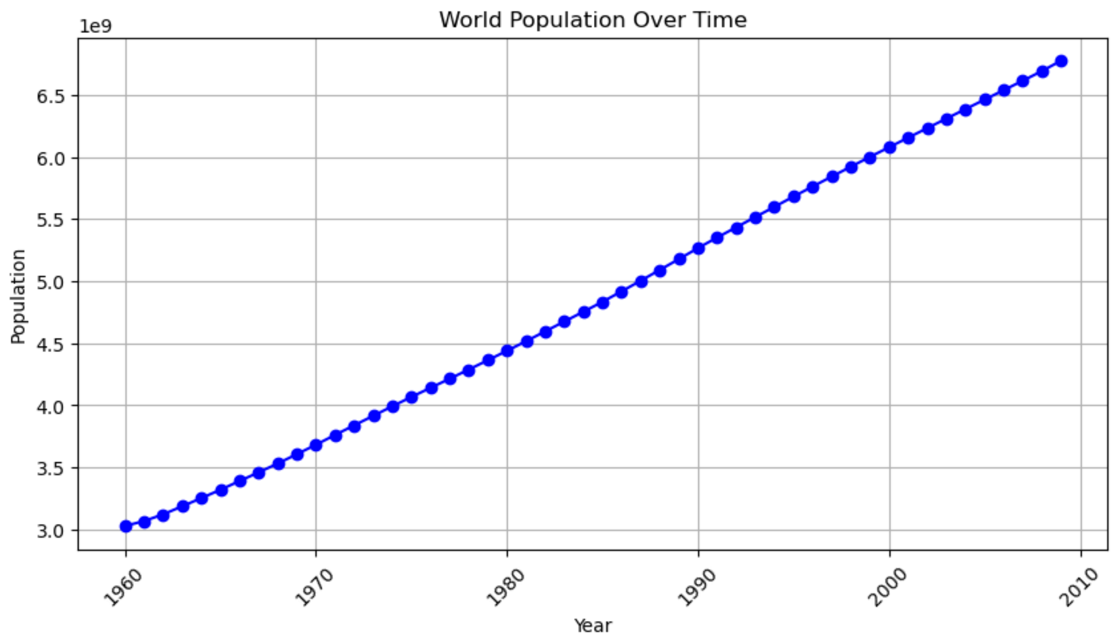
[333]:
```python
import matplotlib.pyplot as plt

# Plotting the data
plt.figure(figsize=(10, 5))
plt.plot(df["Year"], df["Population"], marker="o", linestyle="-", color="b")

# Adding labels and title to the graph
plt.xlabel("Year")
plt.ylabel("Population")
plt.title("World Population Over Time")

# Rotating x-axis labels for better readability
plt.xticks(rotation=45)

# Showing the grid and the plot
plt.grid(True)
plt.show()
```



4. **Complete the following activities from the The Data Wrangling Workshop text.**

   a. The Data Wrangling Workshop: Activity 1.01 page 23

   1. Create a list of 100 random numbers.

```
[334]: import random

       # Creating a list of 100 random numbers between 1 and 1000
       random_numbers_list = [random.randint(1, 1000) for _ in range(100)]

       # Displaying the first 10 numbers to verify
       random_numbers_list[:10]
```

```
[334]: [291, 913, 649, 907, 871, 847, 388, 741, 3, 786]
```

   2. Create a new list from this random list, with numbers that are divisibleby 3.

```
[335]: # Creating a new list with numbers divisible by 3
       divisible_by_3 = [num for num in random_numbers_list if num % 3 == 0]

       # Displaying the first 10 numbers that are divisible by 3
       divisible_by_3[:10]
```

```
[335]: [291, 741, 3, 786, 420, 78, 342, 405, 18, 621]
```

   3. Calculate the length of these two lists and store the difference in a new variable.

```
[336]: # Calculating the difference in length
       difference = len(random_numbers_list) - len(divisible_by_3)

       # Displaying the difference
       difference
```

```
[336]: 59
```

4. Using a loop, perform steps 1, 2, and 3, and find the difference variable 10 times.

```
[337]: # Performing the experiment 10 times
       differences = []

       for _ in range(10):
           random_numbers_list = [random.randint(1, 1000) for _ in range(100)]
           divisible_by_3 = [num for num in random_numbers_list if num % 3 == 0]
           difference = len(random_numbers_list) - len(divisible_by_3)
           differences.append(difference)

       # Display all 10 differences
       differences
```

```
[337]: [68, 71, 68, 64, 67, 65, 56, 70, 72, 68]
```

5. Find the arithmetic mean of these 10 difference values.The output (will vary with each run) should look similar to this:66.3

```
[338]: # Calculating the arithmetic mean of the 10 difference values
       average_difference = sum(differences) / len(differences)

       # Printing the final result
       print(f"Average difference over 10 runs: {average_difference:.1f}")
```

```
Average difference over 10 runs: 66.9
```

b. The Data Wrangling Workshop: Activity 1.02 page 43

1. Create a mutliline_text variable by copying the text from the first chapter of Pride and Prejudice.

```
[339]: import re

       # Creating the variable and storing Chapter I text.
       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 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?" "Bingley." "Is he married or single?" "Oh! Single,␣
  ↪my dear, to be sure! A single man of large fortune; four or five thousand
a year. What a fine thing for our girls!" "How so? How can it affect them?" "My␣
  ↪dear Mr. Bennet," replied his wife, "how can you be so tiresome! You must
know that I am thinking of his marrying one of them." "Is that his design in␣
  ↪settling here?" "Design! Nonsense, how can you talk so! But it is very
likely that he may fall in love with one of them, and therefore you must visit␣
  ↪him as soon as he comes." "I see no occasion for that. You and the girls
may go, or you may send them by themselves, which perhaps will be still better,␣
  ↪for as you are as handsome as any of them, Mr. Bingley may like you the
best of the party." "My dear, you flatter me. I certainly have had my share of␣
  ↪beauty, but I do not pretend to be anything extraordinary now. When a
woman has five grown-up daughters, she ought to give over thinking of her own␣
  ↪beauty." "In such cases, a woman has not often much beauty to think of."
"But, my dear, you must indeed go and see Mr. Bingley when he comes into the␣
  ↪neighbourhood." "It is more than I engage for, I assure you." "But consider
your daughters. Only think what an establishment it would be for one of them.␣
  ↪Sir William and Lady Lucas are determined to go, merely on that account,
for in general, you know, they visit no newcomers. Indeed you must go, for it␣
  ↪will be impossible for us to visit him if you do not." "You are
over-scrupulous, surely. I dare say Mr. Bingley will be very glad to see you;␣
  ↪and I will send a few lines by you to assure him of my hearty consent to
his marrying whichever he chooses of the girls; though I must throw in a good␣
  ↪word for my little Lizzy." "I desire you will do no such thing. Lizzy is
not a bit better than the others; and I am sure she is not half so handsome as␣
  ↪Jane, nor half so good-humoured as Lydia. But you are always giving her
the preference." "They have none of them much to recommend them," replied he;␣
  ↪"they are all silly and ignorant like other girls; but Lizzy has something
more of quickness than her sisters." "Mr. Bennet, how can you abuse your own␣
  ↪children in such a way? You take delight in vexing me. You have no
compassion for my poor nerves." "You mistake me, my dear. I have a high respect␣
  ↪for your nerves. They are my old friends. I have heard you mention
them with consideration these last twenty years at least." "Ah, you do not know␣
  ↪what I suffer." "But I hope you will get over it, and live to see many
```

```
young men of four thousand a year come into the neighbourhood." "It will be no␣
  ↪use to us, if twenty such should come, since you will not visit them."
"Depend upon it, my dear, that when there are twenty, I will visit them all."␣
  ↪Mr. Bennet was so odd a mixture of quick parts, sarcastic humour, reserve,
and caprice, that the experience of three-and-twenty years had been␣
  ↪insufficient to make his wife understand his character. Her mind was less␣
  ↪difficult
to develop. She was a woman of mean understanding, little information, and␣
  ↪uncertain  temper. When she was discontented, she fancied herself nervous.
The business of her life was to get her daughters married; its solace was␣
  ↪visiting and news. I hope Mr. Bingley will like it."""
```

2. Find the type and length of the multiline_text string using the type and len commands.

[340]:
```
# Checking the type of the variable
print("Type of multiline_text:", type(multiline_text))

# Checking the length or the total number of characters in the chapter I
print("Total Length of chachacter:", len(multiline_text))
```

```
Type of multiline_text: <class 'str'>
Total Length of chachacter: 4498
```

3. Remove all new lines and symbols using the replace method.

[341]:
```
# Removing all lines and symbols using regex
cleaned_text_file = re.sub(r"[^\w\s]", "", multiline_text).replace("\n", " ").
  ↪lower()
```

4. Find all of the words in multiline_text using the split method.

[342]:
```
# Splitting text into words
separated_words = cleaned_text_file.split()

# Displaying the first 10 words for confirmation
print(separated_words[:10])
```

```
['it', 'is', 'a', 'truth', 'universally', 'acknowledged', 'that', 'a', 'single',
'man']
```

5. Create a list from this list that will contain only the unique words.

[343]:
```
# Converting to a set which removes duplicates and back to a list
unique_words_list = list(set(separated_words))

#Displaying first 10 unique words for confirmation
print(unique_words_list[:10])
```

```
['pretend', 'park', 'take', 'mrs', 'returned', 'was', 'share', 'do', 'man',
'solace']
```

6. Count the number of times the unique word has appeared in the list using the key and value in dict.

[344]:
```python
# Creating an empty dictionary
word_count_list = {}

# Looping through all words
for word in separated_words:
    word_count_list[word] = word_count_list.get(word, 0) + 1

# Displaying first 10 word counts for confirmation
print(list(word_count_list.items())[:10])
```

```
[('it', 15), ('is', 14), ('a', 21), ('truth', 2), ('universally', 1),
('acknowledged', 1), ('that', 15), ('single', 4), ('man', 4), ('in', 12)]
```

c. The Data Wrangling Workshop: Activity 2.01 page 70

1. Look up the definition of permutations and dropwhile from itertools.

permutations: Generates all possible r-length permulations of the given iterable. Each permulation is returned as a tuple.

dropwhile: Drops elements from an iterable while the predicate is True, then returns the remaining elements one the predicate becomes False.

2. Write an expression to generate all the possible three-digit numbers, using 0, 1,and 2.

[345]:
```python
from itertools import permutations, dropwhile

# Generating all possible 3-digit permutations of (0, 1, 2)
perm_iterator = permutations([0, 1, 2], 3)

# Converting the iterator to a list to display
perm_list = list(perm_iterator)

# Printing all generated permutations for confirmation
print(perm_list)
```

```
[(0, 1, 2), (0, 2, 1), (1, 0, 2), (1, 2, 0), (2, 0, 1), (2, 1, 0)]
```

3. Loop over the iterator expression you generated before. Print each elementreturned by the iterator. Use assert and isinstance to make sure that theelements are of the tuple type.

[346]:
```python
# Looping over each permutation and checking its type
for perm in perm_list:
    assert isinstance(perm, tuple), "Element is not a tuple!"

# Displaying the message only if all the elements are tuples
print("All elements are tuples.")
```

```
All elements are tuples.
```

4. Write the loop again, using dropwhile, with a lambda expression to drop anyleading zeros from the tuples. As an example, (0, 1, 2) will become [0, 2]. Also, cast the output of dropwhile to a list.

[347]:
```python
# Using dropwhile to remove leading zeros
filtered_perms = [list(dropwhile(lambda x: x == 0, perm)) for perm in perm_list]

# Displaying the transformed list
print(filtered_perms)
```

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

5. Check the actual type that dropwhile returns.

[348]:
```python
# Checking the type of one transformed permutation
print("The type dropwhile returns is:", type(filtered_perms[0]))
```

```
The type dropwhile returns is: <class 'list'>
```

6. Combine the preceding code into one block; this time, write a separate function where you will pass the list generated from dropwhile and the function will return the whole number contained in the list. As an example, if you pass [1,2] to the function, it will return 12. Make sure that the return type is indeed a number and not a string. Although this task can be achieved using other tricks, treat the incoming list as a stack in the function and generate the number by reading the individual digits from the stack.

[349]:
```python
# Function to convert a list of digits into a number
def list_to_number(digit_list):
    return int("".join(map(str, digit_list)))

# Applying the function directly while filtering
number_list = [list_to_number(lst) for lst in filtered_perms]

# Displaying the final results
print(number_list)
```

```
[12, 21, 102, 120, 201, 210]
```

d. The Data Wrangling Workshop: Activity 2.02 page 81

1. Import zip_longest from itertools. Create a function to zip header, line, and fillvalue=None.Open the accompanying sales_record.csv file from the GitHub link(https://packt.live/2Yb6iCh) by using r mode inside a with block and check that itis opened.

[350]:
```python
# Importing zip_longest from itertools
from itertools import zip_longest

# Defining the file path
file_path = r"C:\Users\maxim\OneDrive\Desktop\BU\DSC 540\sales_record.csv"
```

```python
# Opening the file to check if it is accessible
with open(file_path, "r") as file:
    print("File opened successfully!")
```

```
File opened successfully!
```

2. Read the first line and use string methods to generate a list of all the column names.

[351]:
```python
# Reading the first line to extract column names
with open(file_path, "r") as file:

    # Generating a list of column names
    headers = file.readline().strip().split(",")

# Displaying the extracted headers
print("Headers:", headers)
```

```
Headers: ['Region', 'Country', 'Item Type', 'Sales Channel', 'Order Priority',
'Order Date', 'Order ID', 'Ship Date', 'Units Sold', 'Unit Price', 'Unit Cost',
'Total Revenue', 'Total Cost', 'Total Profit']
```

3. Start reading the file. Read it line by line.

[352]:
```python
# Creating a list to store the data
data = []

with open(file_path, "r") as file:

    # Extracting headers
    headers = file.readline().strip().split(",")

    for line in file:
        row_dict = parse_csv_line(headers, line)
        data.append(row_dict)
```

4. Read each line and pass that line to a function, along with the list of the headers. The work of the function is to construct a dictionary out of these two and fill up the key: values variables. Keep in mind that a missing value should result in None.

[353]:
```python
# Function to parse a line into a dictionary using headers as keys
def parse_csv_line(headers, line):
    # Splitting the line into values
    values = line.strip().split(",")

    # Handling missing values by using zip_longest
    row_dict = dict(zip_longest(headers, values, fillvalue=None))
    return row_dict

# Processing the file and converting lines into dictionaries
```

```python
data = []

with open(file_path, "r") as file:

    # Extracting headers
    headers = file.readline().strip().split(",")
    for line in file:

        # Parsing each line
        row_dict = parse_csv_line(headers, line)
        data.append(row_dict)

# Displaying the first 5 rows
print(data[:5])
```

```
[{'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': '226716.10', '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': '205.70', 'Unit Cost':
'117.11', 'Total Revenue': '1854591.20', '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'}]
```