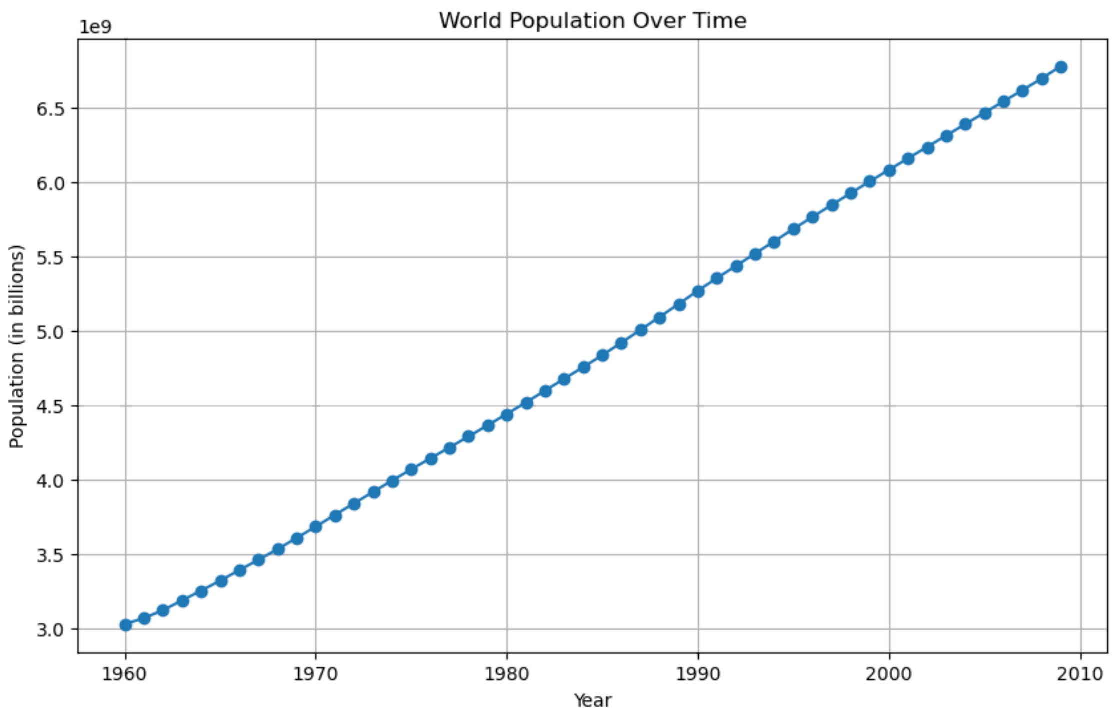
# DataPrep Assignment 1

March 24, 2024

[44]:
```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the Excel file
file_path = '/Users/nickblackford/Desktop/Python/world-population.xlsm'
df = pd.read_excel(file_path)
```

[45]:
```python
# Plotting the graph
plt.figure(figsize=(10, 6))
plt.plot(df['Year'], df['Population'], marker='o')
plt.title('World Population Over Time')
plt.xlabel('Year')
plt.ylabel('Population (in billions)')
plt.grid(True)
plt.show()
```



## 1 Activity 1.01

[46]:
```
#Generate a random list of 100 numbers
import random
random_number_list = [random.randint(0, 100) for x in range(0, 100)]
random_number_list
```

[46]:
```
[13,
 44,
 4,
 23,
 14,
 85,
 67,
 39,
 77,
 22,
 47,
 78,
 60,
 22,
 73,
 47,
 96,
 15,
 82,
 59,
 4,
 23,
 62,
 3,
 6,
 46,
 89,
 0,
 57,
 6,
 81,
 39,
 9,
 52,
 8,
 10,
 52,
48,
56,
98,
56,
32,
88,
82,
47,
58,
87,
60,
48,
26,
42,
100,
17,
1,
43,
94,
34,
62,
44,
95,
19,
41,
74,
97,
67,
53,
75,
93,
68,
98,
87,
77,
96,
28,
58,
36,
81,
33,
48,
42,
88,
36,
2,
82,
 25,
 89,
 33,
 84,
 72,
 99,
 63,
 49,
 29,
 81,
 41,
 76,
 24,
 12,
 78,
 31]
```

[47]:
```
#Create a new list of numbers in random_number_list that are divisible by 3
list_with_divisible_by_3 = [a for a in random_number_list if a % 3 == 0]
list_with_divisible_by_3
```

[47]:
```
[39,
 78,
 60,
 96,
 15,
 3,
 6,
 0,
 57,
 6,
 81,
 39,
 9,
 48,
 87,
 60,
 48,
 42,
 75,
 93,
 87,
 96,
 36,
 81,
 33,
 48,
 42,
 36,
 33,
 84,
 72,
 99,
 63,
 81,
 24,
 12,
 78]
```

[48]:
```
#Calculate the length of both lists and store the 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
difference
```

[48]:
```
63
```

[49]:
```
#Repeat and iterate 10 times
NUMBER_OF_EXPERIMENTS = 10
difference_list = []
for i in range(0, NUMBER_OF_EXPERIMENTS):
    random_number_list = [random.randint(0, 100) for x in range(0, 100)]
    list_with_divisible_by_3 = [a for a in random_number_list if a % 3 == 0]

    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
    difference_list.append(difference)
difference_list
```

[49]:
```
[62, 74, 62, 66, 61, 70, 68, 74, 62, 66]
```

[50]:
```
avg_diff = sum(difference_list) / float(len(difference_list))
avg_diff
```

[50]:
```
66.5
```

## 2 Activity 1.02

[51]:
```
multiline_text = '''The wave crashed and hit the sandcastle head-on. The␣
↪sandcastle began to melt under the waves force and as the wave receded, half␣
↪the sandcastle was gone. The next wave hit, not quite as strong, but still␣
↪managed to cover the remains of the sandcastle and take more of it away. The␣
↪third wave, a big one, crashed over the sandcastle completely covering and␣
↪engulfing it. When it receded, there was no trace the sandcastle ever␣
↪existed and hours of hard work disappeared forever.'''
```

[52]:
```
#type
type(multiline_text)
```

[52]:
```
str
```

[53]:
```
#len
len(multiline_text)
```

[53]:
```
478
```

[54]:
```
# Get ride of line breaks
multiline_text = multiline_text.replace('\n', "")
```

[55]:
```
multiline_text
```

[55]:
```
'The wave crashed and hit the sandcastle head-on. The sandcastle began to melt
under the waves force and as the wave receded, half the sandcastle was gone. The
next wave hit, not quite as strong, but still managed to cover the remains of
the sandcastle and take more of it away. The third wave, a big one, crashed over
the sandcastle completely covering and engulfing it. When it receded, there was
no trace the sandcastle ever existed and hours of hard work disappeared
forever.'
```

[56]:
```
# Remove punctuation
cleaned_multiline_text = ""
for char in multiline_text:
    if char == " ":
        cleaned_multiline_text += char
    elif char.isalnum():  # using the isalnum() method of strings.
        cleaned_multiline_text += char
    else:
        cleaned_multiline_text += " "
```

[57]:
```
cleaned_multiline_text
```

```
[57]: 'The wave crashed and hit the sandcastle head on  The sandcastle began to melt
      under the waves force and as the wave receded  half the sandcastle was gone  The
      next wave hit  not quite as strong  but still managed to cover the remains of
      the sandcastle and take more of it away  The third wave  a big one  crashed over
      the sandcastle completely covering and engulfing it  When it receded  there was
      no trace the sandcastle ever existed and hours of hard work disappeared forever
      '
```

```
[58]: #put words into a list
      list_of_words = cleaned_multiline_text.split()
      list_of_words
```

```
[58]: ['The',
       'wave',
       'crashed',
       'and',
       'hit',
       'the',
       'sandcastle',
       'head',
       'on',
       'The',
       'sandcastle',
       'began',
       'to',
       'melt',
       'under',
       'the',
       'waves',
       'force',
       'and',
       'as',
       'the',
       'wave',
       'receded',
       'half',
       'the',
       'sandcastle',
       'was',
       'gone',
       'The',
       'next',
       'wave',
       'hit',
       'not',
       'quite',
       'as',
       'strong',
       'but',
       'still',
       'managed',
       'to',
       'cover',
       'the',
       'remains',
       'of',
       'the',
       'sandcastle',
       'and',
       'take',
       'more',
       'of',
       'it',
       'away',
       'The',
       'third',
       'wave',
       'a',
       'big',
       'one',
       'crashed',
       'over',
       'the',
       'sandcastle',
       'completely',
       'covering',
       'and',
       'engulfing',
       'it',
       'When',
       'it',
       'receded',
       'there',
       'was',
       'no',
       'trace',
       'the',
       'sandcastle',
       'ever',
       'existed',
       'and',
       'hours',
       'of',
       'hard',
       'work',
       'disappeared',
       'forever']
```

[59]:
```
# calc number of words
len(list_of_words)
```

[59]:
```
85
```

[60]:
```
#find unique number of words
unique_words_as_dict = dict.fromkeys(list_of_words)
len(list(unique_words_as_dict.keys()))
```

[60]:
```
53
```

[61]:
```
#Create dictionary with word, number of occurrences
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
unique_words_as_dict
```

[61]:
```
{'The': 4,
 'wave': 4,
 'crashed': 2,
 'and': 5,
 'hit': 2,
 'the': 8,
 'sandcastle': 6,
 'head': 1,
 'on': 1,
 'began': 1,
 'to': 2,
 'melt': 1,
 'under': 1,
 'waves': 1,
 'force': 1,
 'as': 2,
 'receded': 2,
 'half': 1,
 'was': 2,
 'gone': 1,
 'next': 1,
 'not': 1,
 'quite': 1,
 'strong': 1,
 'but': 1,
 'still': 1,
 'managed': 1,
 'cover': 1,
 'remains': 1,
 'of': 3,
 'take': 1,
 'more': 1,
 'it': 3,
 'away': 1,
 'third': 1,
 'a': 1,
 'big': 1,
 'one': 1,
 'over': 1,
 'completely': 1,
 'covering': 1,
 'engulfing': 1,
 'When': 1,
 'there': 1,
 'no': 1,
 'trace': 1,
 'ever': 1,
 'existed': 1,
 'hours': 1,
 'hard': 1,
 'work': 1,
 'disappeared': 1,
 'forever': 1}
```

[62]:
```
# sort by most frequently used words
top_words = sorted(unique_words_as_dict.items(), key=lambda key_val_tuple:␣
 ↪key_val_tuple[1], reverse=True)
top_words[:25]
```

[62]:
```
[('the', 8),
 ('sandcastle', 6),
 ('and', 5),
 ('The', 4),
 ('wave', 4),
 ('of', 3),
 ('it', 3),
 ('crashed', 2),
 ('hit', 2),
 ('to', 2),
 ('as', 2),
 ('receded', 2),
 ('was', 2),
 ('head', 1),
 ('on', 1),
 ('began', 1),
 ('melt', 1),
 ('under', 1),
 ('waves', 1),
 ('force', 1),
 ('half', 1),
 ('gone', 1),
 ('next', 1),
 ('not', 1),
 ('quite', 1)]
```

## 3 Activity 2.01

[63]:
```
#definitions of permuatations and dropwhile
from itertools import permutations, dropwhile
permutations?
dropwhile?
```

[64]:
```
# all possible 3 digit numbers with 0,1 and 2
for number_tuple in permutations(range(3)):
    print(number_tuple)
    assert isinstance(number_tuple, tuple)
```

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

[65]:
```
# drop leading 0s
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]
```

[66]:
```python
# loop to convert into number format
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

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
```

## 4 Activity 2.02

[67]:
```python
from itertools import zip_longest
```

[68]:
```python
def return_dict_from_csv_line(header, line):
    # Zip
    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
```

[69]:
```python
# Read first line
with open("/Users/nickblackford/Downloads/sales_record.csv", "r") as fd:
    first_line = fd.readline()
    header = first_line.replace("\n", "").split(",")
# Iterate 10 times
    for i, line in enumerate(fd):
        line = line.replace("\n", "").split(",")
        d = return_dict_from_csv_line(header, line)
        print(d)
        if i > 10:
            break
```

```
{'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'}
{'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': '467630.00'}
{'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'}
```

[ ]: