DCS 540 Data Preparation (DSC540-T301 2225-1)

Bellevue University

Assignment: Week 1 & 2 Exercises

Author: Jake Meyer

Date: 03/21/2022

Assignment Instructions:

1. Install the latest versions of either Docker or Anaconda. Your book Data Wrangling with Python uses Docker, however, you are welcome to use whichever distributer you feel comfortable with.
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.
3. Create a line chart with Matplotlib and the following data file.

- a. Data file: world-population.xlsm
- b. (Hint: Python for Data Analysis: Page 19-50 & Data Wrangling with Python: Preface)

1. Complete the following activities:

- a. Data Wrangling with Python: Activity 1 page 17
- b. Data Wrangling with Python: Activity 2 page 31
- c. Data Wrangling with Python: Activity 3 page 49
- d. Data Wrangling with Python: Activity 4 page 59

1. Install the latest versions of either docker or Anaconda. Your book Data Wrangling with Python uses Docker, however, you are welcome to use whichever distributer you feel comfortable with.

I've previously installed Anaconda from a preceeding course and have chosen to use this distributer for the first assignment. However, I've also downloaded Docker to try to get more familiar with it throughout this course.

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.

In [1]:
```
# Create a list (from 0 to 30)
```

```
my_list_1 = [x for x in range(0,31)]
my_list_1
```

Out[1]:
```
[0,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10,
 11,
 12,
 13,
 14,
 15,
 16,
 17,
 18,
 19,
 20,
 21,
 22,
 23,
 24,
 25,
 26,
 27,
 28,
 29,
 30]
```

In [2]:
```
# Iterate over the list

i = 0
while i < len(my_list_1):
    i += 1

# Sort your results (in reverse)

my_list_1.sort(reverse = True)
my_list_1
```

Out[2]:
```
[30,
 29,
 28,
 27,
 26,
 25,
 24,
 23,
 22,
 21,
 20,
 19,
 18,
```

```
17,
16,
15,
14,
13,
12,
11,
10,
9,
8,
7,
6,
5,
4,
3,
2,
1,
0]
```

In [3]:

```
# Generate random numbers

import random

random_list = [random.randint(0,30) for x in range(0,50)]
random_list
```

Out[3]:

```
[25,
 7,
 21,
 12,
 12,
 18,
 30,
 27,
 1,
 0,
 12,
 11,
 22,
 9,
 30,
 16,
 12,
 25,
 23,
 24,
 7,
 25,
 16,
 2,
 23,
 21,
 4,
 24,
 18,
 17,
 0,
 5,
 5,
 23,
```

```
0,
27,
20,
28,
26,
14,
1,
19,
22,
5,
15,
20,
29,
16,
30,
4,
18]
```

In [4]:

```
# Add random numbers to the original list

i = 0
while i < len(random_list):
    my_list_1.append(random_list[i])
    i += 1

# Print your results

print(my_list_1)
```

```
[30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9,
8, 7, 6, 5, 4, 3, 2, 1, 0, 25, 7, 21, 12, 18, 30, 27, 1, 0, 12, 11, 22, 9, 30, 16, 12, 2
5, 23, 24, 7, 25, 16, 2, 23, 21, 4, 24, 18, 17, 0, 5, 5, 23, 0, 27, 20, 28, 26, 14, 1, 1
9, 22, 5, 15, 20, 29, 16, 30, 4, 18]
```

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

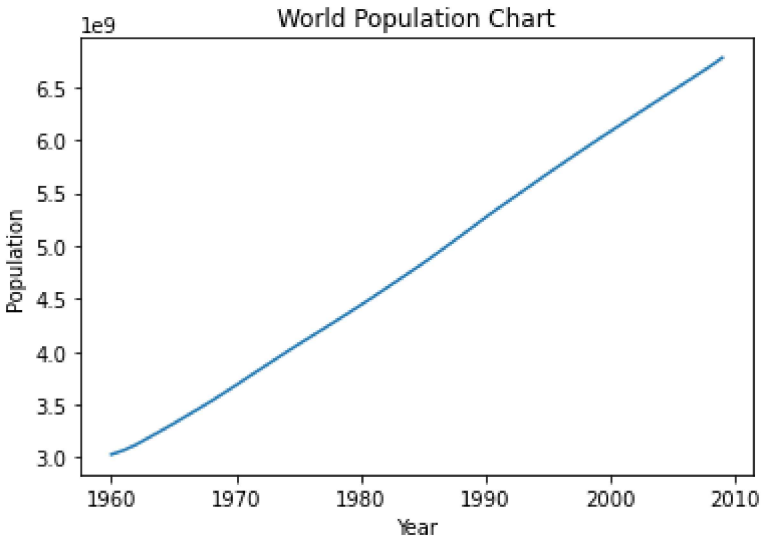
- a. Data file: world-population.xlsm
- b. (Hint: Python for Data Analysis: Page 19-50 & Data Wrangling with Python: Preface)

In [5]:

```
# Import pandas and matplotlib

import pandas as pd
%matplotlib inline
import matplotlib.pyplot as plt

# Read the Data File and store as data frame

df = pd.read_excel(r'C:\Users\jkmey\Documents\Github\DSC540_Course_Assignments\DSC540_W

# Create the line plot using matplotlib

plt.plot(df["Year"],df["Population"])
plt.xlabel('Year')
plt.ylabel('Population')
plt.title('World Population Chart')
```

Out[5]:

```
Text(0.5, 1.0, 'World Population Chart')
```



4a. Data Wrangling with Python: Activity 1 page 17

1. Create a list of 100 random numbers.
2. Create a new list from this random list, with numbers that are divisible by 3.
3. Calculate the length of these two lists and store the difference in a new variable.
4. Using a loop, perform steps 2 and 3 and find the difference variable three times.
5. Find the arithmetic mean of these three difference values.

In [6]:
```
# Create a list of 100 random numbers

random_numbers = [random.randint(0,100) for x in range(0,100)]
```

In [7]:
```
# Create a new list from this random list with numbers that are divisible by 3
random_numbers_2 = [x for x in random_numbers if x % 3 == 0]
```

In [8]:
```
# Calculate the length of these two lists and store the difference in a new variable
length_1 = len(random_numbers)
print("Length of list 1 is:", length_1)
length_2 = len(random_numbers_2)
print("Length of list 2 is:", length_2)
length_dif = length_1 - length_2
print("Length difference is:", length_dif)
```

```
Length of list 1 is: 100
Length of list 2 is: 40
Length difference is: 60
```

In [9]:
```
# Using a loop, perform steps 2 and 3 and find the difference variable three times

i= 0
difference_output = []
for i in range(0,3):
    random_numbers = [random.randint(0,100) for x in range(0,100)]
    random_numbers_2 = [x for x in random_numbers if x % 3 == 0]
    length_1 = len(random_numbers)
```

```
    length_2 = len(random_numbers_2)
    length_dif = length_1 - length_2
    difference_output.append(length_dif)

print("The difference between lists are shown here:",difference_output)
```

```
The difference between lists are shown here: [60, 65, 69]
```

In [10]:

```
# Find the arithmetic mean of these three difference values
from statistics import mean
mean_val = mean(difference_output)
print("Mean for the difference between lists:", round(mean_val, 2))
```

```
Mean for the difference between lists: 64.67
```

4b. Data Wrangling with Python: Activity 2 page 31

1. Create a multiline_text variable by copying the text from the first chapter of Pride and Prejudice
2. Find the type and length of the multiline_text string using the commands type and len
3. Remove all new lines and symbols using the replace function
4. Find all the words in multiline_text using the split function
5. Create a list from this list that will contain only the unique words
6. Count the number of times the unique word has appeared in the list using the key and value in dict
7. Find the top 25 words from the unique words that you have found using the slice function

In [11]:

```
# Create a multiline_text varible by copying the text from the first chapter of Pride a
# Text was taken from Data Wrangling with Python text Github Repository Chapter01/Activ

multiline_text = """It is a truth universally acknowledged, that a single man in posses

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

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

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

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 o

"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

"How so? How can it affect them?"

"My dear Mr. Bennet," replied his wife, "how can you be so tiresome! You must know that

"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

"I see no occasion for that. You and the girls may go, or you may send them by themselv

"My dear, you flatter me. I certainly have had my share of beauty, but I do not pretend

"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 neighbourh

"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 t

"You are over-scrupulous, surely. I dare say Mr. Bingley will be very glad to see you;

"I desire you will do no such thing. Lizzy is not a bit better than the others; and I a

"They have none of them much to recommend them," replied he; "they are all silly and ig

"Mr. Bennet, how can you abuse your own children in such a way? You take delight in vex

"You mistake me, my dear. I have a high respect for your nerves. They are my old friend

"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 yea

"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,
```

```
multiline_text
```

Out[11]:

```
'It is a truth universally acknowledged, that a single man in possession of a good fortu
ne, must be in want of a wife.\n\nHowever little known the feelings or views of such a m
an may be on his first entering a neighbourhood, this truth is so well fixed in the mind
s of the surrounding families, that he is considered the rightful property of some one o
r other of their daughters.\n\n"My dear Mr. Bennet," said his lady to him one day, "have
you heard that Netherfield Park is let at last?"\n\nMr. Bennet replied that he had no
t.\n\n"But it is," returned she; "for Mrs. Long has just been here, and she told me all
about it."\n\nMr. Bennet made no answer.\n\n"Do you not want to know who has taken it?"
cried his wife impatiently.\n\n"You want to tell me, and I have no objection to hearing
it."\n\nThis was invitation enough.\n\n"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 wit
h it, that he agreed with Mr. Morris immediately; that he is to take possession before M
ichaelmas, and some of his servants are to be in the house by the end of next week."\n
\n"What is his name?"\n\n"Bingley."\n\n"Is he married or single?"\n\n"Oh! Single, my dea
```

```
r, to be sure! A single man of large fortune; four or five thousand a year. What a fine
thing for our girls!"\n\n"How so? How can it affect them?"\n\n"My dear Mr. Bennet," repl
ied his wife, "how can you be so tiresome! You must know that I am thinking of his marry
ing one of them."\n\n"Is that his design in settling here?"\n\n"Design! Nonsense, how ca
n you talk so! But it is very likely that he may fall in love with one of them, and ther
efore you must visit him as soon as he comes."\n\n"I see no occasion for that. You and t
he 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 part
y."\n\n"My dear, you flatter me. I certainly have had my share of beauty, but I do not p
retend to be anything extraordinary now. When a woman has five grown-up daughters, she o
ught to give over thinking of her own beauty."\n\n"In such cases, a woman has not often
much beauty to think of."\n\n"But, my dear, you must indeed go and see Mr. Bingley when
he comes into the neighbourhood."\n\n"It is more than I engage for, I assure you."\n\n"B
ut consider your daughters. Only think what an establishment it would be for one of the
m. Sir William and Lady Lucas are determined to go, merely on that account, for in gener
al, you know, they visit no newcomers. Indeed you must go, for it will be impossible for
us to visit him if you do not."\n\n"You are over-scrupulous, surely. I dare say Mr. Bing
ley 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."\n\n"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."\n\n"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."\n\n"Mr. Benne
t, 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."\n\n"You mistake me, my dear. I have a high respe
ct for your nerves. They are my old friends. I have heard you mention them with consider
ation these last twenty years at least."\n\n"Ah, you do not know what I suffer."\n\n"But
I hope you will get over it, and live to see many young men of four thousand a year come
into the neighbourhood."\n\n"It will be no use to us, if twenty such should come, since
you will not visit them."\n\n"Depend upon it, my dear, that when there are twenty, I wil
l visit them all."\n\nMr. Bennet was so odd a mixture of quick parts, sarcastic humour,
reserve, and caprice, that the experience of three-and-twenty years had been insufficien
t 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 wa
s discontented, she fancied herself nervous. The business of her life was to get her dau
ghters married; its solace was visiting and news.'
```

In [12]:

```
# Find the type and length of the multiline_text string using the commands type and len

text_type = type(multiline_text)
text_length = len(multiline_text)
print("multiline_text type is {} and length is {}:".format(text_type, text_length))
```

```
multiline_text type is <class 'str'> and length is 4474:
```

In [13]:

```
# Remove all new lines and symbols using the replace function

multiline_text = multiline_text.replace("\n", "")
multiline_text_clean = ""

for char in multiline_text:
    if char == " ":
        multiline_text_clean += char
    elif char.isalnum():
        multiline_text_clean += char
    else:
        multiline_text_clean
```

```
multiline_text_clean
```

Out[13]:
```
'It is a truth universally acknowledged that a single man in possession of a good fortun
e must be in want of a wifeHowever 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 daughtersMy dear Mr Bennet said his lady to him one day have you heard that Nether
field Park is let at lastMr Bennet replied that he had notBut it is returned she for Mrs
Long has just been here and she told me all about itMr Bennet made no answerDo you not w
ant to know who has taken it cried his wife impatientlyYou want to tell me and I have no
objection to hearing itThis was invitation enoughWhy 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 w
ith it that he agreed with Mr Morris immediately that he is to take possession before Mi
chaelmas and some of his servants are to be in the house by the end of next weekWhat is
his nameBingleyIs he married or singleOh Single my dear to be sure A single man of large
fortune four or five thousand a year What a fine thing for our girlsHow so How can it af
fect themMy dear Mr Bennet replied his wife how can you be so tiresome You must know tha
t I am thinking of his marrying one of themIs that his design in settling hereDesign Non
sense how can you talk so But it is very likely that he may fall in love with one of the
m and therefore you must visit him as soon as he comesI see no occasion for that You and
the girls may go or you may send them by themselves which perhaps will be still better f
or as you are as handsome as any of them Mr Bingley may like you the best of the partyMy
dear you flatter me I certainly have had my share of beauty but I do not pretend to be a
nything extraordinary now When a woman has five grownup daughters she ought to give over
thinking of her own beautyIn such cases a woman has not often much beauty to think ofBut
my dear you must indeed go and see Mr Bingley when he comes into the neighbourhoodIt is
more than I engage for I assure youBut consider your daughters Only think what an establ
ishment it would be for one of them Sir William and Lady Lucas are determined to go mere
ly on that account for in general you know they visit no newcomers Indeed you must go fo
r it will be impossible for us to visit him if you do notYou are overscrupulous surely I
dare say Mr Bingley will be very glad to see you and I will send a few lines by you to a
ssure 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 LizzyI 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 no
r half so goodhumoured as Lydia But you are always giving her the preferenceThey have no
ne 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 sistersMr Bennet how can you ab
use your own children in such a way You take delight in vexing me You have no compassion
for my poor nervesYou 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 leastAh you do not know what I sufferBut I hope you will get over it and live to see
many young men of four thousand a year come into the neighbourhoodIt will be no use to u
s if twenty such should come since you will not visit themDepend upon it my dear that wh
en there are twenty I will visit them allMr Bennet was so odd a mixture of quick parts s
arcastic humour reserve and caprice that the experience of threeandtwenty years had been
insufficient to make his wife understand his character Her mind was less difficult to de
velop 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'
```

In [14]:
```
# Find all the words in multiline_text using the split function

multiline_text_words = multiline_text_clean.split()
len(multiline_text_words)
```

Out[14]:
```
814
```

In [15]:
```
# Create a list from this list that will contain only the unique words

multiline_text_words_unique_dict = dict.fromkeys(multiline_text_words)
multiline_text_words_unique_list = list(multiline_text_words_unique_dict)
len(multiline_text_words_unique_list)
```

Out[15]:
```
349
```

In [16]:
```
# Count the number of times the unique word has appeared in the list using the key and

for word in multiline_text_words:
    if multiline_text_words_unique_dict[word] is None:
        multiline_text_words_unique_dict[word] = 1
    else:
        multiline_text_words_unique_dict[word] += 1
multiline_text_words_unique_dict
```

Out[16]:
```
{'It': 1,
 'is': 12,
 'a': 20,
 'truth': 2,
 'universally': 1,
 'acknowledged': 1,
 'that': 15,
 'single': 2,
 'man': 4,
 'in': 11,
 'possession': 2,
 'of': 28,
 'good': 2,
 'fortune': 3,
 'must': 7,
 'be': 11,
 'want': 3,
 'wifeHowever': 1,
 'little': 3,
 'known': 1,
 'the': 17,
 'feelings': 1,
 'or': 5,
 'views': 1,
 'such': 5,
 'may': 5,
 'on': 3,
 'his': 11,
 'first': 1,
 'entering': 1,
 'neighbourhood': 1,
 'this': 1,
 'so': 8,
 'well': 1,
 'fixed': 1,
 'minds': 1,
 'surrounding': 1,
 'families': 1,
 'he': 11,
 'considered': 1,
 'rightful': 1,
```

```
'property': 1,
'some': 2,
'one': 5,
'other': 2,
'their': 1,
'daughtersMy': 1,
'dear': 8,
'Mr': 6,
'Bennet': 6,
'said': 1,
'lady': 1,
'to': 22,
'him': 4,
'day': 1,
'have': 7,
'you': 23,
'heard': 2,
'Netherfield': 2,
'Park': 1,
'let': 1,
'at': 2,
'lastMr': 1,
'replied': 3,
'had': 3,
'notBut': 1,
'it': 9,
'returned': 1,
'she': 6,
'for': 12,
'Mrs': 2,
'Long': 2,
'has': 5,
'just': 1,
'been': 2,
'here': 1,
'and': 16,
'told': 1,
'me': 5,
'all': 2,
'about': 1,
'itMr': 1,
'made': 1,
'no': 7,
'answerDo': 1,
'not': 7,
'know': 5,
'who': 1,
'taken': 2,
'cried': 1,
'wife': 3,
'impatientlyYou': 1,
'tell': 1,
'I': 15,
'objection': 1,
'hearing': 1,
'itThis': 1,
'was': 8,
'invitation': 1,
'enoughWhy': 1,
'my': 10,
```

```
'says': 1,
'by': 4,
'young': 2,
'large': 2,
'from': 1,
'north': 1,
'England': 1,
'came': 1,
'down': 1,
'Monday': 1,
'chaise': 1,
'four': 3,
'see': 5,
'place': 1,
'much': 3,
'delighted': 1,
'with': 4,
'agreed': 1,
'Morris': 1,
'immediately': 1,
'take': 2,
'before': 1,
'Michaelmas': 1,
'servants': 1,
'are': 8,
'house': 1,
'end': 1,
'next': 1,
'weekWhat': 1,
'nameBingleyIs': 1,
'married': 2,
'singleOh': 1,
'Single': 1,
'sure': 2,
'A': 1,
'five': 2,
'thousand': 2,
'year': 2,
'What': 1,
'fine': 1,
'thing': 2,
'our': 1,
'girlsHow': 1,
'How': 1,
'can': 4,
'affect': 1,
'themMy': 1,
'how': 3,
'tiresome': 1,
'You': 4,
'am': 2,
'thinking': 2,
'marrying': 2,
'themIs': 1,
'design': 1,
'settling': 1,
'hereDesign': 1,
'Nonsense': 1,
'talk': 1,
'But': 2,
```

```
'very': 2,
'likely': 1,
'fall': 1,
'love': 1,
'them': 8,
'therefore': 1,
'visit': 5,
'as': 7,
'soon': 1,
'comesI': 1,
'occasion': 1,
'girls': 3,
'go': 4,
'send': 2,
'themselves': 1,
'which': 1,
'perhaps': 1,
'will': 9,
'still': 1,
'better': 2,
'handsome': 2,
'any': 1,
'Bingley': 3,
'like': 2,
'best': 1,
'partyMy': 1,
'flatter': 1,
'certainly': 1,
'share': 1,
'beauty': 2,
'but': 2,
'do': 4,
'pretend': 1,
'anything': 1,
'extraordinary': 1,
'now': 1,
'When': 2,
'woman': 3,
'grownup': 1,
'daughters': 3,
'ought': 1,
'give': 1,
'over': 2,
'her': 5,
'own': 2,
'beautyIn': 1,
'cases': 1,
'often': 1,
'think': 2,
'ofBut': 1,
'indeed': 1,
'when': 2,
'comes': 1,
'into': 2,
'neighbourhoodIt': 2,
'more': 2,
'than': 3,
'engage': 1,
'assure': 2,
'youBut': 1,
```

```
'consider': 1,
'your': 3,
'Only': 1,
'what': 2,
'an': 1,
'establishment': 1,
'would': 1,
'Sir': 1,
'William': 1,
'Lady': 1,
'Lucas': 1,
'determined': 1,
'merely': 1,
'account': 1,
'general': 1,
'they': 2,
'newcomers': 1,
'Indeed': 1,
'impossible': 1,
'us': 2,
'if': 2,
'notYou': 1,
'overscrupulous': 1,
'surely': 1,
'dare': 1,
'say': 1,
'glad': 1,
'few': 1,
'lines': 1,
'hearty': 1,
'consent': 1,
'whichever': 1,
'chooses': 1,
'though': 1,
'throw': 1,
'word': 1,
'LizzyI': 1,
'desire': 1,
'Lizzy': 2,
'bit': 1,
'others': 1,
'half': 2,
'Jane': 1,
'nor': 1,
'goodhumoured': 1,
'Lydia': 1,
'always': 1,
'giving': 1,
'preferenceThey': 1,
'none': 1,
'recommend': 1,
'silly': 1,
'ignorant': 1,
'something': 1,
'quickness': 1,
'sistersMr': 1,
'abuse': 1,
'children': 1,
'way': 1,
'delight': 1,
```

```
'vexing': 1,
'compassion': 1,
'poor': 1,
'nervesYou': 1,
'mistake': 1,
'high': 1,
'respect': 1,
'nerves': 1,
'They': 1,
'old': 1,
'friends': 1,
'mention': 1,
'consideration': 1,
'these': 1,
'last': 1,
'twenty': 3,
'years': 2,
'leastAh': 1,
'sufferBut': 1,
'hope': 1,
'get': 2,
'live': 1,
'many': 1,
'men': 1,
'come': 2,
'use': 1,
'should': 1,
'since': 1,
'themDepend': 1,
'upon': 1,
'there': 1,
'allMr': 1,
'odd': 1,
'mixture': 1,
'quick': 1,
'parts': 1,
'sarcastic': 1,
'humour': 1,
'reserve': 1,
'caprice': 1,
'experience': 1,
'threeandtwenty': 1,
'insufficient': 1,
'make': 1,
'understand': 1,
'character': 1,
'Her': 1,
'mind': 1,
'less': 1,
'difficult': 1,
'develop': 1,
'She': 1,
'mean': 1,
'understanding': 1,
'information': 1,
'uncertain': 1,
'temper': 1,
'discontented': 1,
'fancied': 1,
'herself': 1,
```

```
 'nervous': 1,
 'The': 1,
 'business': 1,
 'life': 1,
 'its': 1,
 'solace': 1,
 'visiting': 1,
 'news': 1}
```

In [17]:

```
# Find the top 25 words from the unique words that you have found using the slice funct

top_25_words = sorted(multiline_text_words_unique_dict.items(), key = lambda key_val_tu
top_25_words
```

Out[17]:

```
[('of', 28),
 ('you', 23),
 ('to', 22),
 ('a', 20),
 ('the', 17),
 ('and', 16),
 ('that', 15),
 ('I', 15),
 ('is', 12),
 ('for', 12),
 ('in', 11),
 ('be', 11),
 ('his', 11),
 ('he', 11),
 ('my', 10),
 ('it', 9),
 ('will', 9),
 ('so', 8),
 ('dear', 8),
 ('was', 8),
 ('are', 8),
 ('them', 8),
 ('must', 7),
 ('have', 7),
 ('no', 7),
 ('not', 7),
 ('as', 7),
 ('Mr', 6),
 ('Bennet', 6),
 ('she', 6),
 ('or', 5),
 ('such', 5),
 ('may', 5),
 ('one', 5),
 ('has', 5),
 ('me', 5),
 ('know', 5),
 ('see', 5),
 ('visit', 5),
 ('her', 5),
 ('man', 4),
 ('him', 4),
 ('by', 4),
 ('with', 4),
 ('can', 4),
```

```
('You', 4),
('go', 4),
('do', 4),
('fortune', 3),
('want', 3),
('little', 3),
('on', 3),
('replied', 3),
('had', 3),
('wife', 3),
('four', 3),
('much', 3),
('how', 3),
('girls', 3),
('Bingley', 3),
('woman', 3),
('daughters', 3),
('than', 3),
('your', 3),
('twenty', 3),
('truth', 2),
('single', 2),
('possession', 2),
('good', 2),
('some', 2),
('other', 2),
('heard', 2),
('Netherfield', 2),
('at', 2),
('Mrs', 2),
('Long', 2),
('been', 2),
('all', 2),
('taken', 2),
('young', 2),
('large', 2),
('take', 2),
('married', 2),
('sure', 2),
('five', 2),
('thousand', 2),
('year', 2),
('thing', 2),
('am', 2),
('thinking', 2),
('marrying', 2),
('But', 2),
('very', 2),
('send', 2),
('better', 2),
('handsome', 2),
('like', 2),
('beauty', 2),
('but', 2),
('When', 2),
('over', 2),
('own', 2),
('think', 2),
('when', 2),
('into', 2),
```

```
('neighbourhoodIt', 2),
('more', 2),
('assure', 2),
('what', 2),
('they', 2),
('us', 2),
('if', 2),
('Lizzy', 2),
('half', 2),
('years', 2),
('get', 2),
('come', 2),
('It', 1),
('universally', 1),
('acknowledged', 1),
('wifeHowever', 1),
('known', 1),
('feelings', 1),
('views', 1),
('first', 1),
('entering', 1),
('neighbourhood', 1),
('this', 1),
('well', 1),
('fixed', 1),
('minds', 1),
('surrounding', 1),
('families', 1),
('considered', 1),
('rightful', 1),
('property', 1),
('their', 1),
('daughtersMy', 1),
('said', 1),
('lady', 1),
('day', 1),
('Park', 1),
('let', 1),
('lastMr', 1),
('notBut', 1),
('returned', 1),
('just', 1),
('here', 1),
('told', 1),
('about', 1),
('itMr', 1),
('made', 1),
('answerDo', 1),
('who', 1),
('cried', 1),
('impatientlyYou', 1),
('tell', 1),
('objection', 1),
('hearing', 1),
('itThis', 1),
('invitation', 1),
('enoughWhy', 1),
('says', 1),
('from', 1),
('north', 1),
```

```
('England', 1),
('came', 1),
('down', 1),
('Monday', 1),
('chaise', 1),
('place', 1),
('delighted', 1),
('agreed', 1),
('Morris', 1),
('immediately', 1),
('before', 1),
('Michaelmas', 1),
('servants', 1),
('house', 1),
('end', 1),
('next', 1),
('weekWhat', 1),
('nameBingleyIs', 1),
('singleOh', 1),
('Single', 1),
('A', 1),
('What', 1),
('fine', 1),
('our', 1),
('girlsHow', 1),
('How', 1),
('affect', 1),
('themMy', 1),
('tiresome', 1),
('themIs', 1),
('design', 1),
('settling', 1),
('hereDesign', 1),
('Nonsense', 1),
('talk', 1),
('likely', 1),
('fall', 1),
('love', 1),
('therefore', 1),
('soon', 1),
('comesI', 1),
('occasion', 1),
('themselves', 1),
('which', 1),
('perhaps', 1),
('still', 1),
('any', 1),
('best', 1),
('partyMy', 1),
('flatter', 1),
('certainly', 1),
('share', 1),
('pretend', 1),
('anything', 1),
('extraordinary', 1),
('now', 1),
('grownup', 1),
('ought', 1),
('give', 1),
('beautyIn', 1),
```

```
('cases', 1),
('often', 1),
('ofBut', 1),
('indeed', 1),
('comes', 1),
('engage', 1),
('youBut', 1),
('consider', 1),
('Only', 1),
('an', 1),
('establishment', 1),
('would', 1),
('Sir', 1),
('William', 1),
('Lady', 1),
('Lucas', 1),
('determined', 1),
('merely', 1),
('account', 1),
('general', 1),
('newcomers', 1),
('Indeed', 1),
('impossible', 1),
('notYou', 1),
('overscrupulous', 1),
('surely', 1),
('dare', 1),
('say', 1),
('glad', 1),
('few', 1),
('lines', 1),
('hearty', 1),
('consent', 1),
('whichever', 1),
('chooses', 1),
('though', 1),
('throw', 1),
('word', 1),
('LizzyI', 1),
('desire', 1),
('bit', 1),
('others', 1),
('Jane', 1),
('nor', 1),
('goodhumoured', 1),
('Lydia', 1),
('always', 1),
('giving', 1),
('preferenceThey', 1),
('none', 1),
('recommend', 1),
('silly', 1),
('ignorant', 1),
('something', 1),
('quickness', 1),
('sistersMr', 1),
('abuse', 1),
('children', 1),
('way', 1),
('delight', 1),
```

```
('vexing', 1),
('compassion', 1),
('poor', 1),
('nervesYou', 1),
('mistake', 1),
('high', 1),
('respect', 1),
('nerves', 1),
('They', 1),
('old', 1),
('friends', 1),
('mention', 1),
('consideration', 1),
('these', 1),
('last', 1),
('leastAh', 1),
('sufferBut', 1),
('hope', 1),
('live', 1),
('many', 1),
('men', 1),
('use', 1),
('should', 1),
('since', 1),
('themDepend', 1),
('upon', 1),
('there', 1),
('allMr', 1),
('odd', 1),
('mixture', 1),
('quick', 1),
('parts', 1),
('sarcastic', 1),
('humour', 1),
('reserve', 1),
('caprice', 1),
('experience', 1),
('threeandtwenty', 1),
('insufficient', 1),
('make', 1),
('understand', 1),
('character', 1),
('Her', 1),
('mind', 1),
('less', 1),
('difficult', 1),
('develop', 1),
('She', 1),
('mean', 1),
('understanding', 1),
('information', 1),
('uncertain', 1),
('temper', 1),
('discontented', 1),
('fancied', 1),
('herself', 1),
('nervous', 1),
('The', 1),
('business', 1),
('life', 1),
```

```
('its', 1),
('solace', 1),
('visiting', 1),
('news', 1)]
```

Data Wrangling with Python: Activity 3 page 49

1. Look up the definition of permutations and dropwhile from itertools.
2. Write an expression to generate all the possible three-digit numbers using 0, 1, and 2.
3. Loop over the iterator expression you generated before. Print each element that's returned by the iterator. Use assert and isinstance to make sure that the elements are of the tuple type.
4. Write the loop again using dropwhile with a lambda expression to drop any leading zeros from the tuples. Also, cast the output of dropwhile to a list.
5. Check the actual type that dropwhile returns.
6. Combine the preceeding code into one block, and 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.

In [18]:
```
# Look up the definition of permutations from itertools (page 38, Exercise 15)

from itertools import permutations
# from itertools import dropwhile
# from itertools import combinations
# from itertools import repeat
# from itertools import zip_longest

permutations?
```

In [19]:
```
# Look up the definition of dropwhile from itertools (page 38, Exercise 15)

# from itertools import permutations
from itertools import dropwhile
# from itertools import combinations
# from itertools import repeat
# from itertools import zip_longest

dropwhile?
```

In [20]:
```
# Write an expression to generate all the possible three-digit numbers using 0, 1, and

possibilities = permutations(range(3))
possibilities
```

Out[20]:
```
<itertools.permutations at 0x1b7c39fecc0>
```

In [21]:
```
# Loop over the iterator expression you generated before. Print each element that's ret
# Use assert and isinstance to make sure that the elements are of the tuple type.

for poss_tuple in permutations(range(3)):
    print(poss_tuple)
    assert isinstance(poss_tuple, tuple)
```

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

In [22]:

```
# Write the loop again using dropwhile with a lambda expression to drop any leading zer
# Also, cast the output of dropwhile to a list.

for poss_tuple in permutations(range(3)):
    poss_tuple = list(dropwhile(lambda x: x <= 0, poss_tuple))
    print(poss_tuple)
```

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

In [23]:

```
# Check the actual type that dropwhile returns.

for poss_tuple in permutations(range(3)):
    poss_tuple = list(dropwhile(lambda x: x <= 0, poss_tuple))
    print(isinstance(poss_tuple, list))
```

```
True
True
True
True
True
True
```

In [24]:

```
# Combine the preceeding code into one block,
# and this time write a separate function where you will pass the list generated from d
# and the function will return the whole number contained in the list.

import math


def convert_to_number(number_stack):
    last_number = 0
    for i in range(0, len(number_stack)):
        last_number += (number_stack.pop() * (math.pow(10, i)))
    return last_number
```

In [25]:

```
for poss_tuple in permutations(range(3)):
    number_stack = list(dropwhile(lambda x: x <= 0, poss_tuple))
    print(convert_to_number(number_stack))
```

```
12.0
21.0
102.0
120.0
```

```
201.0
210.0
```

Data Wrangling with Python: Activity 4 page 59

1. Import zip_longest from itertools. Create a join function to zip header, line and fillvalue = None.
2. Open the accompanying sales_record.csv file from GitHub link by using r mode inside a with block and first check that it is opened.
3. Read the first line and use string methods to generate a list of all the column names.
4. Start reading the file. Read it line by line.
5. Read each line and pass that line to a function, along with the list of the headers.

In [26]:
```
# Import zip_longest from itertools.

from itertools import zip_longest
```

In [27]:
```
# Create a join function to zip header, line and fillvalue = None.

def csv_to_dict(header, line):
    zip_line = zip_longest(header,line, fillvalue=None)
    ret_dict = {kv[0]: kv[1] for kv in zip_line}
    return ret_dict
```

In [28]:
```
# Open the accompanying sales_record.csv file from GitHub link by using r mode inside a
# First check that it is open
# Read the first line and use string methods to generate a list of all the column names

with open("sales_record.csv", "r") as file:
    line_one = file.readline()
    header = line_one.replace("\n", "").split(",")
    print(header)
```

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

In [29]:
```
# Start reading the file. Read it line by line.
# Read each line and pass that line to a function, along with the list of the headers.

with open("sales_record.csv", "r") as file:
    line_one = file.readline()
    header = line_one.replace("\n", "").split(",")
    for i, line in enumerate(file):
        line = line.replace("\n", "").split(",")
        d = csv_to_dict(header, line)
        print(d)
        if i > 3:
            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': '87
```

999.84', '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'}